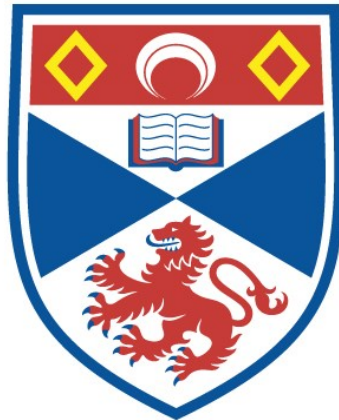


# NORMALISATION TECHNIQUES IN PROOF THEORY AND CATEGORY THEORY

Taher Tawfik Ahmed Hamza

A Thesis Submitted for the Degree of PhD  
at the  
University of St Andrews



1986

Full metadata for this item is available in  
St Andrews Research Repository  
at:  
<http://research-repository.st-andrews.ac.uk/>

Please use this identifier to cite or link to this item:  
<http://hdl.handle.net/10023/13371>

This item is protected by original copyright

NORMALISATION TECHNIQUES IN  
PROOF THEORY AND CATEGORY THEORY.  
AN IMPLEMENTATION AND APPLICATIONS.

A thesis submitted to the university  
of St. Andrews for the degree of  
Doctor of philosophy

by

Taher Tawfik Ahmed Hamza

Department of Computational Science,  
University of St. Andrews.

November 1985.



ProQuest Number: 10167251

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



ProQuest 10167251

Published by ProQuest LLC (2017). Copyright of the Dissertation is held by the Author.

All rights reserved.

This work is protected against unauthorized copying under Title 17, United States Code  
Microform Edition © ProQuest LLC.

ProQuest LLC.  
789 East Eisenhower Parkway  
P.O. Box 1346  
Ann Arbor, MI 48106 – 1346

74  
4397

I Taher Tawfik Hamza hereby certify that this thesis has been composed by myself, that it is a record of my own work, and that it has not been accepted in partial or complete fulfillment of any other degree or professional qualification.

Signed .

.....

Date ..18...iii..1986

I was admitted to the Faculty of Science of the University of St. Andrews under Ordinance No. 12 on April 1982 and as a candidate for the degree of Ph.D. on October 1983.

signed .

...

Date ...18...iii..1986

In submitting this thesis to the University of St Andrews I understand that I am giving permission for it to be made available for use in accordance with the regulations of the University Library for the time being in force, subject to any copyright vested in the work not being affected thereby. I also understand that the title and abstract will be published, and that a copy of the work may be made and supplied to any bona fide library or research worker.

CERTIFICATE

I hereby certify that the candidate has fulfilled the conditions of the Resolution and Regulations appropriate to the degree of Ph.D.

signature of Supervisor

Date 18 iii 86

## ACKNOWLEDGEMENTS

I would like to express indebtedness and sincere thanks to my supervisor, Dr R. Dyckhoff, for his invaluable assistance in the preparation of this work. He has always been a great help and spared no effort in giving me advice, guidance and supervision of this work. I also thank the Egyptian government for the financial support I have received during this study.

Finally I owe apology to my wife and my children Sulaf and Ahmed for lack of attendance during the preparation of this thesis.

## ABSTRACT

The word problem for the free categories with some structure generated by a category  $X$  can be solved using proof-theoretical means. These free categories give a semantics in which derivations of GENTZEN's propositional sequent calculus can be interpreted by means of arrows of these categories.

In this thesis we describe, implement and document the cut-elimination and the normalization techniques in proof theory as outlined in SZABO [1978]; we show how these are used in order to solve, mechanically, the word problem for the free categories with structure of : cartesian, bicartesian, distributive bicartesian, cartesian closed, and bicartesian closed. This implementation is extended by a procedure to interpret intuitionistic propositional sequent derivations as arrows of the above categories. Implementation of these techniques has forced us to modify the techniques in various inessential ways.

The description and the representation in the syntax of our implementation of the above categories is contained in chapters 1 - 5, where each chapter describes one theory and concludes with examples of the system in use to represent concepts and solve simple word problems from category theory ( of various types ). Appendix 1 contains some apparent printing errors we have observed in the work done by SZABO. The algorithms used in the proof of the cut-elimination theorems and normalization through chapters 1 - 5 are collected in appendices 2 - 4. Appendices 5 - 8 concern the implementation and its user manual.



**CONTENTS**  
 =====

Acknowledgements .....		I
Abstract .....		ii
Contents .....		iii
Chapter 0	Introduction .....	1
Chapter 1	Cartesian Category Theory .....	7
Chapter 2	Bicartesian Category Theory .....	38
Chapter 3	Distributive Bicartesian Category Theory .....	68
Chapter 4	Cartesian Closed Category Theory .....	91
Chapter 5	Bicartesian Closed Category Theory .....	119
Chapter 6	Notes on the Implementation .....	137
Conclusion .....		145
References .....		146
Appendix 1	Apparent errors in the printing of SZABO [1978] .....	A - 1
Appendix 2	Cut Elimination Algorithm .....	A - 4
Appendix 3	Contraction & Interchange Elimination Algorithm .....	A - 12
Appendix 4	Normalization Algorithm .....	A - 26
Appendix 5	User Manual .....	A - 43
Appendix 6	Predicate Index .....	A - 61
Appendix 7	Maintenance Information .....	A - 63
Appendix 8	Program Listing .....	

Chapter 0

INTRODUCTION

The purpose of this thesis is the modification and implementation of the cut-elimination and normalization algorithms described by SZABO in his book, Algebra of Proofs [1978] ( hereinafter referred to only as SZABO ). This implementation is for the assistance both of category theorists and logicians; for the first, as a step towards the automation of reasoning about morphisms in categories, for the second as a practical method for transformation of proofs in GENTZEN's propositional sequent calculus as expounded by SZABO. Our implementation is in C-Prolog on a VAX 11/750 computer under Unix BSD 4.2.

The connection between category theory and proof theory behind our implementation was used by LAMBEK [1968 , 1969 , 1972], MANN [1975] and SZABO. This connection shows that, on the one hand, category theory gives rise to non-trivial semantics for derivations in the intuitionistic Gentzen systems, by constructing the free category of a certain type, having as objects the formulae of the relevant deductive system and as arrows, the equivalence classes of proofs; this free category is an algebraic model of the class of proofs in this deductive system. On the other hand proof theory provides both a suitable syntax in which arrows of a free category can be represented, and also techniques, such as cut-elimination and normalization, for transforming such representations. These techniques can be used to solve the word problem in, for example, free cartesian (-closed, ...) categories: equality of arrows is checked by checking their normal representations for syntactic equality.

( The word problem for a left adjoint  $F : \text{Cat} \rightarrow C$  to a forgetful functor  $U : C \rightarrow \text{Cat}$  is defined as the problem of finding an algorithm which decides the equality of arrows of the category  $F(X)$  relative to the equality of arrows of the category  $X$  )

In the following, I introduce the two main techniques behind our implementation.

a) Cut-elimination

When Gentzen [1969] examined the specific properties of the 'natural calculus', he was led to the conjecture that it should be possible to bring purely logical proofs into a certain 'normal form' in which all

concepts required for the proof would in some sense appear in the conclusion of the proof. In order to enunciate and prove his theorem which he called the 'Hauptsatz', however, Gentzen had to abandon his natural 'assumption' calculi and formulate logistic 'sequent' calculi in which the logical rules can be divided into 'structural' and 'operational' rules. The Hauptsatz then refers to the fact that one of the 'structural' inference rules of the calculus, the 'cut' rule, can be eliminated from purely logical proofs. The cut rule as stated by Gentzen is :

$$\frac{G \rightarrow H, a \quad a, D \rightarrow L}{G, D \rightarrow H, L}$$

SZABO gives a detailed algorithm for the elimination of cuts from proofs in a propositional sequent calculus, with a slightly different 'cut' rule:

$$\frac{G \rightarrow F \text{ a E} \quad D \text{ a L} \rightarrow H}{D G L \rightarrow F H E}$$

#### b) Normalization

SZABO defined a monotone partial ordering relation ( $\geq$ ) on the class of GENTZEN derivations. A derivation  $p$  is defined to be normal if it is cut-free and if  $p \geq q$  implies that  $p = q$ . A derivation  $p$ , say, reduces to  $q$  if there exists a finite sequence  $\langle p_1, \dots, p_n \rangle$  of derivations such that  $p = p_1$ ,  $p_n = q$  and  $p_1 \geq p_2 \geq \dots \geq p_n$  and  $p$ , say, reduces immediately to  $q$ , if  $p = q$  or if  $p \geq q$  by virtue of precisely one of the defining conditions of ' $\geq$ '. Every derivation  $p$  which represents an arrow of one of the categories in chapters 1 - 5, reduces to a cut-free derivation  $q$ , and an induction on the number of violations of the conditions, in appendices 3 and 4, occurring in  $q$  shows that  $p$  reduces to a normal, and hence unique normal, derivation  $r$ .

It is worth mentioning the relation between this approach of SZABO (which we have implemented), and other approaches to such problems. KREISEL [1971] and PRAWITZ [1971] have developed Gentzen's approach of 'natural deduction'; PRAWITZ defined an equivalence relation on proofs in the natural deduction systems and proved some theorems for the normalization of these proofs. MANN [1975] showed that the equivalence relation defined by PRAWITZ for the natural deduction proofs is the same as the equivalence relation defined by SZABO for the sequent calculus

proofs, in the following ways : firstly, the equivalence classes (under PRAWITZ's definition) can be used to form the maps of a cartesian closed category, and secondly, under the usual translation from the sequent calculus systems to the natural deduction systems, two equivalent proofs (under the SZABO definition) translate to two equivalent proofs ( under the PRAWITZ definition ). Although it is acknowledged that the systems of natural deduction are in many ways much easier to work with, ' and in most respects more ' natural ' , than the calculus of sequents, we have nevertheless chosen the sequent calculus approach because there are very explicit algorithms for cut-elimination, normalization, representation of arrows by proofs and interpretations of sequent calculus proofs by means of arrows : these are explicit enough for convenient implementation.

LAMBEK and SCOTT [1984] gave an equational presentation for the cartesian closed categories, by defining " cartesian closed category " as a deductive system with an equivalence relation between proofs, in order to represent the arrows of the cartesian closed categories by equivalence classes of these proofs and satisfy a set of equations. They showed how to construct a cartesian closed category freely generated by a graph  $X$  using the deductive system technique. They constructed the free cartesian closed category with weak natural numbers object generated by a typed lambda-calculus and showed how to solve the validity problem ( i.e to prove the equality between arrows ) in the cartesian closed categories. A detailed proof was given for the case of the free cartesian closed category with weak natural numbers object generated by pure typed lambda-calculus ( by a pure typed lambda-calculus is meant a typed lambda-calculus with only two basic types 1 and  $N$  ). The problem of deciding equality between two arrows reduces to deciding the equality of their corresponding terms in the pure typed lambda-calculus. They gave a solution for the latter problem.

CURIEN [1985] has an equational axiomatic system for cartesian closed categories, and showed how to construct a free cartesian closed category  $CCC(K)$  generated by a set  $K$  of basic types. He proved some equivalence theorems between  $CCC(K)$  and the typed lambda - calculus generated by the set  $K$  of basic types. Finally he pointed out that this translation to typed lambda- calculus can be used to solve the validity problem for cartesian closed categories using the confluence of the typed lambda-calculus (c.f POTTINGER [1979]) and the noetherian property as shown for the similar case of the free cartesian closed category over the empty graph by LAMBEK and SCOTT .

Most of the notations, definitions and theorems used for category theory in this work are as in MACLANE [1971]; the others for logic as in GENTZEN [1969]. For lack of the convenient printer characters for Greek letters, we use the following notations, especially in appendices 2, 3, 4, corresponding to the Greek letters used by SZABO :

a	for	$\alpha$		b	for	$\beta$
c	for	$\gamma$		d	for	$\delta$
G	for	$\Gamma$		D	for	$\Delta$
L	for	$\Lambda$		Z	for	$\Xi$
P	for	$\Pi$		F	for	$\Phi$
E	for	$\Psi$		H	for	$\Theta$
O	for	$\Omega$				

The thesis is divided into six chapters and eight appendices. The first five chapters correspond to five classes of category : cartesian, bicartesian, distributive bicartesian, cartesian closed and bicartesian closed categories respectively. We follow SZABO in discussing each theory while tidying up some trivial printing errors and other obscurities. In the theory of cartesian categories, for example, we extend the unlabelled deductive system of SZABO by a new rule of inference (interchange in the antecedent) which is required for the elimination of cuts during the normalization process. All the extensions we have done, and the changes in the definition of the normal form, in each theory, will be discussed in more detail later, in the first five chapters. We end each chapter with examples and illustrations thereof arising in the theory being studied, in order to demonstrate the use of the system. Chapter six contains notes on the implementation..

In appendix 1, we list some apparent errors in the printing of SZABO; appendix 2 contains the parts of SZABO's cut-elimination algorithm which have been implemented. Appendix 3 consists of new algorithms, some of them to eliminate contractions and interchanges, both in the succedent and the antecedent, in the cases of : cartesian, bicartesian, and distributive bicartesian categories; and others to transform proofs into proofs with restricted applications of the thinning rule. These algorithms correspond to the changes in the definition of the normal form, as well as to prove some lemmas and corollaries ( stated without

proof in SZABO ). Appendix 4 collects together those parts of SZABO's normalization algorithm which we have implemented, in order to complete the description of the normalization. Appendix 5 is a user manual for the system; appendix 6 is an index to the predicates used in the program. In appendix 7, we give some information on how to maintain the system and in appendix 8, we list the implementation program itself.

Prolog is a high-level programming language with powerful pattern-matching facilities. Since our system consists of many transformation rules, to be chosen according to the LHS of a rule matching a fragment of a proof, the pattern-matching facilities of Prolog suggested Prolog as a natural candidate for use as the language of the system. Prolog has a convenient interactive user interface, which allowed a speedy development of the earlier phases of the system.

In addition to the substantial space that 'Prolog' requires to run, our system requires much space. The following table shows the space used by Prolog system itself and the run time since Prolog started, the space used by our implementation and the time taken by Prolog to restore the saved implementation to its top level in order to be ready for use, and the space used and the run time taken in the computation of some simple and complicated examples.

( all examples obtained at one randomly chosen session )

	space used	run time
Prolog	58100 bytes	1.70 sec
the system	398702 bytes	12.23 sec
example 1 in (1.6)	518 bytes	4.57 sec
(3.6)	45112 bytes	250.36 sec
example 2 in (4.6)	182680 bytes	688.04 sec

In chapter 3, we cannot give examples to test the equality of arrows in the theory of distributive bicartesian categories. This is because the representations of the arrows of this theory are too complicated, and need too much space to be computed; the system responds that there is not enough space to complete the computation. The most troublesome part of the whole technique appears to lie in the complexity of the cut-elimination algorithm. In some of the transformation rules ( like (C.3),..... in appendix 2 ), the movement of an instance of 'cut' up

to the leaves of the tree introduces two more 'cuts' and these, by turn, may need to introduce more instances of 'cut' until all the instances of 'cut' reach the leaves of the tree (whence they can be removed, finally).

A modified version of our implementation has been already done, as a step towards overcoming the complexity related to the space and the run time of the computation with our system. All the examples related to category theory mentioned herein have been redone in this new version, without any logical differences being apparent. The following results show that the new version indeed reduces the space and the run time used in computation with our system.

( Note these examples are the same mentioned above and the values between brackets are the old ones )

	space used		run time	
example 1 in (1.6)	424 (518)	bytes	3.70 (4.57)	sec
(3.6)	44585 (45112)	bytes	149.77 (250.36)	sec
example 2 in (4.6)	157196 (182680)	bytes	171.03 (688.04)	sec

( Note : the program listed in Appendix 8 is the old version )

Since the purpose of the present thesis is not the presentation of a substantially new algorithm but the implementation, modification and application of SZABO's algorithms, some of the material in the following chapters deliberately bears a close resemblance to SZABO's descriptive and theoretical material. It would have been unnecessarily confusing either to introduce an entirely different description of the algorithms or to omit such description altogether.

## Chapter 1

### CARTESIAN CATEGORY THEORY

This chapter deals with the connection between the class of cartesian categories and the class of derivations in a particular Gentzen deductive system. By a cartesian category we mean a category with a terminal object  $T$  and a binary product function  $\times$  (but not a category with finite limits).

We follow SZABO in discussing this theory, and tidy up some obscurities. These lie essentially in the definition of the unlabelled deductive system relevant to the theory of cartesian categories. The need for the "interchange in the antecedent" rule (R4) in order to perform large contractions will be discussed in more detail later in (1.5.1). We add a lemma (1.5.3); this is required by the definition of normal form in SZABO, but not there stated. The algorithmic proof of the lemma is implemented in our system. As a result of adding rule (R4) to the definition of the class of unlabelled 'cartesian-derivations', we have restated, and constructed an algorithmic proof of lemma (1.5.4) which was stated without proof in SZABO. This proof is implemented in our system to complete the the definition of the normal form. We have added some clauses to the proof of the cut-elimination theorem to handle the addition of rule (R4).

This chapter is divided into six sections as follows. Section (1.1) is a definition of what is meant by a cartesian category. The construction of the free cartesian category  $Fc(X)$  over a category  $X$ , is described through subsections of the section (1.2); in (1.2.1) we define the cartesian formulae over a category  $X$ , in section (1.2.2) the class  $LDC(X)$  of (cartesian) labelled derivations over  $X$  is defined, in (1.2.3) we define an equivalence relation on the class  $LDC(X)$ , in (1.2.4) we define the equivalence classes and finally in (1.2.5)  $Fc(X)$  is defined via this equivalence relation. Section (1.3) is divided into two subsections; (1.3.1) contains the definition of the class  $Dc(X)$  of (cartesian) unlabelled derivations, in order to represent the arrows of  $Fc(X)$  by means of unlabelled derivations with cuts as an alternative to using labelled derivations. (1.3.2) describes how derivations of  $Dc(X)$  are



represented in our Prolog implementation. We conclude this section with some examples of derivations in  $Dc(X)$  and of how to use the system to check the correctness of these derivations.

Section (1.4) is concerned with the semantics of the class  $Dc(X)$  and is divided into two subsections : (1.4.1) contains an interpretation function from  $Dc(X)$  to  $ArFc(X)$  followed by two examples from the system to illustrate this interpretation function; (1.4.2) contains a proof to show that  $Dc(X)$  is adequate, in the sense that every arrow in  $Fc(X)$  is representable by means of some derivations in  $Dc(X)$ , and also some examples from the system to illustrate this representation. Section (1.5) is divided into five subsections; (1.5.1) is a justification for introducing the interchange rule in the definition of  $Dc(X)$ ; in (1.5.2), (1.5.3), (1.5.4) and (1.5.5) we prove some theorems for eliminating cuts, contractions, and interchanges and getting normal forms for derivations of  $Dc(X)$ . We end every subsection with some examples in operation with the system to illustrate the processes being studied in each case. We conclude, in section (1.6), with some examples from category theory in operation with the system.

### (1.1) Definition

A cartesian category is a category  $C$  with the following structure:

(1) A bifunctor  $(-)\uparrow(-) : C \times C \longrightarrow C$ .

(2) A distinguished object  $T \in \text{Ob } C$ .

(3) Two adjunctions  $\alpha_\pi$  and  $\alpha_\tau$ , where

$$\alpha_\pi = \{ \alpha_\pi(A,B,C) : C(A, B \uparrow C) \longrightarrow C(A,B) \times C(A,C) \in \text{Ar Ens} \mid A,B,C \in \text{Ob } C \},$$

and

$$\alpha_\tau = \{ \alpha_\tau(A) : C(A,T) \longrightarrow \{e\} \in \text{Ar Ens} \mid A \in \text{Ob } C \}.$$

### (1.2) The Free Cartesian Category $Fc(X)$

Let  $X$  be a fixed but arbitrary small category. The free cartesian category  $Fc(X)$  is constructed as follows :-

#### (1.2.1) Definition

The class  $cL(X)$  of (cartesian) formulae over  $X$  is defined by

- (i) Every object of  $X$  is a formula ;
- (ii)  $T$  is a formula ;
- (iii) If  $A, B$  are formulae, then  $A \uparrow B$  is ;
- (iv) The class  $cL(X)$  is generated by (i) to (iii) .

(1.2.2) The Labelled Derivations LDc(X)

The class of (cartesian) labelled derivations over  $X$ , and their conclusions, are simultaneously defined by

- i) for objects  $A, B$  of  $X$  and  $f$  in  $X(A, B)$ ,  
 $LA1(f)$  is a derivation  
with conclusion  $f : A \dashrightarrow B$ ;
- ii) for  $A$  in  $cL(X)$ , not an object of  $X$ ,  
 $LA2(A)$  is a derivation,  
with conclusion  $1(A) : A \dashrightarrow A$ ;
- iii) for  $A$  in  $cL(X)$ ,  
 $LA3(A)$  is a derivation,  
with conclusion  $\tau(A) : A \dashrightarrow T$ ;
- iv) for  $A, B$  in  $cL(X)$ ,  
 $LA4(A, B)$  is a derivation,  
with conclusion  $\pi_{\text{left}}(A, B) : A \uparrow B \dashrightarrow A$ ;
- v) for  $A, B$  in  $cL(X)$ ,  
 $LA5(A, B)$  is a derivation,  
with conclusion  $\pi_{\text{right}}(A, B) : A \uparrow B \dashrightarrow B$ ;
- vi) for derivations  $F, G$  with conclusions  
 $f : A \rightarrow B$ ,  $g : B \rightarrow C$  respectively,  
 $COMP(F, G, B)$  is a derivation,  
with conclusion  $h : A \rightarrow C$ , where  $h$  is the  
composition of  $f$  and  $g$ ;
- vii) for derivations  $F, G$  with conclusions  
 $f : A \rightarrow B$ ,  $g : A \rightarrow C$  respectively,  
 $ANGLE(F, G, B \uparrow C)$  is a derivation,  
with conclusion  $\langle f, g \rangle : A \rightarrow B \uparrow C$ .
- viii) The class of the derivations  $LDc(X)$  is generated by i to vii.

NB

In clauses vi) & vii) above, it is necessary to mention the active formulae in the derivation, for a technical reason in the clauses vi) and vii) of the proof of completeness theorem.

(1.2.3) Definition

The relation  $\equiv$  is defined as the smallest equivalence relation on  $LDc(X)$  satisfying

- i) If  $F \equiv G$  and  $H \equiv K$  with conclusions  $f, g : A \rightarrow B$ ,  
 $h, k : B \rightarrow C$ ,  
then  $\text{COMP}(F, H, B) \equiv \text{COMP}(G, K, B)$  ;
- ii) If  $F \equiv G$  and  $H \equiv K$  with conclusions  $f, g : A \rightarrow B$ ,  
 $h, k : A \rightarrow C$ ,  
then  $\text{ANGLE}(F, H, B \uparrow C) \equiv \text{ANGLE}(G, K, B \uparrow C)$  ;
- iii) For  $F$  with conclusion  $f : A \rightarrow B$ , both  
 $\text{COMP}(\text{LA2}(A), F, A) \equiv F$  and  
 $\text{COMP}(F, \text{LA2}(B), B) \equiv F$  ;
- iv) For  $F, G, H$  with conclusions  $f : A \rightarrow B$ ,  $g : B \rightarrow C$ ,  
 $h : C \rightarrow D$  respectively,  
 $\text{COMP}(F, \text{COMP}(G, H, C), B) \equiv \text{COMP}(\text{COMP}(F, G, B), H, C)$  ;
- v) For  $F, G$  with conclusions  $f : A \rightarrow B$ ,  $g : A \rightarrow C$  respectively,  
 $\text{COMP}(\text{ANGLE}(F, G, B \uparrow C), \text{LA4}(B, C), B \uparrow C) \equiv F$  and  
 $\text{COMP}(\text{ANGLE}(F, G, B \uparrow C), \text{LA5}(B, C), B \uparrow C) \equiv G$  ;
- vi) For  $F$  with conclusion  $f : A \rightarrow B \uparrow C$ ,  
 $\text{ANGLE}(\text{COMP}(F, \text{LA4}(B, C), B \uparrow C),$   
 $\text{COMP}(F, \text{LA5}(B, C), B \uparrow C), B \uparrow C) \equiv F$  ;
- vii) For  $F$  with conclusion  $f : A \rightarrow T$ ,  
 $\text{LA3}(A) \equiv F$  .

(1.2.4) Definition

$[F]$  : denotes the  $\equiv$ -class of  $F$  .

(1.2.5) Definition

We define  $\text{Fc}(X)$  to be the category having :

- i) as objects , the cartesian formulae over  $X$  ;
- ii) as arrows , the  $\equiv$ -classes of  $\text{LDc}(X)$  ;
- iii) as dom , the function defined by  
 $\text{dom}([F]) = A$  where  $F$  has conclusion  $f : A \rightarrow B$  ;
- iv) as cod , the function defined by  
 $\text{cod}([F]) = B$  where  $F$  has conclusion  $f : A \rightarrow B$  ;
- v) as comp , the function defined by  
 $\text{comp}([G], [F]) = [\text{COMP}(F, G, B)]$  ,  
where  $F$  has conclusion  $f : A \rightarrow B$  ,  
 $G$  has conclusion  $g : B \rightarrow C$  ;
- vi) as id , the function defined by  
 $\text{id}(A) = [\text{LA1}(\text{id}(A))] \text{ for } A \text{ in } \text{Ob}(X)$  , and  
 $[\text{LA2}(A)] \text{ for } A \text{ not in } \text{Ob}(X)$  ;

- vii) as  $T$  ,            the formula  $T$  ;
- viii) as  $\tau$  ,            the function defined by  
 $\tau(A) = [ LA3(A) ]$  ;
- ix) as  $\pi_{left}$  , the function defined by  
 $\pi_{left}(A,B) = [ LA4(A,B) ]$  ;
- x) as  $\pi_{right}$  , the function defined by  
 $\pi_{right}(A,B) = [ LA5(A,B) ]$  ;
- xi) as  $\langle -, - \rangle$  , the function defined by  
 $\langle [F], [G] \rangle = [ ANGLE( F, G, cod([F]) \uparrow cod([G])) ]$   
 when  $dom( [F] ) = dom( [G] )$  .

It is now routine to check that, with this structure,  $Fc(X)$  is a category, is cartesian, and is free cartesian over  $X$ . There is the obvious embedding  $X \rightarrow Fc(X)$  defined by  $A \mapsto A$ ;  $f \mapsto [ LA1(f) ]$ .

### (1.3) The Unlabelled Derivations $Dc(X)$

Instead of using labelled derivations  $LDC(X)$ , an alternative approach is to use unlabelled (cartesian) derivations  $Dc(X)$  to represent the arrows of  $Fc(X)$  .

#### (1.3.1) Definition

We define below the class  $Dc(X)$  of unlabelled (cartesian) derivations.

In the following, we use

$a, b, c, d, e \dots$	for formulae over $X$ ;
$D, G, L \dots$	for lists thereof ;
$G \rightarrow D \dots$	for sequents with antecedent $G$ and succedent $D$ ;
$p, p1, p2 \dots$	for derivations .

Derivations are really trees of sequents. We represent these inadequately using terms, called unlabelled ( cartesian ) derivations ; at the same time, we define the relationship between terms and particular sequents called conclusions then, from a derivation and a conclusion the tree of sequents can be recovered. The point is that the structural rules operate on formulae which may occur in various places, and our derivation terms only indicate the formula and not its occurrences. The conclusion however makes the occurrence clear. Thus, a derivation tree has a unique conclusion; our derivation terms do not necessarily have unique conclusions, but satisfy a relationship with sequents defined inductively as follows :

- i) if  $a, b$  are objects of  $X$  , and if  $f$  is in  $X(a, b)$  , then

$A1(f)$  is a derivation , with conclusion  $a \xrightarrow{f} b$  ;

- ii)  $A2$  is a derivation , with conclusion  $[ ] \rightarrow T$  ;

- iii) ( cut )  
 if  $p_1, p_2$  are derivations, with conclusions  $G \rightarrow a$ ,  $DaL \rightarrow c$  then  
 $R1(p_1, p_2, a)$  is a derivation with conclusion  $DGL \rightarrow c$  ;
- iv) ( thinning in the antecedent )  
 if  $p$  is a derivation with conclusion  $GD \rightarrow b$  , then  
 $R2(p, a)$  is a derivation with conclusion  $GaD \rightarrow b$  ;
- v) ( contraction in the antecedent )  
 if  $p$  is a derivation with conclusion  $GaaD \rightarrow b$  , then  
 $R3(p, a)$  is a derivation with conclusion  $GaD \rightarrow b$  ;
- vi) ( Interchange in the antecedent )  
 if  $p$  is a derivation with conclusion  $GabD \rightarrow c$  , then  
 $R4(p, b, a)$  is a derivation with conclusion  $GbaD \rightarrow c$  ;
- vii) ( and-introduction in the succedent )  
 if  $p_1, p_2$  are derivations with conclusions  $G \rightarrow a$ ,  $G \rightarrow b$ , then  
 $R10(p_1, p_2, a \uparrow b)$  is a derivation with conclusion  $G \rightarrow (a \uparrow b)$ ;
- viii) ( and-introduction in the antecedent )  
 if  $p$  is a derivation with conclusion  $GabD \rightarrow c$  , then  
 $R11(p, a \uparrow b)$  is a derivation with conclusion  $G(a \uparrow b)D \rightarrow c$  ;
- ix) The class of the unlabelled derivations  $Dc(X)$  is generated by  
 (i) to (viii) .

### (1.3.2) Representation of $Dc(X)$ in Prolog

We now give representations of the unlabelled derivations in Prolog, where a derivation  $p$  is represented by a step-collection of steps; these steps represent the conclusions of all subderivations of the derivation  $p$ . Each step is given a name followed by a list of two elements e.g.

$$g0 = [cut(g1, g2, a), [x, y] => [z]]$$

The first is a predicate which is the name of the operation being applied to the conclusion(s) of the subderivation(s), with arguments representing the name(s) given to the conclusion(s) of the subderivation(s) and the name of the active formula(e) . [ By active formula is meant the cut, thinning, contraction, interchange, or and-introduction formula ] . The second is composed of two lists, for the antecedent and succedent of the sequent, separated by the symbol  $=>$ .

We use the following abbreviations :-

& for  $\uparrow$

tr for T in A2 ( Prolog uses upper-case for variables )

true     for the name of the constant in A2  
 cut     for the name of the operation in  $R1(p1,p2,a)$   
 th     for the name of the operation in  $R2(p,a)$   
 con     for the name of the operation in  $R3(p,a)$   
 inc     for the name of the operation in  $R4(p,b,a)$   
 ais     for the name of the operation in  $R10(p1,p2,a \uparrow b)$   
 aia     for the name of the operation in  $R11(p,a \uparrow b)$  .

We use the meta-variables  $g1, g2, g3 \dots$  for the names of the steps.

- i)  $A1(f)$  is represented by
 
$$g1 = [f, [a] \Rightarrow [b]]$$
- ii)  $A2$  is represented by
 
$$g1 = [true, [] \Rightarrow [tr]]$$
- iii)  $R1(p1,p2,a)$  is represented by a step-collection with the last step,
 
$$g3 = [cut(g1,g2,a), DGL \Rightarrow [c]]$$
 where  $g1, g2$  are names of the last steps of the step-collections for the derivations  $p1, p2$  respectively .
- iv)  $R2(p,a)$  is represented by a step-collection with the last step,
 
$$g2 = [th(g1,a), GaD \Rightarrow [b]]$$
 where  $g1$  is the name of the last step in the step-collection for the derivation  $p$  .
- v)  $R3(p,a)$  is represented by a step-collection with the last step,
 
$$g2 = [con(g1,a), GaD \Rightarrow [b]]$$
 where  $g1$  is the name of the last step in the step-collection for the derivation  $p$  .
- vi)  $R4(p,b,a)$  is represented by a step-collection with the last step,
 
$$g2 = [inc(g1,b,a), GbaD \Rightarrow [c]]$$
 where  $g1$  is the name of the last step in the step-collection for the derivation  $p$  .
- vii)  $R10(p1,p2,a \uparrow b)$  is represented by a step-collection with the last step ,
 
$$g3 = [ais(g1,g2,a \& b), G \Rightarrow [a \& b]]$$
 where  $g1,g2$  are names of the last steps of the step-collections for the derivations  $p1, p2$  respectively .
- viii)  $R11(p,a \uparrow b)$  is represented by a step-collection with the last step,
 
$$g2 = [aia(g1,a \& b), Ga\&bD \Rightarrow [c]]$$
 where  $g1$  is the name of the last step of the step-collection for the derivation  $p$  .

We give some examples of unlabelled derivations, their representations in the implementation language and how to check the correctness of a proof using our implementation (proof checker part).

Note : we write the derivations in tree form .

Example 1      The derivation

$$\frac{\frac{a \xrightarrow{f} b \quad a \xrightarrow{g} c}{a \rightarrow b \uparrow c} \quad \frac{e \xrightarrow{h} d}{b \uparrow c, e \rightarrow d}}{a, e \rightarrow d}$$

is represented by

```
l=[f,[a]=>[b]]
m=[g,[a]=>[c]]
n=[ais(l,m,b&c),[a]=>[b&c]]
o=[h,[e]=>[d]]
p=[th(o,b&c),[b&c,e]=>[d]]
q=[cut(n,p,b&c),[a,e]=>[d]]
```

The following is input and output session with our system to check the correctness of the above proof. The command "theory(cart)" is used to inform the system that the following proof is constructed according to the definition of  $Dc(X)$  in the theory of cartesian categories. The command "prove(Z)", where Z represents a proof step, is required to prove the correctness of each step in the proof.

```

TH>szabo
Cut Elimination and Normal Form Program under Prolog
For help use the help command

yes
| ?- theory(cart).
yes
| ?- prove(l = [f, [a] => [b]]).
yes
| ?- prove(m = [g, [a] => [c]]).
yes
| ?- prove(n = [ais(l,m,b & c), [a] => [b & c]]).
yes
| ?- prove(o = [h, [e] => [d]]).
yes
| ?- prove(p = [th(o,b & c), [b & c,e] => [d]]).
yes
| ?- prove(q = [cut(n,p,b & c), [a,e] => [d]]).
yes
| ?-

```

Example 2      The derivation

$$\frac{\frac{a \rightarrow a \quad b \xrightarrow{g} c}{a, b \rightarrow a \quad a, b \rightarrow c} \quad \frac{a, c \xrightarrow{h} e}{a \uparrow c \rightarrow e}}{a, b \rightarrow a \uparrow c \quad d, a \uparrow c \rightarrow e} \quad \frac{d, a, b \rightarrow e}{a, d, b \rightarrow e}$$

is represented by

```

l=[id(a),[a]=>[a]]
m = [th(l,b),[a,b]=>[a]]
n = [g,[b]=>[c]]
o = [th(n,a),[a,b]=>[c]]
p = [als(m,o,a&c),[a,b]=>[a&c]]
q = [h,[c]=>[e]]
r = [th(q,a),[a,c]=>[e]]
s = [ala(r,a&c),[a&c]=>[e]]
t = [th(s,d),[d,a&c]=>[e]]
u = [cut(p,t,a&c),[d,a,b]=>[e]]
v = [inc(u,a,d),[a,d,b]=>[e]]

```

```

yes
| ?- theory(cart).
yes
| ?- prove(l = [id(a), [a] => [a]]).
yes
| ?- prove(m = [th(l,b), [a,b] => [a]]).
yes
| ?- prove(n = [g, [b] => [c]]).
yes
| ?- prove(o = [th(n,a), [a,b] => [c]]).
yes
| ?- prove(p = [als(m,o,a & c), [a,b] => [a & c]]).
yes
| ?- prove(q = [h, [c] => [e]]):
yes
| ?- prove(r = [th(q,a), [a,c] => [e]]).
yes
| ?- prove(s = [ala(r,a & c), [a & c] => [e]]).
yes
| ?- prove(t = [th(s,d), [d,a & c] => [e]]).
yes
| ?- prove(u = [cut(p,t,a & c), [d,a,b] => [e]]).
yes
| ?- prove(v = [inc(u,a,d), [a,d,b] => [e]]).
yes
| ?-

```

#### (1.4) The Semantics of Dc(X)

In this section, we interpret the derivations of the class  $Dc(X)$  as arrows of  $Fc(X)$  and prove the completeness theorem. In order to perform this interpretation, the following canonical arrows of  $Fc(X)$  are required ( By a canonical arrow in  $Fc(X)$  is meant, any arrow which can be represented canonically by the adjunctions  $\alpha_\pi$  and  $\alpha_\pi$ . This arrow does not depend on the arrows of the underlying category  $X$ , but on its objects. ) :

i)  $\delta(A) : A \rightarrow A \uparrow A$  for all  $A$  in  $ObFc(X)$ , where

$$\alpha_\pi^{-1} ( \langle Id(A), Id(A) \rangle ) = \delta(A) .$$

ii)  $\alpha(A,B,C) : A \uparrow (B \uparrow C) \rightarrow (A \uparrow B) \uparrow C$  for all  $A,B,C$  in  $ObFc(X)$ , where



$$\begin{aligned} & ( (\text{pi\_left}(A, B \uparrow C), \text{comp}(\text{pi\_left}(B, C), \text{pi\_right}(A, B \uparrow C))), \\ & \text{comp}(\text{pi\_right}(B, C), \text{pi\_right}(A, B \uparrow C))) = \text{alpha}(A, B, C) . \end{aligned}$$

iii)  $\text{sigma}(A, B) : A \uparrow B \rightarrow B \uparrow A$  for all  $A, B$  in  $\text{Ob Fc}(X)$ , where  
 $\langle \text{pi\_right}(A, B), \text{pi\_left}(A, B) \rangle = \text{sigma}(A, B)$  .

#### (1.4.1) Definition

We shall shortly define a function  $S$  from unlabelled derivations  $\text{Dc}(X)$  to  $\text{ArFc}(X)$  .

First, note that if  $p$  has conclusion  $G \rightarrow a$ , then  $S(p)$  is to be an arrow with codomain  $a$  and domain the product  $\text{Pi}(G)$  defined by

$$\begin{aligned} \text{Pi}(G) = & \text{if } G = \text{nil} \\ & \text{then } T \\ & \text{else if } G = [a] \text{ then } a \\ & \text{else } (\text{Pi}(\text{fst}(G)) \uparrow (\text{lst}(G))) \end{aligned}$$

where

$\text{fst}(G)$  is the sublist of  $G$  consisting of all except the last element,  
 $\text{lst}(G)$  is the last element of the list  $G$  .

Second, if  $D$  and  $G$  are lists of formulae, then  $\text{Pi}(DG)$  is canonically identified to  $\text{Pi}(D) \uparrow \text{Pi}(G)$ , using induction on the length of  $G$  :

i)  $G = \text{nil}$

$$\text{Pi}(D \text{ nil}) = \text{Pi}(D) \stackrel{=}{=} \text{Pi}(D) \uparrow T = \text{Pi}(D) \uparrow \text{Pi}(\text{nil})$$

ii)  $G = [a]$

Let  $D'$  be the list resulting from appending  $[a]$  to the list  $D$

$$\text{Pi}(D[a]) = \text{Pi}(D') = \text{Pi}(D) \uparrow a = \text{Pi}(D) \uparrow \text{Pi}([a])$$

where the proof of the middle step is from definition of  $\text{Pi}$ .

iii) Suppose that :

$$\text{Pi}(DG) \stackrel{=}{=} \text{Pi}(D) \uparrow \text{Pi}(G)$$

and let  $G'$  be the list resulting from prepending the list  $G$  to  $[b]$   
then

$$\begin{aligned} \text{Pi}(DG') &= \text{Pi}(DG) \uparrow b \\ &\stackrel{=}{=} (\text{Pi}(D) \uparrow \text{Pi}(G)) \uparrow \text{Pi}([b]) \\ &\stackrel{=}{=} \text{Pi}(D) \uparrow (\text{Pi}(G) \uparrow \text{Pi}([b])) \quad [\text{using alpha\_star}] \\ &\stackrel{=}{=} \text{Pi}(D) \uparrow \text{Pi}(G[b]) \\ &= \text{Pi}(D) \uparrow \text{Pi}(G') \end{aligned}$$

and this completes the proof by induction on the length of  $G$ .

Now,  $S$  is defined as follows :

- 1)  $S(A1(f)) = [ LA1(f) ]$  for  $f$  in  $X(A,B)$  ;
- 2)  $S(A2) = [ LA2(T) ]$  ;
- 3)  $S(R1(p1,p2,a)) = \text{comp}( S(p2), (( I1 \uparrow S(p1) ) \uparrow I2 ) )$  , where  $I1$  is the identity on  $\text{Pi}(D)$  ,  $I2$  is the identity on  $\text{Pi}(L)$ , and we make the canonical identifications of  $\text{Pi}(DGL)$  with  $( \text{Pi}(D) \uparrow \text{Pi}(G) ) \uparrow \text{Pi}(L)$  and of  $\text{Pi}(DaL)$  with  $( \text{Pi}(D) \uparrow a ) \uparrow \text{Pi}(L)$  ,  $D,G,a,L$  being as in the definition of  $R1(p1,p2,a)$  ;
- 4)  $S(R2(p,a)) = \text{comp}( S(p), ( \text{pi\_left}( \text{Pi}(G), a ) \uparrow I ) )$  , where  $I$  is the identity on  $\text{Pi}(D)$  , and we make the canonical identifications of  $\text{Pi}(GaD)$  with  $( \text{Pi}(G) \uparrow a ) \uparrow \text{Pi}(D)$  and of  $\text{Pi}(GD)$  with  $\text{Pi}(G) \uparrow \text{Pi}(D)$  ,  $G,a,D$  being as in the definition of  $R2(p,a)$  ;
- 5)  $S(R3(p,a)) = \text{comp}( \text{comp}( S(p), ( \text{alpha}( \text{Pi}(G), a, a ) \uparrow I2 ) ) , (( I1 \uparrow \text{delta}(a) ) \uparrow I2 ) )$  , where  $I1$  is the identity on  $\text{Pi}(G)$  ,  $I2$  is the identity on  $\text{Pi}(D)$  and we make the canonical identifications of  $\text{Pi}(Ga \uparrow aD)$  with  $( \text{Pi}(G) \uparrow ( a \uparrow a ) ) \uparrow \text{Pi}(D)$ ,  $\text{Pi}(GaD)$  with  $( \text{Pi}(G) \uparrow a ) \uparrow \text{Pi}(D)$  and of  $\text{Pi}(GaaD)$  with  $( ( \text{Pi}(G) \uparrow a ) \uparrow a ) \uparrow \text{Pi}(D)$ ,  $G,a,D$  being as in the definition of  $R3(p,a)$  ;
- 6)  $S(R4(p,b,a)) = \text{comp}( \text{comp}( \text{comp}( S(p), ( \text{alpha}( \text{Pi}(G), a, b ) \uparrow I2 ) ) , ( ( I1 \uparrow \text{sigma\_inverse}(a,b) ) \uparrow I2 ) ) , ( \text{alpha\_inverse}( \text{Pi}(G), b, a ) \uparrow I2 ) )$  , where  $I1$  is the identity on  $\text{Pi}(G)$  ,  $I2$  is the identity on  $\text{Pi}(D)$ ,  $\text{sigma\_inverse}$  is the inverse arrow for  $\text{sigma}$ ,  $\text{alpha\_inverse}$  is the inverse arrow for  $\text{alpha}$  and we make the canonical identifications of  $\text{Pi}(GbaD)$  with  $( ( \text{Pi}(G) \uparrow b ) \uparrow a ) \uparrow \text{Pi}(D)$ ,  $\text{Pi}(Gb \uparrow aD)$  with  $( \text{Pi}(G) \uparrow ( b \uparrow a ) ) \uparrow \text{Pi}(D)$ ,  $\text{Pi}(Ga \uparrow bD)$  with  $( \text{Pi}(G) \uparrow ( a \uparrow b ) ) \uparrow \text{Pi}(D)$  and of  $\text{Pi}(GabD)$  with  $( ( \text{Pi}(G) \uparrow a ) \uparrow b ) \uparrow \text{Pi}(D)$  ,  $G,b,a,D$  being as in the definition of  $R4(p,b,a)$  ;
- 7)  $S(R10(p1,p2,a \uparrow b)) = \text{comp}( ( S(p1) \uparrow S(p2) ) , \text{delta}( \text{Pi}(G) ) )$  , where we make the canonical identifications of  $\text{Pi}(GG)$  with  $\text{Pi}(G) \uparrow \text{Pi}(G)$  ,  $G$  being as in the definition of  $R10(p1,p2,a \uparrow b)$  ;

8)  $S( R11(p, a \uparrow b) ) = \text{comp}( S(p), ( \text{alpha}( \text{Pi}(G), a, b ) \uparrow I ) )$ , where  $I$  is the identity on  $\text{Pi}(D)$  and we make the canonical identifications of

$\text{Pi}(GabD)$  with  $( (\text{Pi}(G) \uparrow a) \uparrow b) \uparrow \text{Pi}(D)$  and of  
 $\text{Pi}(Ga \uparrow bD)$  with  $( \text{Pi}(G) \uparrow (a \uparrow b) ) \uparrow \text{Pi}(D)$   
 $G, a, b, D$  being as in the definition of  $R11(p, a \uparrow b)$ .

We give some examples for derivations of  $\text{Dc}(X)$  and their interpretations in  $\text{ArFc}(X)$  using the implementation.

Example 1

Let  $a, b, c, d$  be objects of the category  $X$ ,  $f$  in  $X(a, b)$ ,  $g$  in  $X(c, d)$ ; then the derivation

$$\frac{\frac{\frac{f}{a \rightarrow b}}{c, a \rightarrow b}}{\frac{c, a, c \rightarrow d}{c, c, a \rightarrow d}} \quad \frac{\frac{g}{c \rightarrow d}}{b, c \rightarrow d}}{c, a \rightarrow d}}{c \uparrow a \rightarrow d}$$

is interpreted in  $\text{ArFc}(X)$  as follows :

We insert the above proof into the system as described in (1.3.2) Example 1, then using the command "  $\text{Interpr}(Z, H)$  ", where  $Z$  represents the name of the last step of the proof and  $H$  will be instantiated as the arrow computed to interpret the derivation.

```

yes
| ?- theory(cart).
yes
| ?- prove(l = [f, [a] => [b]]).
yes
| ?- prove(l1 = [th(l, c), [c, a] => [b]]).
yes
| ?- prove(m = [g, [c] => [d]]).
yes
| ?- prove(m1 = [th(m, b), [b, c] => [d]]).
yes
| ?- prove(n = [cut(l1, m1, b), [c, a, c] => [d]]).
yes
| ?- prove(o = [inc(n, a, c), [a, c, c] => [d]]).
yes
| ?- prove(p = [con(o, c), [a, c] => [d]]).
yes
| ?- prove(q = [ala(p, a & c), [a & c] => [d]]).
yes
| ?- interpr(q, H).

H = comp(comp(comp(comp(g, pi_right(b, c)),
                        product(comp(f, pi_right(c, a)),
                                id(c))),
                        product(sigma_inverse(c, a), id(c))),
        alpha(a, c, c)),
    product(id(a), delta(c)))

yes
| ?-

```

Example 2

Suppose  $a, b$  are objects of the category  $X$ . Then the derivation

$$\frac{\frac{\frac{b \rightarrow b}{a, b \rightarrow b}}{a \uparrow b \rightarrow b} \quad \frac{\frac{a \rightarrow a}{a, b \rightarrow a}}{a \uparrow b \rightarrow a}}{a \uparrow b \rightarrow b \uparrow a}$$

is interpreted as follows :

```

yes
| ?- theory(cart).
yes
| ?- prove(l = [id(b), [b] => [b]]).
yes
| ?- prove(l1 = [th(l,a), [a,b] => [b]]).
yes
| ?- prove(l2 = [aia(l1,a & b), [a & b] => [b]]).
yes
| ?- prove(m = [id(a), [a] => [a]]).
yes
| ?- prove(m1 = [th(m,b), [a,b] => [a]]).
yes
| ?- prove(m2 = [aia(m1,a & b), [a & b] => [a]]).
yes
| ?- prove(n = [ais(l2,m2,b & a), [a & b] => [b & a]]).
yes
| ?- Interpr(n,H).

H = comp(product(pi_right(a,b),pi_left(a,b)),delta(a&b))

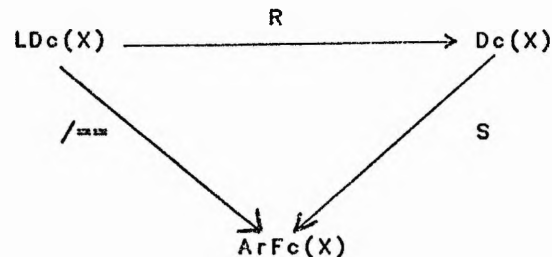
```

The following theorem asserts that the class  $Dc(X)$  of unlabelled derivations as defined in (1.3.1) is adequate in the sense that every arrow in the free cartesian category  $Fc(X)$  is representable by means of some derivation in  $Dc(X)$  :

(1.4.2) The Completeness Theorem for  $Dc(X)$ 

There is a function

$R : LDc(X) \dashrightarrow Dc(X)$ , so that



commutes.

Proof by induction on structure of  $F$  in  $LDc(X)$ , define  $R(F)$  by :  
note that  $I(A)$  below is defined by

$$I(A) = \begin{array}{l} \text{if } A \text{ is in } Ob(X) \\ \text{then } LA1(id(A)) \\ \text{else } LA2(A) \end{array}$$

- i) when  $F = LA1(f)$  for  $f$  in  $X(A,B)$ ,  $A,B$  in  $Ob(X)$ ,  
 define  $R(F) = A1(f)$   
 then  $S(R(F)) = S(A1(f)) = [LA1(f)] = [F]$
- ii) when  $F = LA2(A)$  for  $A$  in  $cL(X)$ , not an object of  $X$ , we use  
 Induction on structure of  $A$  :  
 $A$  is either  $T$  : put  $R(F) = A2$   
 then  $S(R(F)) = S(A2) = [LA2(T)] = [F]$   
or is  $B \uparrow C$  : put  
 $R(F) = R11(R10(R2(R(I(B))), C),$   
 $R2(R(I(C)), B), B \uparrow C), B \uparrow C)$
- iii) when  $F$  is  $LA3(A)$  for  $A$  in  $cL(X)$ , put  
 $R(F) = R2(A2, A)$
- iv) when  $F$  is  $LA4(A,B)$  for  $A,B$  in  $cL(X)$ , put  
 $R(F) = R11(R2(R(LA2(A)), B), A \uparrow B)$
- v) when  $F$  is  $LA5(A,B)$  for  $A,B$  in  $cL(X)$ , put  
 $R(F) = R11(R2(R(LA2(B)), A), A \uparrow B)$
- vi) when  $F$  is  $COMP(G,H,A)$ , put  
 $R(F) = R1(R(G), R(H), A)$
- vii) when  $F$  is  $ANGLE(G,H,A \uparrow B)$ , put  
 $R(F) = R10(R(G), R(H), A \uparrow B)$

We now illustrate the algorithm in this proof by showing how some canonical arrows of  $Fc(X)$  are represented as unlabelled derivations. these have been implemented in the system .

Example 1 For the derivation  $F = ANGLE(I(A), I(A), A \uparrow A)$   
 representing the arrow

$$\delta(A) : A \rightarrow A \uparrow A$$

the representation  $R(F)$  is

$$R10(R(LA2(A)), R(LA2(A)), A \uparrow A)$$

which can only be further simplified if we know the structure of  $A$ .

Example 2 For a derivation

$$F = ANGLE( ANGLE( LA4(A, B \uparrow C),$$

$$COMP( LA5(A, B \uparrow C), LA4(B, C), B \uparrow C ),$$

$$A \uparrow B ),$$

$$COMP( LA5(A, B \uparrow C), LA5(B, C), B \uparrow C ),$$

$$(A \uparrow B) \uparrow C ) )$$

representing the arrow

$\text{alpha}(A,B,C) : A \uparrow (B \uparrow C) \rightarrow (A \uparrow B) \uparrow C$   
 the representation  $R(F)$  is  
 $R10(R10(R11(R2(R(LA2(A)), B \uparrow C), A \uparrow (B \uparrow C)),$   
 $R1(R11(R2(R(LA2(B \uparrow C)), A), A \uparrow (B \uparrow C)),$   
 $R11(R2(R(LA2(B)), C), B \uparrow C),$   
 $B \uparrow C),$   
 $A \uparrow B),$   
 $R1(R11(R2(R(LA2(B \uparrow C)), A), A \uparrow (B \uparrow C)),$   
 $R11(R2(R(LA2(C)), B), B \uparrow C),$   
 $B \uparrow C),$   
 $(A \uparrow B) \uparrow C).$

### Example 3

For derivations  $F, G$  representing the arrows

$f : A \rightarrow B, g : C \rightarrow D$  respectively

then the derivation

$\text{PRODUCT}(F, G) = \text{ANGLE}(\text{COMP}(LA4(A, C), F, A),$   
 $\text{COMP}(LA5(A, C), G, C),$   
 $B \uparrow D)$

representing the arrow

$f \uparrow g : A \uparrow C \rightarrow B \uparrow D$

the representation  $R(\text{PRODUCT}(F, G))$  is

$R10(R1(R11(R2(R(LA2(A)), C), A \& C), R(F), A),$   
 $R1(R11(R2(R(LA2(C)), A), A \& C), R(G), C),$   
 $B \uparrow D)$

### Example 4

For the derivation

$F = \text{ANGLE}(LA5(A, B), LA4(A, B), B \uparrow A)$

representing the arrow

$\text{sigma}(A, B) : A \uparrow B \rightarrow B \uparrow A$

the representation  $R(F)$  is

$R10(R11(R2(R(LA2(B)), A), A \uparrow B),$   
 $R11(R2(R(LA2(A)), B), A \uparrow B),$   
 $B \uparrow A)$

For the derivation

$F = \text{ANGLE}(LA5(B, A), LA4(B, A), A \uparrow B)$

representing the arrow

$\text{sigma\_inverse}(A, B) : B \uparrow A \rightarrow A \uparrow B$

the representation  $R(F)$  is

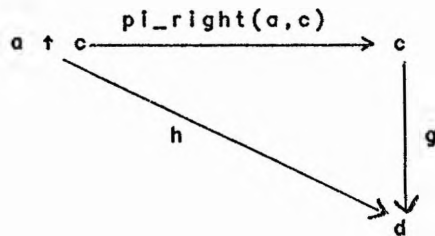
$R10(R11(R2(R(LA2(A)), B), B \uparrow A),$   
 $R11(R2(R(LA2(B)), A), B \uparrow A),$   
 $A \uparrow B)$

The following examples are sessions of input and output with the implementation to explain how to use the system to produce unlabelled derivations as representations for arrows of  $Fc(X)$ . First, the command "arrow(a,f,b)", where  $f$  in  $X(a,b)$  is used in the construction of the arrow in question, will store this arrow for later use. After inserting all arrows of the category  $X$  used in the construction of the arrow in question, we use the command "rep\_of(Y)" where  $Y$  represents the arrow to be represented. Then it will respond with the representation.

Example 5

For  $g$  in  $X(c,d)$ , the arrow  
 $h = \text{comp}(g, \text{pi\_right}(a,c)) : a \uparrow c \rightarrow d$

In the diagram



is represented by an unlabelled derivation as follows :

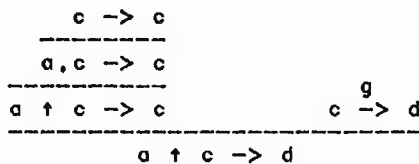
```

yes
| ?- arrow(c,g,d).
yes
| ?- rep_of(comp(g, pi_right(a,c))).

The arrow is :
=====
comp(g,pi_right(a,c))
representation :
=====
f5=[id(c),[c]=>[c]]
f4=[th(f5,a),[a,c]=>[c]]
f3=[aia(f4,a&c),[a&c]=>[c]]
f2=[g,[c]=>[d]]
f1=[cut(f3,f2,c),[a&c]=>[d]]

yes
| ?-
    
```

which in tree form is :

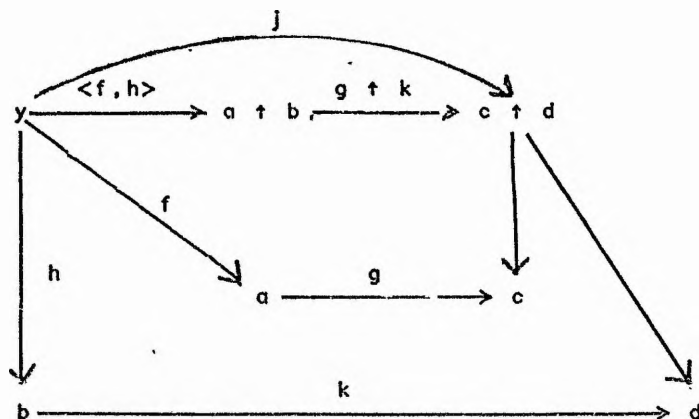


Example 6

For arrows  $f$  in  $X(y,a)$ ,  $g$  in  $X(a,c)$ ,  $h$  in  $X(y,b)$  and  $k$  in  $X(b,d)$ , where  $X$  is a category. Then the arrow

$J = \text{comp}(g \uparrow k, \langle f, h \rangle) : y \rightarrow c \uparrow d$ ,

In the diagram



is represented by an unlabelled derivation as follows :

```

| ?- theory(cart).
yes
| ?- arrow(y,f,a).
yes
| ?- arrow(a,g,c).
yes
| ?- arrow(y,h,b).
yes
| ?- arrow(b,k,d).
yes
| ?- rep_of(comp(product(g,k), angle(f,h))).

```

The arrow is :

```

=====
comp(product(g,k),angle(f,h))

```

representation :

```

=====
f14=[f,[y]=>[a]]
f15=[h,[y]=>[b]]
f13=[ais(f14,f15,a&b),[y]=>[a&b]]
f7=[id(a),[a]=>[a]]
f6=[th(f7,b),[a,b]=>[a]]
f5=[aia(f6,a&b),[a&b]=>[a]]
f3=[g,[a]=>[c]]
f8=[cut(f5,f3,a),[a&b]=>[c]]
f11=[id(b),[b]=>[b]]
f10=[th(f11,a),[a,b]=>[b]]
f9=[aia(f10,a&b),[a&b]=>[b]]
f4=[k,[b]=>[d]]
f12=[cut(f9,f4,b),[a&b]=>[d]]
f2=[ais(f8,f12,c&d),[a&b]=>[c&d]]
f1=[cut(f13,f2,a&b),[y]=>[c&d]]

```

which in tree form is :

$$\begin{array}{c}
 \frac{y \xrightarrow{f} a \quad y \xrightarrow{h} b}{y \rightarrow a \uparrow b} \\
 \frac{\frac{\frac{a \rightarrow a}{a, b \rightarrow a}}{a \uparrow b \rightarrow a} \quad \frac{\frac{b \rightarrow b}{a, b \rightarrow b}}{a \uparrow b \rightarrow b} \quad \frac{b \xrightarrow{k} d}{b \rightarrow d}}{a \uparrow b \rightarrow c \quad a \uparrow b \rightarrow d} \\
 \frac{a \uparrow b \rightarrow c \quad a \uparrow b \rightarrow d}{a \uparrow b \rightarrow c \uparrow d} \\
 \frac{a \uparrow b \rightarrow c \uparrow d}{y \rightarrow c \uparrow d}
 \end{array}$$

### (1.5) The Syntax of $Fc(X)$

The most important part of the theory of cartesian categories, for our implementation, is the syntax of  $Fc(X)$ . This is because the arrows in the free cartesian category can be represented by proofs with cuts; then these representations can be manipulated to eliminate cuts and obtain their normal representations.



(1.5.1) Interchange rule and its justification :

We observe in SZABO's algorithm for cut-elimination that in some clauses like [(C.3),(C.14),...], there is a step which has a contraction for say, G . However G may be of length greater than one and in this case, we must perform interchange between the objects of the sequence GG, before carrying out contraction to get the sequence G . This explains the inclusion of the inference rule ' interchange in the antecedent ' in the unlabelled ( cartesian ) derivations class, and this additional rule leads to changes in the definition of the normalization .

The following example is to justify the need for the rule of interchange in the antecedent :-

$$\text{alpha}(A,B,C) : A \uparrow (B \uparrow C) \text{ -----} \rightarrow (A \uparrow B) \uparrow C$$

and this arrow is canonical in the theory of cartesian categories .  
By the completeness theorem we get its equivalent unlabelled proof in the tree representation as :

$$\begin{array}{c}
 \frac{\frac{\frac{b \rightarrow b \quad c \rightarrow c}{b,c \rightarrow b \quad b,c \rightarrow c}}{b,c \rightarrow b \uparrow c}}{b \uparrow c \rightarrow b \uparrow c} \quad \frac{b \rightarrow b}{b,c \rightarrow b} \quad \frac{c \rightarrow c}{b,c \rightarrow c}}{\frac{a \rightarrow a \quad a,b \uparrow c \rightarrow b \uparrow c}{a,b \uparrow c \rightarrow a} \quad \frac{a \uparrow (b \uparrow c) \rightarrow b \uparrow c \quad b \uparrow c \rightarrow b}{a \uparrow (b \uparrow c) \rightarrow b} \quad \frac{b \rightarrow b \quad b,c \rightarrow b \uparrow c}{b \uparrow c \rightarrow b \uparrow c} \quad \frac{c \rightarrow c}{b,c \rightarrow c}}{\frac{a \uparrow (b \uparrow c) \rightarrow a \quad a \uparrow (b \uparrow c) \rightarrow b \uparrow c}{a \uparrow (b \uparrow c) \rightarrow a} \quad \frac{a \uparrow (b \uparrow c) \rightarrow b \uparrow c \quad b \uparrow c \rightarrow c}{a \uparrow (b \uparrow c) \rightarrow b \uparrow c} \quad \frac{a \uparrow (b \uparrow c) \rightarrow b \uparrow c \quad b \uparrow c \rightarrow c}{a \uparrow (b \uparrow c) \rightarrow b \uparrow c} \quad \frac{a \uparrow (b \uparrow c) \rightarrow b \uparrow c \quad b \uparrow c \rightarrow c}{a \uparrow (b \uparrow c) \rightarrow b \uparrow c}}{\frac{a \uparrow (b \uparrow c) \rightarrow (a \uparrow b)}{\frac{a \uparrow (b \uparrow c) \rightarrow (a \uparrow b) \quad a \uparrow (b \uparrow c) \rightarrow c}{a \uparrow (b \uparrow c) \rightarrow (a \uparrow b) \uparrow c}}
 \end{array}$$

Applying clause (C.42) from the cut-elimination algorithm described in Appendix (3) to the cut step in the above tree marked in a box, we get

$$\begin{array}{c}
 \frac{\frac{\frac{b \rightarrow b \quad c \rightarrow c}{b,c \rightarrow b \quad b,c \rightarrow c}}{b,c \rightarrow b \uparrow c}}{b \uparrow c \rightarrow b \uparrow c} \quad \frac{b \rightarrow b}{b,c \rightarrow b} \quad \frac{c \rightarrow c}{b,c \rightarrow c}}{\frac{a \rightarrow a \quad a,b \uparrow c \rightarrow b \uparrow c}{a,b \uparrow c \rightarrow a} \quad \frac{a \uparrow (b \uparrow c) \rightarrow b \uparrow c \quad b \uparrow c \rightarrow b}{a \uparrow (b \uparrow c) \rightarrow b} \quad \frac{b \rightarrow b \quad b,c \rightarrow b \uparrow c}{b \uparrow c \rightarrow b \uparrow c} \quad \frac{c \rightarrow c}{b,c \rightarrow c}}{\frac{a \uparrow (b \uparrow c) \rightarrow a \quad a \uparrow (b \uparrow c) \rightarrow b \uparrow c}{a \uparrow (b \uparrow c) \rightarrow a} \quad \frac{a \uparrow (b \uparrow c) \rightarrow b \uparrow c \quad b \uparrow c \rightarrow c}{a \uparrow (b \uparrow c) \rightarrow b \uparrow c} \quad \frac{a \uparrow (b \uparrow c) \rightarrow b \uparrow c \quad b \uparrow c \rightarrow c}{a \uparrow (b \uparrow c) \rightarrow b \uparrow c} \quad \frac{a \uparrow (b \uparrow c) \rightarrow b \uparrow c \quad b \uparrow c \rightarrow c}{a \uparrow (b \uparrow c) \rightarrow b \uparrow c}}{\frac{a \uparrow (b \uparrow c) \rightarrow (a \uparrow b)}{\frac{a \uparrow (b \uparrow c) \rightarrow (a \uparrow b) \quad a \uparrow (b \uparrow c) \rightarrow c}{a \uparrow (b \uparrow c) \rightarrow (a \uparrow b) \uparrow c}}
 \end{array}$$

Applying clause (C.34) from the cut\_elimination algorithm to the cut step marked in a box above, we get

$$\begin{array}{c}
 \frac{\frac{\frac{b \rightarrow b \quad c \rightarrow c}{b, c \rightarrow b \quad b, c \rightarrow c} \quad b \rightarrow b}{b, c \rightarrow b \quad b, c \rightarrow c} \quad \frac{b \rightarrow b \quad c \rightarrow c}{b, c \rightarrow b \quad b, c \rightarrow c}}{\frac{b \uparrow c \rightarrow b \uparrow c \quad b \uparrow c \rightarrow b}{b, c \rightarrow b \quad b, c \rightarrow c}} \\
 \frac{\frac{a \rightarrow a}{a, b \uparrow c \rightarrow a} \quad \frac{b \uparrow c \rightarrow b}{a, (b \uparrow c) \rightarrow b}}{\frac{a \uparrow (b \uparrow c) \rightarrow a}{a \uparrow (b \uparrow c) \rightarrow b}} \quad \frac{\frac{b \uparrow c \rightarrow b \uparrow c \quad c \rightarrow c}{a, b \uparrow c \rightarrow b \uparrow c} \quad \frac{c \rightarrow c}{b, c \rightarrow c}}{\frac{a \uparrow (b \uparrow c) \rightarrow b \uparrow c \quad b \uparrow c \rightarrow c}}{a \uparrow (b \uparrow c) \rightarrow c}} \\
 \frac{a \uparrow (b \uparrow c) \rightarrow a \quad a \uparrow (b \uparrow c) \rightarrow b}{a \uparrow (b \uparrow c) \rightarrow (a \uparrow b)} \quad \frac{a \uparrow (b \uparrow c) \rightarrow b \uparrow c \quad b \uparrow c \rightarrow c}{a \uparrow (b \uparrow c) \rightarrow c} \\
 \frac{a \uparrow (b \uparrow c) \rightarrow (a \uparrow b) \quad a \uparrow (b \uparrow c) \rightarrow c}{a \uparrow (b \uparrow c) \rightarrow (a \uparrow b) \uparrow c}
 \end{array}$$

Applying clause (C.42) we get

$$\begin{array}{c}
 \frac{\frac{\frac{b \rightarrow b \quad c \rightarrow c \quad b \rightarrow b}{b, c \rightarrow b \quad b, c \rightarrow c \quad b, c \rightarrow b} \quad \frac{b \rightarrow b \quad c \rightarrow c}{b, c \rightarrow b \quad b, c \rightarrow c}}{\frac{b, c \rightarrow b \uparrow c \quad b \uparrow c \rightarrow b}{b, c \rightarrow b \quad b, c \rightarrow c}} \\
 \frac{\frac{a \rightarrow a}{a, b \uparrow c \rightarrow a} \quad \frac{b \uparrow c \rightarrow b}{a, (b \uparrow c) \rightarrow b}}{\frac{a \uparrow (b \uparrow c) \rightarrow a}{a \uparrow (b \uparrow c) \rightarrow b}} \quad \frac{\frac{b \uparrow c \rightarrow b \uparrow c \quad c \rightarrow c}{a, b \uparrow c \rightarrow b \uparrow c} \quad \frac{c \rightarrow c}{b, c \rightarrow c}}{\frac{a \uparrow (b \uparrow c) \rightarrow b \uparrow c \quad b \uparrow c \rightarrow c}}{a \uparrow (b \uparrow c) \rightarrow c}} \\
 \frac{a \uparrow (b \uparrow c) \rightarrow a \quad a \uparrow (b \uparrow c) \rightarrow b}{a \uparrow (b \uparrow c) \rightarrow (a \uparrow b)} \quad \frac{a \uparrow (b \uparrow c) \rightarrow b \uparrow c \quad b \uparrow c \rightarrow c}{a \uparrow (b \uparrow c) \rightarrow c} \\
 \frac{a \uparrow (b \uparrow c) \rightarrow (a \uparrow b) \quad a \uparrow (b \uparrow c) \rightarrow c}{a \uparrow (b \uparrow c) \rightarrow (a \uparrow b) \uparrow c}
 \end{array}$$

Applying clause (C.14) to the cut step marked in a box in the above tree we get the following equivalent tree. This uses the rule 'interchange in the antecedent' for the objects b,c at the step indicated below.

$$\begin{array}{c}
 \frac{\frac{\frac{c \rightarrow c \quad b \rightarrow b}{b, c \rightarrow c \quad b, c \rightarrow b} \quad b \rightarrow b}{b, c \rightarrow b \quad b, b, c \rightarrow b}}{\frac{b, c, b, c \rightarrow b}{b, b, c, c \rightarrow b}} \\
 \frac{\frac{a \rightarrow a}{a, b \uparrow c \rightarrow a} \quad \frac{b \uparrow c \rightarrow b}{a, (b \uparrow c) \rightarrow b}}{\frac{a \uparrow (b \uparrow c) \rightarrow a}{a \uparrow (b \uparrow c) \rightarrow b}} \quad \frac{\frac{b \rightarrow b \quad c \rightarrow c}{b, c \rightarrow b \quad b, c \rightarrow c} \quad \frac{b \rightarrow b \quad c \rightarrow c}{b, c \rightarrow b \uparrow c}}{\frac{b \uparrow c \rightarrow b \uparrow c \quad c \rightarrow c}{a, b \uparrow c \rightarrow b \uparrow c} \quad \frac{c \rightarrow c}{b, c \rightarrow c}} \\
 \frac{a \uparrow (b \uparrow c) \rightarrow a \quad a \uparrow (b \uparrow c) \rightarrow b}{a \uparrow (b \uparrow c) \rightarrow (a \uparrow b)} \quad \frac{a \uparrow (b \uparrow c) \rightarrow b \uparrow c \quad b \uparrow c \rightarrow c}{a \uparrow (b \uparrow c) \rightarrow c} \\
 \frac{a \uparrow (b \uparrow c) \rightarrow (a \uparrow b) \quad a \uparrow (b \uparrow c) \rightarrow c}{a \uparrow (b \uparrow c) \rightarrow (a \uparrow b) \uparrow c}
 \end{array}$$

We continue the processes of the cut-elimination algorithm until, we get the following cut-free equivalent proof :

$$\begin{array}{c}
 \begin{array}{c}
 \frac{a \rightarrow a}{a, b \uparrow c \rightarrow a} \\
 \frac{a \uparrow (b \uparrow c) \rightarrow a}{a \uparrow (b \uparrow c) \rightarrow a}
 \end{array}
 \quad
 \begin{array}{c}
 \frac{b \rightarrow b}{b, c \rightarrow b} \\
 \frac{b, c \rightarrow b}{b, c, c \rightarrow b} \\
 \frac{b, c, c \rightarrow b}{b, c, b, c \rightarrow b} \\
 \frac{b, c, b, c \rightarrow b}{b, b, c, c \rightarrow b} \\
 \frac{b, b, c, c \rightarrow b}{b, c, c \rightarrow b} \\
 \frac{b, c, c \rightarrow b}{b, c \rightarrow b} \\
 \frac{b, c \rightarrow b}{b \uparrow c \rightarrow b} \\
 \frac{b \uparrow c \rightarrow b}{a, b \uparrow c \rightarrow b} \\
 \frac{a, b \uparrow c \rightarrow b}{a \uparrow (b \uparrow c) \rightarrow b}
 \end{array}
 \quad
 \begin{array}{c}
 \frac{c \rightarrow c}{b, c \rightarrow c} \\
 \frac{b, c \rightarrow c}{b, c, c \rightarrow c} \\
 \frac{b, c, c \rightarrow c}{b, c, b, c \rightarrow c} \\
 \frac{b, c, b, c \rightarrow c}{b, b, c, c \rightarrow c} \\
 \frac{b, b, c, c \rightarrow c}{b, c, c \rightarrow c} \\
 \frac{b, c, c \rightarrow c}{b, c \rightarrow c} \\
 \frac{b, c \rightarrow c}{b \uparrow c \rightarrow c} \\
 \frac{b \uparrow c \rightarrow c}{a, b \uparrow c \rightarrow c} \\
 \frac{a, b \uparrow c \rightarrow c}{a \uparrow (b \uparrow c) \rightarrow c}
 \end{array}
 \\
 \hline
 \frac{a \uparrow (b \uparrow c) \rightarrow a \quad a \uparrow (b \uparrow c) \rightarrow b \quad a \uparrow (b \uparrow c) \rightarrow c}{a \uparrow (b \uparrow c) \rightarrow a \uparrow b}
 \\
 \hline
 \frac{a \uparrow (b \uparrow c) \rightarrow a \uparrow b}{a \uparrow (b \uparrow c) \rightarrow (a \uparrow b) \uparrow c}
 \end{array}$$

(1.5.2) The Cut-Elimination Theorem for Dc(X)

Every p in Dc(X) is equivalent to a cut-free q in Dc(X) .

proof

By the clauses (C.1), (C.2.1) , (C.2.3) , (C.3) , (C.14) , (C.18), (C.19), (C.26), (C.27), (C.34), (C.35) and (C.42) as in SZABO [1978] , in addition to the clauses (C.20) and (C.36) needed because of the additional rule (R4) in the definition of Dc(X) , of the algorithm described in appendix (2), every derivation of Dc(X) containing a cut reduces to a cut-free one .

###

The following example is to illustrate the cut-elimination theorem and to explain how to use the system to eliminate cuts from derivations of Dc(X). As in (1.3.2) Example 1 we insert the proof, using the command " composite(-,-,-) " to insert all possible compositions for the arrows of the category X used in the construction of the derivation, where the third argument is the composition arrow of the arrows represented by the first and the second arguments of the command. Finally the command " cut\_free(Z) ", where Z represents the name of the last step in the inserted proof, will produce the equivalent cut-free one .

Example

Suppose a,b,c,e are objects of the category X; and that f in X(b,c), h in X(b,e) with comp(h,g) = k. Then the derivation

$$\begin{array}{c}
 \frac{a \rightarrow a}{a, b \rightarrow a} \qquad \frac{b \xrightarrow{g} c}{a, b \rightarrow c} \qquad \frac{c \xrightarrow{h} e}{a, c \rightarrow e} \\
 \hline
 a, b \rightarrow a \uparrow c \qquad \frac{a \uparrow c \rightarrow e}{d, a \uparrow c \rightarrow e} \\
 \hline
 d, a, b \rightarrow e
 \end{array}$$

is transformed to a cut-free one as follows :

```

yes
| ?- theory(cart).
yes
| ?- composite(h,g,k).
yes
| ?- prove(l = [id(a), [a] => [a]]).
yes
| ?- prove(m = [th(l,b), [a,b] => [a]]).
yes
| ?- prove(n = [g, [b] => [c]]).
yes
| ?- prove(o = [th(n,a), [a,b] => [c]]).
yes
| ?- prove(p = [ais(m,o,a & c), [a,b] => [a & c]]).
yes
| ?- prove(q = [h, [c] => [e]]).
yes
| ?- prove(r = [th(q,a), [a,c] => [e]]).
yes
| ?- prove(s = [aia(r,a & c), [a & c] => [e]]).
yes
| ?- prove(t = [th(s,d), [d,a & c] => [e]]).
yes
| ?- prove(u = [cut(p,t,a & c), [d,a,b] => [e]]).
yes
| ?- cut_free(u).

```

Proof is as follows :

```

=====
l=[id(a),[a]=>[a]]
m=[th(l,b),[a,b]=>[a]]
n=[g,[b]=>[c]]
o=[th(n,a),[a,b]=>[c]]
p=[ais(m,o,a&c),[a,b]=>[a&c]]
q=[h,[c]=>[e]]
r=[th(q,a),[a,c]=>[e]]
s=[aia(r,a&c),[a&c]=>[e]]
t=[th(s,d),[d,a&c]=>[e]]
u=[cut(p,t,a&c),[d,a,b]=>[e]]

```

cut\_free Proof is :

```

=====
f26=[k,[b]=>[e]]
f25=[th(f26,a),[a,b]=>[e]]
f22=[th(f25,b),[a,b,b]=>[e]]
f8=[th(f22,a),[a,b,a,b]=>[e]]
f11=[inc(f8,a,b),[a,a,b,b]=>[e]]
f12=[con(f11,a),[a,b,b]=>[e]]
f3=[con(f12,b),[a,b]=>[e]]
u=[th(f3,d),[d,a,b]=>[e]]

```

```

yes
| ?-

```

which in tree form is :

$$\begin{array}{c}
 \text{K} \\
 \frac{b \rightarrow e}{a, b \rightarrow e} \\
 \frac{a, b \rightarrow e}{a, b, b \rightarrow e} \\
 \frac{a, b, b \rightarrow e}{a, b, a, b \rightarrow e} \\
 \frac{a, b, a, b \rightarrow e}{a, a, b, b \rightarrow e} \\
 \frac{a, a, b, b \rightarrow e}{a, b, b \rightarrow e} \\
 \frac{a, b, b \rightarrow e}{a, b \rightarrow e} \\
 \frac{a, b \rightarrow e}{d, a, b \rightarrow e}
 \end{array}$$

(1.5.3) Lemma

If  $p$  derives  $G \rightarrow a$  and  $T$  is the only atomic subformula of  $a$ , then  $p$  can be reduced to the form

$$\frac{\frac{g}{\rightarrow a}}{G \rightarrow a} \quad (K)$$

where (K) denotes the instances of (R2) required to derive  $G \rightarrow a$  from  $\rightarrow a$ .

Proof :

The proof follows by induction on the construction of  $a$  :

i)  $a = T$

$$G \xrightarrow{P} T \quad > \quad \frac{\rightarrow T}{G \rightarrow T}$$

ii)  $a = c \uparrow d$

$$G \xrightarrow{P} c \uparrow d \quad > \quad \frac{\frac{\frac{p1}{\rightarrow c} \quad \frac{p2}{\rightarrow d}}{\rightarrow c \uparrow d}}{G \rightarrow c \uparrow d}$$

###

(1.5.4) Lemma

For every cut-free derivation  $p$  in  $Dc(X)$  there exists an equivalent cut-free  $q$  in  $Dc(X)$  containing no subderivations constructed with (R3) or (R4) and where the only subderivations constructed with (R2), if any, have atomic active formulae .

Proof

By the clauses (E.1) , (E.2) , (E.3) , (E.6) , (E.10) , (E.18) and (E.37) and the special cases of the clauses (E.9) and (E.17) in which  $F$  and  $E$  are empty sequences, of the algorithm described in appendix(3), every derivation of  $Dc(X)$  containing subderivations constructed with

(R3) or (R4) reduces to a contraction and interchange - free one .  
 And by the clause (E.41.1) of the same appendix, every derivation of  $Dc(X)$  containing subderivations constructed with R2 with active formulae of the form  $a \uparrow b$  reduces to a derivation where the active formulae of subderivations constructed with (R2) are atomic . These clauses preserve the equivalence relation between proofs as proved in SZABO.

##

The following examples are to illustrate the algorithm in appendix 3 for the above lemma. Insert a derivation of  $Dc(X)$  to the system in the same way as in (1.3.2) example 1, then use the command "con\_inc\_free(Z)", where Z represents the name of the last step of the inserted proof. The system responds with an equivalent contraction and interchange-free one where in each instance of the 'thinning in the antecedent' rule, if any, the active formula is atomic .

Example 1

The derivation ( from Example of 1.5.2 )

$$\begin{array}{r}
 \begin{array}{c}
 \text{b} \xrightarrow{\text{k}} \text{e} \\
 \hline
 \text{a, b} \xrightarrow{\quad} \text{e} \\
 \hline
 \text{a, b, b} \xrightarrow{\quad} \text{e} \\
 \hline
 \text{a, b, a, b} \xrightarrow{\quad} \text{e} \\
 \hline
 \text{a, a, b, b} \xrightarrow{\quad} \text{e} \\
 \hline
 \text{a, b, b} \xrightarrow{\quad} \text{e} \\
 \hline
 \text{a, b} \xrightarrow{\quad} \text{e} \\
 \hline
 \text{d, a, b} \xrightarrow{\quad} \text{e}
 \end{array}
 \end{array}$$

is transformed to a contraction and interchange free one as follows :-

```

yes
| ?- theory(cart).
yes
| ?- prove(l = [k, [b] => [e]]).
yes
| ?- prove(m = [th(l,a), [a,b] => [e]]).
yes
| ?- prove(n = [th(m,b), [a,b,b] => [e]]).
yes
| ?- prove(o = [th(n,a), [a,b,a,b] => [e]]).
yes
| ?- prove(p = [inc(o,a,b), [a,a,b,b] => [e]]).
yes
| ?- prove(q = [con(p,a), [a,b,b] => [e]]).
yes
| ?- prove(r = [con(q,b), [a,b] => [e]]).
yes
| ?- prove(u = [th(r,d), [d,a,b] => [e]]).
yes
| ?- con_inc_free(u).

```

Proof is as follows :

=====

l=[k,[b]=>[e]]

```

m = [th(l, a), [a, b] => [e]]
n = [th(m, b), [a, b, b] => [e]]
o = [th(n, a), [a, b, a, b] => [e]]
p = [inc(o, a, b), [a, a, b, b] => [e]]
q = [con(p, a), [a, b, b] => [e]]
r = [con(q, b), [a, b] => [e]]
u = [th(r, d), [d, a, b] => [e]]

```

contraction and interchange free proof is :

```

l = [k, [b] => [e]]
r = [th(l, a), [a, b] => [e]]
u = [th(r, d), [d, a, b] => [e]]

```

```

yes
| ?-

```

which in tree form is :

$$\begin{array}{c}
 \begin{array}{c}
 \begin{array}{c}
 b \xrightarrow{k} e \\
 \hline
 a, b \xrightarrow{\quad} e \\
 \hline
 d, a, b \xrightarrow{\quad} e
 \end{array}
 \end{array}
 \end{array}$$

### Example 2

Let  $a$  be an object of the category  $X$ ; then the derivation

$$\begin{array}{c}
 a \rightarrow a \\
 \hline
 a, b \uparrow c \rightarrow a \\
 \hline
 b \uparrow c, a \rightarrow a \\
 \hline
 (b \uparrow c) \uparrow a \rightarrow a
 \end{array}$$

uses the 'interchange' rule and has an occurrence of the 'thinning in the antecedent' rule with non-atomic active formula  $b \uparrow c$ . This can be transformed as follows :

```

yes
| ?- theory(cart).
yes
| ?- prove(l = [id(a), [a] => [a]]).
yes
| ?- prove(m = [th(l, b & c), [a, b & c] => [a]]).
yes
| ?- prove(n = [inc(m, b & c, a), [b & c, a] => [a]]).
yes
| ?- prove(o = [dia(n, (b & c) & a), [(b & c) & a] => [a]]).
yes
| ?- con_inc_free(o).

```

Proof is as follows :

```

l = [id(a), [a] => [a]]
m = [th(l, b & c), [a, b & c] => [a]]
n = [inc(m, b & c, a), [b & c, a] => [a]]
o = [dia(n, (b & c) & a), [(b & c) & a] => [a]]

```

• contraction and interchange free proof is :

```

=====
l=[id(a),[a]=>[a]]
f1=[th(l,c),[c,a]=>[a]]
m=[th(f1,b),[b,c,a]=>[a]]
n=[aia(m,b&c),[b&c,a]=>[a]]
o=[aia(n,(b&c)&a),[(b&c)&a]=>[a]]
=====

```

which in tree form is :

$$\begin{array}{c}
 a \rightarrow a \\
 \hline
 c, a \rightarrow a \\
 \hline
 b, c, a \rightarrow a \\
 \hline
 b \uparrow c, a \rightarrow a \\
 \hline
 (b \uparrow c) \uparrow a \rightarrow a
 \end{array}$$

#### (1.5.5) Definition

A derivation  $p$  in  $Dc(X)$  is normal if it satisfies the following conditions :-

- (1)  $p$  contains no subderivations constructed with (R1), (R3) or (R4).
- (2) If  $p$  derives  $G \rightarrow a$  and  $T$  is the only atomic subformula of  $a$ , then  $p$  is of the form

$$\frac{q \rightarrow a}{G \rightarrow a} \quad (K)$$

where (K) denotes the instances of (R2) required to derive  $G \rightarrow a$  from  $\rightarrow a$ .

- (3)  $p$  contains only subderivations constructed with (R2), if any, in which the active formulae are atomic.
- (4) Using  $\Leftarrow$  to denote 'priority', and saying  $R_i \Leftarrow R_j$  iff whenever a choice exists as to whether 'rule'  $R_i$  or  $R_j$  is used,  $R_i$  is used first:

for normality it is required that

$$(R_{10}) \Leftarrow (R_{11}) \Leftarrow (R_2) .$$

#### (1.5.6) The Normalization Theorem for $Dc(X)$

Every  $p$  in  $Dc(X)$  is equivalent to a unique normal  $q$  in  $Dc(X)$ .

Proof

By combining the proof of the Cut-Elimination theorem(1.5.2), lemma (1.5.3), lemma (1.5.4) and the following conditions of appendix (4) : (D.1), (D.8), (D.53) and the special cases of (D.7) and (D.49) with  $F$  and  $E$  are empty, as in SZABO, every  $p$  in  $Dc(X)$  reduces to a unique equivalent normal derivation  $q$  in  $Dc(X)$  .

###



Given a derivation  $p$  in  $Dc(X)$ , its normal form can be obtained by the use of our system as follows :

Insert the derivation  $p$  to the system in the same way as done in (1.5.2); instead of using the command " cut\_free(Z) " use the general command " normal(Z) " where  $Z$  represents the name of the last step in the inserted proof. The system starts by eliminating cuts and produces a cut-free one; then it continues to eliminate interchanges and contractions from the cut-free one and satisfy the condition on the active formulae of thinnings and produce an equivalent proof which is cut-free, interchange-free, contraction-free and with atomic thinning formulae. Finally the proof resulting from the above processes is transformed by the part of the implementation of the algorithm in appendix (4), to a normal proof.

Example :-

For objects  $a, b, c$  of the category  $X$ ; the derivation

$$\begin{array}{c}
 \frac{a \rightarrow a}{a, b \rightarrow a} \quad \frac{b \rightarrow b}{a, b \rightarrow b} \\
 \hline
 a, b \rightarrow a \uparrow b \\
 \hline
 a \uparrow b \rightarrow a \uparrow b \\
 \hline
 a \uparrow b, c \rightarrow a \uparrow b \\
 \hline
 (a \uparrow b) \uparrow c \rightarrow a \uparrow b \\
 \hline
 (a \uparrow b) \uparrow c \rightarrow a \\
 \hline
 (a \uparrow b) \uparrow c \rightarrow a \uparrow c
 \end{array}
 \qquad
 \begin{array}{c}
 \frac{a \rightarrow a}{a, b \rightarrow a} \\
 \hline
 a \uparrow b \rightarrow a \\
 \hline
 a \uparrow b, c \rightarrow a \\
 \hline
 (a \uparrow b) \uparrow c \rightarrow a \\
 \hline
 (a \uparrow b) \uparrow c \rightarrow a \uparrow c
 \end{array}
 \qquad
 \begin{array}{c}
 \frac{c \rightarrow c}{a \uparrow b, c \rightarrow c} \\
 \hline
 (a \uparrow b) \uparrow c \rightarrow c \\
 \hline
 (a \uparrow b) \uparrow c \rightarrow a \uparrow c
 \end{array}$$

is transformed to its equivalent normal form as follows :

```

yes
| ?- theory(cart).
yes
| ?- prove(l = [id(a), [a] => [a]]).
yes
| ?- prove(l1 = [th(l,b), [a,b] => [a]]).
yes
| ?- prove(m = [id(b), [b] => [b]]).
yes
| ?- prove(m1 = [th(m,a), [a,b] => [b]]).
yes
| ?- prove(n = [ais(l1,m1,a & b), [a,b] => [a & b]]).
yes
| ?- prove(o = [aia(n,a & b), [a & b] => [a & b]]).
yes
| ?- prove(p = [th(o,c), [a & b,c] => [a & b]]).
yes
| ?- prove(q = [aia(p,(a&b) & c), [(o&b) & c] => [a&b]]).
yes
| ?- prove(r = [id(a), [a] => [a]]).
yes
| ?- prove(r1 = [th(r,b), [a,b] => [a]]).
yes
| ?- prove(r2 = [aia(r1,a & b), [a & b] => [a]]).
yes
| ?- prove(s = [cut(q,r2,a & b), [(a & b) & c] => [a]]).
yes
| ?- prove(t = [id(c), [c] => [c]]).

```

```

yes
| ?- prove(t1 = [th(t, a & b), [a & b, c] => [c]]).
yes
| ?- prove(t2 = [aia(t1, (a&b) & c), [(a&b) & c] => [c]]).
yes
| ?- prove(u = [ais(s, t2, a & c), [(a & b) & c] => [a & c]]).
yes
| ?- normal(u).

```

Proof is as follows :

```

=====
l=[id(a), [a]=>[a]]
l1=[th(l, b), [a, b]=>[a]]
m=[id(b), [b]=>[b]]
m1=[th(m, a), [a, b]=>[b]]
n=[ais(l1, m1, a&b), [a, b]=>[a&b]]
o=[aia(n, a&b), [a&b]=>[a&b]]
p=[th(o, c), [a&b, c]=>[a&b]]
q=[aia(p, (a&b)&c), [(a&b)&c]=>[a&b]]
r=[id(a), [a]=>[a]]
r1=[th(r, b), [a, b]=>[a]]
r2=[aia(r1, a&b), [a&b]=>[a]]
s=[cut(q, r2, a&b), [(a&b)&c]=>[a]]
t=[id(c), [c]=>[c]]
t1=[th(t, a&b), [a&b, c]=>[c]]
t2=[aia(t1, (a&b)&c), [(a&b)&c]=>[c]]
u=[ais(s, t2, a&c), [(a&b)&c]=>[a&c]]

```

cut\_free Proof is :

```

=====
f32=[id(a), [a]=>[a]]
f29=[th(f32, b), [a, b]=>[a]]
f26=[th(f29, b), [a, b, b]=>[a]]
f14=[th(f26, a), [a, b, a, b]=>[a]]
f17=[inc(f14, a, b), [a, a, b, b]=>[a]]
f18=[con(f17, a), [a, b, b]=>[a]]
f9=[con(f18, b), [a, b]=>[a]]
f6=[aia(f9, a&b), [a&b]=>[a]]
f3=[th(f6, c), [a&b, c]=>[a]]
s=[aia(f3, (a&b)&c), [(a&b)&c]=>[a]]
t=[id(c), [c]=>[c]]
t1=[th(t, a&b), [a&b, c]=>[c]]
t2=[aia(t1, (a&b)&c), [(a&b)&c]=>[c]]
u=[ais(s, t2, a&c), [(a&b)&c]=>[a&c]]

```

contraction and interchange free proof is :

```

=====
f32=[id(a), [a]=>[a]]
f9=[th(f32, b), [a, b]=>[a]]
f6=[aia(f9, a&b), [a&b]=>[a]]
f3=[th(f6, c), [a&b, c]=>[a]]
s=[aia(f3, (a&b)&c), [(a&b)&c]=>[a]]
t=[id(c), [c]=>[c]]
f34=[th(t, b), [b, c]=>[c]]
f33=[th(f34, a), [a, b, c]=>[c]]
t1=[aia(f33, a&b), [a&b, c]=>[c]]
t2=[aia(t1, (a&b)&c), [(a&b)&c]=>[c]]
u=[ais(s, t2, a&c), [(a&b)&c]=>[a&c]]

```

Normal proof is :

```

=====
f32=[id(a), [a]=>[a]]
f9=[th(f32, b), [a, b]=>[a]]
f6=[aia(f9, a&b), [a&b]=>[a]]
f3=[th(f6, c), [a&b, c]=>[a]]
t=[id(c), [c]=>[c]]
f34=[th(t, b), [b, c]=>[c]]
f33=[th(f34, a), [a, b, c]=>[c]]
t1=[aia(f33, a&b), [a&b, c]=>[c]]
s=[ais(f3, t1, a&c), [a&b, c]=>[a&c]]
u=[aia(s, (a&b)&c), [(a&b)&c]=>[a&c]]

```

```

yes
| ?-

```

which in tree form is :

$$\begin{array}{c}
 \frac{a \rightarrow a}{a, b \rightarrow a} \\
 \frac{a \uparrow b \rightarrow a}{a \uparrow b, c \rightarrow a} \\
 \hline
 (a \uparrow b) \uparrow c \rightarrow a \uparrow c
 \end{array}
 \qquad
 \begin{array}{c}
 \frac{c \rightarrow c}{b, c \rightarrow c} \\
 \frac{a, b, c \rightarrow c}{a \uparrow b, c \rightarrow c} \\
 \hline
 a \uparrow b, c \rightarrow c
 \end{array}$$

(1.5.7) The CHURCH-ROSSER Theorem for Dc(X)

If  $p = q$ , then there exists a normal  $r$  in  $Dc(X)$  such that  $p \geq r$  and  $q \geq r$ .

(1.6) Application to Category Theory

In this section we give some examples of the operation of our implementation based on arrows in the theory of cartesian categories . Proofs in  $Dc(X)$  correspond to these arrows; their cut-free equivalents, contraction and interchange-free equivalents and their normal forms are produced. We also give some examples to prove the commutativity of diagrams in the theory of cartesian categories using the implementation.

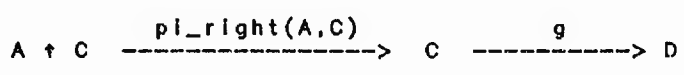
The command " norm\_rep(X) " where X represents arrow from  $Fc(X)$ , is used to find the normal representation of this arrow. It will start with constructing the representing proof for this arrow. Finally it works on this proof and produces the normal proof representing the given arrow, as follows :

Example 1

The normal representation of the arrow

$$h = \text{comp}(g, \text{pi\_right}(a,c))$$

in the digram



where  $g$  in  $X(c,d)$ , can be obtained as follows :

```

*-----*
| yes
| | ?- theory(cart).
| yes
| | ?- arrow(c,g,d).
| yes
| | ?- norm_rep(comp(g, pi_right(a,c))).
|
*-----*

```

The arrow is :

```
=====
comp(g,pl_right(a,c))
```

Proof is as follows :

```
=====
f5=[id(c).[c]=>[c]]
f4=[th(f5,a).[a,c]=>[c]]
f3=[aia(f4,a&c).[a&c]=>[c]]
f2=[g.[c]=>[d]]
f1=[cut(f3,f2,c).[a&c]=>[d]]
```

cut\_free Proof is :

```
=====
f11=[g.[c]=>[d]]
f8=[th(f11,a).[a,c]=>[d]]
f1=[aia(f8,a&c).[a&c]=>[d]]
```

contraction and interchange free proof is :

```
=====
f11=[g.[c]=>[d]]
f8=[th(f11,a).[a,c]=>[d]]
f1=[aia(f8,a&c).[a&c]=>[d]]
```

Normal proof is :

```
=====
f11=[g.[c]=>[d]]
f8=[th(f11,a).[a,c]=>[d]]
f1=[aia(f8,a&c).[a&c]=>[d]]
```

```
yes
| ?-
```

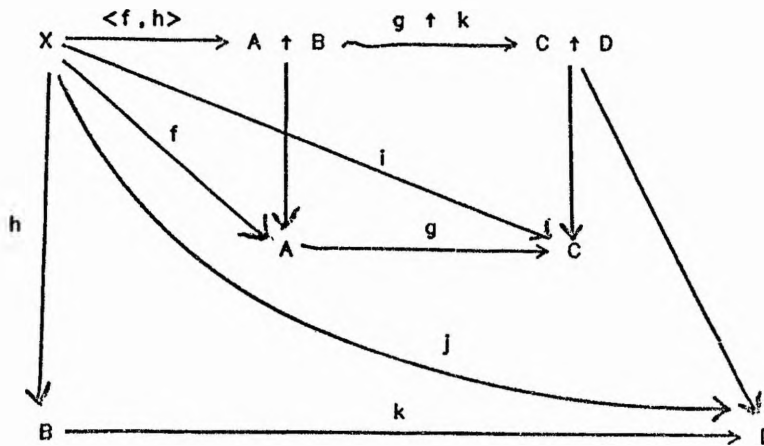
which in tree form is :

$$\frac{\frac{c \xrightarrow{g} d}{a, c \rightarrow d}}{a \uparrow c \rightarrow d}$$

The following is a session with the system to prove commutativity of diagrams in the theory of cartesian categories, in other words to prove the equality of arrows by proving the equality of their normal representations. The command "commutative" is to be used for this purpose; after typing this command the system waits to read the LHS arrow of the equality in question and then responds by computing its normal representation as in the above example. The system waits again to read the RHS arrow of the equality and responds by computing its normal representation. Finally it compares these two normal representations and responds with the conclusion as follows :

Example 2

In the following diagram



$$\langle gf, kh \rangle = (g \uparrow k) . \langle f, h \rangle$$

and this can be proved by the use of the system as follows :

```

yes
| ?- theory(cart).
yes
| ?- arrow(x,f,a).
yes
| ?- arrow(x,h,b).
yes
| ?- arrow(x,i,c).
yes
| ?- arrow(x,j,d).
yes
| ?- arrow(a,g,c).
yes
| ?- arrow(b,k,d).
yes
| ?- composite(g,f,i).
yes
| ?- composite(k,h,j).
yes
| ?- commutative.
Insert the first arrow :angle(comp(g,f), comp(k,h)).

```

The arrow is :

=====

angle(comp(g,f),comp(k,h))

Representation :

=====

```

f4=[f,[x]=>[a]]
f3=[g,[a]=>[c]]
f2=[cut(f4,f3,a),[x]=>[c]]
f7=[h,[x]=>[b]]
f6=[k,[b]=>[d]]
f5=[cut(f7,f6,b),[x]=>[d]]
f1=[ais(f2,f5,c&d),[x]=>[c&d]]

```

Cut\_free proof is :

=====

```

f2=[i,[x]=>[c]]
f5=[j,[x]=>[d]]
f1=[ais(f2,f5,c&d),[x]=>[c&d]]

```

Contraction and Interchange free proof is :

=====

```

f2=[i,[x]=>[c]]

```

```
f5=[j,[x]=>[d]]
f1=[ais(f2,f5,c&d),[x]=>[c&d]]
```

Normal proof is :

```
f2=[i,[x]=>[c]]
f5=[j,[x]=>[d]]
f1=[ais(f2,f5,c&d),[x]=>[c&d]]
```

Insert the second arrow :  $\text{comp}(\text{product}(g,k), \text{angle}(f,h))$ .

The arrow is :

```
comp(product(g,k),angle(f,h))
```

Representation :

```
f21=[f,[x]=>[a]]
f22=[h,[x]=>[b]]
f20=[ais(f21,f22,a&b),[x]=>[a&b]]
f14=[id(a),[a]=>[a]]
f13=[th(f14,b),[a,b]=>[a]]
f12=[ala(f13,a&b),[a&b]=>[a]]
f10=[g,[a]=>[c]]
f15=[cut(f12,f10,a),[a&b]=>[c]]
f18=[id(b),[b]=>[b]]
f17=[th(f18,a),[a,b]=>[b]]
f16=[ala(f17,a&b),[a&b]=>[b]]
f11=[k,[b]=>[d]]
f19=[cut(f16,f11,b),[a&b]=>[d]]
f9=[ais(f15,f19,c&d),[a&b]=>[c&d]]
f8=[cut(f20,f9,a&b),[x]=>[c&d]]
```

Cut\_free proof is :

```
f51=[i,[x]=>[c]]
f45=[th(f51,x),[x,x]=>[c]]
f37=[con(f45,x),[x]=>[c]]
f62=[j,[x]=>[d]]
f56=[th(f62,x),[x,x]=>[d]]
f40=[con(f56,x),[x]=>[d]]
f8=[ais(f37,f40,c&d),[x]=>[c&d]]
```

contraction and interchange free proof is :

```
f37=[i,[x]=>[c]]
f40=[j,[x]=>[d]]
f8=[ais(f37,f40,c&d),[x]=>[c&d]]
```

Normal proof is :

```
f37=[i,[x]=>[c]]
f40=[j,[x]=>[d]]
f8=[ais(f37,f40,c&d),[x]=>[c&d]]
```

The answer is :

```
angle(comp(g,f),comp(k,h)) =
comp(product(g,k),angle(f,h))
```

because they have the same normal representation .

```
yes
| ?-
```

## CHAPTER 2

## BICARTESIAN CATEGORY THEORY

In this chapter, we extend the representation of the class of small cartesian categories in chapter 1 to represent the class of small bicartesian categories. We extend the implementation of the theory of cartesian categories, in order to handle the extension of the algorithms for representing arrows of the free bicartesian category  $Fbc(X)$  over  $X$  by means of derivations from the class  $Dbc(X)$  of unlabelled 'bicartesian-derivations' over  $X$ ; interpreting derivations of this class as arrows of  $Fbc(X)$ , and the others for cut-eliminations and normal form (2.5).

This chapter is divided in a similar way to that of the previous chapter. We tidy up some obscurities; these lie essentially in the definition of the unlabelled deductive system relevant to the theory of bicartesian categories. The need for the "interchange in the succedent" rule in order to perform large contractions will be discussed in more detail in (2.5.1).

As a consequence of adding rule (R7) "interchange in the succedent", we add more clauses to the proof of the cut-elimination theorem (2.5.2). Corollaries (2.5.6) and (2.5.7), concerning the normal representation for the terminal and the initial arrows in  $Fbc(X)$ , were stated without proof in SZABO; we have constructed and implemented their proofs in the system. As in the previous chapter, we modified lemma (2.5.8), which was stated without proof in SZABO, to transform all the subderivations constructed by (R4) and (R7) to their equivalent interchange-free ones; the proof has been constructed and implemented in the system.

## (2.1) Definition

A bicartesian category is a cartesian category  $C$  with the following additional structure :-

- (4) A bifunctor  $(-) \vee (-) : C \times C \rightarrow C$ .
- (5) A distinguished object  $\perp \in \text{Ob } C$ .
- (6) Two adjunction  $\alpha_\sigma$  and  $\alpha_\iota$ , where

$$\alpha_\sigma = \{ \alpha_\sigma(A, B, C) : C(A \vee B, C) \rightarrow C(A, C) \times C(B, C) \in \text{ArEns} \\ | A, B, C \in \text{Ob } \underline{C} \} .$$

and

$$\alpha_\perp = \{ \alpha_\perp(A) : C(\perp, A) \rightarrow \{*\} \in \text{ArEns} | A \in \text{Ob } \underline{C} \} .$$

## (2.2) The Free Bicartesian Category Fbc(X)

Let  $X$  be a fixed but arbitrary small category. The free bicartesian category Fbc(X) is constructed as follows :-

### (2.2.1) Definition

The class  $bcL(X)$  of (bicartesian) formulae over  $X$  is the smallest class satisfying the conditions of Definition (1.2.1) and the following additional conditions :-

- v)  $\perp$  is a formula ;
- vi)  $A \vee B$  is a formula when  $A, B$  are ;
- vii) The class  $bcL(X)$  is generated by i) to vi) .

### (2.2.2) The Labelled Derivations LDbc(X)

The class of (bicartesian) labelled derivations  $LDbc(X)$  over  $X$  is defined inductively by (i) - (viii) of (1.2.2) and the following:

- (ix) For  $A$  in  $bcL(X)$  ,  
 $LA6(A)$  is a derivation ,  
 with conclusion  $\tau_{\text{star}}(A) : \perp \rightarrow A$  ;
- (x) For  $A, B$  in  $bcL(X)$  ,  
 $LA7(A, B)$  is a derivation ,  
 with conclusion  $\pi_{\text{left\_star}}(A, B) : A \rightarrow A \vee B$  ;
- (xi) For  $A, B$  in  $bcL(X)$  ,  
 $LA8(A, B)$  is a derivation ,  
 with conclusion  $\pi_{\text{right\_star}}(A, B) : B \rightarrow A \vee B$  ;
- (xii) For derivations  $F, G$  with conclusions  $f : A \rightarrow C$  and  $g : B \rightarrow C$  respectively  
 $SQUARE(F, G, A \vee B)$  is a derivation  
 with conclusion  $[f, g] : A \vee B \rightarrow C$  .

### (2.2.3) Definition

The relation  $\equiv$  is defined as the smallest equivalence relation on  $LDbc(X)$  satisfying the conditions from (i) to (viii) of definition (1.2.3) and the following additional conditions :-



- ix) If  $F \equiv G$  and  $H \equiv K$  with conclusions  $f, g : A \rightarrow C$ ,  
 $h, k : B \rightarrow C$   
then  $\text{SQUARE}( F, H, A \vee B ) \equiv \text{SQUARE}( G, K, A \vee B )$  ;
- x) For  $F, G$  with conclusions  $f : A \rightarrow C, g : B \rightarrow C$  respectively  
 $\text{COMP}( \text{LA7}(A, B), \text{SQUARE}( F, G, A \vee B ), A \vee B ) \equiv F$  and  
 $\text{COMP}( \text{LA8}(A, B), \text{SQUARE}( F, G, A \vee B ), A \vee B ) \equiv G$  ;
- xi) For  $F$  with conclusion  $f : A \vee B \rightarrow C$ ,  
 $\text{SQUARE}( \text{COMP}( \text{LA7}( A, B ), F, A \vee B ),$   
 $\text{COMP}( \text{LA8}( A, B ), F, A \vee B ),$   
 $A \vee B ) \equiv F$  ;
- xii) For  $F$  with conclusion  $f : \perp \rightarrow A$ ,  
 $F \equiv \text{LA6}( A )$  .

#### (2.2.4) Definition

We define  $\text{Fbc}(X)$  to be the category having :

- i) as objects, the bicartesian formulae over  $X$  ;
- ii) as arrows, the  $\equiv$ -classes of  $\text{LDbc}(X)$  ;
- iii) to xi) as in the Definition (1.2.5) of  $\text{Fc}(X)$  ;
- xii) as  $\perp$ , the formula  $\perp$  ;
- xiii) as  $\text{tau\_star}$ , the function defined by  
 $\text{tau\_star}( A ) = [ \text{LA6}( A ) ]$  ;
- xiv) as  $\text{pi\_left\_star}$ , the function defined by  
 $\text{pi\_left\_star}( A, B ) = [ \text{LA7}( A, B ) ]$  ;
- xv) as  $\text{pi\_right\_star}$ , the function defined by  
 $\text{pi\_right\_star}( A, B ) = [ \text{LA8}( A, B ) ]$  ;
- xvi) as  $[ -, - ]$ , the function defined by  
 $[[F], [G]] = [\text{SQUARE}(F, G, \text{dom}([F]) \vee \text{dom}([G]))]$   
when  $\text{cod}( [F] ) = \text{cod}( [G] )$  .

It is now routine to check that, with this structure,  $\text{Fbc}(X)$  is a category, is bicartesian, and is free bicartesian over  $X$ . There is the obvious embedding  $X \rightarrow \text{Fbc}(X)$  defined by  $A \rightarrow A$ ;  $f \rightarrow [\text{LA1}(f)]$ .

#### (2.3) The Unlabelled Derivations $\text{Dbc}(X)$

We now extend the class  $\text{Dc}(X)$  of unlabelled cartesian derivations to the class  $\text{Dbc}(X)$  of unlabelled bicartesian derivations to represent the arrows of  $\text{Fbc}(X)$  .

(2.3.1) Definition

The class  $Dbc(X)$  of unlabelled ( bicartesian ) derivations over a category  $X$  is defined inductively as follows :-

Note : we use the same notations for formulae , lists , sequents and derivations as in (1.3) .

- i) If  $a, b$  are objects of  $X$  , and if  $f$  is in  $X(a, b)$  , then  $A1(f)$  is a derivation with conclusion  $a \xrightarrow{f} b$  ;
- ii)  $A2$  is a derivation with conclusion  $[] \rightarrow T$  ;
- iii)  $A3$  is a derivation with conclusion  $\perp \rightarrow []$  ;
- iv) ( cut )
  - a) if  $p1, p2$  are derivations with conclusions  $G \rightarrow a, DaL \rightarrow F$ , then  
 $R1(p1, p2, a)$  is a derivation with conclusion  $DGL \rightarrow F$
  - b) if  $p1, p2$  are derivations with conclusions  $G \rightarrow FaE, a \rightarrow H$ , then  
 $R1(p1, p2, a)$  is a derivation with conclusion  $G \rightarrow FHE$  ;
- v) ( thinning in the antecedent )
 

If  $p$  is a derivation with conclusion  $GD \rightarrow F$  , then  
 $R2(p, a)$  is a derivation with conclusion  $GaD \rightarrow F$  ;
- vi) ( contraction in the antecedent )
 

If  $p$  is a derivation with conclusion  $GaaD \rightarrow F$  , then  
 $R3(p, a)$  is a derivation with conclusion  $GaD \rightarrow F$  ;
- vii) ( interchange in the antecedent )
 

If  $p$  is a derivation with conclusion  $GabD \rightarrow F$  , then  
 $R4(p, b, a)$  is a derivation with conclusion  $GbaD \rightarrow F$  ;
- viii) ( thinning in the succedent )
 

If  $p$  is a derivation with conclusion  $G \rightarrow FE$  , then  
 $R5(p, a)$  is a derivation with conclusion  $G \rightarrow FaE$  ;
- ix) ( contraction in the succedent )
 

If  $p$  is a derivation with conclusion  $G \rightarrow FaaE$  , then  
 $R6(p, a)$  is a derivation with conclusion  $G \rightarrow FaE$  ;
- x) ( interchange in the succedent )
 

If  $p$  is a derivation with conclusion  $G \rightarrow FabE$  , then  
 $R7(p, b, a)$  is a derivation with conclusion  $G \rightarrow FbaE$  ;

- x1) ( and Introduction in the succedent )  
 If  $p_1, p_2$  are derivations with conclusions  $G \rightarrow a, G \rightarrow b$ , then  
 $R10(p_1, p_2, a \uparrow b)$  is a derivation with conclusion  $G \rightarrow (a \uparrow b)$ ;
- xii) ( and Introduction in the antecedent )  
 If  $p$  is a derivation with conclusion  $G \uparrow b \rightarrow F$ , then  
 $R11(p, a \uparrow b)$  is a derivation with conclusion  $G(a \uparrow b) \rightarrow F$ ;
- xiii) ( or Introduction in the antecedent )  
 If  $p_1, p_2$  are derivations with conclusions  $a \rightarrow F, b \rightarrow F$ , then  
 $R12(p_1, p_2, a \vee b)$  is a derivation with conclusion  $(a \vee b) \rightarrow F$ ;
- xiv) ( or Introduction in the succedent )  
 If  $p$  is a derivation with conclusion  $G \rightarrow F \vee b$ , then  
 $R13(p, a \vee b)$  is a derivation with conclusion  $G \rightarrow F(a \vee b)$ ;
- xv) The class of unlabelled derivations  $Dbc(X)$  is generated by  
 (i) to (xiv) .

### (2.3.2) Representation of $Dbc(X)$ in Prolog

We use the technique mentioned in (1.3.2) to represent the unlabelled derivations  $Dbc(X)$  in the Prolog. We extend the abbreviations used in (1.3.2) by the following :-

```
#      for v
bo      for 1 in A3
bottom  for the name of the constant in A3
th      for the name of the operation in R2 and R5 ( thinning )
con     for the name of the operation in R3 and R6 ( contraction )
inc     for the name of the operation in R4 and R7 ( interchange )
ois     for the name of the operation in R12 ( or introduction
                                               in the succedent )
oia     for the name of the operation in R13 ( or introduction
                                               in the antecedent )
```

i) to viii) similar to (1.3.2)

ix) A3 is represented by

```
g1 = [bottom, [bo] => []]
```

x)  $R5(p, a)$  is represented by a step-collection with the last step,

```
g2 = [th(g1, a), G => FaE]
```

where  $g1$  is the name of the last step in the step-collection for the derivation  $p$  .



```

g8 = [ais(g6,g7,c&d), [a] => [c&d]]
g9 = [i, [b] => [c]]
g10 = [j, [b] => [d]]
g11 = [ais(g9,g10,c&d), [b] => [c&d]]
g12 = [oia(g8,g11,a#b), [a#b] => [c&d]]
g13 = [cut(g5,g12,a#b), [x,y] => [c&d,z]]

```

and the use of the system to check its correctness as follows :

```

yes
| ?- theory(bicart).
yes
| ?- prove(g1 = [f, [x] => [a]]).
yes
| ?- prove(g2 = [th(g1,y), [x,y] => [a]]).
yes
| ?- prove(g3 = [th(g2,b), [x,y] => [a,b]]).
yes
| ?- prove(g4 = [th(g3,z), [x,y] => [a,b,z]]).
yes
| ?- prove(g5 = [ois(g4,a # b), [x,y] => [a # b,z]]).
yes
| ?- prove(g6 = [g, [a] => [c]]).
yes
| ?- prove(g7 = [h, [a] => [d]]).
yes
| ?- prove(g8 = [ais(g6,g7,c & d), [a] => [c & d]]).
yes
| ?- prove(g9 = [i, [b] => [c]]).
yes
| ?- prove(g10 = [j, [b] => [d]]).
yes
| ?- prove(g11 = [ais(g9,g10,c & d), [b] => [c & d]]).
yes
| ?- prove(g12 = [oia(g8,g11,a # b), [a # b] => [c & d]]).
yes
| ?- prove(g13 = [cut(g5,g12,a # b), [x,y] => [c & d,z]]).
yes
| ?-

```

#### (2.4) The Semantics of Dbc(X)

In this section, we extend the definition of the interpretation function in (1.4) to interpret the derivations of Dbc(X) as arrows of Fbc(X) and prove the completeness theorem. In order to define this interpretation, we extend the canonical arrows in (1.4) by the following ones of Fbc(X), determined by  $\alpha_{\sigma}$  :-

iv)  $\delta_{\text{star}}(A) : A \vee A \rightarrow A$  for all  $A$  in  $\text{ObFbc}(X)$ , where

$$\alpha_{\sigma}^{-1}(\langle \text{Id}(A), \text{Id}(A) \rangle) = \delta_{\text{star}}(A).$$

v)  $\alpha_{\text{star}}(A,B,C) : A \vee (B \vee C) \rightarrow (A \vee B) \vee C$

for all  $A, B, C$  in  $\text{ObFbc}(X)$ , where

$$\alpha_{\sigma}^{-1}(\text{comp}(\text{pi\_left\_star}(A \vee B,C), \text{pi\_left\_star}(A,B))) ,$$

$$\alpha_{\sigma}^{-1}(\text{comp}(\text{pi\_left\_star}(A \vee B,C), \text{pi\_right\_star}(A,B))) ,$$

$$\text{pi\_right\_star}(A \vee B,C)) = \alpha_{\text{star}}(A,B,C)$$

vi)  $\text{sigma\_star}(A,B) : A \vee B \rightarrow B \vee A$  for all  $A,B$  in  $\text{ObFbc}(X)$ , where  
 $\alpha_G^{-1} ( \langle \text{pi\_right\_star}(B,A), \text{pi\_left\_star}(B,A) \rangle )$   
 $= \text{sigma\_star}(A,B)$  .

(2.4.1) Definition

We define a function  $S$  from the class  $\text{Dbc}(X)$  of unlabelled derivations to  $\text{ArFbc}(X)$  satisfying conditions (1) - (8) of (1.4.1) and the following additional conditions :

Note that if a derivation  $p$  has conclusion  $G \rightarrow F$ , then  $S(p)$  is an arrow with domain  $\text{Pi}(G)$  ( where  $\text{Pi}$  as in (1.4.1) ) and codomain  $\text{Un}(F)$  defined by

$$\text{Un}( F ) = \begin{array}{l} \text{if } F = \text{nil} \\ \text{then } T \\ \text{else if } F = [a] \text{ then } a \\ \text{else } ( \text{Un}( \text{fst}(F) ) \vee \text{lst}(F) ) \end{array}$$

where

$\text{fst}(F)$  and  $\text{lst}(F)$  similar to that in (1.4.1) .

- 9)  $S( A3 ) = [ \text{LA2}(1) ]$  ;
- 10)  $S( R1(p1,p2,a) ) = \text{comp}( ( I1 \vee S(p2) ) \vee I2 , S(p1) )$  , where  
 $I1$  is the identity on  $\text{Un}(F)$  ,  
 $I2$  is the identity on  $\text{Un}(E)$ , and we make the canonical identifications of  
 $\text{Un}(FHE)$  with  $( \text{Un}(F) \vee \text{Un}(H) ) \vee \text{Un}(E)$  and of  
 $\text{Un}(FaE)$  with  $( \text{Un}(F) \vee a ) \vee \text{Un}(E)$  ,  
 $F,H,a,E$  being as in part(b) of the definition of  $R1(p1,p2,a)$  ;
- 11)  $S( R5(p,a) ) = \text{comp}( ( \text{pi\_left\_star}(\text{Un}(F),a) \vee I ) , S(p) )$  ,  
 where  $I$  is the identity on  $E$  , and we make the canonical identifications of  
 $\text{Un}(FaE)$  with  $( \text{Un}(F) \vee a ) \vee \text{Un}(E)$  and of  
 $\text{Un}(FE)$  with  $\text{Un}(F) \vee \text{Un}(E)$  ,  
 $F,a,E$  being as in the definition of  $R5(p,a)$  ;
- 12)  $S( R6(p,a) ) = \text{comp}( \text{comp}( ( I1 \vee \text{delta\_star}(a) ) \vee I2 ) , \text{alpha\_star\_inverse}(\text{Un}(F),a,a) \vee I2 ) , S(p) )$  , where  
 $I1$  is the identity on  $\text{Un}(F)$  ,  
 $I2$  is the identity on  $\text{Un}(E)$  ,  
 $\text{alpha\_star\_inverse}$  is the inverse of  $\text{alpha\_star}$   
 and we make the canonical identifications of

$Un(FaE)$  with  $(Un(F) \vee a) \vee Un(E)$  ,  
 $Un(Fa \ aE)$  with  $(Un(F) \vee (a \vee a)) \vee Un(E)$  and of  
 $Un(FaaE)$  with  $((Un(F) \vee a) \vee a) \vee Un(E)$  ,  
 $F, a, E$  being as in the definition of  $R6(p, a)$  ;

13)  $S(R7(p, b, a)) = \text{comp}(\text{comp}(\text{comp}(\alpha\_star(Un(F), b, a) \vee I2 ,$   
 $(I1 \vee \sigma\_star(a, b)) \vee I2),$   
 $\alpha\_star\_inverse(Un(F), a, b) \vee I2),$   
 $S(p))$  , where

$I1$  is the identity on  $Un(F)$  ,  
 $I2$  is the identity on  $Un(E)$  and we make  
 the canonical identifications of  
 $Un(FbaE)$  with  $((Un(F) \vee b) \vee a) \vee Un(E)$  ,  
 $Un(Fb \vee aE)$  with  $(Un(F) \vee (b \vee a)) \vee Un(E)$  ,  
 $Un(Fa \vee bE)$  with  $(Un(F) \vee (a \vee b)) \vee Un(E)$  and of  
 $Un(FabE)$  with  $((Un(F) \vee a) \vee b) \vee Un(E)$  ,  
 $F, a, b, E$  being as in the definition of  $R7(p, b, a)$  .

14)  $S(R12(p1, p2, a \vee b)) = \text{comp}(\delta\_star(Un(F)) , S(p1) \vee S(p2))$  ,  
 we make the canonical identifications of  
 $Un(FF)$  with  $Un(F) \vee Un(F)$  ,  
 $F$  being as in the definition of  $R12(p1, p2, a \vee b)$  ;

15)  $S(R13(p, a \vee b)) = \text{comp}(\alpha\_star\_inverse(Un(F), a, b) \vee I, S(p))$  ,  
 where  $I$  is the identity on  $Un(E)$  and we  
 make the canonical identifications of  
 $Un(Fa \vee bE)$  with  $(Un(F) \vee (a \vee b)) \vee Un(E)$  and of  
 $Un(FabE)$  with  $((Un(F) \vee a) \vee b) \vee Un(E)$  ,  
 $F, a, b, G$  being as in the definition of  $R13(p, a \vee b)$  .

The following is an example for a derivation of  $Dbc(X)$ , and its interpretation in  $ArFbc(X)$  by the use of the implementation .

#### Example

For arrows  $f$  in  $X(a, c)$ ,  $g$  in  $X(c, a)$ , and objects  $a, b, c$  of the category  $X$ , the derivation

$$\begin{array}{c}
 \begin{array}{ccc}
 c \rightarrow c & a \xrightarrow{f} c & \\
 \hline
 c \vee a \rightarrow c & & 
 \end{array}
 &
 \begin{array}{ccc}
 c \xrightarrow{g} a & a \rightarrow a & \\
 \hline
 c \vee a \rightarrow a & & 
 \end{array}
 \\
 \hline
 c \vee a \rightarrow c \vee a & & \\
 \hline
 c \vee a \rightarrow c \vee a & & \\
 \hline
 c \vee a \rightarrow c \vee a, c & & \\
 \hline
 c \vee a \rightarrow (c \vee a) \vee c & & 
 \end{array}$$

is interpreted in  $\text{ArFbc}(X)$  as follows :

```

yes
| ?- theory(bicart).
yes
| ?- prove(l = [id(c), [c] => [c]]).
yes
| ?- prove(m = [f, [a] => [c]]).
yes
| ?- prove(n = [ola(l,m,c # a), [c # a] => [c]]).
yes
| ?- prove(o = [g, [c] => [a]]).
yes
| ?- prove(p = [id(a), [a] => [a]]).
yes
| ?- prove(q = [ola(o,p,c # a), [c # a] => [a]]).
yes
| ?- prove(r = [ais(n,q,c & a), [c # a] => [c & a]]).
yes
| ?- prove(s = [th(r,c), [c # a] => [c & a,c]]).
yes
| ?- prove(t = [ols(s,(c&a) # c), [c#a] => [(c & a) # c]]).
yes
| ?- interpr(t,H).

H = comp(pi_left_star(c&a,c),
         comp(product(comp(delta_star(c),union(id(c),f)),
                       comp(delta_star(a),union(g,id(a))))),
         delta(c#a)))

yes
| ?-

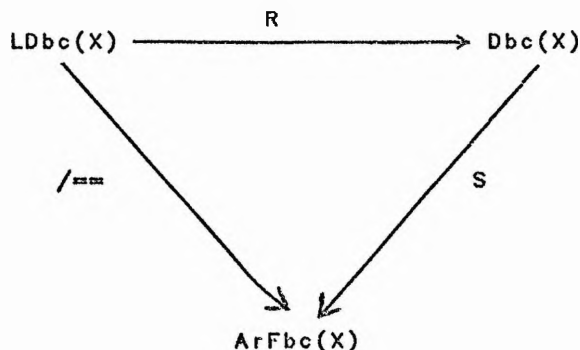
```

The following theorem proves that the class  $\text{Dbc}(X)$  constructed for the theory of bicartesian categories is adequate, in the sense that every arrow in the free bicartesian category  $\text{Fbc}(X)$  can be represented by means of some derivation in  $\text{Dbc}(X)$  :

#### (2.4.2) The Completeness Theorem for $\text{Dbc}(X)$

There is a function

$R : \text{LDbc}(X) \longrightarrow \text{Dbc}(X)$ , so that



commutes.



Proof

We define a function  $R$  to satisfy conditions (i) - (vii) of (1.4.2) and the following

Note that  $I(A)$  in this proof and the following examples is as defined in (1.4.2) .

viii) when  $F = LA2(A)$  for  $A$  in  $bcL(X)$ , not an object of  $X$ , we use induction on structure of  $A$  :

A is either  $T$  as in (ii) of (1.5.2)  
or  $B \uparrow C$   
or  $\perp$  : put  $R(F) = A3$   
then  $S(R(F)) = S(A3) = [LA2(\perp)] = [F]$   
or  $B \vee C$  : put  
 $R(F) = R13( R12( R5( R(I(B)), C ),$   
 $R5( R(I(C)), B ),$   
 $B \vee C )$   
 $B \vee C )$

ix) when  $F$  is  $LA6(A)$  for  $A$  in  $bcL(X)$ , put  
 $R(F) = R5(A3, A)$

x) when  $F$  is  $LA7(A,B)$  for  $A,B$  in  $bcL(X)$ , put  
 $R(F) = R13( R5( R(I(A)), B ), A \vee B )$

xi) when  $F$  is  $LA8(A,B)$  for  $A,B$  in  $bcL(X)$ , put  
 $R(F) = R13( R5( R(I(B)), A ), A \vee B )$

xii) when  $F$  is  $SQUARE(G,H,A \vee B)$ , put  
 $R(F) = R12( R(G), R(H), A \vee B )$

###

We give some examples to illustrate the algorithm in this proof by showing how some arrows of  $Fbc(X)$  are representable by means of unlabelled (bicartesian) derivations. ( These particular arrows, and their representations are built into the system for the user's convenience ).

Example 1

For the derivation

$$F = SQUARE( I(A), I(A), A \vee A )$$

representing the arrow

$$\delta_{\text{star}}(A) : A \vee A \rightarrow A ,$$

the representation  $R(F)$  is

$$R12( R( I(A) ) , R( I(A) ) , A \vee A )$$

Example 2

For a derivation

$$F = \text{SQUARE}(\text{COMP}(\text{LA7}(A \vee B, C), \text{LA7}(A, B), A \vee B), \\ \text{SQUARE}(\text{COMP}(\text{LA7}(A \vee B, C), \text{LA8}(A, B), A \vee B), \\ \text{LA8}(A \vee B, C), \\ B \vee C), \\ A \vee (B \vee C)))$$

representing the arrow

$$\alpha_{\text{star}}(A, B, C) : A \vee (B \vee C) \rightarrow (A \vee B) \vee C$$

the representation  $R(F)$  is

$$\begin{aligned} &R12(R1(R13(R5(R(I(A \vee B))), C), (A \vee B) \vee C), \\ &R13(R5(R(I(A))), B), A \vee B), \\ &A \vee B), \\ &R12(R1(R13(R5(R(I(A \vee B))), C), (A \vee B) \vee C), \\ &R13(R5(R(I(B))), A), A \vee B), \\ &A \vee B), \\ &R13(R5(R(I(C))), A \vee B), (A \vee B) \vee C), \\ &B \vee C), \\ &A \vee (B \vee C)) \end{aligned}$$
Example 3

For derivations  $F, G$  representing the arrows

$$f : A \rightarrow B, \quad g : C \rightarrow D \quad \text{respectively}$$

then for the derivation

$$\text{UNION}(F, G) = \text{SQUARE}(\text{COMP}(F, \text{LA7}(B, D), B), \\ \text{COMP}(G, \text{LA8}(B, D), D), \\ A \vee C)$$

representing the arrow

$$f \vee g : A \vee C \rightarrow B \vee D$$

the representation  $R(\text{UNION}(F, G))$  is

$$\begin{aligned} &R12(R1(R(F)), \\ &R13(R5(R(I(B))), D), B \vee D), \\ &B), \\ &R1(R(G)), \\ &R13(R5(R(I(D))), B), B \vee D), \\ &D), \\ &A \vee C) \end{aligned}$$

Example 4

For the derivation

$$F = \text{SQUARE}( \text{LA8}(B,A), \text{LA7}(B,A), A \vee B )$$

representing the arrow

$$\text{sigma\_star}(A,B) : A \vee B \rightarrow B \vee A$$

the representation  $R(F)$  is

$$\begin{aligned} &R12( R13( R5( R( I(A) ), B), B \vee A), \\ &R13( R5( R( I(B) ), A), B \vee A), \\ &A \vee B ) \end{aligned}$$

For the derivation

$$F = \text{SQUARE}( \text{LA8}(A,B), \text{LA7}(A,B), B \vee A )$$

representing the arrow

$$\text{sigma\_star\_inverse}(A,B) : B \vee A \rightarrow A \vee B$$

the representation  $R(F)$  is

$$\begin{aligned} &R12( R13( R5( R( I(B) ), A), A \vee B), \\ &R13( R5( R( I(A) ), B), A \vee B), \\ &B \vee A ) \end{aligned}$$

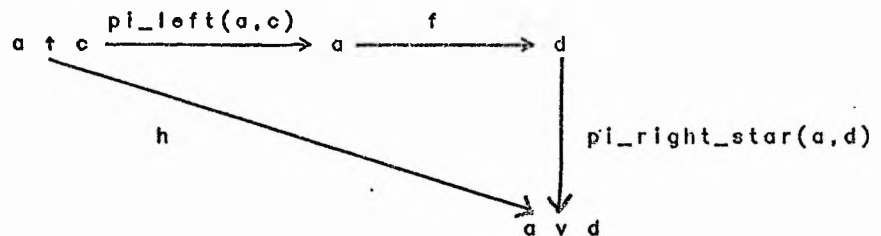
The following example is a session of input and output with the system to represent an arrow of  $\text{Fbc}(X)$  by means of some unlabelled bicartesian derivation :

Example 5

For  $f$  in  $X(a,d)$ , the arrow

$$h = \text{comp}(\text{comp}(\text{pi\_right\_star}(a,d),f),\text{pi\_left}(a,c)) : a \uparrow c \rightarrow a \vee d$$

in the diagram



is represented by an unlabelled derivation as follows :

```

yes
| ?- theory(bicart).
yes
| ?- arrow(a,f,d).
yes
| ?- rep_of(comp(comp(pi_right_star(a,d),f),pi_left(a,c))).

```

The arrow is

```

=====
comp(comp(pi_right_star(a,d),f),pi_left(a,c))

```

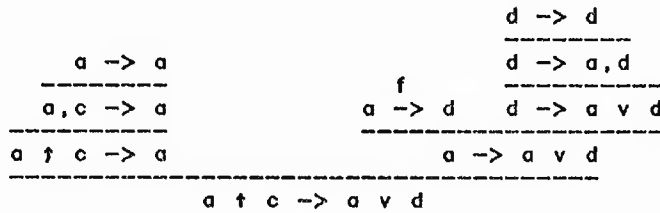
Representation :-

=====

```
f9=[id(a),[a]=>[a]]
f8=[th(f9,c),[a,c]=>[a]]
f7=[aia(f8,a&c),[a&c]=>[a]]
f6=[f,[a]=>[d]]
f5=[id(d),[d]=>[d]]
f4=[th(f5,a),[d]=>[a,d]]
f3=[ois(f4,a#d),[d]=>[a#d]]
f2=[cut(f6,f3,d),[a]=>[a#d]]
f1=[cut(f7,f2,a),[a&c]=>[a#d]]
```

yes  
| ?-

which in tree form is :



(2.5) The Syntax of Fbc(X)

=====

After representing the arrows of the free bicartesian category Fbc(X) by unlabelled derivations with cuts, we are able to manipulate these derivations to eliminate cuts and obtain their normal representations.

(2.5.1) Interchange in the succedent rule and its justification

Likewise as in (1.5.1), we observe in SZABO'S algorithm for cut-elimination that in some clauses like [ (C.4), (C.15),... ], there is a big contraction say for, H since H may be of length greater than one. In this case we must perform interchanges first between the objects of the sequence HH and then using the contraction rule (R6) to obtain H. This explains the inclusion of the inference rule 'interchange in the succedent' in the definition of the class Dbc(X) of unlabelled (bicartesian) derivations. As a result of the inclusion of this rule, there are a few changes in the definition of the normalization.

The following example is to justify the need for the rule 'interchange in the succedent' :-

$$\alpha\_star(A,B,C) : A \vee (B \vee C) \longrightarrow (A \vee B) \vee C$$

and this is a canonical arrow of the free bicartesian category Fbc(X). Using the completeness theorem, its unlabelled representation derivation is as follows :



$$\begin{array}{c}
 \frac{a \rightarrow a}{a \rightarrow a, b} \quad \frac{a \rightarrow a}{a \rightarrow a, b} \quad \frac{b \rightarrow b}{b \rightarrow a, b} \\
 \hline
 \frac{a \rightarrow a \vee b \quad a \vee b \rightarrow a, b}{a \rightarrow a, b} \\
 \hline
 \frac{a \rightarrow a \vee b}{a \rightarrow a \vee b, c} \\
 \hline
 \frac{a \rightarrow (a \vee b) \vee c}{a \vee (b \vee c) \rightarrow (a \vee b) \vee c}
 \end{array}
 \qquad
 \begin{array}{c}
 \frac{a \rightarrow a}{a \rightarrow a, b} \quad \frac{b \rightarrow b}{b \rightarrow a, b} \\
 \hline
 \frac{a \vee b \rightarrow a, b}{a \vee b \rightarrow a \vee b} \\
 \hline
 \frac{b \rightarrow b \quad a \vee b \rightarrow a \vee b}{b \rightarrow a, b \quad a \vee b \rightarrow a \vee b, c} \\
 \hline
 \frac{b \rightarrow a \vee b \quad a \vee b \rightarrow (a \vee b) \vee c}{b \rightarrow (a \vee b) \vee c} \\
 \hline
 \frac{b \vee c \rightarrow (a \vee b) \vee c}{a \vee (b \vee c) \rightarrow (a \vee b) \vee c}
 \end{array}$$

Applying clause (C.15) to the cut in the above marked box , we get the following equivalent tree. This uses the rule ' interchange in the succedent ' for the objects a,b at the step indicated bellow .

$$\begin{array}{c}
 \frac{a \rightarrow a}{a \rightarrow a, b} \quad \frac{a \rightarrow a}{a \rightarrow a, b} \quad \frac{b \rightarrow b}{b \rightarrow a, b} \\
 \hline
 \frac{a \rightarrow a, b, b \quad b \rightarrow a, b}{a \rightarrow a, b, a, b} \\
 \hline
 \frac{a \rightarrow a, b, b}{a \rightarrow a, b, b} \\
 \hline
 \frac{a \rightarrow a, b}{a \rightarrow a, b} \\
 \hline
 \frac{a \rightarrow a \vee b}{a \rightarrow a \vee b, c} \\
 \hline
 \frac{a \rightarrow (a \vee b) \vee c}{a \vee (b \vee c) \rightarrow (a \vee b) \vee c}
 \end{array}
 \qquad
 \begin{array}{c}
 \frac{a \rightarrow a}{a \rightarrow a, b} \quad \frac{b \rightarrow b}{b \rightarrow a, b} \\
 \hline
 \frac{a \vee b \rightarrow a, b}{a \vee b \rightarrow a \vee b} \\
 \hline
 \frac{b \rightarrow b \quad a \vee b \rightarrow a \vee b}{b \rightarrow a, b \quad a \vee b \rightarrow a \vee b, c} \\
 \hline
 \frac{b \rightarrow a \vee b \quad a \vee b \rightarrow (a \vee b) \vee c}{b \rightarrow (a \vee b) \vee c} \\
 \hline
 \frac{b \vee c \rightarrow (a \vee b) \vee c}{a \vee (b \vee c) \rightarrow (a \vee b) \vee c}
 \end{array}$$

We continue the processes of the cut-elimination algorithm until , we get the following equivalent cut-free proof :

$$\begin{array}{c}
 \frac{a \rightarrow a}{a \rightarrow a, b} \\
 \hline
 \frac{a \rightarrow a, b}{a \rightarrow a, b, b} \\
 \hline
 \frac{a \rightarrow a, b, b}{a \rightarrow a, b, a, b} \\
 \hline
 \frac{a \rightarrow a, b, a, b}{a \rightarrow a, a, b, b} \\
 \hline
 \frac{a \rightarrow a, a, b, b}{a \rightarrow a, b, b} \\
 \hline
 \frac{a \rightarrow a, b, b}{a \rightarrow a, b} \\
 \hline
 \frac{a \rightarrow a, b}{a \rightarrow a \vee b} \\
 \hline
 \frac{a \rightarrow a \vee b}{a \rightarrow a \vee b, c} \\
 \hline
 \frac{a \rightarrow a \vee b, c}{a \rightarrow (a \vee b) \vee c} \\
 \hline
 \frac{a \rightarrow (a \vee b) \vee c}{a \vee (b \vee c) \rightarrow (a \vee b) \vee c}
 \end{array}
 \qquad
 \begin{array}{c}
 \frac{b \rightarrow b}{b \rightarrow a, b} \\
 \hline
 \frac{b \rightarrow a, b}{b \rightarrow a, b, b} \\
 \hline
 \frac{b \rightarrow a, b, b}{b \rightarrow a, b, a, b} \\
 \hline
 \frac{b \rightarrow a, b, a, b}{b \rightarrow a, a, b, b} \\
 \hline
 \frac{b \rightarrow a, a, b, b}{b \rightarrow a, b, b} \\
 \hline
 \frac{b \rightarrow a, b, b}{b \rightarrow a, b} \\
 \hline
 \frac{b \rightarrow a, b}{b \rightarrow a \vee b} \\
 \hline
 \frac{b \rightarrow a \vee b}{b \rightarrow a \vee b, c} \\
 \hline
 \frac{b \rightarrow a \vee b, c}{b \rightarrow (a \vee b) \vee c} \\
 \hline
 \frac{b \rightarrow (a \vee b) \vee c}{b \vee c \rightarrow (a \vee b) \vee c} \\
 \hline
 \frac{b \vee c \rightarrow (a \vee b) \vee c}{a \vee (b \vee c) \rightarrow (a \vee b) \vee c}
 \end{array}$$

(2.5.2) The Cut-Elimination Theorem for Dbc(X)

Every  $p$  in  $\text{Dbc}(X)$  is equivalent to a cut-free  $q$  in  $\text{Dbc}(X)$ .

Proof

By Theorem (1.5.2), together with the following clauses of the algorithm in appendix (2):

(C.4), (C.7), (C.8), (C.9), (C.12), (C.15), (C.21), (C.22), (C.29), (C.37), (C.38), (C.43) and (C.44) as in SZABO [1978], in addition to clauses (C.23) and (C.39) needed because of the additional rule (R7) in the definition of  $\text{Dbc}(X)$ , there is an equivalent cut-free derivation for every derivation of the class  $\text{Dbc}(X)$  of unlabelled derivations.

###

The following example is to illustrate the above cut-elimination theorem:

Example:

For  $f$  in  $X(a,c)$ ,  $g$  in  $X(d,b)$ , where  $a,b,c,d$  are objects of a category  $X$ , the derivation

$$\begin{array}{c}
 \frac{a \rightarrow a}{a, d \rightarrow a} \quad \frac{d \rightarrow d}{a, d \rightarrow d} \quad \frac{b \rightarrow b}{b \rightarrow b, a} \quad \frac{a \rightarrow a \quad b \rightarrow b}{a \rightarrow a, b \quad b \rightarrow a, b} \\
 \frac{a \uparrow d \rightarrow a \quad a \xrightarrow{f} c}{a \uparrow d \rightarrow c} \quad \frac{a \uparrow d \rightarrow d \quad d \xrightarrow{g} b}{a \uparrow d \rightarrow b} \quad \frac{c, b \rightarrow b}{c \uparrow b \rightarrow b} \quad \frac{b \rightarrow b, a \quad a \rightarrow a, b}{b \rightarrow a \vee b} \quad \frac{a \vee b \rightarrow a, b}{a \vee b \rightarrow a \vee b} \\
 \frac{a \uparrow d \rightarrow c \quad a \uparrow d \rightarrow b}{a \uparrow d \rightarrow c \uparrow b} \quad \frac{c \uparrow b \rightarrow b \quad b \rightarrow a \vee b}{c \uparrow b \rightarrow (a \vee b) \vee c} \\
 \frac{a \uparrow d \rightarrow c \uparrow b \quad c \uparrow b \rightarrow (a \vee b) \vee c}{a \uparrow d \rightarrow (a \vee b) \vee c}
 \end{array}$$

is transformed to a cut-free one as follows:

```

yes
| ?- theory(bicart).
yes
| ?- prove(s1 = [id(a), [a] => [a]]).
yes
| ?- prove(s2 = [th(s1,d), [a,d] => [a]]).
yes
| ?- prove(s3 = [ala(s2,a & d), [a & d] => [a]]).
yes
| ?- prove(s4 = [f, [a] => [c]]).
yes
| ?- prove(s5 = [cut(s3,s4,a), [a & d] => [c]]).
yes
| ?- prove(s6 = [id(d), [d] => [d]]).
yes
| ?- prove(s7 = [th(s6,a), [a,d] => [d]]).
yes
| ?- prove(s8 = [ala(s7,a & d), [a & d] => [d]]).

```

```

yes
| ?- prove(s9 = [g, [d] => [b]]).
yes
| ?- prove(s10 = [cut(s8,s9,d), [a & d] => [b]]).
yes
| ?- prove(s11 = [ais(s5,s10,c & b), [a & d] => [c & b]]).
yes
| ?- prove(s12 = [id(b), [b] => [b]]).
yes
| ?- prove(s13 = [th(s12,c), [c,b] => [b]]).
yes
| ?- prove(s14 = [aia(s13,c & b), [c & b] => [b]]).
yes
| ?- prove(s15 = [id(b), [b] => [b]]).
yes
| ?- prove(s16 = [th(s15,a), [b] => [a,b]]).
yes
| ?- prove(s17 = [ois(s16,a # b), [b] => [a # b]]).
yes
| ?- prove(s18 = [d(a), [a] => [a]]).
yes
| ?- prove(s19 = [th(s18,b), [a] => [a,b]]).
yes
| ?- prove(s20 = [id(b), [b] => [b]]).
yes
| ?- prove(s21 = [th(s20,a), [b] => [a,b]]).
yes
| ?- prove(s22 = [oia(s19,s21,a # b), [a # b] => [a,b]]).
yes
| ?- prove(s23 = [ois(s22,a # b), [a # b] => [a # b]]).
yes
| ?- prove(s24 = [th(s23,c), [a # b] => [a # b,c]]).
yes
| ?- prove(s25 = [ois(s24,(a#b)#c), [a#b] => [(a#b)#c]]).
yes
| ?- prove(s26 = [cut(s17,s25,a # b), [b] => [(a#b)#c]]).
yes
| ?- prove(s27 = [cut(s14,s26,b), [c&b] => [(a#b)#c]]).
yes
| ?- prove(s28 = [cut(s11,s27,c & b), [a&d] => [(a#b)#c]]).
yes
| ?- cut_free(s28).

```

Proof is as follows :

=====

```

s1=[id(a),[a]=>[a]]
s2=[th(s1,d),[a,d]=>[a]]
s3=[aia(s2,a&d),[a&d]=>[a]]
s4=[f,[a]=>[c]]
s5=[cut(s3,s4,a),[a&d]=>[c]]
s6=[id(d),[d]=>[d]]
s7=[th(s6,a),[a,d]=>[d]]
s8=[aia(s7,a&d),[a&d]=>[d]]
s9=[g,[d]=>[b]]
s10=[cut(s8,s9,d),[a&d]=>[b]]
s11=[ais(s5,s10,c&b),[c&d]=>[c&b]]
s12=[id(b),[b]=>[b]]
s13=[th(s12,c),[c,b]=>[b]]
s14=[aia(s13,c&b),[c&b]=>[b]]
s15=[id(b),[b]=>[b]]
s16=[th(s15,a),[b]=>[a,b]]
s17=[ois(s16,a#b),[b]=>[a#b]]
s18=[id(a),[a]=>[a]]
s19=[th(s18,b),[a]=>[a,b]]
s20=[id(b),[b]=>[b]]
s21=[th(s20,a),[b]=>[a,b]]
s22=[oia(s19,s21,a#b),[a#b]=>[a,b]]
s23=[ois(s22,a#b),[a#b]=>[a#b]]
s24=[th(s23,c),[a#b]=>[a#b,c]]
s25=[ois(s24,(a#b)#c),[a#b]=>[(a#b)#c]]
s26=[cut(s17,s25,a#b),[b]=>[(a#b)#c]]
s27=[cut(s14,s26,b),[c&b]=>[(a#b)#c]]
s28=[cut(s11,s27,c&b),[a&d]=>[(a#b)#c]]

```

Cut\_free proof is :

=====

```

f115=[g,[d]=>[b]]
f112=[th(f115,a),[a,d]=>[b]]

```





- (1) If  $a = \sim T$  and  $b = \sim \perp$ , then  $T$  and  $\perp$  are the only atomic subformula of  $a$  and  $b$ .
- (2)  $a \uparrow b = \sim T$  iff  $a = \sim b = \sim T$ .
- (3)  $a \vee b = \sim T$  iff  $a(b) = \sim T$  and  $b(a) = \sim \perp$ .
- (4)  $a \uparrow b = \sim \perp$  iff  $a(b) = \sim T$  and  $b(a) = \sim \perp$ .
- (5)  $a \vee b = \sim \perp$  iff  $a = \sim b = \sim \perp$ .

(2.5.6) Corollary

If  $a = \sim T$ , there exists a unique derivation  $p$  of  $\rightarrow a$  consisting at most of instances of (A2), (R5), (R10), and (R13).

Proof:

By induction on the construction of  $a$  :

- i)  $a = T$  as in (1.5.3)
- ii)  $a = c \uparrow d$

- iii)  $a = c \vee d$

- 1)  $c = \sim T$  and  $d = \sim \perp$

$$\frac{\frac{q}{\rightarrow c}}{\rightarrow c, d}}{\rightarrow c \vee d}$$

- 2)  $c = \sim \perp$  and  $d = \sim T$

$$\frac{\frac{r}{\rightarrow d}}{\rightarrow c, d}}{\rightarrow c \vee d}$$

###

(2.5.7) Corollary

If  $b = \sim \perp$ , there exists a unique derivation  $p$  of  $b \rightarrow$  consisting at most of instances of (A3), (R2), (R11), and (R12).

Proof:

By induction on the construction of  $b$  :

- i)  $b = \perp$

$\rightarrow \perp$

- ii)  $b = c \uparrow d$

- 1)  $c = \sim T$  and  $d = \sim \perp$

$$\frac{\frac{q}{d \rightarrow}}{c, d \rightarrow}}{c \uparrow d \rightarrow}$$

2)  $c = \sim \perp$  and  $d = \sim \top$

$$\frac{\frac{c \xrightarrow{r}}{\quad}}{c, d \rightarrow} \quad \frac{\quad}{c \uparrow d \rightarrow}$$

III)  $b = c \vee d$  where  $c = \sim d = \sim \perp$

$$\frac{c \xrightarrow{q} \quad \quad \quad d \xrightarrow{r}}{c \vee d \rightarrow}$$

###

### (2.5.8) Lemma

For every cut-free  $p$  in  $\text{Dbc}(X)$  there exists an equivalent cut-free  $q$  in  $\text{Dbc}(X)$  containing no subderivations constructed with (R3), (R4), (R6) or (R7), and no subderivations constructed with (R2) or (R5) whose active formulae are of the form  $a \uparrow b$  or  $a \vee b$  respectively.

#### Proof

By Lemma (1.5.4) and the following clauses of the algorithm in appendix (3) :

(E.4), (E.5), (E.7), (E.8), (E.12), (E.15), (E.16), (E.20), (E.23), (E.24), (E.25), (E.26), (E.28), (E.30), (E.33), (E.35), (E.38) and the special cases of the clauses (E.29) and (E.34) in which  $G$  and  $D$  are empty sequences, of the the same algorithm in Appendix (3); every derivation of  $\text{Dbc}(X)$  containing subderivations constructed with (R3), (R4), (R6) or (R7) reduces to contraction and interchange-free one . By clauses (E.41.1) and (E.42.2) of Appendix (3), every derivation of  $\text{Dbc}(X)$  containing subderivation constructed with (R2) or (R5) with active formula of the form  $a \uparrow b$  or  $a \vee b$  respectively reduces to an equivalent one where there is no instances of (R2) or (R5) with active formula of the form  $a \uparrow b$  or  $a \vee b$  respectively .

###

The following example for some derivation of  $\text{Dbc}(X)$  which have the contraction and interchange rules in its construction, is to illustrate the above lemma .

#### Example

The cut\_free derivation resulted from Example of (2.5.2) is transformed to a contraction and interchange free one as follows :

```

yes
| ?- theory(bicart).
yes
| ?- prove(s1 = [g, [d] => [b]]).
yes
| ?- prove(s2 = [th(s1,a), [a,d] => [b]]).
yes
| ?- prove(s3 = [aia(s2,a & d), [a & d] => [b]]).
yes
| ?- prove(s4 = [th(s3,a), [a,a & d] => [b]]).
yes
| ?- prove(s5 = [th(s4,d), [a,d,a & d] => [b]]).
yes
| ?- prove(s6 = [aia(s5,a & d), [a & d,a & d] => [b]]).
yes
| ?- prove(s7 = [con(s6,a & d), [a & d] => [b]]).
yes
| ?- prove(s8 = [th(s7,a), [a & d] => [a,b]]).
yes
| ?- prove(s9 = [th(s8,b), [a & d] => [a,b,b]]).
yes
| ?- prove(s10 = [th(s9,a), [a & d] => [a,b,a,b]]).
yes
| ?- prove(s11 = [inc(s10,a,b), [a & d] => [a,a,b,b]]).
yes
| ?- prove(s12 = [con(s11,a), [a & d] => [a,b,b]]).
yes
| ?- prove(s13 = [con(s12,b), [a & d] => [a,b]]).
yes
| ?- prove(s14 = [oia(s13,a # b), [a & d] => [a # b]]).
yes
| ?- prove(s15 = [th(s14,c), [a & d] => [a # b,c]]).
yes
| ?- prove(s16 = [ois(s15,(a#b)#c), [a&d] => [(a#b)#c]]).
yes
| ?- con_inc_free(s16).

```

Proof is as follows :

```

s1=[g,[d]=>[b]]
s2=[th(s1,a),[a,d]=>[b]]
s3=[aia(s2,a&d),[a&d]=>[b]]
s4=[th(s3,a),[a,a&d]=>[b]]
s5=[th(s4,d),[a,d,a&d]=>[b]]
s6=[aia(s5,a&d),[a&d,a&d]=>[b]]
s7=[con(s6,a&d),[a&d]=>[b]]
s8=[th(s7,a),[a&d]=>[a,b]]
s9=[th(s8,b),[a&d]=>[a,b,b]]
s10=[th(s9,a),[a&d]=>[a,b,a,b]]
s11=[inc(s10,a,b),[a&d]=>[a,a,b,b]]
s12=[con(s11,a),[a&d]=>[a,b,b]]
s13=[con(s12,b),[a&d]=>[a,b]]
s14=[oia(s13,a#b),[a&d]=>[a#b]]
s15=[th(s14,c),[a&d]=>[a#b,c]]
s16=[ois(s15,(a#b)#c),[a&d]=>[(a#b)#c]]

```

Contraction and Interchange free proof is :

```

s1=[g,[d]=>[b]]
s2=[th(s1,a),[a,d]=>[b]]
s7=[aia(s2,a&d),[a&d]=>[b]]
s13=[th(s7,a),[a&d]=>[a,b]]
s14=[oia(s13,a#b),[a&d]=>[a#b]]
s15=[th(s14,c),[a&d]=>[a#b,c]]
s16=[ois(s15,(a#b)#c),[a&d]=>[(a#b)#c]]

```

```

yes
| ?-

```

which in tree form is :

$$\begin{array}{c}
 \frac{d \xrightarrow{g} b}{a, d \rightarrow b} \\
 \frac{a \uparrow d \rightarrow b}{a \uparrow d \rightarrow a, b} \\
 \frac{a \uparrow d \rightarrow a, b}{a \uparrow d \rightarrow a \vee b} \\
 \frac{a \uparrow d \rightarrow a \vee b}{a \uparrow d \rightarrow a \vee b, c} \\
 \frac{a \uparrow d \rightarrow a \vee b, c}{a \uparrow d \rightarrow (a \vee b) \vee c}
 \end{array}$$

(2.5.9) Definition

A derivation  $p$  of  $\text{Dbc}(X)$  is normal if it satisfies the following conditions :-

- (1)  $p$  contains no subderivations constructed with (R1), (R3), (R6), (R4) or (R7) .
- (2) If  $p$  derives  $G \rightarrow F$  , and if one of the disjunctions of the formulae of  $F$  is isomorphic to  $T$ , then  $p$  is of the form

$$\frac{\frac{q}{\rightarrow F}}{G \rightarrow F} \quad (K)$$

where  $q$  is the unique derivation of  $\rightarrow F$  compatible with corollary (2.5.6), and where (K) consists of the instances of (R2) required to derive  $G \rightarrow F$  from  $\rightarrow F$ .

- (3) If  $p$  derives  $G \rightarrow F$  , and if one of the conjunctions of the formulae of  $G$  is isomorphic to  $T$  , and no disjunctions of the formulae of  $F$  is isomorphic to  $T$ , then  $p$  is of the form

$$\frac{G \xrightarrow{q}}{G \rightarrow F} \quad (K)$$

where  $q$  is the unique derivation of  $G \rightarrow$  compatible with corollary (2.5.7), and where (K) consists of the instances of (R5) required to derive  $G \rightarrow F$  from  $G \rightarrow$  .

- (4)  $p$  contains no subderivations constructed with (R2) or (R5) whose active formulae are of the form  $a \uparrow b$  or  $a \vee b$  respectively .
- (5) Using the relation  $\equiv$  as defined in (1.5.5) ; for normality in  $\text{Dbc}(X)$  it is required that

$$(R5) \equiv (R13) \equiv (R10) \equiv (R12) \equiv (R11) \equiv (R2)$$

(2.5.10) The Normalization Theorem for Dbc(X)

For every derivation  $p$  in  $\text{Dbc}(X)$  there is a unique equivalent normal derivation  $q$  in  $\text{Dbc}(X)$ .

Proof

We extend the Normalization theorem (1.5.6) by the Cut-Elimination theorem (2.5.2), the above Lemmas and corollaries and the following additional clauses of Appendix (4) :

(D.4), (D.10), (D.34), (D.36), (D.38), (D.55), (D.62) and the special cases of clauses (D.37) and (D.59) with  $G$  and  $D$  are empty sequences; this proves the theorem.

###

For a derivation  $p$  of  $\text{Dbc}(X)$ , we can use the implementation to get its equivalent normal form as follows :

Example

For  $f$  in  $X(a,c)$ ,  $g$  in  $X(b,d)$ ,  $h$  in  $X(c,b)$ ,  $k$  in  $X(d,c)$ ,  $i1$  in  $X(a,b) = \text{comp}(h,f)$ ,  $i2$  in  $X(c,d) = \text{comp}(g,h)$ ,  $i3$  in  $X(b,c) = \text{comp}(k,g)$ ,  $i4$  in  $X(d,b) = \text{comp}(h,k)$ , where  $a,b,c,d$  in  $\text{Ob}(X)$ , the derivation

$$\begin{array}{c}
 \begin{array}{c}
 \frac{a \rightarrow a}{a,b \rightarrow a} \\
 \frac{a,b \rightarrow a \quad f}{a \uparrow b \rightarrow a \rightarrow c} \\
 \frac{a \uparrow b \rightarrow a \rightarrow c}{a \uparrow b \rightarrow c}
 \end{array}
 \quad
 \begin{array}{c}
 \frac{b \rightarrow b}{a,b \rightarrow b} \\
 \frac{a,b \rightarrow b \quad g}{a \uparrow b \rightarrow b \rightarrow d} \\
 \frac{a \uparrow b \rightarrow b \rightarrow d}{a \uparrow b \rightarrow d}
 \end{array}
 \quad
 \begin{array}{c}
 \frac{c \rightarrow c}{c,d \rightarrow c} \\
 \frac{c,d \rightarrow c \quad c \rightarrow c,d}{c,d \rightarrow c \rightarrow c \rightarrow d} \\
 \frac{c,d \rightarrow c \rightarrow c \rightarrow d}{c \uparrow d \rightarrow c}
 \end{array}
 \quad
 \begin{array}{c}
 \frac{b \rightarrow b}{b \rightarrow b,c} \\
 \frac{b \rightarrow b,c \quad h}{c \rightarrow b} \\
 \frac{c \rightarrow b \quad b \rightarrow b \vee c}{c \rightarrow b \vee c} \\
 \frac{c \rightarrow b \vee c \quad c \rightarrow c \rightarrow d}{c \vee d \rightarrow b \vee c} \\
 \frac{c \vee d \rightarrow b \vee c}{c \rightarrow b \vee c}
 \end{array}
 \quad
 \begin{array}{c}
 \frac{c \rightarrow c}{d \rightarrow c} \\
 \frac{d \rightarrow c \quad k}{c \rightarrow b \vee c} \\
 \frac{c \rightarrow b \vee c \quad d \rightarrow b \vee c}{d \rightarrow b \vee c} \\
 \frac{d \rightarrow b \vee c}{c \rightarrow b \vee c}
 \end{array}
 \end{array}$$

$$\frac{a \uparrow b \rightarrow c \quad a \uparrow b \rightarrow d \quad c \uparrow d \rightarrow c}{a \uparrow b \rightarrow c \uparrow d} \quad \frac{c \uparrow d \rightarrow c}{c \uparrow d \rightarrow b \vee c}$$

$$\frac{a \uparrow b \rightarrow c \uparrow d \quad c \uparrow d \rightarrow b \vee c}{a \uparrow b \rightarrow b \vee c}$$

is transformed to its equivalent normal form as follows :

```

yes
| ?- theory(bicart).
yes
| ?- composite(h,f,i1).
yes
| ?- composite(g,h,i2).
yes
| ?- composite(k,g,i3).
yes
| ?- composite(h,k,i4).
yes
| ?- prove(g1 = [id(a), [a] => [a]]).
yes
| ?- prove(g2 = [th(g1,b), [a,b] => [a]]).
yes
| ?- prove(g3 = [aia(g2,a & b), [a & b] => [a]]).
yes

```

```

| ?- prove(g4 = [f, [a] => [c]]).
yes
| ?- prove(g5 = [cut(g3,g4,a), [a & b] => [c]]).
yes
| ?- prove(g6 = [id(b), [b] => [b]]).
yes
| ?- prove(g7 = [th(g6,a), [a,b] => [b]]).
yes
| ?- prove(g8 = [aia(g7,a & b), [a & b] => [b]]).
yes
| ?- prove(g9 = [g, [b] => [d]]).
yes
| ?- prove(g10 = [cut(g8,g9,b), [a & b] => [d]]).
yes
| ?- prove(g11 = [ais(g5,g10,c & d), [a & b] => [c & d]]).
yes
| ?- prove(g12 = [id(c), [c] => [c]]).
yes
| ?- prove(g13 = [th(g12,d), [c,d] => [c]]).
yes
| ?- prove(g14 = [aia(g13,c & d), [c & d] => [c]]).
yes
| ?- prove(g15 = [id(c), [c] => [c]]).
yes
| ?- prove(g16 = [th(g15,d), [c] => [c,d]]).
yes
| ?- prove(g17 = [ois(g16,c # d), [c] => [c # d]]).
yes
| ?- prove(g18 = [h, [c] => [b]]).
yes
| ?- prove(g19 = [id(b), [b] => [b]]).
yes
| ?- prove(g20 = [th(g19,c), [b] => [b,c]]).
yes
| ?- prove(g21 = [ois(g20,b # c), [b] => [b # c]]).
yes
| ?- prove(g22 = [cut(g18,g21,b), [c] => [b # c]]).
yes
| ?- prove(g23 = [k, [d] => [c]]).
yes
| ?- prove(g24 = [id(c), [c] => [c]]).
yes
| ?- prove(g25 = [th(g24,b), [c] => [b,c]]).
yes
| ?- prove(g26 = [ois(g25,b # c), [c] => [b # c]]).
yes
| ?- prove(g27 = [cut(g23,g26,c), [d] => [b # c]]).
yes
| ?- prove(g28 = [oia(g22,g27,c # d), [c # d] => [b # c]]).
yes
| ?- prove(g29 = [cut(g17,g28,c # d), [c] => [b # c]]).
yes
| ?- prove(g30 = [cut(g14,g29,c), [c & d] => [b # c]]).
yes
| ?- prove(g31 = [cut(g11,g30,c & d), [a & b] => [b # c]]).
yes
| ?- normal(g31).

```

Proof is as follows :

```

g1=[id(a),[a]=>[a]]
g2=[th(g1,b),[a,b]=>[a]]
g3=[aia(g2,a&b),[a&b]=>[a]]
g4=[f,[a]=>[c]]
g5=[cut(g3,g4,a),[a&b]=>[c]]
g6=[id(b),[b]=>[b]]
g7=[th(g6,a),[a,b]=>[b]]
g8=[aia(g7,a&b),[a&b]=>[b]]
g9=[g,[b]=>[d]]
g10=[cut(g8,g9,b),[a&b]=>[d]]
g11=[ais(g5,g10,c&d),[a&b]=>[c&d]]
g12=[id(c),[c]=>[c]]
g13=[th(g12,d),[c,d]=>[c]]
g14=[aia(g13,c&d),[c&d]=>[c]]
g15=[id(c),[c]=>[c]]
g16=[th(g15,d),[c]=>[c,d]]
g17=[ois(g16,c#d),[c]=>[c#d]]
g18=[h,[c]=>[b]]
g19=[id(b),[b]=>[b]]
g20=[th(g19,c),[b]=>[b,c]]

```

```

g21=[ois(g20,b#c),[b]=>[b#c]]
g22=[cut(g18,g21,b),[c]=>[b#c]]
g23=[k,[d]=>[c]]
g24=[id(c),[c]=>[c]]
g25=[th(g24,b),[c]=>[b,c]]
g26=[ois(g25,b#c),[c]=>[b#c]]
g27=[cut(g23,g26,c),[d]=>[b#c]]
g28=[oia(g22,g27,c#d),[c#d]=>[b#c]]
g29=[cut(g17,g28,c#d),[c]=>[b#c]]
g30=[cut(g14,g29,c),[c&d]=>[b#c]]
g31=[cut(g11,g30,c&d),[a&b]=>[b#c]]

```

Cut\_free proof is :

```

=====
f124=[i1,[a]=>[b]]
f121=[th(f124,b),[a,b]=>[b]]
f118=[aia(f121,a&b),[a&b]=>[b]]
f115=[th(f118,b),[a&b,b]=>[b]]
f112=[th(f115,a),[a&b,a,b]=>[b]]
f100=[oia(f112,a&b),[a&b,a&b]=>[b]]
f95=[con(f100,a&b),[a&b]=>[b]]
f92=[th(f95,c),[a&b]=>[b,c]]
f89=[th(f92,c),[a&b]=>[b,c,c]]
f86=[ois(f89,b#c),[a&b]=>[b#c,c]]
f83=[th(f86,b),[a&b]=>[b#c,b,c]]
f80=[ois(f83,b#c),[a&b]=>[b#c,b#c]]
g31=[con(f80,b#c),[a&b]=>[b#c]]

```

Contraction and Interchange free proof is :

```

=====
f124=[i1,[a]=>[b]]
f121=[th(f124,b),[a,b]=>[b]]
f95=[aia(f121,a&b),[a&b]=>[b]]
f80=[th(f95,c),[a&b]=>[b,c]]
g31=[ois(f80,b#c),[a&b]=>[b#c]]

```

Normal proof is :

```

=====
f124=[i1,[a]=>[b]]
f121=[th(f124,c),[a]=>[b,c]]
f95=[ois(f121,b#c),[a]=>[b#c]]
f80=[th(f95,b),[a,b]=>[b#c]]
g31=[aia(f80,a&b),[a&b]=>[b#c]]

```

```

yes
| ?-

```

which in tree form is :

$$\begin{array}{c}
 i1 \\
 \hline
 a \rightarrow b \\
 \hline
 a \rightarrow b, c \\
 \hline
 a \rightarrow b \vee c \\
 \hline
 a, b \rightarrow b \vee c \\
 \hline
 a \wedge b \rightarrow b \vee c
 \end{array}$$

#### (2.5.11) The CHURCH-ROSSER Theorem for $Dbc(X)$

If  $p = g$ , then there exists a normal  $r$  in  $Dbc(X)$  such that  $p \succ r$  and  $q \succ r$ .

#### (2.6) Application to Category Theory

In this section we illustrate the use of the system to represent some arrows of  $Fbc(X)$ , to obtain their normal representations, and (where possible) to show the equality of arrows of  $Fbc(X)$ .



Example 1

The normal representation of the following arrow

$$a \vee (b \vee c) \xrightarrow{\text{id}(a) \vee \text{tau}(b \vee c)} a \vee \top$$

is obtained as follows :

```
yes
| ?- theory(bicart).
```

```
yes
| ?- norm_rep(union(id(a),tau(b # c))).
```

The arrow is :

```
=====
union(id(a),tau(b#c))
```

Proof is as follows :

```
=====
f2=[id(a),[a]=>[a]]
f7=[id(a),[a]=>[a]]
f6=[th(f7,tr),[a]=>[a,tr]]
f5=[ois(f6,a#tr),[a]=>[a#tr]]
f8=[cut(f2,f5,a),[a]=>[a#tr]]
f4=[true,[]=>[tr]]
f3=[th(f4,b#c),[b#c]=>[tr]]
f11=[id(tr),[tr]=>[tr]]
f10=[th(f11,a),[tr]=>[a,tr]]
f9=[ois(f10,a#tr),[tr]=>[a#tr]]
f12=[cut(f3,f9,tr),[b#c]=>[a#tr]]
f1=[ois(f8,f12,a#b#c),[a#b#c]=>[a#tr]]
```

Cut\_free proof is :

```
=====
f18=[id(a),[a]=>[a]]
f15=[th(f18,tr),[a]=>[a,tr]]
f8=[ois(f15,a#tr),[a]=>[a#tr]]
f27=[true,[]=>[tr]]
f24=[th(f27,b#c),[b#c]=>[tr]]
f21=[th(f24,a),[b#c]=>[a,tr]]
f12=[ois(f21,a#tr),[b#c]=>[a#tr]]
f1=[ois(f8,f12,a#b#c),[a#b#c]=>[a#tr]]
```

Contraction and Interchange free proof is :

```
=====
f28=[true,[]=>[tr]]
f30=[th(f28,a),[a]=>[a,tr]]
f15=[th(f30,a),[a]=>[a,tr]]
f8=[ois(f15,a#tr),[a]=>[a#tr]]
f34=[true,[]=>[tr]]
f36=[th(f34,a),[a]=>[a,tr]]
f37=[th(f36,b),[b]=>[a,tr]]
f40=[true,[]=>[tr]]
f39=[th(f40,a),[a]=>[a,tr]]
f38=[th(f39,c),[c]=>[a,tr]]
f21=[ois(f37,f38,b#c),[b#c]=>[a,tr]]
f12=[ois(f21,a#tr),[b#c]=>[a#tr]]
f1=[ois(f8,f12,a#b#c),[a#b#c]=>[a#tr]]
```

Normal proof is :

```
=====
f28=[true,[]=>[tr]]
f30=[th(f28,a),[a]=>[a,tr]]
f15=[ois(f30,a#tr),[a]=>[a#tr]]
f8=[th(f15,a),[a]=>[a#tr]]
```



```
yes
| ?- arrow(d,k,o).
```

```
yes
| ?- composite(g,f,h).
```

```
yes
| ?- composite(j,l,k).
```

```
yes
| ?- commutative.
```

Insert the first arrow : union(h,k).

The arrow is :

```
union(h,k)
```

Proof is as follows :

```
f2=[h,[a]=>[c]]
f6=[id(c).[c]=>[c]]
f5=[th(f6,o).[c]=>[c,o]]
f4=[ois(f5,c#o).[c]=>[c#o]]
f7=[cut(f2,f4,c).[a]=>[c#o]]
f3=[k,[d]=>[o]]
f10=[id(o).[o]=>[o]]
f9=[th(f10,c).[o]=>[c,o]]
f8=[ois(f9,c#o).[o]=>[c#o]]
f11=[cut(f3,f8,o).[d]=>[c#o]]
f1=[oia(f7,f11,a#d).[a#d]=>[c#o]]
```

Cut\_free proof is :

```
f17=[h,[a]=>[c]]
f14=[th(f17,o).[a]=>[c,o]]
f7=[ois(f14,c#o).[a]=>[c#o]]
f23=[k,[d]=>[o]]
f20=[th(f23,c).[d]=>[c,o]]
f11=[ois(f20,c#o).[d]=>[c#o]]
f1=[oia(f7,f11,a#d).[a#d]=>[c#o]]
```

Contraction and interchange free proof is :

```
f17=[h,[a]=>[c]]
f14=[th(f17,o).[a]=>[c,o]]
f7=[ois(f14,c#o).[a]=>[c#o]]
f23=[k,[d]=>[o]]
f20=[th(f23,c).[d]=>[c,o]]
f11=[ois(f20,c#o).[d]=>[c#o]]
f1=[oia(f7,f11,a#d).[a#d]=>[c#o]]
```

Normal proof is :

```
f17=[h,[a]=>[c]]
f14=[th(f17,o).[a]=>[c,o]]
f7=[ois(f14,c#o).[a]=>[c#o]]
f23=[k,[d]=>[o]]
f20=[th(f23,c).[d]=>[c,o]]
f11=[ois(f20,c#o).[d]=>[c#o]]
f1=[oia(f7,f11,a#d).[a#d]=>[c#o]]
```

Insert the second arrow : comp(union(g,j),union(f,i)).

The arrow is :

```
comp(union(g,j),union(f,i))
```

• Proof is as follows :

```

f37=[f,[a]=>[b]]
f41=[id(b),[b]=>[b]]
f40=[th(f41,e),[b]=>[b,e]]
f39=[ois(f40,b#e),[b]=>[b#e]]
f42=[cut(f37,f39,b),[a]=>[b#e]]
f38=[i,[d]=>[e]]
f45=[id(e),[e]=>[e]]
f44=[th(f45,b),[e]=>[b,e]]
f43=[ois(f44,b#e),[e]=>[b#e]]
f46=[cut(f38,f43,e),[d]=>[b#e]]
f36=[ois(f42,f46,a#d),[a#d]=>[b#e]]
f26=[g,[b]=>[c]]
f30=[id(c),[c]=>[c]]
f29=[th(f30,o),[c]=>[c,o]]
f28=[ois(f29,c#o),[c]=>[c#o]]
f31=[cut(f26,f28,c),[b]=>[c#o]]
f27=[j,[e]=>[o]]
f34=[id(o),[o]=>[o]]
f33=[th(f34,c),[o]=>[c,o]]
f32=[ois(f33,c#o),[o]=>[c#o]]
f35=[cut(f27,f32,o),[e]=>[c#o]]
f25=[ois(f31,f35,b#e),[b#e]=>[c#o]]
f24=[cut(f36,f25,b#e),[a#d]=>[c#o]]

```

Cut\_free proof is :

```

f105=[h,[a]=>[c]]
f104=[th(f105,o),[a]=>[c,o]]
f101=[th(f104,o),[a]=>[c,o,o]]
f98=[ois(f101,c#o),[a]=>[c#o,o]]
f95=[th(f98,c),[a]=>[c#o,c,o]]
f81=[ois(f95,c#o),[a]=>[c#o,c#o]]
f82=[con(f81,c#o),[a]=>[c#o]]
f134=[k,[d]=>[o]]
f131=[th(f134,c),[d]=>[c,o]]
f128=[th(f131,o),[d]=>[c,o,o]]
f125=[ois(f128,c#o),[d]=>[c#o,o]]
f122=[th(f125,c),[d]=>[c#o,c,o]]
f110=[ois(f122,c#o),[d]=>[c#o,c#o]]
f111=[con(f110,c#o),[d]=>[c#o]]
f24=[ois(f82,f111,a#d),[a#d]=>[c#o]]

```

Contraction and interchange free proof is :

```

f105=[h,[a]=>[c]]
f81=[th(f105,o),[a]=>[c,o]]
f82=[ois(f81,c#o),[a]=>[c#o]]
f128=[k,[d]=>[o]]
f110=[th(f128,c),[d]=>[c,o]]
f111=[ois(f110,c#o),[d]=>[c#o]]
f24=[ois(f82,f111,a#d),[a#d]=>[c#o]]

```

Normal proof is :

```

f105=[h,[a]=>[c]]
f81=[th(f105,o),[a]=>[c,o]]
f82=[ois(f81,c#o),[a]=>[c#o]]
f128=[k,[d]=>[o]]
f110=[th(f128,c),[d]=>[c,o]]
f111=[ois(f110,c#o),[d]=>[c#o]]
f24=[ois(f82,f111,a#d),[a#d]=>[c#o]]

```

The answer is :

```

union(h,k) =
comp(union(g,j),union(f,i))

```

because they have the same normal representation .

```

yes
| ?-

```

## CHAPTER 3

## DISTRIBUTIVE BICARTESIAN CATEGORY THEORY

In this chapter, we study the effect of the distributivity of  $\uparrow$  over  $\vee$ ; and define the class of small distributive bicartesian categories. We describe our representation of this class in Prolog, and the extensions to the implementation of bicartesian category theory required. Rules such as (R1) "cut rule", (R10) "and-introduction in the succedent" and (R12) "or-introduction in the antecedent" now become more general, and extra transformation rules are required in the cut-elimination and normalization algorithms. We describe these changes and their implementations.

Corollaries (3.5.5) and (3.5.6), concerning the terminal and initial objects in  $Fdbc(X)$ , are stated without proof in SZABO; we have constructed and implemented their proofs in the system in order to be able to automate derivations containing terminal or initial objects. Lemma (3.5.7) is a modification of a similar lemma, stated without proof in SZABO, to meet the changes introduced in the previous chapters by adding (R4) and (R7); we have constructed and implemented the proof of this lemma in the system, in order to remove the contractions and interchanges from derivations in the class  $Ddbc(X)$  according to the definition of the normal form (3.5.8).

## (3.1) Definition

A distributive bicartesian category is a cartesian category with the following additional structure:

- (4) A bifunctor  $(-) \vee (-) : C \times C \rightarrow C$ .
- (5) A distinguished object  $\perp \in Ob C$ .
- (6) Two adjunctions  $\alpha_\delta$  and  $\alpha_i$ , where

$$\alpha_\delta = \{ \alpha_\delta(A, B, C, D) : C(A \uparrow (B \vee C), D) \rightarrow C(A \uparrow B, D) \times C(A \uparrow C, D) \in ArEns \mid A, B, C, D \in Ob C \}$$

and

$$\alpha_i = \{ \alpha_i(A) : C(\perp, A) \rightarrow \{*\} \in ArEns \mid A \in Ob C \}.$$

### (3.2) The Free Distributive Bicartesian Category Fdbc(X)

Let  $X$  be a fixed but arbitrary small category. The free distributive bicartesian category Fdbc(X) is constructed as follows :

#### (3.2.1) Definition

The class  $dbcL(X)$  of ( distributive bicartesian ) formulae over a category  $X$  is the same as the class  $bcL(X)$  of (bicartesian) formulae over  $X$  as defined in (2.2.1) .

#### (3.2.2) The labelled Derivations LDdbc(X)

The class of ( distributive bicartesian ) labelled derivations  $LDdbc(X)$  over a category  $X$  is an extension to the class  $LDbc(X)$  in (2.2.2) by the replacement of (xii) by

xii') for derivations  $F, G$  with conclusions  $f : A \uparrow B \rightarrow D$  and  $g : A \uparrow C \rightarrow D$  respectively

$ROUND( F, G, A \uparrow (B \vee C) )$  is a derivation with conclusion  $( f, g ) : A \uparrow (B \vee C) \rightarrow D$  .

#### (3.2.3) Definition

The relation  $==$  is defined as the smallest equivalence relation on  $LDdbc(X)$  satisfying the conditions of Definition (1.2.3) and the following additional conditions :

[ Note : we use the notation

$$LA(A) = \begin{cases} \text{if } A \text{ is an object of } X \\ \text{then } LA1(Id(A)) \\ \text{else } LA2(A) . \end{cases}$$

ix) If  $F == G$  and  $H == K$  with conclusions  $f, g : A \uparrow B \rightarrow D$  and  $h, k : A \uparrow C \rightarrow D$

then

$$ROUND( F, H, A \uparrow (B \vee C) ) == ROUND( G, K, A \uparrow (B \vee C) ) ;$$

x) For  $F, G$  with conclusions  $f : A \uparrow B \rightarrow D$  ,  $g : A \uparrow C \rightarrow D$  respectively

then

$$\begin{aligned} &COMP( PRODUCT( LA(A), LA7(B,C) ), \\ &ROUND( F, G, A \uparrow (B \vee C) ), \\ &A \uparrow (B \vee C) ) == F \end{aligned}$$

and

$$\begin{aligned} &COMP( PRODUCT( LA(A), LAB(B,C) ), \\ &ROUND( F, G, A \uparrow (B \vee C) ), \\ &A \uparrow (B \vee C) ) == G \end{aligned}$$

- xi) For  $F$  with conclusion  $f : A \uparrow (B \vee C) \rightarrow D$   
 $\text{ROUND}(\text{COMP}(\text{PRODUCT}(\text{LA}(A), \text{LA7}(B,C)), F, A \uparrow (B \vee C)),$   
 $\text{COMP}(\text{PRODUCT}(\text{LA}(A), \text{LAB}(B,C)), F, A \uparrow (B \vee C)),$   
 $A \uparrow (B \vee C)) \quad \equiv \quad F \quad ;$
- xii) For  $F$  with conclusion  $f : \perp \rightarrow A$  ,  
 $F \equiv \text{LA6}(A)$  .

### (3.2.4) Definition

$\text{Fdbc}(X)$  is defined to be the category having :

- i) as objects , the distributive bicartesian formulae over  $X$  ;  
 ii) as arrows , the  $\equiv$ -classes of  $\text{LDdbc}(X)$  ;  
 iii) to xv) as in the definition of  $\text{Fbc}(X)$  ;  
 xvi) as  $(-, -)$  , the function defined by  
 $( [F], [G] ) = [ \text{ROUND}( F, G, A \uparrow (B \vee C) ) ]$   
 where  $\text{dom}([F]) = A \uparrow B$  ,  
 $\text{dom}([G]) = A \uparrow C$  and  
 $\text{cod}([F]) = \text{cod}([G])$  .

It is now routine to check that, with this structure,  $\text{Fdbc}(X)$  is a category, is distributive bicartesian, and is free over  $X$ . There is the obvious embedding  $X \rightarrow \text{Fdbc}(X)$  defined by  $A \rightarrow A; f \rightarrow [\text{LA1}(f)]$ .

### (3.3) The Unlabelled Derivations $\text{Ddbc}(X)$

The class  $\text{Ddbc}(X)$  of unlabelled derivations ( for representation of the free distributive bicartesian category over  $X$  ) is defined as  $\text{Dbc}(X)$  in (2.3) with extra generality for the constructors (R1), (R10) and (R12) as follows :

- iv') (cut)  
 if  $p_1, p_2$  are derivations with conclusions  $G \rightarrow \text{FaE}, \text{DaL} \rightarrow H$   
 then  
 $\text{R1}(p_1, p_2, a)$  is a derivation with conclusion  $\text{DGL} \rightarrow \text{FHE}$  ;
- xi') ( and-introduction in the succedent )  
 if  $p_1, p_2$  are derivations with conclusions  $G \rightarrow \text{FaE}, G \rightarrow \text{FbE}$   
 then  
 $\text{R10}(p_1, p_2, a \uparrow b)$  is a derivation with conclusion  $G \rightarrow \text{Fa} \uparrow \text{bE}$ ;
- xiii') ( or-introduction in the antecedent )  
 if  $p_1, p_2$  are derivations with conclusions  $\text{GaD} \rightarrow F, \text{GbD} \rightarrow F$   
 then  
 $\text{R12}(p_1, p_2, a \vee b)$  is a derivation with conclusion  $\text{Ga} \vee \text{bD} \rightarrow F$ .

The following is an example of a derivation in  $\text{Ddbc}(X)$ , its representation in Prolog and the use of the implementation to check the correctness of this derivation :

Example

For objects  $a, b, c$  in the category  $X$ , the derivation

$$\begin{array}{c}
 \begin{array}{c}
 \frac{a \rightarrow a}{a \rightarrow a, b} \\
 \frac{a \rightarrow a, b}{a \rightarrow c, a, b} \\
 \hline
 a, a \uparrow b \rightarrow c, a, b
 \end{array}
 \qquad
 \begin{array}{c}
 \frac{c \rightarrow c}{c, a \rightarrow c} \\
 \frac{c, a \rightarrow c}{c, a, b \rightarrow c} \\
 \frac{c, a, b \rightarrow c}{c, a \uparrow b \rightarrow c} \\
 \hline
 a, (a \uparrow b) \vee c \rightarrow c, a, b
 \end{array}
 \qquad
 \begin{array}{c}
 \frac{a \rightarrow a}{a \rightarrow a, b} \\
 \frac{a \rightarrow a, b}{a \rightarrow c, a, b} \\
 \hline
 a, c \rightarrow c, a, b
 \end{array}
 \end{array}$$

is represented by

```

l = [id(a), [a] => [a]]
l1 = [th(l,b), [a] => [a,b]]
l2 = [th(l1,c), [a] => [c,a,b]]
m = [id(c), [c] => [c]]
m1 = [th(m,a), [c,a] => [c]]
m2 = [th(m1,b), [c,a,b] => [c]]
m3 = [oia(m2,a & b), [c,a & b] => [c]]
n = [cut(l2,m3,c), [a,a & b] => [c,a,b]]
o = [th(l2,c), [a,c] => [c,a,b]]
p = [oia(n,o,(a & b) # c), [a,(a & b) # c] => [c,a,b]]

```

and the following is a session with the system to check the correctness of the above derivation :

```

yes
| ?- theory(dbicart).
yes
| ?- prove(l = [id(a), [a] => [a]]).
yes
| ?- prove(l1 = [th(l,b), [a] => [a,b]]).
yes
| ?- prove(l2 = [th(l1,c), [a] => [c,a,b]]).
yes
| ?- prove(m = [id(c), [c] => [c]]).
yes
| ?- prove(m1 = [th(m,a), [c,a] => [c]]).
yes
| ?- prove(m2 = [th(m1,b), [c,a,b] => [c]]).
yes
| ?- prove(m3 = [oia(m2,a & b), [c,a & b] => [c]]).
yes
| ?- prove(n = [cut(l2,m3,c), [a,a & b] => [c,a,b]]).
yes
| ?- prove(o = [th(l2,c), [a,c] => [c,a,b]]).
yes
| ?- prove(p = [oia(n,o,(a&b)#c), [a,(a&b)#c] => [c,a,b]]).
yes
| ?-

```



(3.4) The Semantics of Ddbc(X)

We now define an interpretation function to interpret the derivations of Ddbc(X) in ArFdbc(X) and prove the completeness theorem. For this purpose we extend the canonical arrows in (1.4) and (2.4) by the following canonical arrows of Fdbc(X) :

$$\text{vii) } \text{delta\_left}(A,B,C) : A \uparrow (B \vee C) \rightarrow (A \uparrow B) \vee (A \uparrow C)$$

for all  $A,B,C$  in  $\text{ObFdbc}(X)$  , and where

$$\text{delta\_left}(A,B,C) = v((A \uparrow B) \vee (A \uparrow C))(\text{Id}((A \uparrow B) \vee (A \uparrow C))),$$

and where  $v$  is the natural isomorphism determined by the compositions

$$\begin{array}{ccc} \underline{C}((A \uparrow B) \vee (A \uparrow C), D) & \xrightarrow{\alpha_\sigma} & \underline{C}(A \uparrow B, D) \times \underline{C}(A \uparrow C, D) \\ & \searrow v & \downarrow \alpha_\delta^{-1} \\ & & \underline{C}(A \uparrow (B \vee C), D) \end{array}$$

with  $\underline{C} = \text{Fdbc}(X)$  .

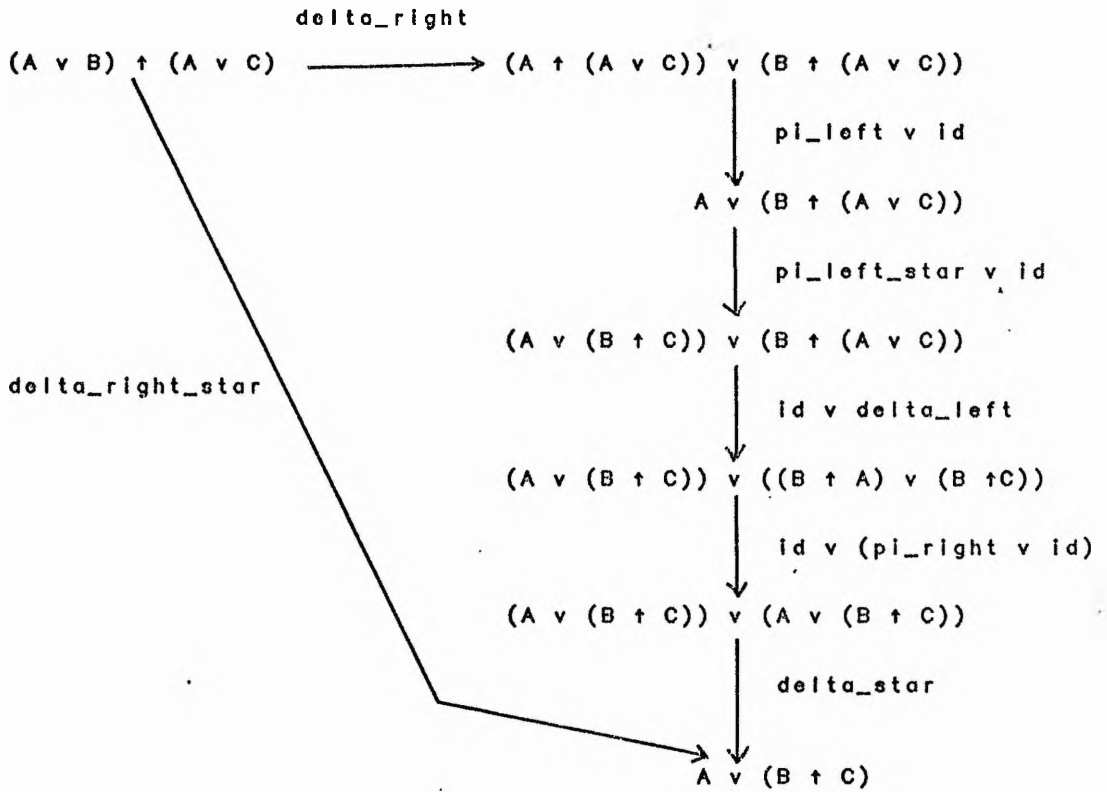
$$\text{viii) } \text{delta\_right}(A,B,C) : (A \vee B) \uparrow C \rightarrow (A \uparrow C) \vee (B \uparrow C)$$

for all  $A,B,C$  in  $\text{ObFdbc}(X)$ , with  $\text{delta\_right}$  defined by the compositions

$$\begin{array}{ccccc} (A \vee B) \uparrow C & \xrightarrow{\text{sigma}} & C \uparrow (A \vee B) & \xrightarrow{\text{delta\_left}} & (C \uparrow A) \vee (C \uparrow B) \\ & \searrow \text{delta\_right} & & & \downarrow \text{sigma} \vee \text{sigma} \\ & & & & (A \uparrow C) \vee (B \uparrow C) \end{array}$$

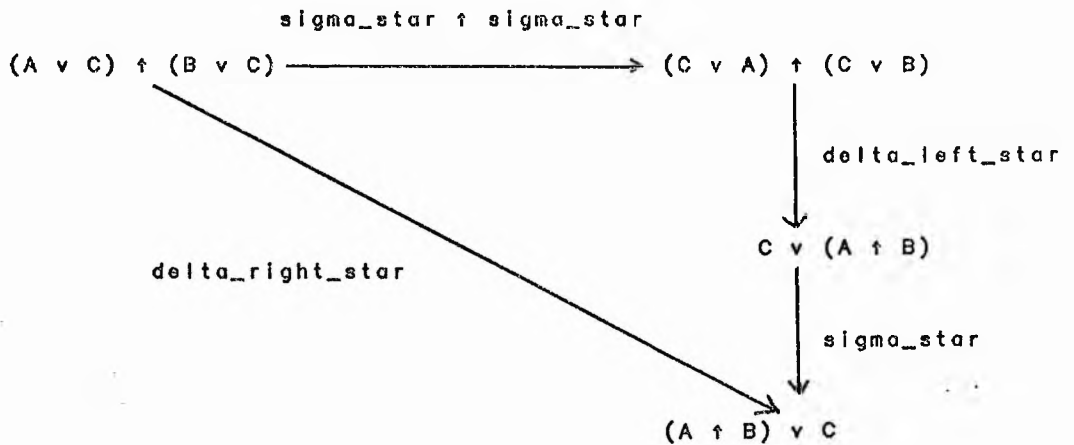
$$\text{ix) } \text{delta\_left\_star}(A,B,C) : (A \vee B) \uparrow (A \vee C) \rightarrow A \vee (B \uparrow C)$$

for all  $A,B,C$  in  $\text{ObFdbc}(X)$ , with  $\text{delta\_left\_star}$  defined by the compositions



x)  $\text{delta\_right\_star}(A,B,C) : (A \vee C) \uparrow (B \vee C) \rightarrow (A \uparrow B) \vee C$

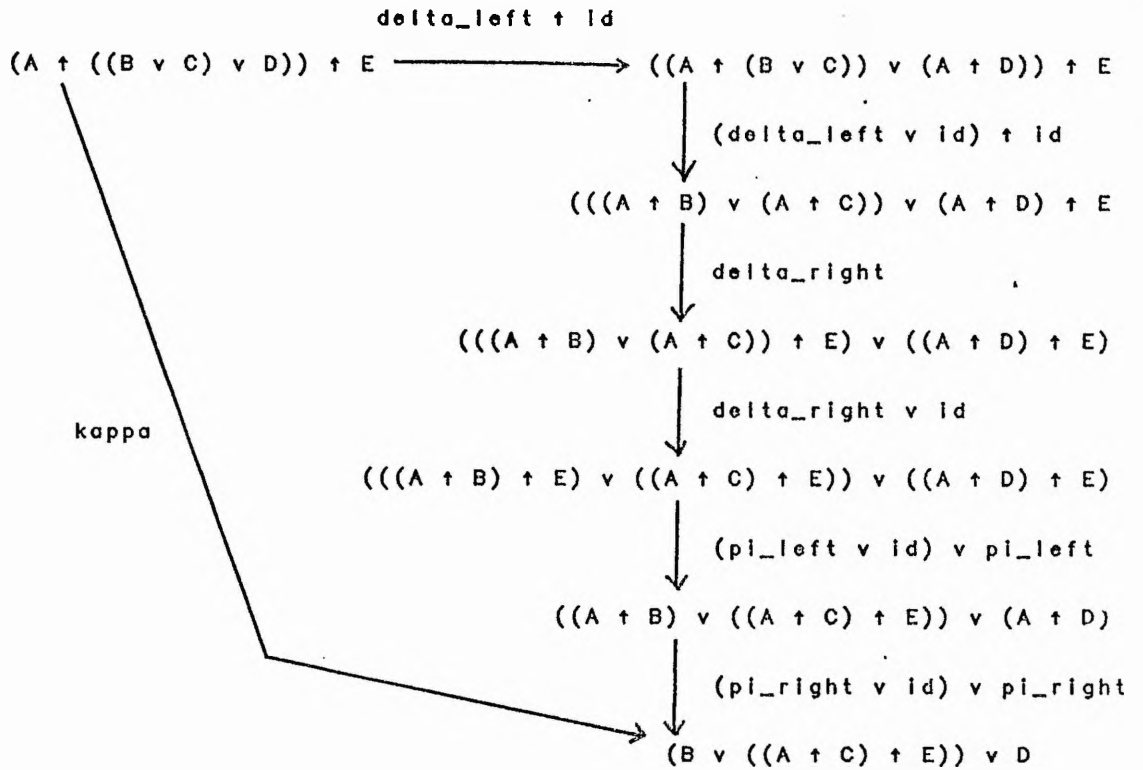
for all  $A,B,C$  in  $\text{ObFdbc}(X)$ , with  $\text{delta\_right\_star}$  defined by the compositions



xi)  $\text{kappa}(A,B,C,D,E) : (A \uparrow ((B \vee C) \vee D)) \uparrow E \rightarrow$

$(B \vee ((A \uparrow C) \uparrow E)) \vee D$

for all  $A,B,C,D,E$  in  $\text{ObFdbc}(X)$ , with  $\text{kappa}$  defined by the compositions



### (3.4.1) Definition

We extend the interpretation function  $S$  defined in (2.4.1) to interpret derivations of  $Ddbc(X)$  in  $ArFdbc(X)$  and satisfy the following generalizations of conditions (3), (10), (7) and (14) mentioned in (1.4.1) and (2.4.1) :

$$\begin{aligned}
 10') \quad S(R1(p1, p2, a)) &= \text{comp}(\text{comp}((I1 \vee S(p2)) \vee I2, \\
 &\quad \text{kappa}(PI(D), \text{Un}(F), a, \text{Un}(E), PI(L))), \\
 &\quad (I3 \uparrow S(p1)) \uparrow I4) \quad , \quad \text{where} \\
 &I1 \text{ is the identity on } \text{Un}(F) \quad , \\
 &I2 \text{ is the identity on } \text{Un}(E) \quad , \\
 &I3 \text{ is the identity on } PI(D) \quad , \\
 &I4 \text{ is the identity on } PI(L) \text{ and} \\
 &G, D, L, a, F, H, E \text{ are as in (3.3) condition } (iv') \text{.}
 \end{aligned}$$

$$\begin{aligned}
 7') \quad S(R10(p1, p2, a \uparrow b)) &= \text{comp}(\text{comp}(\text{comp}((\text{delta\_left\_star}(\text{Un}(F), a, b) \vee I), \\
 &\quad \text{delta\_right\_star}(\text{Un}(F) \vee a, \\
 &\quad \text{Un}(F) \vee b, \text{Un}(E))), \\
 &\quad S(p1) \uparrow S(p2) \quad ), \\
 &\quad \text{delta}(PI(G)) \quad ) \quad , \quad \text{where} \\
 &I \text{ is the identity on } \text{Un}(E) \text{ and} \\
 &G, F, a, b, E \text{ are as in (3.3) condition } (xi') \text{.}
 \end{aligned}$$

14')  $S(R12(p1,p2,a \vee b) = \text{comp}(\text{comp}(\text{comp}(\text{delta\_star}(\text{Un}(F)),$   
 $S(p1) \vee S(p2)),$   
 $\text{delta\_right}(\text{Pi}(G) \uparrow a, \text{Pi}(G) \uparrow b, \text{Pi}(D))),$   
 $(\text{delta\_left}(\text{Pi}(G), a, b) \uparrow I)),$  where  
 $I$  is the identity over  $\text{Pi}(D)$ , and  
 $G, a, b, D, F$  being as in (3.3) condition xlii').

The following example is a session with the system to interpret a derivation of the class  $\text{Ddbc}(X)$  in  $\text{ArFdbc}(X)$  :

Example

Let  $a, b$  be objects of the category  $X$ ,  $f$  in  $X(b, a)$ ; then the derivation

$$\begin{array}{c}
 \frac{a \rightarrow a}{a \rightarrow a, d} \\
 \frac{a \rightarrow a, d \quad b \xrightarrow{f} a}{a \rightarrow b, a, d} \\
 \frac{a \rightarrow b, a, d}{a \rightarrow a, a, d} \\
 \frac{a \rightarrow a, a, d}{a \rightarrow a, a \vee d} \\
 \frac{a \rightarrow a, a \vee d \quad a \rightarrow a}{a \rightarrow a \uparrow b, a \vee d} \\
 \frac{a \rightarrow a \uparrow b, a \vee d}{a \rightarrow (a \uparrow b) \vee (a \vee d)}
 \end{array}$$

is interpreted in  $\text{ArFdbc}(X)$  as follows :

```

yes
| ?- theory(dbicart).

yes
| ?- prove(l = [ld(a), [a] => [a]]).

yes
| ?- prove(l1 = [th(l,d), [a] => [a,d]]).

yes
| ?- prove(l2 = [th(l1,b), [a] => [b,a,d]]).

yes
| ?- prove(m = [f, [b] => [a]]).

yes
| ?- prove(n = [cut(l2,m,b), [a] => [a,a,d]]).

yes
| ?- prove(n1 = [ois(n,a # d), [a] => [a,a # d]]).

yes
| ?- prove(o = [th(l,b), [a] => [b,a]]).

yes
| ?- prove(o1 = [th(o,d), [a] => [b,a,d]]).

yes
| ?- prove(o2 = [ois(o1,a # d), [a] => [b,a # d]]).

yes
| ?- prove(p = [ais(n1,o2,a & b), [a] => [a&b,a#d]]).

yes
| ?- prove(q = [ois(p,(a&b)#(a#d)), [a] => [(a&b)#(a#d)]]).

```



The following examples of some canonical arrows of  $Fdbc(X)$  and their representations ( unlabelled derivations ) are to illustrate the above theorem; they are also implemented as a primitive arrows in the system.

Example 1

For a derivation  $F$  with conclusion

$$\text{delta\_left}(A,B,C) : A \uparrow (B \vee C) \rightarrow (A \uparrow B) \vee (A \uparrow C)$$

the representation  $R(F)$  is

$$R( \text{ROUND}( \text{LA7}(A \uparrow B, A \uparrow C), \text{LA8}(A \uparrow B, A \uparrow C), A \uparrow (B \vee C)) )$$

Example 2

For a derivation  $F$  with conclusion

$$\text{delta\_right}(A,B,C) : (A \vee B) \uparrow C \rightarrow (A \uparrow C) \vee (B \uparrow C)$$

the representation  $R(F)$  is

$$R( \text{COMP}( \text{UNION}(F1, F2), \\ \text{COMP}(F3, F4, C \uparrow (A \vee B)), \\ (C \uparrow A) \vee (C \uparrow B) ) )$$

where

$F1, F2, F3$  and  $F4$  are derivations with conclusions

$$\text{sigma}(C,A) : C \uparrow A \rightarrow A \uparrow C,$$

$$\text{sigma}(C,B) : C \uparrow B \rightarrow B \uparrow C,$$

$$\text{delta\_left}(C,A,B) : C \uparrow (A \vee B) \rightarrow (C \uparrow A) \vee (C \uparrow B) \quad \text{and}$$

$$\text{sigma}(A \vee B,C) : (A \vee B) \uparrow C \rightarrow C \uparrow (A \vee B) \quad \text{respectively.}$$

Example 3

For a derivation  $F$  with conclusion

$$\text{delta\_left\_star}(A,B,C) : (A \vee B) \uparrow (A \vee C) \rightarrow A \vee (B \uparrow C)$$

the representation  $R(F)$  is

$$R( \text{COMP}( \text{COMP}( \text{COMP}( \text{COMP}( \text{COMP}(F1, \\ \text{UNION}( \text{LA2}(A \vee (B \uparrow C)), \\ \text{UNION}( \text{LA5}(B,A), \text{LA2}(B \uparrow C) ) ), \\ (A \vee (B \uparrow C)) \vee (A \vee (B \uparrow C)) ), \\ \text{UNION}( \text{LA2}(A \vee (B \uparrow C)), F2 ), \\ (A \vee (B \uparrow C)) \vee ((B \uparrow A) \vee (B \uparrow C)) ), \\ \text{UNION}( \text{LA7}(A, B \uparrow C), \text{LA2}(B \uparrow (A \vee C)) ), \\ (A \vee (B \uparrow C)) \vee (B \uparrow (A \vee C)) ), \\ \text{UNION}( \text{LA4}(A, A \vee C), \text{LA2}(B \uparrow (A \vee C)) ), \\ A \vee (B \uparrow (A \vee C)) ) )$$

$F2,$

$$(A \uparrow (A \vee C)) \vee (B \uparrow (A \vee C)) ) )$$

where

F1 and F2 are derivations with conclusions

$\text{delta\_star}(A \vee (B \uparrow C)) : (A \vee (B \uparrow C)) \vee (A \vee (B \uparrow C)) \rightarrow A \vee (B \uparrow C)$

and

$\text{delta\_right}(A, B, A \vee C) : (A \vee B) \uparrow (A \vee C) \rightarrow \left\{ \begin{array}{l} A \uparrow (A \vee C) \\ B \uparrow (A \vee C) \end{array} \right\} \vee$

respectively .

#### Example 4

For a derivation F with conclusion

$\text{delta\_right\_star}(A, B, C) : (A \vee B) \uparrow (B \vee C) \rightarrow (A \uparrow B) \vee C$

the representation R(F) is

$R(\text{COMP}(\text{COMP}(F1, F2, C \vee (A \uparrow B)),$   
 $\text{PRODUCT}(F3, F4),$   
 $(C \vee A) \uparrow (C \vee B) ) )$

where

F1, F2, F3 and F4 are derivations with conclusions

$\text{sigma\_star}(C, A \uparrow B) : C \vee (A \uparrow B) \rightarrow (A \uparrow B) \vee C$  ,

$\text{delta\_left\_star}(C, A, B) : (C \vee A) \uparrow (C \vee B) \rightarrow C \vee (A \uparrow B)$  ,

$\text{sigma\_star}(A, C) : A \vee C \rightarrow C \vee A$  and

$\text{sigma\_star}(B, C) : B \vee C \rightarrow C \vee B$

respectively .

#### Example 5

For a derivation F with conclusion

$\text{kappa}(A, B, C, D, E) : (A \uparrow ((B \vee C) \vee D)) \uparrow E \rightarrow (B \vee ((A \uparrow C) \uparrow E)) \vee D$

the representation R(F) is

$R(\text{COMP}(\text{COMP}(\text{COMP}(\text{COMP}(\text{COMP}(\text{UNION}(\text{UNION}(\text{LA5}(A, B),$   
 $\text{LA2}((A \uparrow C) \uparrow E)),$   
 $\text{LA5}(A, D)),$   
 $\text{UNION}(\text{UNION}(\text{LA4}(A \uparrow B, E)$   
 $\text{LA2}((A \uparrow C) \uparrow D)),$   
 $\text{LA4}(A \uparrow D, E)),$   
 $((A \uparrow B) \vee ((A \uparrow C) \uparrow E)) \vee (A \uparrow D)),$   
 $\text{UNION}(F1, \text{LA2}((A \uparrow D) \uparrow E)),$   
 $((A \uparrow B) \uparrow E) \vee ((A \uparrow C) \uparrow E) \vee ((A \uparrow D) \uparrow E)),$   
 $F2,$   
 $((A \uparrow B) \vee (A \uparrow C)) \uparrow E \vee ((A \uparrow D) \uparrow E) ),$   
 $\text{PRODUCT}(\text{UNION}(F3, \text{LA2}(A \uparrow D)), \text{LA2}(E) ),$   
 $((A \uparrow B) \vee (A \uparrow C)) \vee (A \uparrow D) \uparrow E ),$   
 $\text{PRODUCT}(F4, \text{LA2}(E) ),$   
 $((A \uparrow (B \vee C)) \vee (A \uparrow D)) \uparrow E ) )$

where

F1, F2, F3 and F4 are derivations with conclusions

$$\text{delta\_right}(A \uparrow B, A \uparrow C, E) : ((A \uparrow B) \vee (A \uparrow C)) \uparrow E \rightarrow ((A \uparrow B) \uparrow E) \vee ((A \uparrow C) \uparrow E) ,$$

$$\text{delta\_right}((A \uparrow B) \vee (A \uparrow C), A \uparrow D, E) :$$

$$(((A \uparrow B) \vee (A \uparrow C)) \vee (A \uparrow D)) \uparrow E \rightarrow (((A \uparrow B) \vee (A \uparrow C)) \uparrow E) \vee ((A \uparrow D) \uparrow E) ,$$

$$\text{delta\_left}(A, B, C) : A \uparrow (B \vee C) \rightarrow (A \uparrow B) \vee (A \uparrow C) \quad \text{and}$$

$$\text{delta\_left}(A, B \vee C, D) : A \uparrow ((B \vee C) \vee D) \rightarrow (A \uparrow (B \vee C)) \vee (A \uparrow D)$$

respectively .

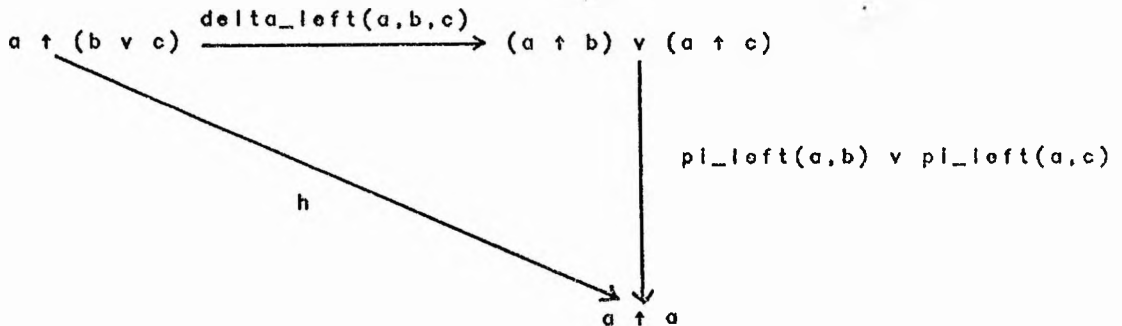
The following example is a session of input and output with the system to obtain an unlabelled distributive bicartesian representation for some arrow of  $\text{Fdbc}(X)$  :

#### Example 6

For objects  $a, b, c$  of the category  $X$  , the arrow

$$h = \text{comp}(\text{union}(\text{pi\_left}(a, b), \text{pi\_left}(a, c)), \text{delta\_left}(a, b, c))$$

in the diagram



is represented by an unlabelled derivation as follows :

```
yes
| ?- theory(dbicart).
```

```
yes
| ?- rep_of(comp(union(pi_left(a,b),pi_left(a,c)),
                    delta_left(a,b,c))).
```

The arrow is :

```
comp(union(pi_left(a,b),pi_left(a,c)),delta_left(a,b,c))
```

Representation :-

```
f43=[id(a),[a]=>[a]]
f39=[th(f43,b),[a,b]=>[a]]
f44=[id(b),[b]=>[b]]
f40=[th(f44,a),[a,b]=>[b]]
```



```

f37= ais(f39, f40, a&b), [a, b] => [a&b]
f24= id(a), [a] => [a]
f22= th(f24, b), [a, b] => [a]
f25= id(b), [b] => [b]
f23= th(f25, a), [a, b] => [b]
f21= ais(f22, f23, a&b), [a, b] => [a&b]
f20= aia(f21, a&b), [a&b] => [a&b]
f19= th(f20, a&c), [a&b] => [a&b, a&c]
f18= ois(f19, (a&b)#a&c), [a&b] => [(a&b)#a&c]
f35= cut(f37, f18, a&b), [a, b] => [(a&b)#a&c]
f45= id(a), [a] => [a]
f41= th(f45, c), [a, c] => [a]
f46= id(c), [c] => [c]
f42= th(f46, a), [a, c] => [c]
f38= ais(f41, f42, a&c), [a, c] => [a&c]
f32= id(a), [a] => [a]
f30= th(f32, c), [a, c] => [a]
f33= id(c), [c] => [c]
f31= th(f33, a), [a, c] => [c]
f29= ais(f30, f31, a&c), [a, c] => [a&c]
f28= aia(f29, a&c), [a&c] => [a&c]
f27= th(f28, a&b), [a&c] => [a&b, a&c]
f26= ois(f27, (a&b)#a&c), [a&c] => [(a&b)#a&c]
f36= cut(f38, f26, a&c), [a, c] => [(a&b)#a&c]
f34= oia(f35, f36, b#c), [a, b#c] => [(a&b)#a&c]
f17= aia(f34, a&b#c), [a&b#c] => [(a&b)#a&c]
f5= id(a), [a] => [a]
f4= th(f5, b), [a, b] => [a]
f3= aia(f4, a&b), [a&b] => [a]
f11= id(a), [a] => [a]
f10= th(f11, a), [a] => [a, a]
f9= ois(f10, a#a), [a] => [a#a]
f12= cut(f3, f9, a), [a&b] => [a#a]
f8= id(a), [a] => [a]
f7= th(f8, c), [a, c] => [a]
f6= aia(f7, a&c), [a&c] => [a]
f15= id(a), [a] => [a]
f14= th(f15, a), [a] => [a, a]
f13= ois(f14, a#a), [a] => [a#a]
f16= cut(f6, f13, a), [a&c] => [a#a]
f2= oia(f12, f16, (a&b)#a&c), [(a&b)#a&c] => [a#a]
f1= cut(f17, f2, (a&b)#a&c), [a&b#c] => [a#a]

yes
| ?-

```

### (3.5) The Syntax of Fdbc(X)

The advantages of Ddbc(X) over LDdbc(X) lie in the fact that the derivations of Ddbc(X) code their own labels and provide cut-free representations of ArFdbc(X) as follows .

#### (3.5.1) The Cut-Elimination Theorem for Ddbc(X)

Every derivation in Ddbc(X) is equivalent to a cut-free in Ddbc(X).

##### Proof

Because of the generality of rules (R1), (R10) and (R12) in Ddbc(X) we must review cases (C.3), (C.4), (C.8-9), (C.12), (C.14), (C.15), (C.18), (C.19), (C.20), (C.21), (C.22), (C.26), (C.27), (C.29), (C.34), (C.35), (C.36), (C.37), (C.38), (C.42), (C.43) and (C.44) mentioned in (1.5.2) and (2.5.2) to become general cases; the above clauses and the following additional ones of the same algorithm of appendix (2) : (C.28) and (C.41) prove the theorem .

###



(E.17), (E.29) and (E.34), in order to meet the generality of rules (R10) and (R12) and the following additional clauses (E.11), (E.19), (E.27), (E.32), (E.41.2) and (E.42.1) in Appendix (3), we can prove the lemma .

###

(3.5.8) Definition

A derivation  $p$  in  $Ddbc(X)$  is defined to be normal if it is normal in the sense of definition (2.5.9) after replacing condition (4) by the more general form

- 4')  $p$  contains no subderivations constructed with (R2) or (R5) whose active formulae are of the form  $a \uparrow b$  or  $a \vee b$  .

(3.5.9) The Normalization Theorem for Ddbc(X)

Every derivation  $p$  in  $Ddbc(X)$  is equivalent to a unique normal derivation  $q$  in  $Ddbc(X)$  .

Proof

The theorem follows from theorem (2.5.10) with clauses (D.7), (D.37), (D.49) and (D.59) generalized to allow non-empty sequences  $F, E, G$  and  $D$ , together with the following conditions of Appendix (4) : (D.9) , (D.35) , (D.48) , (D.50) , (D.51) , (D.54) and (D.58) .

###

The following examples are sessions of input and output with the system for derivations of  $Ddbc(X)$  and their equivalents of cut-free, contraction & interchange-free and normal form derivations; in order to illustrate the above processes and to satisfy the definition of normal form :

Example 1

For an arrow  $f$  in  $X(a,d)$ , where  $a,b,c,d$  are objects of  $X$  , the derivation

$$\begin{array}{c}
 \begin{array}{ccc}
 \frac{a \rightarrow a}{a,b \rightarrow a} & \frac{\begin{array}{c} f \\ a \rightarrow d \end{array}}{a,c \rightarrow d} & d \rightarrow d \\
 \frac{a,b \rightarrow a, d}{a,b \rightarrow a, d} & \frac{a,c \rightarrow d}{a,c \rightarrow a, d} & \frac{d \rightarrow d}{d, b \rightarrow d} \\
 \frac{a,b \vee c \rightarrow a, d}{a,b \vee c \rightarrow a, d} & c, d, b \rightarrow d & \frac{b \rightarrow b}{c \uparrow a, b \rightarrow b} \\
 \frac{c, a, b \vee c, b \rightarrow a, d}{c \uparrow a, b \vee c, b \rightarrow a, d} & & \frac{c \uparrow a, b \vee c, b \rightarrow b}{c \uparrow a, b \vee c, b \rightarrow b, d} \\
 \frac{c \uparrow a, b \vee c, b \rightarrow a, d}{c \uparrow a, b \vee c, b \rightarrow a \uparrow b, d} & & \\
 \frac{(c \uparrow a) \uparrow (b \vee c), b \rightarrow a \uparrow b, d}{(c \uparrow a) \uparrow (b \vee c), b \rightarrow (a \uparrow b) \vee d} & & 
 \end{array}
 \end{array}$$

is transformed to its normal form as follows :

```

yes
| ?- theory(dblcart).
yes
| ?- prove(s1 = [id(a), [a] => [a]]).
yes
| ?- prove(s2 = [th(s1,b), [a,b] => [a]]).
yes
| ?- prove(s3 = [th(s2,d), [a,b] => [a,d]]).
yes
| ?- prove(s4 = [f, [a] => [d]]).
yes
| ?- prove(s5 = [th(s4,c), [a,c] => [d]]).
yes
| ?- prove(s6 = [th(s5,a), [a,c] => [a,d]]).
yes
| ?- prove(s7 = [ois(s3,s6,b # c), [a,b # c] => [a,d]]).
yes
| ?- prove(s8 = [id(d), [d] => [d]]).
yes
| ?- prove(s9 = [th(s8,b), [d,b] => [d]]).
yes
| ?- prove(s10 = [th(s9,c), [c,d,b] => [d]]).
yes
| ?- prove(s11 = [cut(s7,s10,d), [c,a,b#c,b] => [a,d]]).
yes
| ?- prove(s12 = [aia(s11,c&a), [c&a,b#c,b] => [a,d]]).
yes
| ?- prove(s13 = [id(b), [b] => [b]]).
yes
| ?- prove(s14 = [th(s13,c&a), [c&a,b] => [b]]).
yes
| ?- prove(s15 = [th(s14,b#c), [c&a,b#c,b] => [b]]).
yes
| ?- prove(s16 = [th(s15,d), [c&a,b#c,b] => [b,d]]).
yes
| ?- prove(s17 = [ais(s12,s16,a&b), [c&a,b#c,b] => [a&b,d]]).
yes
| ?- prove(s18=[aia(s17,(c&a)&(b#c)), [(c&a)&(b#c),b]=>[a&b,d]]).
yes
| ?- prove(s19 = [ois(s18,(a&b)#d), [(c&a)&(b#c),b]=>[(a&b)#d]]).
yes
| ?- normal(s19).

```

Proof is as follows :

=====

```

s1=[id(a),[a]=>[a]]
s2=[th(s1,b),[a,b]=>[a]]
s3=[th(s2,d),[a,b]=>[a,d]]
s4=[f,[a]=>[d]]
s5=[th(s4,c),[a,c]=>[d]]
s6=[th(s5,a),[a,c]=>[a,d]]
s7=[ois(s3,s6,b#c),[a,b#c]=>[a,d]]
s8=[id(d),[d]=>[d]]
s9=[th(s8,b),[d,b]=>[d]]
s10=[th(s9,c),[c,d,b]=>[d]]
s11=[cut(s7,s10,d),[c,a,b#c,b]=>[a,d]]
s12=[aia(s11,c&a),[c&a,b#c,b]=>[a,d]]
s13=[id(b),[b]=>[b]]
s14=[th(s13,c&a),[c&a,b]=>[b]]
s15=[th(s14,b#c),[c&a,b#c,b]=>[b]]
s16=[th(s15,d),[c&a,b#c,b]=>[b,d]]
s17=[ais(s12,s16,a&b),[c&a,b#c,b]=>[a&b,d]]
s18=[aia(s17,(c&a)&(b#c)),[(c&a)&(b#c),b]=>[a&b,d]]
s19=[ois(s18,(a&b)#d),[(c&a)&(b#c),b]=>[(a&b)#d]]

```

Cut\_free proof is :

=====

```

s1=[id(a),[a]=>[a]]
f13=[th(s1,b),[a,b]=>[a]]
f9=[th(f13,d),[a,b]=>[a,d]]
f19=[f,[a]=>[d]]
f16=[th(f19,c),[a,c]=>[d]]

```

```

f12=[th(f16,a),[a,c]=>[a,d]]
f6=[oia(f9,f12,b#c),[a,b#c]=>[a,d]]
f3=[th(f6,b),[a,b#c,b]=>[a,d]]
s11=[th(f3,c),[c,a,b#c,b]=>[a,d]]
s12=[aia(s11,c&a),[c&a,b#c,b]=>[a,d]]
s13=[ld(b),[b]=>[b]]
s14=[th(s13,c&a),[c&a,b]=>[b]]
s15=[th(s14,b#c),[c&a,b#c,b]=>[b]]
s16=[th(s15,d),[c&a,b#c,b]=>[b,d]]
s17=[ois(s12,s16,a&b),[c&a,b#c,b]=>[a&b,d]]
s18=[aia(s17,(c&a)&b#c),[(c&a)&b#c,b]=>[a&b,d]]
s19=[ois(s18,(a&b)#d),[(c&a)&b#c,b]=>[(a&b)#d]]

```

Contraction and Interchange free proof is :

```

s1=[ld(a),[a]=>[a]]
f13=[th(s1,b),[a,b]=>[a]]
f9=[th(f13,d),[a,b]=>[a,d]]
f19=[f,[a]=>[d]]
f16=[th(f19,c),[a,c]=>[d]]
f12=[th(f16,a),[a,c]=>[a,d]]
f6=[oia(f9,f12,b#c),[a,b#c]=>[a,d]]
f3=[th(f6,b),[a,b#c,b]=>[a,d]]
s11=[th(f3,c),[c,a,b#c,b]=>[a,d]]
s12=[aia(s11,c&a),[c&a,b#c,b]=>[a,d]]
s13=[ld(b),[b]=>[b]]
f21=[th(s13,a),[a,b]=>[b]]
f20=[th(f21,c),[c,a,b]=>[b]]
s14=[aia(f20,c&a),[c&a,b]=>[b]]
f22=[th(s14,b),[c&a,b,b]=>[b]]
s13=[ld(b),[b]=>[b]]
f21=[th(s13,a),[a,b]=>[b]]
f20=[th(f21,c),[c,a,b]=>[b]]
f24=[aia(f20,c&a),[c&a,b]=>[b]]
f23=[th(f24,c),[c&a,c,b]=>[b]]
s15=[oia(f22,f23,b#c),[c&a,b#c,b]=>[b]]
s16=[th(s15,d),[c&a,b#c,b]=>[b,d]]
s17=[ais(s12,s16,a&b),[c&a,b#c,b]=>[a&b,d]]
s18=[aia(s17,(c&a)&b#c),[(c&a)&b#c,b]=>[a&b,d]]
s19=[ois(s18,(a&b)#d),[(c&a)&b#c,b]=>[(a&b)#d]]

```

Normal proof is :

```

s1=[ld(a),[a]=>[a]]
f13=[th(s1,d),[a]=>[a,d]]
f9=[th(f13,b),[a,b]=>[a,d]]
f19=[f,[a]=>[d]]
f16=[th(f19,a),[a]=>[a,d]]
f12=[th(f16,c),[a,c]=>[a,d]]
f6=[oia(f9,f12,b#c),[a,b#c]=>[a,d]]
f3=[th(f6,b),[a,b#c,b]=>[a,d]]
s11=[th(f3,c),[c,a,b#c,b]=>[a,d]]
s12=[aia(s11,c&a),[c&a,b#c,b]=>[a,d]]
s13=[ld(b),[b]=>[b]]
f21=[th(s13,d),[b]=>[b,d]]
f20=[th(f21,a),[a,b]=>[b,d]]
s14=[th(f20,c),[c,a,b]=>[b,d]]
f22=[aia(s14,c&a),[c&a,b]=>[b,d]]
s15=[th(f22,b),[c&a,b,b]=>[b,d]]
s13=[ld(b),[b]=>[b]]
f21=[th(s13,d),[b]=>[b,d]]
f20=[th(f21,a),[a,b]=>[b,d]]
f24=[th(f20,c),[c,a,b]=>[b,d]]
f23=[aia(f24,c&a),[c&a,b]=>[b,d]]
f25=[th(f23,c),[c&a,c,b]=>[b,d]]
s16=[oia(s15,f25,b#c),[c&a,b#c,b]=>[b,d]]
s17=[ais(s12,s16,a&b),[c&a,b#c,b]=>[a&b,d]]
s18=[ois(s17,(a&b)#d),[c&a,b#c,b]=>[(a&b)#d]]
s19=[aia(s18,(c&a)&b#c),[(c&a)&b#c,b]=>[(a&b)#d]]

```

yes  
| ?-

which in tree form is :

$$\begin{array}{c}
 \frac{a \rightarrow a}{a \rightarrow a, d} \quad \frac{a \xrightarrow{f} d}{a \rightarrow a, d} \\
 \frac{a, b \rightarrow a, d}{a, b \vee c \rightarrow a, d} \quad \frac{a, c \rightarrow a, d}{a, b \vee c \rightarrow a, d} \\
 \frac{a, b \vee c \rightarrow a, d}{a, b \vee c, b \rightarrow a, d} \quad \frac{c, a, b \rightarrow b, d}{c, a, b \rightarrow b, d} \\
 \frac{c, a, b \vee c, b \rightarrow a, d}{c \uparrow a, b \vee c, b \rightarrow a, d} \quad \frac{c \uparrow a, b \rightarrow b, d}{c \uparrow a, b, b \rightarrow b, d} \quad \frac{c \uparrow a, b \rightarrow b, d}{c \uparrow a, c, b \rightarrow b, d} \\
 \frac{c \uparrow a, b \vee c, b \rightarrow a, d}{c \uparrow a, b \vee c, b \rightarrow (a \uparrow b) \vee d} \\
 \frac{c \uparrow a, b \vee c, b \rightarrow (a \uparrow b) \vee d}{(c \uparrow a) \uparrow (b \vee c), b \rightarrow (a \uparrow b) \vee d}
 \end{array}$$

### Example 2

For objects  $a, b, c, d$  in the category  $X$ , the derivation

$$\begin{array}{c}
 \frac{a \rightarrow a}{a, b \rightarrow a} \quad \frac{b \rightarrow b}{a, b \rightarrow b} \quad \frac{d \rightarrow d}{d, a \uparrow b \rightarrow d} \quad \frac{a \rightarrow a}{a, b \rightarrow a} \quad \frac{b \rightarrow b}{a, b \rightarrow b} \\
 \frac{a, b \rightarrow a \quad a, b \rightarrow b}{a, b \rightarrow a \uparrow b} \quad \frac{d, a \uparrow b \rightarrow d}{d, a \uparrow b \rightarrow d \uparrow (a \uparrow b)} \\
 \frac{d \rightarrow d}{d, a \uparrow b \rightarrow d} \quad \frac{a \uparrow b \rightarrow a \uparrow b}{d, a \uparrow b \rightarrow a \uparrow b} \quad \frac{d \uparrow (a \uparrow b) \rightarrow d \uparrow (a \uparrow b)}{d \uparrow (a \uparrow b) \rightarrow d \uparrow (a \uparrow b), d \uparrow c} \\
 \frac{d, a \uparrow b \rightarrow d \uparrow (a \uparrow b)}{d, a \uparrow b \rightarrow (d \uparrow (a \uparrow b)) \vee (d \uparrow c)}
 \end{array}$$

is transformed to a normal form as follows :

```

yes
| ?- theory(dbicart).
yes
| ?- prove(s1 = [id(d), [d] => [d]]).
yes
| ?- prove(s2 = [th(s1, (a&b)#c), [(a&b)#c, d] => [d]]).
yes
| ?- prove(s3 = [aia(s2, ((a&b)#c)&d], [((a&b)#c)&d] => [d]]).
yes
| ?- prove(s4 = [id(a), [a] => [a]]).
yes
| ?- prove(s5 = [th(s4, b), [a, b] => [a]]).
yes
| ?- prove(s6 = [id(b), [b] => [b]]).
yes
| ?- prove(s7 = [th(s6, a), [a, b] => [b]]).
yes
| ?- prove(s8 = [ais(s5, s7, a&b), [a, b] => [a&b]]).
yes
| ?- prove(s9 = [aia(s8, a&b), [a&b] => [a&b]]).
yes
| ?- prove(s10 = [th(s9, c), [a&b] => [a&b, c]]).
yes
| ?- prove(s11 = [id(c), [c] => [c]]).
yes
| ?- prove(s12 = [th(s11, a&b), [c] => [a&b, c]]).
yes

```

```

| ?- prove(s13 = [oia(s10,s12,(a&b)#c), [(a&b)#c] => [a&b,c]]).
yes
| ?- prove(s14 = [ois(s13,(a&b)#c), [(a&b)#c] => [(a&b)#c]]).
yes
| ?- prove(s15 = [th(s14,d), [(a&b)#c,d] => [(a&b)#c]]).
yes
| ?- prove(s16 = [aia(s15,((a&b)#c)&d), [((a&b)#c)&d] =>
[(a&b)#c]]).
yes
| ?- prove(s17 = [ais(s3,s16,d&((a&b)#c)), [((a&b)#c)&d] =>
[d&((a&b)#c)]]).
yes
| ?- prove(s18 = [th(s1,a&b), [d,a&b] => [d]]).
yes
| ?- prove(s19 = [th(s9,d), [d,a&b] => [a&b]]).
yes
| ?- prove(s20 = [ais(s18,s19,d&(a&b)), [d,a&b] => [d&(a&b)]]).
yes
| ?- prove(s21 = [aia(s20,d&(a&b)), [d&(a&b)] => [d&(a&b)]]).
yes
| ?- prove(s22 = [th(s21,d&c), [d&(a&b)] => [d&(a&b),d&c]]).
yes
| ?- prove(s23 = [ois(s22,(d&(a&b))#(d&c)), [d&(a&b)] =>
[(d&(a&b))#(d&c)]]).
yes
| ?- prove(s24 = [cut(s20,s23,d&(a&b)), [d,a&b] =>
[(d&(a&b))#(d&c)]]).
yes
| ?- normal(s24).

```

Proof is as follows :

=====

```

f15=[id(d),[d]=>[d]]
s18=[th(f15,a&b),[d,a&b]=>[d]]
f30=[id(a),[a]=>[a]]
f27=[th(f30,b),[a,b]=>[a]]
s6=[id(b),[b]=>[b]]
f32=[th(s6,a),[a,b]=>[b]]
s8=[ais(f27,f32,a&b),[a,b]=>[a&b]]
f24=[aia(s8,a&b),[a&b]=>[a&b]]
f21=[th(f24,d),[d,a&b]=>[a&b]]
f12=[ais(s18,f21,d&a&b),[d,a&b]=>[d&a&b]]
f2=[id(d),[d]=>[d]]
f14=[th(f2,a&b),[d,a&b]=>[d]]
f29=[id(a),[a]=>[a]]
f7=[th(f29,b),[a,b]=>[a]]
f34=[id(b),[b]=>[b]]
f31=[th(f34,a),[a,b]=>[b]]
f26=[ais(f7,f31,a&b),[a,b]=>[a&b]]
f23=[aia(f26,a&b),[a&b]=>[a&b]]
s19=[th(f23,d),[d,a&b]=>[a&b]]
f11=[ais(f14,s19,d&a&b),[d,a&b]=>[d&a&b]]
s21=[aia(f11,d&a&b),[d&a&b]=>[d&a&b]]
s22=[th(s21,d&c),[d&a&b]=>[d&a&b,d&c]]
s23=[ois(s22,(d&a&b))#d&c],[d&a&b]=>[(d&a&b)#d&c]]
s24=[cut(f12,s23,d&a&b),[d,a&b]=>[(d&a&b)#d&c]]

```

Cut\_free proof is :

=====

```

f181=[id(d),[d]=>[d]]
f178=[th(f181,a&b),[d,a&b]=>[d]]
f175=[th(f178,b),[d,a&b,b]=>[d]]
f172=[th(f175,a),[d,a&b,a,b]=>[d]]
f169=[th(f172,b),[d,a&b,a,b,b]=>[d]]
f166=[th(f169,a),[d,a&b,a,b,a,b]=>[d]]
f163=[inc(f166,a,b),[d,a&b,a,a,b,b]=>[d]]
f160=[con(f163,a),[d,a&b,a,b,b]=>[d]]
f157=[con(f160,b),[d,a&b,a,b]=>[d]]
f154=[aia(f157,a&b),[d,a&b,a&b]=>[d]]
f148=[th(f154,d),[d,a&b,d,a&b]=>[d]]
f138=[id(a),[a]=>[a]]
f135=[th(f138,b),[a,b]=>[a]]
f132=[th(f135,b),[a,b,b]=>[a]]
f126=[th(f132,a),[a,b,a,b]=>[a]]
f145=[id(b),[b]=>[b]]
f144=[th(f145,a),[a,b]=>[b]]
f141=[th(f144,b),[a,b,b]=>[b]]
f129=[th(f141,a),[a,b,a,b]=>[b]]

```

```

f102=[als(f126,f129,a&b),[a,b,a,b]=>[a&b]]
f105=[inc(f102,a,b),[a,a,b,b]=>[a&b]]
f106=[con(f105,a),[a,b,b]=>[a&b]]
f97=[con(f106,b),[a,b]=>[a&b]]
f188=[aia(f97,a&b),[a&b]=>[a&b]]
f187=[th(f188,d),[d,a&b]=>[a&b]]
f184=[th(f187,a&b),[d,a&b,a&b]=>[a&b]]
f151=[th(f184,d),[d,a&b,d,a&b]=>[a&b]]
f45=[als(f148,f151,d&a&b),[d,a&b,d,a&b]=>[d&a&b]]
f48=[inc(f45,d,a&b),[d,d,a&b,a&b]=>[d&a&b]]
f49=[con(f48,d),[d,a&b,a&b]=>[d&a&b]]
f40=[con(f49,a&b),[d,a&b]=>[d&a&b]]
f37=[th(f40,d&c),[d,a&b]=>[d&a&b,d&c]]
s24=[ois(f37,(d&a&b)#d&c),[d,a&b]=>[(d&a&b)#d&c]]

```

Contraction and Interchange free proof is :

```

=====
f181=[id(d),[d]=>[d]]
f190=[th(f181,b),[d,b]=>[d]]
f189=[th(f190,a),[d,a,b]=>[d]]
f49=[aia(f189,a&b),[d,a&b]=>[d]]
f138=[id(a),[a]=>[a]]
f106=[th(f138,b),[a,b]=>[a]]
f145=[id(b),[b]=>[b]]
f193=[th(f145,a),[a,b]=>[b]]
f97=[als(f106,f193,a&b),[a,b]=>[a&b]]
f195=[aia(f97,a&b),[a&b]=>[a&b]]
f196=[th(f195,d),[d,a&b]=>[a&b]]
f40=[als(f49,f196,d&a&b),[d,a&b]=>[d&a&b]]
f197=[th(f40,d),[d,a&b]=>[d&a&b,d]]
f215=[id(d),[d]=>[d]]
f214=[th(f215,b),[d,b]=>[d]]
f213=[th(f214,a),[d,a,b]=>[d]]
f201=[aia(f213,a&b),[d,a&b]=>[d]]
f231=[id(a),[a]=>[a]]
f224=[th(f231,b),[a,b]=>[a]]
f237=[id(b),[b]=>[b]]
f225=[th(f237,a),[a,b]=>[b]]
f222=[als(f224,f225,a&b),[a,b]=>[a&b]]
f216=[aia(f222,a&b),[a&b]=>[a&b]]
f202=[th(f216,d),[d,a&b]=>[a&b]]
f199=[als(f201,f202,d&a&b),[d,a&b]=>[d&a&b]]
f198=[th(f199,c),[d,a&b]=>[d&a&b,c]]
f37=[als(f197,f198,d&c),[d,a&b]=>[d&a&b,d&c]]
s24=[ois(f37,(d&a&b)#d&c),[d,a&b]=>[(d&a&b)#d&c]]

```

Normal proof is :

```

=====
f181=[id(d),[d]=>[d]]
f190=[th(f181,d),[d]=>[d,d]]
f189=[th(f190,b),[d,b]=>[d,d]]
f49=[th(f189,a),[d,a,b]=>[d,d]]
f40=[aia(f49,a&b),[d,a&b]=>[d,d]]
f138=[id(a),[a]=>[a]]
f106=[th(f138,d),[a]=>[a,d]]
f97=[th(f106,b),[a,b]=>[a,d]]
f145=[id(b),[b]=>[b]]
f193=[th(f145,d),[b]=>[b,d]]
f245=[th(f193,a),[a,b]=>[b,d]]
f195=[als(f97,f245,a&b),[a,b]=>[a&b,d]]
f196=[aia(f195,a&b),[a&b]=>[a&b,d]]
f243=[th(f196,d),[d,a&b]=>[a&b,d]]
f197=[als(f40,f243,d&a&b),[d,a&b]=>[d&a&b,d]]
f215=[id(d),[d]=>[d]]
f214=[th(f215,c),[d]=>[d,c]]
f213=[th(f214,b),[d,b]=>[d,c]]
f201=[th(f213,a),[d,a,b]=>[d,c]]
f199=[aia(f201,a&b),[d,a&b]=>[d,c]]
f231=[id(a),[a]=>[a]]
f224=[th(f231,c),[a]=>[a,c]]
f222=[th(f224,b),[a,b]=>[a,c]]
f237=[id(b),[b]=>[b]]
f225=[th(f237,c),[b]=>[b,c]]
f244=[th(f225,a),[a,b]=>[b,c]]
f216=[als(f222,f244,a&b),[a,b]=>[a&b,c]]
f202=[aia(f216,a&b),[a&b]=>[a&b,c]]
f242=[th(f202,d),[d,a&b]=>[a&b,c]]
f198=[als(f199,f242,d&a&b),[d,a&b]=>[d&a&b,c]]

```



```
f37=[ois(f197,f198,d&c),[d,a&b]=>[d&a&b,d&c]]
s24=[ois(f37,(d&a&b)#d&c),[d,a&b]=>[(d&a&b)#d&c]]
```

which in tree form is :

	$a \rightarrow a$	$b \rightarrow b$		$a \rightarrow a$	$b \rightarrow b$
$d \rightarrow d$	$a \rightarrow a, d$	$b \rightarrow b, d$	$d \rightarrow d$	$a \rightarrow a, c$	$b \rightarrow b, c$
$d \rightarrow d, d$	$a, b \rightarrow a, d$	$a, b \rightarrow b, d$	$d \rightarrow d, c$	$a, b \rightarrow a, c$	$a, b \rightarrow b, c$
$d, b \rightarrow d, d$	$a, b \rightarrow a \uparrow b, d$		$d, b \rightarrow d, c$	$a, b \rightarrow a \uparrow b, c$	
$d, a, b \rightarrow d, d$	$a \uparrow b \rightarrow a \uparrow b, d$		$d, a, b \rightarrow d, c$	$a \uparrow b \rightarrow a \uparrow b, c$	
$d, a \uparrow b \rightarrow d, d$	$d, a \uparrow b \rightarrow a \uparrow b, d$		$d, a \uparrow b \rightarrow d, c$	$d, a \uparrow b \rightarrow a \uparrow b, c$	
$d, a \uparrow b \rightarrow d \uparrow (a \uparrow b), d$			$d, a \uparrow b \rightarrow d \uparrow (a \uparrow b), c$		
$d, a \uparrow b \rightarrow (d \uparrow (a \uparrow b)) \vee (d \uparrow c)$					

### (3.5.10) The CHURCH-ROSSER Theorem for $Ddbc(X)$

If  $p = q$ , then there exists a normal  $r$  in  $Ddbc(X)$  such that  $p \geq r$  and  $q \geq r$ .

### (3.6) Application to Category Theory

In this section, we use the system to represent some arrows of  $Fdbc(X)$  and obtain their normal representations.

Example The normal representation of the following arrow

$$a \uparrow (b \vee c) \xrightarrow{\text{delta\_left}(a,b,c)} (a \uparrow b) \vee (a \uparrow c)$$

is obtained as follows :

```
yes
| ?- theory(dbicart).
yes
| ?- norm_rep(delta_left(a,b,c)).
```

The arrow is :

```
delta_left(a,b,c)
```

Proof is as follows :

```
f27=[id(a),[a]=>[a]]
f23=[th(f27,b),[a,b]=>[a]]
f28=[id(b),[b]=>[b]]
f24=[th(f28,a),[a,b]=>[b]]
f21=[ois(f23,f24,a&b),[a,b]=>[a&b]]
f8=[id(a),[a]=>[a]]
f6=[th(f8,b),[a,b]=>[a]]
f9=[id(b),[b]=>[b]]
f7=[th(f9,a),[a,b]=>[b]]
f5=[ois(f6,f7,a&b),[a,b]=>[a&b]]
f4=[ois(f5,a&b),[a&b]=>[a&b]]
f3=[th(f4,a&c),[a&b]=>[a&b,a&c]]
f2=[ois(f3,(a&b)#a&c),[a&b]=>[(a&b)#a&c]]
f19=[cut(f21,f2,a&b),[a,b]=>[(a&b)#a&c]]
f29=[id(a),[a]=>[a]]
f25=[th(f29,c),[a,c]=>[a]]
f30=[id(c),[c]=>[c]]
f26=[th(f30,a),[a,c]=>[c]]
```

```

f22= [a]s(f25, f26, a&c). [a, c] => [a&c]
f16= [d(a), [a] => [a]]
f14= th(f16, c). [a, c] => [a]
f17= [d(c), [c] => [c]]
f15= th(f17, a). [a, c] => [c]
f13= [a]s(f14, f15, a&c). [a, c] => [a&c]
f12= [a]a(f13, a&c). [a&c] => [a&c]
f11= th(f12, a&b). [a&c] => [a&b, a&c]
f10= [a]s(f11, (a&b)#a&c). [a&c] => [(a&b)#a&c]
f20= cut(f22, f10, a&c). [a, c] => [(a&b)#a&c]
f18= [a]a(f19, f20, b#c). [a, b#c] => [(a&b)#a&c]
f1= [a]a(f18, a&b#c). [a&b#c] => [(a&b)#a&c]

```

Cut-free proof is :

```

f77= [d(a), [a] => [a]]
f74= th(f77, b). [a, b] => [a]
f71= th(f74, b). [a, b, b] => [a]
f65= th(f71, a). [a, b, a, b] => [a]
f84= [d(b), [b] => [b]]
f83= th(f84, a). [a, b] => [b]
f80= th(f83, b). [a, b, b] => [b]
f80= th(f80, a). [a, b, a, b] => [b]
f41= [a]s(f65, f68, a&b). [a, b, a, b] => [a&b]
f44= [inc(f41, a, b). [a, a, b, b] => [a&b]
f45= [con(f44, a). [a, b, b] => [a&b]
f36= [con(f45, b). [a, b] => [a&b]
f33= th(f36, a&c). [a, b] => [a&b, a&c]
f19= [a]s(f33, (a&b)#a&c). [a, b] => [(a&b)#a&c]
f131= [d(a), [a] => [a]]
f128= th(f131, c). [a, c] => [a]
f125= th(f128, c). [a, c, c] => [a]
f119= th(f125, a). [a, c, a, c] => [a]
f138= [d(c), [c] => [c]]
f137= th(f138, a). [a, c] => [c]
f134= th(f137, c). [a, c, c] => [c]
f122= th(f134, a). [a, c, a, c] => [c]
f95= [a]s(f119, f122, a&c). [a, c, a, c] => [a&c]
f98= [inc(f95, a, c). [a, a, c, c] => [a&c]
f99= [con(f98, a). [a, c, c] => [a&c]
f90= [con(f99, c). [a, c] => [a&c]
f87= th(f90, a&b). [a, c] => [a&b, a&c]
f20= [a]s(f87, (a&b)#a&c). [a, c] => [(a&b)#a&c]
f18= [a]a(f19, f20, b#c). [a, b#c] => [(a&b)#a&c]
f1= [a]a(f18, a&b#c). [a&b#c] => [(a&b)#a&c]

```

Contraction and Interchange free proof is :

```

f77= [d(a), [a] => [a]]
f45= th(f77, b). [a, b] => [a]
f84= [d(b), [b] => [b]]
f141= th(f84, a). [a, b] => [b]
f36= [a]s(f45, f141, a&b). [a, b] => [a&b]
f142= th(f36, a). [a, b] => [a&b, a]
f153= [d(a), [a] => [a]]
f146= th(f153, b). [a, b] => [a]
f159= [d(b), [b] => [b]]
f147= th(f159, a). [a, b] => [b]
f144= [a]s(f146, f147, a&b). [a, b] => [a&b]
f143= th(f144, c). [a, b] => [a&b, c]
f33= [a]s(f142, f143, a&c). [a, b] => [a&b, a&c]
f19= [a]s(f33, (a&b)#a&c). [a, b] => [(a&b)#a&c]
f131= [d(a), [a] => [a]]
f99= th(f131, c). [a, c] => [a]
f138= [d(c), [c] => [c]]
f162= th(f138, a). [a, c] => [c]
f90= [a]s(f99, f162, a&c). [a, c] => [a&c]
f163= th(f90, a). [a, c] => [a, a&c]
f174= [d(a), [a] => [a]]
f167= th(f174, c). [a, c] => [a]
f180= [d(c), [c] => [c]]
f168= th(f180, a). [a, c] => [c]
f165= [a]s(f167, f168, a&c). [a, c] => [a&c]
f164= th(f165, b). [a, c] => [b, a&c]
f87= [a]s(f163, f164, a&b). [a, c] => [a&b, a&c]
f20= [a]s(f87, (a&b)#a&c). [a, c] => [(a&b)#a&c]
f18= [a]a(f19, f20, b#c). [a, b#c] => [(a&b)#a&c]
f1= [a]a(f18, a&b#c). [a&b#c] => [(a&b)#a&c]

```

Normal proof is :

```

f77= [id(a), [a] => [a]]
f45= [th(f77, a), [a] => [a, a]]
f36= [th(f45, b), [a, b] => [a, a]]
f84= [id(b), [b] => [b]]
f141= [th(f84, a), [b] => [b, a]]
f184= [th(f141, a), [a, b] => [b, a]]
f142= [ais(f36, f184, a&b), [a, b] => [a&b, a]]
f153= [id(a), [a] => [a]]
f146= [th(f153, c), [a] => [a, c]]
f144= [th(f146, b), [a, b] => [a, c]]
f159= [id(b), [b] => [b]]
f147= [th(f159, c), [b] => [b, c]]
f183= [th(f147, a), [a, b] => [b, c]]
f143= [ais(f144, f183, a&b), [a, b] => [a&b', c]]
f33= [ais(f142, f143, a&c), [a, b] => [c&b, a&c]]
f19= [ois(f33, (a&b)#a&c), [a, b] => [(a&b)#a&c]]
f131= [id(a), [a] => [a]]
f99= [th(f131, a), [a] => [a, a]]
f90= [th(f99, c), [a, c] => [a, a]]
f138= [id(c), [c] => [c]]
f162= [th(f138, a), [c] => [a, c]]
f182= [th(f162, a), [a, c] => [a, c]]
f163= [ais(f90, f182, a&c), [a, c] => [a, a&c]]
f174= [id(a), [a] => [a]]
f167= [th(f174, b), [a] => [b, a]]
f165= [th(f167, c), [a, c] => [b, a]]
f180= [id(c), [c] => [c]]
f168= [th(f180, b), [c] => [b, c]]
f181= [th(f168, a), [a, c] => [b, c]]
f164= [ais(f165, f181, a&c), [a, c] => [b, a&c]]
f87= [ais(f163, f164, a&b), [a, c] => [a&b, a&c]]
f20= [ois(f87, (a&b)#a&c), [a, c] => [(a&b)#a&c]]
f18= [ois(f19, f20, b#c), [a, b#c] => [(a&b)#a&c]]
f1= [ais(f18, a&b#c), [a&b#c] => [(a&b)#a&c]]
    
```

yes  
|  
?-

which in tree form is :

$a \rightarrow a$	$b \rightarrow b$	$a \rightarrow a$	$b \rightarrow b$	$a \rightarrow a$	$c \rightarrow c$	$a \rightarrow a$	$c \rightarrow c$
$a \rightarrow a, a$	$b \rightarrow b, a$	$a \rightarrow a, c$	$b \rightarrow b, c$	$a \rightarrow a, a$	$c \rightarrow a, c$	$a \rightarrow b, a$	$c \rightarrow b, c$
$a, b \rightarrow a, a$	$a, b \rightarrow b, a$	$a, b \rightarrow a, c$	$a, b \rightarrow b, c$	$a, c \rightarrow a, a$	$a, c \rightarrow a, c$	$a, c \rightarrow b, a$	$a, c \rightarrow b, c$
$a, b \rightarrow a \uparrow b, a$	$a, b \rightarrow a \uparrow b, c$	$a, c \rightarrow a, a \uparrow c$	$a, c \rightarrow b, a \uparrow c$				
$a, b \rightarrow a \uparrow b, a \uparrow c$	$a, c \rightarrow a \uparrow b, a \uparrow c$						
$a, b \rightarrow (a \uparrow b) \vee (a \uparrow c)$	$a, c \rightarrow (a \uparrow b) \vee (a \uparrow c)$						
$a, b \vee c \rightarrow (a \uparrow b) \vee (a \uparrow c)$							
$a \uparrow (b \vee c) \rightarrow (a \uparrow b) \vee (a \uparrow c)$							

## Chapter 4

## CARTESIAN CLOSED CATEGORY THEORY

In this chapter, we represent the class of small cartesian closed categories and implement SZABO's algorithms for representing arrows from the free cartesian closed category  $Fccl(X)$  over  $X$  by means of derivations from the class  $Dccl(X)$  of unlabelled 'cartesian-closed-derivations' over  $X$ ; for interpreting derivations of this class as arrows of  $Fccl(X)$ ; for elimination of cuts in, and normalization of, such derivations; and, finally, for giving the normal representation of an arrow of  $Fccl(X)$ .

Several lemmas concerning the cartesian closed category theory are stated without proof in SZABO :

- i) Lemma (4.5.4), concerning derivations with terminal arrows of the category  $Fccl(X)$ ;
- ii) Lemma (4.5.5), concerning derivations which have instances of (R2) "thinning in the antecedent", (R3) "contraction in the antecedent" and (R4) "interchange in the antecedent" with active formulae of the form  $a \uparrow b$ ;
- iii) Lemma (4.5.6), concerning derivations of  $Dccl(X)$  which have instances of (R14) "hook introduction in the antecedent" with the property that the active formulae in their right premisses are terminal objects;

we have constructed and implemented their proofs in the system to satisfy the definition of the normal form (4.5.7).

## (4.1) Definition

A cartesian closed category is a cartesian category  $C$  with the following additional structure :

(4) A bifunctor  $(-) \Rightarrow (-) : op(C) \times C \rightarrow C$ .

(5) An adjunction  $\alpha_\lambda$ , where

$$\alpha_\lambda = \{ \alpha_\lambda(A, B, C) : C(A \uparrow B, C) \rightarrow C(B, A \Rightarrow C) \in ArEns \mid A, B, C \in ObC \}.$$

(4.2) The Free Cartesian Closed Category  $Fccl(X)$ 

Let  $X$  be a fixed but arbitrary small category. The free cartesian closed category  $Fccl(X)$  is constructed as follows :

(4.2.1) Definition

The class  $cclL(X)$  of (cartesian closed) formulae over a category  $X$  is defined to contain the class  $cL(X)$  of (cartesian) formulae and

v) If  $A, B$  are formulae, then  $A \Rightarrow B$  is .

(4.2.2) The Labelled Derivations  $LDccl(X)$ 

The class  $LDccl(X)$  of (cartesian closed) labelled derivations over  $X$  and their conclusions, are simultaneously defined by conditions (i) - (viii) of (1.2.2) and the following additional conditions :

ix) for derivations  $F, G$  with conclusions

$f : A \rightarrow B$  ,  $g : C \rightarrow D$  respectively,

$EPSILON(F, G)$  is a derivation ,

with conclusion  $epsilon\_left(f, g) : A \uparrow (B \Rightarrow C) \rightarrow D$  ;

x) for a derivation  $F$  with conclusion  $f : A \uparrow B \rightarrow C$

$ALPHA(F)$  is a derivation ,

with conclusion  $alpha\_right(f) : B \rightarrow (A \Rightarrow C)$  .

(4.2.3) Definition

The relation  $==$  is defined to be the smallest equivalence relation on  $LDccl(X)$  satisfying conditions (i) - (viii) of (1.2.3) and the following additional conditions

ix) If  $F == G$  and  $H == K$  , with conclusions  $f, g : A \rightarrow B$  ,  
 $h, k : C \rightarrow D$  ,

then

$EPSILON(F, H) == EPSILON(G, K)$  ;

x) IF  $F == G$  , with conclusions  $f, g : A \uparrow B \rightarrow C$

then

$ALPHA(F) == ALPHA(G)$  ;

xi) For derivations  $F, G, H, K$  with conclusions

$f : A \rightarrow B$  ,  $g : C \rightarrow D$  ,

$h : B \rightarrow E$  ,  $k : J \rightarrow C$  respectively

[ Note for derivations  $F, G$  as above we define the derivation  
 $HOOK(F, G) = ALPHA( EPSILON(F, G) )$  , with conclusion

$f \Rightarrow g = alpha\_right(epsilon\_left(f, g)) : (B \Rightarrow C) \rightarrow (A \Rightarrow D)$  ]

then after introducing the definition of the derivation  $HOOK$  we have :

$COMP( HOOK(H, K), HOOK(F, G), B \Rightarrow C ) ==$   
 $HOOK( COMP(F, H, B), COMP(K, G, C) )$  ;

- xii) For  $F$  with conclusion  $f : A \uparrow B \rightarrow C$  ,  
 $\text{COMP}(\text{PRODUCT}(I(A), \text{ALPHA}(F)) ,$   
 $\text{EPSILON}(I(A), I(C)) ,$   
 $A \uparrow (A \Rightarrow C) ) == F$   
 where  $I(A)$  is as defined in (1.4.2) ;
- xiii) For  $G$  with conclusion  $g : B \rightarrow (A \Rightarrow C)$  ,  
 $\text{ALPHA}(\text{COMP}(\text{PRODUCT}(I(A), G) ,$   
 $\text{EPSILON}(I(A), I(C)) ,$   
 $A \uparrow (A \Rightarrow C) ) ) == G$  ;
- xiv) For  $F, G$  with conclusions  $f : D \rightarrow C$  and  
 $g : A \uparrow B \rightarrow D$  respectively  
 $\text{COMP}(\text{ALPHA}(G) , \text{HOOK}(I(A), F) , A \Rightarrow D) ==$   
 $\text{ALPHA}(\text{COMP}(G, F, D) )$  ;
- xv) For  $F, G, H$  with conclusions  
 $f : A \rightarrow D$  ,  $g : D \uparrow E \rightarrow C$  ,  $h : B \rightarrow E$  respectively  
 $\text{COMP}(\text{COMP}(H, \text{ALPHA}(G) , E) ,$   
 $\text{HOOK}(F, I(C)) ,$   
 $D \Rightarrow C) ==$   
 $\text{ALPHA}(\text{COMP}(\text{PRODUCT}(F, H) , G, D \uparrow E) )$  ;
- xvi) For  $F, G, H$  with conclusions  
 $f : E \rightarrow (C \Rightarrow D)$  ,  $g : A \rightarrow C$  ;  $h : D \rightarrow B$  respectively  
 $\text{COMP}(\text{COMP}(\text{EPSILON}(I(C), I(D)) ,$   
 $\text{PRODUCT}(G, F) ,$   
 $C \uparrow (C \Rightarrow D) ) ,$   
 $H ,$   
 $D) ==$   
 $\text{COMP}(\text{PRODUCT}(I(A), \text{COMP}(F, \text{HOOK}(G, H) , C \Rightarrow D) ) ,$   
 $\text{EPSILON}(I(A), I(B)) ,$   
 $A \uparrow (A \Rightarrow B) )$  .

#### (4.2.4) Definition

The free cartesian closed category  $\text{Fccl}(X)$  is defined to be the category having :

- i) as objects , the cartesian closed formulae over  $X$  ;  
 ii) as arrows , the  $==$ -class of  $\text{LDccl}(X)$  ;  
 iii) - xi) as in the definition (1.2.5) ;  
 xii) as  $\text{epsilon\_left}(-, -)$ , the function defined by  
 $\text{epsilon\_left}([F], [G]) = [\text{EPSILON}(F, G)]$  ;

xiii) as  $\text{alpha\_right}(-,-)$  , the function defined by  

$$\text{alpha\_right}([F]) = [\text{ALPHA}(F)] .$$

It is now routine to check that, with this structure,  $\text{Fccl}(X)$  is a category, is cartesian closed, and is free cartesian closed over  $X$ . There is the obvious embedding  $X \rightarrow \text{Fccl}(X)$  defined by  $A \mapsto A; f \mapsto [\text{LA1}(f)]$ .

#### (4.3) The Unlabelled Derivations $\text{Dccl}(X)$

We define the class  $\text{Dccl}(X)$  of unlabelled ( cartesian closed ) derivations as another way to represent these arrows of  $\text{Fccl}(X)$  by composition free derivations .

##### (4.3.1) Definition

The class  $\text{Dccl}(X)$  is defined inductively by the conditions in the definition (1.3.1) and the following additional rules :

- x) ( hook introduction in the antecedent )  
 If  $p_1, p_2$  are derivations with conclusions  $G \rightarrow a, \text{DbL} \rightarrow F$ , then  $\text{R14}(p_1, p_2, a \Rightarrow b)$  is a derivation with conclusion  $\text{DG}(a \Rightarrow b)\text{L} \rightarrow F$ ;
- xi) ( hook introduction in the succedent )  
 If  $p$  is a derivation with conclusion  $\text{GaD} \rightarrow b$  , then  $\text{R15}(p, a \Rightarrow b)$  is a derivation with conclusion  $\text{GD} \rightarrow (a \Rightarrow b)$ .

##### (4.3.2) Representation of $\text{Dccl}(X)$ in Prolog

In order to represent derivations of  $\text{Dccl}(X)$  in Prolog we extend the representation in (1.3.2) by the following :

we extend the abbreviations in (1.3.2) by

$\rightarrow$  for  $\Rightarrow$   
 $\text{hia}$  for the name of the operation in  $\text{R14}(p_1, p_2, a \Rightarrow b)$   
 $\text{his}$  for the name of the operation in  $\text{R15}(p, a \Rightarrow b)$  .

- ix)  $\text{R14}(p_1, p_2, a \Rightarrow b)$  is represented by a step-collection with the last step ,

$\text{g3} = [\text{hia}(g_1, g_2, a \rightarrow b), \text{DGA} \rightarrow \text{bL} \Rightarrow F]$

where  $g_1, g_2$  are names of the last steps of the step-collections for the derivations  $p_1, p_2$  respectively .

- x)  $\text{R15}(p, a \Rightarrow b)$  is represented by a step-collection with the last step.

$$g2 = [\text{his}(g1, a \rightarrow b), GD \Rightarrow [a \rightarrow b]]$$

where  $g1$  is the name of the last step of the step-collection for the derivation  $p$ .

As an example of a derivation of the class  $\text{Dccl}(X)$  and the use of the system to check its correctness, consider the following :

#### Example

The derivation

$$\begin{array}{c}
 \frac{c \rightarrow c}{b, c \rightarrow c} \\
 \frac{b, c \rightarrow c}{b, e, c \rightarrow c} \\
 \frac{a \rightarrow a \quad b, c \rightarrow (e \Rightarrow c)}{a, a \Rightarrow b, c \rightarrow (e \Rightarrow c)} \\
 \frac{a, (a \Rightarrow b) \uparrow c \rightarrow (e \Rightarrow c)}{(a \Rightarrow b) \uparrow c, a \rightarrow (e \Rightarrow c)}
 \end{array}$$

can be checked for correctness correct using the system as follows :

```

yes
| ?- theory(cartcl).

yes
| ?- prove(l = [id(a), [a] => [a]]).

yes
| ?- prove(m = [id(c), [c] => [c]]).

yes
| ?- prove(m1 = [th(m,b), [b,c] => [c]]).

yes
| ?- prove(m2 = [th(m1,e), [b,e,c] => [c]]).

yes
| ?- prove(m3 = [his(m2,e -> c), [b,c] => [e -> c]]).

yes
| ?- prove(n = [hia(l,m3,a -> b), [a,a -> b,c] => [e -> c]]).

yes
| ?- prove(o = [ala(n,(a->b)&c), [a,(a->b)&c] => [e -> c]]).

yes
| ?- prove(p = [inc(o,(a->b)&c,a), [(a->b)&c,a] => [e->c]]).

yes
| ?-

```

#### (4.4) The Semantics of Dccl(X)

In this section, we define a function to interpret derivations of  $\text{Dccl}(X)$  as arrows of  $\text{Fccl}(X)$  and prove the completeness theorem in the sense that every arrow of  $\text{Fccl}(X)$  is representable by means of some



derivation of  $Dccl(X)$ . For the sake of this interpretation we extend the canonical arrows in (1.4) by the following canonical arrows of  $Fccl(X)$ , determined by the adjunction  $\alpha_\lambda$ :

iv)  $\epsilonpsilon(A,B) : A \dagger (A \Rightarrow B) \rightarrow B$  for all  $A,B$  of  $ObFccl(X)$ , where  $\epsilonpsilon(A,B) = \alpha_\lambda^{-1}(A, A \Rightarrow B, B)(id(A \Rightarrow B))$

v)  $\iota(A,B) : B \rightarrow A \Rightarrow (A \dagger B)$  for all  $A,B$  of  $ObFccl(X)$ , where  $\iota(A,B) = \alpha_\lambda(A, B, A \dagger B)(id(A \dagger B))$ .

#### (4.4.1) Definition

We extend the definition of the function  $S$  in (1.4.1) to interpret derivations of  $Dccl(X)$  as arrows of  $Fccl(X)$  as follows:

(9)  $S(R14(p1,p2,a \Rightarrow b)) = \text{comp}(\text{comp}(\text{comp}(S(p2), (I1 \dagger \epsilonpsilon(a,b)) \dagger I2), ((I1 \dagger S(p1)) \dagger I3) \dagger I2), \text{alpha\_inverse}(\Pi(D), \Pi(G), a \Rightarrow b) \dagger I2)$

where

$I1$  is the identity on  $\Pi(D)$ ,

$I2$  is the identity on  $\Pi(L)$  and

$I3$  is the identity on  $a \Rightarrow b$ ;

(10)  $S(R15(p, a \Rightarrow b)) = \text{comp}(\text{comp}(\text{comp}(I1 \Rightarrow S(p), I1 \Rightarrow (\text{sigma}(a, \Pi(G)) \dagger I2)), I1 \Rightarrow \text{alpha}(a, \Pi(G), \Pi(D))), \iota(a, \Pi(G) \dagger \Pi(D)))$ , where

$I1$  is the identity on  $a$  and

$I2$  is the identity on  $\Pi(D)$ .

The following is an example for a session with the system; as input is a derivation of  $Dccl(X)$  and as output is an arrow of  $Fccl(X)$  as interpretation for that derivation.

#### Example

For the arrows  $f$  in  $X(a,c)$ ,  $g$  in  $X(b,c)$  and  $e$  in  $Ob(X)$ , the derivation

$$\begin{array}{c}
 \frac{e \rightarrow e}{a, e \rightarrow e} \\
 \frac{a, e, e \rightarrow e}{a, e \rightarrow (c \Rightarrow e)} \\
 \hline
 a, a, e, b \rightarrow c \\
 \hline
 a, e, b \rightarrow c \\
 \hline
 a, e \dagger b \rightarrow c
 \end{array}
 \qquad
 \begin{array}{c}
 \frac{b \xrightarrow{g} c}{e, b \rightarrow c} \\
 \frac{a \xrightarrow{f} c \quad e, b \rightarrow c}{a, c \Rightarrow e, b \rightarrow c} \\
 \hline
 a, a, e, b \rightarrow c \\
 \hline
 a, e, b \rightarrow c \\
 \hline
 a, e \dagger b \rightarrow c
 \end{array}$$

is interpreted in  $\text{ArFccl}(X)$  as follows :

```

yes
| ?- theory(cartcl).

yes
| ?- prove(l = [id(e), [e] => [e]]).

yes
| ?- prove(l1 = [th(l,a), [a,e] => [e]]).

yes
| ?- prove(l2 = [th(l1,c), [a,e,c] => [e]]).

yes
| ?- prove(l3 = [his(l2,c -> e), [a,e] => [c -> e]]).

yes
| ?- prove(m = [f, [a] => [c]]).

yes
| ?- prove(n = [g, [b] => [c]]).

yes
| ?- prove(n1 = [th(n,e), [e,b] => [c]]).

yes
| ?- prove(o = [hia(m,n1,c -> e), [a,c -> e,b] => [c]]).

yes
| ?- prove(p = [cut(l3,o,c -> e), [a,a,e,b] => [c]]).

yes
| ?- prove(q = [con(p,a), [a,e,b] => [c]]).

yes
| ?- prove(r = [aia(q,e & b), [a,e & b] => [c]]).

yes
| ?- interpr(r,H).

H = comp(comp(comp(comp(comp(comp(g,pi_right(e,b)),
                                product(epsilon(c,e),id(b))),
                                product(product(f,id(c->e)),id(b))),
                                product(product(id(a),
                                                comp(comp(hook(id(c),
                                                            comp(pi_right(a,e),
                                                                pi_left(a&e,c))),
                                                                hook(id(c),sigma(c,a&e))),
                                                                lta(c,a&e))),
                                                id(b))),
                                product(delta(a),id(e&b))),
                                alpha(a,e,b))

yes
| ?-

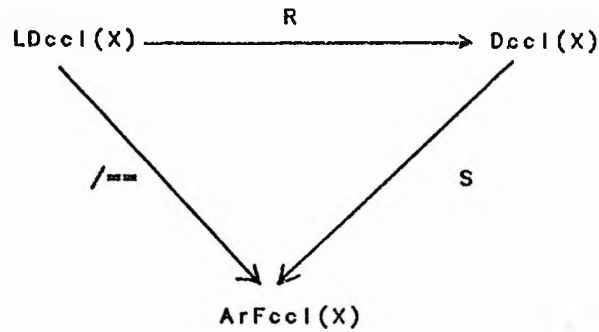
```

In order to prove that the class  $\text{Dccl}(X)$  of the unlabelled cartesian closed derivations is adequate in the sense that every arrow in  $\text{Fccl}(X)$  is representable by means of derivation in this class, we prove the following theorem :

#### (4.4.2) The Completeness Theorem for $\text{Dccl}(X)$

There is a function

$R : \text{LDccl}(X) \longrightarrow \text{Dccl}(X)$  , so that



commutes .

Proof :

We extend the definition of  $R(F)$  for  $F$  of  $\text{LDcc1}(X)$  in (1.4.2) by the following :

We extend iii) of (1.4.2) by

iii) or is  $B \Rightarrow C$  : put

$$\begin{aligned}
 R(F) = R15( R14( R( I(B) ), \\
 R( I(C) ), B \Rightarrow C ), \\
 B \Rightarrow C )
 \end{aligned}$$

viii) when  $F$  is  $\text{EPSILON}(G, H)$  , put

$$\begin{aligned}
 R(F) = R11( R14( R(G), R(H), B \Rightarrow C ), \\
 A \uparrow (B \Rightarrow C) )
 \end{aligned}$$

ix) when  $F$  is  $\text{ALPHA}(G)$  , put

$$\begin{aligned}
 R(F) = R15( R1( R10( R2( R( I(A) ), B ), \\
 R2( R( I(B) ), A ), \\
 A \uparrow B ), \\
 R(G), \\
 A \uparrow B ), \\
 A \Rightarrow C )
 \end{aligned}$$

We extend the examples in (1.4.2) by the following ones for some arrows of  $\text{Fccl}(X)$  in order to illustrate the above algorithm. These arrows and their representations are built in to the system, for convenience.

Example 1

For the derivation  $F = \text{EPSILON}(I(A), I(B))$  representing the arrow

$$\text{epsilon}(A, B) : A \uparrow (A \Rightarrow B) \rightarrow B$$

the representation  $R(F)$  is

$$\begin{aligned}
 R11( R14( R( I(A) ), R( I(B) ), A \Rightarrow B ), \\
 A \uparrow (A \Rightarrow B) )
 \end{aligned}$$

Example 2

For the derivation  $F = \text{ALPHA}(\text{LA2}(A \uparrow B))$  representing the arrow

$$\text{Ita}(A,B) : B \rightarrow A \Rightarrow (A \uparrow B)$$

the representation  $R(F)$  is

$$\begin{aligned} &R15( R10( R2( R( I(A) ), B ), \\ &\quad R2( R( I(B) ), A ), \\ &\quad A \uparrow B ), \\ &A \Rightarrow (A \uparrow B) ). \end{aligned}$$

Example 3 For derivations  $F, G$  representing the arrows

$$f : A \rightarrow B, \quad g : C \rightarrow D \text{ respectively,}$$

with the derivation

$$\text{HOOK}(F, G) = \text{ALPHA}(\text{EPSILON}(F, G))$$

representing the arrow

$$f \Rightarrow g : (B \Rightarrow C) \rightarrow (A \Rightarrow D)$$

then the representation  $R(\text{HOOK}(F, G))$  is

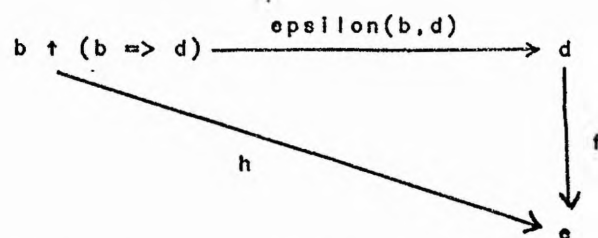
$$\begin{aligned} &R( \text{ALPHA}( \text{EPSILON}( F, G ) ) ). \\ &R15( R1( R10( R2( R( I(A) ), B ), \\ &\quad R2( R( I(B) ), A ), \\ &\quad A \uparrow B ), \\ &R11( R14( R( G ), R( H ), B \Rightarrow C ), \\ &\quad A \uparrow (B \Rightarrow C) ) \\ &A \uparrow B ), \\ &A \Rightarrow C ) \end{aligned}$$

The following is a representation of an arrow of  $\text{Fccl}(X)$  by means of a derivation of the class  $\text{Dccl}(X)$  :

Example 4 for  $f$  in  $X(d,e)$ , the arrow

$$h = \text{comp}(f, \text{epsilon}(b,d)) : b \uparrow (b \Rightarrow d) \rightarrow e$$

in the diagram



is represented by a derivation of  $\text{Dccl}(X)$  as follows :

```

yes
| ?- theory(cartcl).
yes
| ?- arrow(d,f,e).
yes
| ?- rop_of(comp(f,epsilon(b,d))).

```

The arrow is :

```

=====
comp(f,epsilon(b,d))

```

Representation :-

```

f5=[id(b), [b]=>[b]]
f6=[id(d), [d]=>[d]]
f4=[hid(f5, f6, b->d), [b, b->d]=>[d]]
f3=[aid(f4, b, b->d), [b&(b->d)]=>[d]]
f2=[f, [d]=>[e]]
f1=[cut(f3, f2, d), [b&(b->d)]=>[e]]

```

yes  
| ?-

which in tree form is :

$$\begin{array}{c}
 \frac{b \rightarrow b \quad d \rightarrow d}{b, b \Rightarrow d \rightarrow d} \\
 \frac{b \uparrow (b \Rightarrow d) \rightarrow d \quad \frac{d \xrightarrow{f} e}{d \rightarrow e}}{b \uparrow (b \Rightarrow d) \rightarrow e}
 \end{array}$$

#### (4.5) The Syntax of Fccl(X)

After representing the arrows of Fccl(X) by means of derivations of the class Dccl(X), we are able to manipulate these derivations to eliminate cuts and obtain their normal representations.

##### (4.5.1) The Cut-Elimination Theorem for Dccl(X)

There exists for every derivation  $p$  in Dccl(X) an equivalent cut-free derivation  $q$  in Dccl(X).

Proof

By the proof of (1.5.2) and the following additional clauses of Appendix (2) :

(C.5), (C.16), (C.30), C.31) and (C.45), every derivation in Dccl(X) has an equivalent cut-free one.

###

##### (4.5.2) Definition

Let  $L(T)$  be the subclass of the class of (cartesian closed) formulae  $cccl(X)$  generated by the conditions :

- 1)  $T$  is in  $L(T)$ .
- 2) If  $a, b$  are in  $L(T)$ , then  $a \uparrow b$  is in  $L(T)$ .
- 3) If  $a$  is in  $L(T)$  and  $b$  is in  $cccl(X)$ , then  $(b \Rightarrow a)$  is in  $L(T)$ .

##### (4.5.3) Lemma

$a \approx T$  in Fccl(X) iff  $a$  is in  $L(T)$ .

(4.5.4) Lemma

If  $p$  derives  $G \rightarrow a$  and  $a = \sim T$ , then  $p$  is of the form

$$\frac{\frac{q}{\rightarrow a}}{G \rightarrow a} \quad (K)$$

Proof:

The proof is by induction on the construction of the formula  $a$  :

- i)  $a = T$  as in (1.5.3)
- ii)  $a = c \uparrow d$
- iii)  $a = c \Rightarrow d$  where  $d = \sim T$  and  $b$  is in  $ObFocl(X)$ ,

$$\frac{\frac{\frac{g}{\rightarrow d}}{c \rightarrow d}}{\rightarrow (c \Rightarrow d)} \\ G \rightarrow (c \Rightarrow d)$$

This completes the induction proof.

###

(4.5.5) Lemma

For every cut-free derivation  $p$  in  $Docl(X)$ , there exists an equivalent cut-free  $q$  in  $Docl(X)$ , containing no instances of (R2), (R3) and (R4) whose active formulae are of the form  $a \uparrow b$ .

Proof:

The proof in case of (R2) follows from clause (E.41.1) in Appendix 3. In cases of (R3) and (R4) the proof follows from the proof of lemma (1.5.4) and the clauses (E.13), (E.14), (E.21) and (E.22) in Appendix (3). There are some cases in which an instance of (R14) followed by an instance of (R3) or (R4) with active formulae of the form  $a \uparrow b$  and these active formulae are not in the antecedent of one of the premisses of (R14) but each one in the antecedent of each premiss like

$$\frac{\frac{a \uparrow b, G \rightarrow c \quad D, a \uparrow b, d, L \rightarrow e}{D, a \uparrow b, a \uparrow b, G, c \Rightarrow d, L \rightarrow e}}{D, a \uparrow b, G, c \Rightarrow d, L \rightarrow e}$$

In these cases we use the priority described in the clauses mentioned below in (4.5.8) to move the instance of (R11) applied to construct  $a \uparrow b$  below the application of the instance of (R14) and then using the clauses of lemma (1.5.4) again to remove the instances of (R3) and (R4) with active formulae of the form  $a \uparrow b$ .

###

(4.5.6) Lemma

Every cut-free  $p$  in  $Dccl(X)$  is equivalent to a cut-free  $q$  in  $Dccl(X)$  containing no instance of (R14) with the property that the active formulae in their right premisses are isomorphic to  $T$ .

Proof:

Let  $p$  is a derivation of the form

$$\frac{G \rightarrow a \quad \text{DbL} \xrightarrow{p1} c}{DG a \Rightarrow b L \rightarrow c}$$

where  $b$  is isomorphic to  $T$ , then it is equivalent to

$$\frac{G \rightarrow a \quad \text{DbL} \xrightarrow{p2} c}{DG a \Rightarrow b L \rightarrow c}$$

where  $p2$  is the unique cut-free derivation obtained by applying the cut elimination algorithm of appendix (2) to the derivation

$$\frac{m \rightarrow b \quad \text{DbL} \xrightarrow{p1} c}{DL \rightarrow c}$$

where  $m$  is the unique derivation of  $\rightarrow b$  by lemma (4.5.4), then using clause (D.11.4) in appendix (4) to remove the instance of (R14).  
###

(4.5.7) Definition

A derivation  $p$  in  $Dccl(X)$  is normal if it satisfies the following conditions :

- (1) All subderivations of  $p$  not containing instances of (R14) and (R15) are normal in the sense of the definition (1.5.4) with the condition that the active formulae of instances of (R2) are atomic replace by the condition that the active formulae of instances of (R2) are either atomic or of the form  $a \Rightarrow b$ .
- (2)  $p$  contains no instance of (R14) whose active formula in the right-hand premiss is isomorphic to  $T$ .
- (3) If  $p$  derives  $G \rightarrow a$  and  $a \approx \sim T$ , then  $p$  is of the form

$$\frac{\rightarrow a}{G \rightarrow a} \quad (K)$$

where  $q$  is the unique derivation of  $\rightarrow a$ , and where (K) consists of the instances of (R2) required to derive  $G \rightarrow a$  from  $\rightarrow a$ .

(4)  $p$  contains no instances of (R2), (R3) and (R4) whose active formulae are of the form  $a \uparrow b$ .

(5) Using the relation  $=<$  as defined in (1.5.5); for normality in  $Dccl(X)$  it is required that

$$(R10) =< (R15) =< (R14) =< (R3) =< (R11) =< (R2) =< (R4).$$

#### (4.5.8) The Normalization Theorem For $Dccl(X)$

For every derivation  $p$  in the class  $Dccl(X)$  there is a unique equivalent normal derivation  $q$  in  $Dccl(X)$ .

#### Proof

By combining the results of the Cut-Elimination theorem (4.5.1), the theorem (1.5.6), the above lemmas and the following additional clauses of the algorithm in appendix (4) :

(D.2), (D.3), (D.11), (D.12), (D.15), (D.16), (D.18), (D.19), (D.22), (D.23), (D.28), (D.29), (D.32), (D.33), (D.52), (D.56), (D.57), (D.64), (D.65), (D.73), (D.75), (D.77), (D.79), (D.81.3), (D.82.3) and (D.82.4), there is a unique equivalent normal form for every derivation in  $Dccl(X)$ .

###

The following example is a session with the system to construct the normal equivalent proof for a derivation of the class  $Dccl(X)$ , illustrating above theorems for cut-elimination and normalization :

#### Example 1

Let  $a, b, c, d$  be objects of the category  $X$ ,  $f$  in  $X(a, b)$ ,  $g$  in  $X(c, d)$ ; then the following derivation which has an instance of (R14) whose active formula in the right-hand premiss is  $(T \uparrow T)$  which is isomorphic to  $T$  :

$$\begin{array}{c}
 \begin{array}{c}
 f \\
 a \rightarrow b \\
 \hline
 a, T \rightarrow b \\
 \hline
 a, e, T \rightarrow b \\
 \hline
 a, e, T, j \rightarrow b \\
 \hline
 a \uparrow e, T, j \rightarrow b \\
 \hline
 a \uparrow e, T, T, j \rightarrow b \\
 \hline
 a \uparrow e, T \uparrow T, j \rightarrow b
 \end{array} \\
 \begin{array}{c}
 g \\
 c \rightarrow d \\
 \hline
 a \uparrow e, c, d \Rightarrow (T \uparrow T), j \rightarrow b
 \end{array}
 \end{array}$$



is transformed to its equivalent normal form, as follows :

```

yes
| ?- theory(cartel).
yes
| ?- prove(s1 = [g, [c] => [d]]).
yes
| ?- prove(s2 = [f, [a] => [b]]).
yes
| ?- prove(s3 = [th(s2,tr), [a,tr] => [b]]).
yes
| ?- prove(s4 = [th(s3,e), [a,e,tr] => [b]]).
yes
| ?- prove(s5 = [th(s4,j), [a,e,tr,j] => [b]]).
yes
| ?- prove(s6 = [aia(s5,a & e), [a & e,tr,j] => [b]]).
yes
| ?- prove(s7 = [th(s6,tr), [a & e,tr,tr,j] => [b]]).
yes
| ?- prove(s8 = [aia(s7,tr & tr), [a & e,tr & tr,j] => [b]]).
yes
| ?- prove(s=[hia(s1,s8,d->(tr&tr)), [a&e,c,d->(tr&tr),j] => [b]]).
yes
| ?- normal(s).

```

Proof is as follows :

```

s1=[g,[c]==>[d]]
s2=[f,[a]==>[b]]
s3=[th(s2,tr),[a,tr]==>[b]]
s4=[th(s3,e),[a,e,tr]==>[b]]
s5=[th(s4,j),[a,e,tr,j]==>[b]]
s6=[aia(s5,a&e),[a&e,tr,j]==>[b]]
s7=[th(s6,tr),[a&e,tr,tr,j]==>[b]]
s8=[aia(s7,tr&tr),[a&e,tr&tr,j]==>[b]]
s=[hia(s1,s8,d->tr&tr),[a&e,c,d->tr&tr,j]==>[b]]

```

Cut\_free proof is :

```

s1=[g,[c]==>[d]]
s2=[f,[a]==>[b]]
s3=[th(s2,tr),[a,tr]==>[b]]
s4=[th(s3,e),[a,e,tr]==>[b]]
s5=[th(s4,j),[a,e,tr,j]==>[b]]
s6=[aia(s5,a&e),[a&e,tr,j]==>[b]]
s7=[th(s6,tr),[a&e,tr,tr,j]==>[b]]
s8=[aia(s7,tr&tr),[a&e,tr&tr,j]==>[b]]
s=[hia(s1,s8,d->tr&tr),[a&e,c,d->tr&tr,j]==>[b]]

```

Normal proof is :

```

f21=[f,[a]==>[b]]
f18=[th(f21,j),[a,j]==>[b]]
f15=[th(f18,e),[a,e,j]==>[b]]
f10=[aia(f15,a&e),[a&e,j]==>[b]]
f5=[th(f10,d->tr&tr),[a&e,d->tr&tr,j]==>[b]]
s=[th(f5,c),[a&e,c,d->tr&tr,j]==>[b]]

```

```

yes
| ?-

```

which in tree form is

$$\begin{array}{c}
 \frac{f}{a \rightarrow b} \\
 \hline
 \frac{a, j \rightarrow b}{a, e, j \rightarrow b} \\
 \hline
 \frac{a \uparrow e, j \rightarrow b}{a \uparrow e, d \Rightarrow (T \uparrow T), j \rightarrow b} \\
 \hline
 \frac{a \uparrow e, d \Rightarrow (T \uparrow T), j \rightarrow b}{a \uparrow e, c, d \Rightarrow (T \uparrow T), j \rightarrow b}
 \end{array}$$



which in tree form is

$$\frac{\frac{\frac{a \xrightarrow{f} b}{\quad} \quad \quad \quad c \xrightarrow{g} d}{\quad} \quad \quad \quad a, b \Rightarrow c \rightarrow d}{\quad} \quad \quad \quad a \uparrow (b \Rightarrow c) \rightarrow d$$

### Example 3

Let  $a, b, c, d$  be objects of the category  $X$ ; then the following derivation :

$$\frac{\frac{\frac{b \rightarrow b}{\quad} \quad \quad \quad c, b \rightarrow b}{\quad} \quad \quad \quad c \uparrow b \rightarrow b \quad \quad \quad \frac{\frac{\frac{a \rightarrow a}{\quad} \quad \quad \quad a, b \rightarrow a}{\quad} \quad \quad \quad a \uparrow b \rightarrow a}{\quad} \quad \quad \quad a \uparrow b, c \rightarrow a}{\quad} \quad \quad \quad a \uparrow b, d, c \rightarrow a}{\quad} \quad \quad \quad a \uparrow b, c \uparrow b, b \Rightarrow d, c \rightarrow a}{\quad} \quad \quad \quad c \uparrow b, a \uparrow b, b \Rightarrow d, c \rightarrow a$$

is transformed to its equivalent normal form, as follows :

```

yes
| ?- theory(cartci).
yes
| ?- prove(s1 = [id(b), [b] => [b]]).
yes
| ?- prove(s2 = [th(s1,c), [c,b] => [b]]).
yes
| ?- prove(s3 = [ala(s2,c & b), [c & b] => [b]]).
yes
| ?- prove(s4 = [id(a), [a] => [a]]).
| ?- prove(s5 = [th(s4,b), [a,b] => [a]]).
yes
| ?- prove(s6 = [ala(s5,a & b), [a & b] => [a]]).
yes
| ?- prove(s7 = [th(s6,c), [a & b,c] => [a]]).
yes
| ?- prove(s8 = [th(s7,d), [a&b,d,c] => [a]]).
yes
| ?- prove(s9 = [hia(s3,s8,b->d), [a&b,c&b,b->d,c] => [a]]).
yes
| ?- prove(s = [inc(s9,c&b,a&b), [c&b,a&b,b->d,c] => [a]]).
yes
| ?- normal(s).

```

Proof is as follows :

=====

```

s1=[id(b),[b]=>[b]]
s2=[th(s1,c),[c,b]=>[b]]
s3=[ala(s2,c&b),[c&b]=>[b]]
s4=[id(a),[a]=>[a]]
s5=[th(s4,b),[a,b]=>[a]]
s6=[ala(s5,a&b),[a&b]=>[a]]
s7=[th(s6,c),[a&b,c]=>[a]]
s8=[th(s7,d),[a&b,d,c]=>[a]]
s9=[hia(s3,s8,b->d),[a&b,c&b,b->d,c]=>[a]]
s=[inc(s9,c&b,a&b),[c&b,a&b,b->d,c]=>[a]]

```

Cut\_free proof is :

=====

```

s1=[id(b),[b]=>[b]]
s2=[th(s1,c),[c,b]=>[b]]
s3=[ala(s2,c&b),[c&b]=>[b]]
s4=[id(a),[a]=>[a]]

```

```

s5=[th(s4,b),[a,b]=>[a]]
s6=[aia(s5,a&b),[a&b]=>[a]]
s7=[th(s6,c),[a&b,c]=>[a]]
s8=[th(s7,d),[a&b,d,c]=>[a]]
s9=[hia(s3,s8,b->d),[a&b,c&b,b->d,c]=>[a]]
s=[nc(s9,c&b,a&b),[c&b,a&b,b->d,c]=>[a]]

```

Normal proof is :

=====

```

s4=[ld(a),[a]=>[a]]
s5=[ti(s4,b),[a,b]=>[a]]
s6=[aia(s5,a&b),[a&b]=>[a]]
s7=[th(s6,c),[a&b,c]=>[a]]
s3=[th(s7,b->d),[a&b,b->d,c]=>[a]]
f2=[th(s3,b),[b,a&b,b->d,c]=>[a]]
f1=[th(f2,c),[c,b,a&b,b->d,c]=>[a]]
s=[aia(f1,c&b),[c&b,a&b,b->d,c]=>[a]]

```

```

yes
| ?-

```

which in tree form is

$$\begin{array}{c}
 a \rightarrow a \\
 \hline
 a, b \rightarrow a \\
 \hline
 a \uparrow b \rightarrow a \\
 \hline
 a \uparrow b, c \rightarrow a \\
 \hline
 a \uparrow b, b \Rightarrow d, c \rightarrow a \\
 \hline
 b, a \uparrow b, b \Rightarrow d, c \rightarrow a \\
 \hline
 c, b, a \uparrow b, b \Rightarrow d, c \rightarrow a \\
 \hline
 c \uparrow b, a \uparrow b, b \Rightarrow d, c \rightarrow a
 \end{array}$$

#### Example 4

For  $f$  in  $X(a,b)$ ,  $g$  in  $X(c,d)$  and  $c,d,e$  are objects of the category  $X$ ; the derivation

$$\begin{array}{c}
 \begin{array}{c}
 c \rightarrow c \\
 \hline
 c, e \rightarrow c \\
 \hline
 c \uparrow e \rightarrow c
 \end{array}
 \quad
 \begin{array}{c}
 g \\
 c \rightarrow d
 \end{array}
 \quad
 \begin{array}{c}
 e \rightarrow e \\
 \hline
 c, e \rightarrow e \\
 \hline
 c \uparrow e \rightarrow c \quad e \rightarrow e \\
 \hline
 c \uparrow e \rightarrow e
 \end{array} \\
 \hline
 \begin{array}{c}
 f \\
 a \rightarrow b
 \end{array}
 \quad
 \begin{array}{c}
 c \uparrow e \rightarrow d \quad c \uparrow e \rightarrow e \\
 \hline
 a, b \Rightarrow (c \uparrow e) \rightarrow d \uparrow e \\
 \hline
 a \uparrow (b \Rightarrow (c \uparrow e)) \rightarrow d \uparrow e
 \end{array}
 \quad
 \begin{array}{c}
 d \rightarrow d \\
 \hline
 d, e \rightarrow d \\
 \hline
 d \uparrow e \rightarrow d
 \end{array} \\
 \hline
 a \uparrow (b \Rightarrow (c \uparrow e)) \rightarrow d
 \end{array}$$

is transformed to its equivalent normal form, as follows :

```

yes
| ?- theory(cartcl).
yes
| ?- prove(s1 = [f, [a] => [b]]).
yes
| ?- prove(s2 = [ld(c), [c] => [c]]).
yes
| ?- prove(s3 = [th(s2,e), [c,e] => [c]]).
yes
| ?- prove(s4 = [aia(s3,c & e), [c & e] => [c]]).

```

```

yes.
| ?- prove(s5 = [g, [c] => [d]]).
yes
| ?- prove(s6 = [cut(s4,s5,c), [c & e] => [d]]).
yes
| ?- prove(s7 = [id(e), [e] => [e]]).
yes
| ?- prove(s8 = [th(s7,c), [c,e] => [e]]).
yes
| ?- prove(s9 = [aia(s8,c & e), [c & e] => [e]]).
yes
| ?- prove(s10 = [id(e), [e] => [e]]).
yes
| ?- prove(s11 = [cut(s9,s10,e), [c & e] => [e]]).
yes
| ?- prove(s12 = [ais(s6,s11,d & e), [c & e] => [d & e]]).
yes
| ?- prove(s13 = [hia(s1,s12,b->(c&e)), [a,b -> (c&e)] => [d&e]]).
yes
| ?- prove(s14 = [aia(s13,a&(b->(c&e))),[a&(b->(c&e))] => [d&e]]).
yes
| ?- prove(s15 = [id(d), [d] => [d]]).
yes
| ?- prove(s16 = [th(s15,e), [d,e] => [d]]).
yes
| ?- prove(s17 = [aia(s16,d & e), [d & e] => [d]]).
yes
| ?- prove(s = [cut(s14,s17,d&e), [a&(b->(c&e))] => [d]]).
yes
| ?- normal(s).

```

Proof is as follows :

```

=====
s1=[f,[a]=>[b]]
s2=[id(c),[c]=>[c]]
s3=[th(s2,e),[c,e]=>[c]]
s4=[aia(s3,c&e),[c&e]=>[c]]
s5=[g,[c]=>[d]]
s6=[cut(s4,s5,c),[c&e]=>[d]]
s7=[id(e),[e]=>[e]]
s8=[th(s7,c),[c,e]=>[e]]
s9=[aia(s8,c&e),[c&e]=>[e]]
s10=[id(e),[e]=>[e]]
s11=[cut(s9,s10,e),[c&e]=>[e]]
s12=[ais(s6,s11,d&e),[c&e]=>[d&e]]
s13=[hia(s1,s12,b->c&e),[a,b->c&e]=>[d&e]]
s14=[aia(s13,a&(b->c&e)),[a&(b->c&e)]=>[d&e]]
s15=[id(d),[d]=>[d]]
s16=[th(s15,e),[d,e]=>[d]]
s17=[aia(s16,d&e),[d&e]=>[d]]
s=[cut(s14,s17,d&e),[a&(b->c&e)]=>[d]]

```

Cut\_free proof is :

```

=====
f19=[f,[a]=>[b]]
f48=[g,[c]=>[d]]
f45=[th(f48,e),[c,e]=>[d]]
f42=[aia(f45,c&e),[c&e]=>[d]]
f39=[th(f42,e),[c&e,e]=>[d]]
f36=[th(f39,c),[c&e,c,e]=>[d]]
f24=[aia(f36,c&e),[c&e,c&e]=>[d]]
f18=[con(f24,c&e),[c&e]=>[d]]
f15=[hia(f19,f18,b->c&e),[a,b->c&e]=>[d]]
s=[aia(f15,a&(b->c&e)),[a&(b->c&e)]=>[d]]

```

Normal proof is :

```

=====
f19=[f,[a]=>[b]]
f48=[g,[c]=>[d]]
f45=[th(f48,e),[c,e]=>[d]]
f18=[aia(f45,c&e),[c&e]=>[d]]
f15=[hia(f19,f18,b->c&e),[a,b->c&e]=>[d]]
s=[aia(f15,a&(b->c&e)),[a&(b->c&e)]=>[d]]

```

which in tree form is :

$$\begin{array}{c}
 \begin{array}{c}
 \begin{array}{c}
 \begin{array}{c}
 c \xrightarrow{g} d \\
 \hline
 c, e \rightarrow d \\
 \hline
 c \uparrow e \rightarrow d
 \end{array} \\
 \hline
 a \xrightarrow{f} b \quad \quad \quad c \uparrow e \rightarrow d \\
 \hline
 a, b \Rightarrow (c \uparrow e) \rightarrow d \\
 \hline
 a \uparrow (b \Rightarrow (c \uparrow e)) \rightarrow d
 \end{array}
 \end{array}
 \end{array}$$

(4.5.9) The CHURCH-ROSSER Theorem for  $Dccl(X)$

If  $p \equiv q$ , then there exists a normal  $r$  in  $Dccl(X)$  such that  $p \geq r$  and  $q \geq r$ .

(4.6) Applications to Category theory

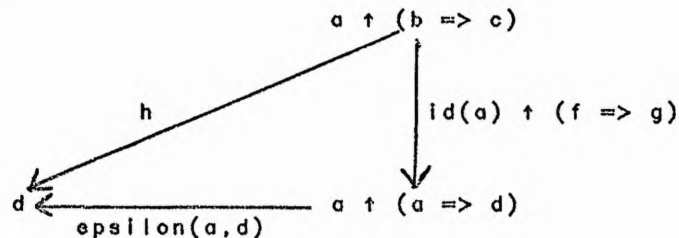
In this section, we illustrate the use of the system to represent some arrows of  $Fccl(X)$ , to obtain their normal representations and (where possible) to show the equality of arrows of  $Fccl(X)$ .

Example 1

For arrows  $f$  in  $X(a,b)$  and  $g$  in  $X(c,d)$ , the normal representation of the arrow

$$h = \text{comp}(\text{epsilon}(a,d), \text{product}(\text{id}(a), \text{hook}(f,g))) : a \uparrow (b \Rightarrow c) \rightarrow d$$

In the diagram



is obtained, as follows :

```

yes
| ?- theory(cartcl).
yes
| ?- arrow(a,f,b).
yes
| ?- arrow(c,g,d).
yes
| ?- norm_rep(comp(epsilon(a,d),product(id(a),hook(f,g)))).
The arrow is :
=====

```

```

comp(epsilon(a,d),product(id(a),hook(f,g)))

```

```

Proof is as follows :
=====

```

```

f24= [id(a), [a] => [a]]
f23= [th(f24, b->c), [a, b->c] => [a]]
f22= [aia(f23, a&(b->c)), [a&(b->c)] => [a]]
f7= [id(a), [a] => [a]]
f25= [cut(f22, f7, a), [a&(b->c)] => [a]]
f30= [id(b), [b] => [b]]
f31= [id(c), [c] => [c]]
f29= [hia(f30, f31, b->c), [b, b->c] => [c]]
f20= [his(f29, b->c), [b->c] => [b->c]]
f27= [th(f20, a), [a, b->c] => [b->c]]
f26= [aia(f27, a&(b->c)), [a&(b->c)] => [b->c]]
f17= [id(a), [a] => [a]]

```

```

f15=[th(f17,b->c),[a,b->c]=>[a]]
f20=[id(b),[b]=>[b]]
f21=[id(c),[c]=>[c]]
f19=[hia(f20,f21,b->c),[b,b->c]=>[c]]
f18=[his(f19,b->c),[b->c]=>[b->c]]
f16=[th(f18,a),[a,b->c]=>[b->c]]
f14=[ais(f15,f16,a&(b->c)),[a,b->c]=>[a&(b->c)]]
f10=[f,[a]=>[b]]
f11=[g,[c]=>[d]]
f12=[hia(f10,f11,b->c),[a,b->c]=>[d]]
f9=[aia(f12,a&(b->c)),[a&(b->c)]=>[d]]
f13=[cut(f14,f9,a&(b->c)),[a,b->c]=>[d]]
f8=[his(f13,a->d),[b->c]=>[a->d]]
f32=[cut(f26,f8,b->c),[a&(b->c)]=>[a->d]]
f6=[ais(f25,f32,a&(a->d)),[a&(b->c)]=>[a&(a->d)]]
f4=[id(a),[a]=>[a]]
f5=[id(d),[d]=>[d]]
f3=[hia(f4,f5,a->d),[a,a->d]=>[d]]
f2=[aia(f3,a&(a->d)),[a&(a->d)]=>[d]]
f1=[cut(f6,f2,a&(a->d)),[a&(b->c)]=>[d]]

```

Cut\_free proof is :

```

f251=[f,[a]=>[b]]
f248=[th(f251,b->c),[a,b->c]=>[b]]
f245=[aia(f248,a&(b->c)),[a&(b->c)]=>[b]]
f242=[th(f245,b->c),[a&(b->c),b->c]=>[b]]
f239=[th(f242,a),[a&(b->c),a,b->c]=>[b]]
f235=[aia(f239,a&(b->c)),[a&(b->c),a&(b->c)]=>[b]]
f236=[g,[c]=>[d]]
f232=[hia(f235,f236,b->c),[a&(b->c),a&(b->c),b->c]=>[d]]
f229=[th(f232,a),[a&(b->c),a&(b->c),a,b->c]=>[d]]
f226=[aia(f229,a&(b->c)),[a&(b->c),a&(b->c),a&(b->c)]=>[d]]
f223=[th(f226,a),[a&(b->c),a&(b->c),a,a&(b->c)]=>[d]]
f220=[inc(f223,a,a&(b->c)),[a&(b->c),a,a&(b->c)]=>[d]]
f217=[th(f220,b->c),[a&(b->c),a,b->c,a&(b->c),a&(b->c)]=>[d]]
f202=[aia(f217,a&(b->c)),[a&(b->c),a&(b->c),a&(b->c),a&(b->c)]=>[d]]
f197=[con(f202,a&(b->c)),[a&(b->c),a&(b->c),a&(b->c)]=>[d]]
f147=[con(f197,a&(b->c)),[a&(b->c),a&(b->c)]=>[d]]
f1=[con(f147,a&(b->c)),[a&(b->c)]=>[d]]

```

Normal proof is :

```

f251=[f,[a]=>[b]]
f236=[g,[c]=>[d]]
f248=[hia(f251,f236,b->c),[a,b->c]=>[d]]
f1=[aia(f248,a&(b->c)),[a&(b->c)]=>[d]]

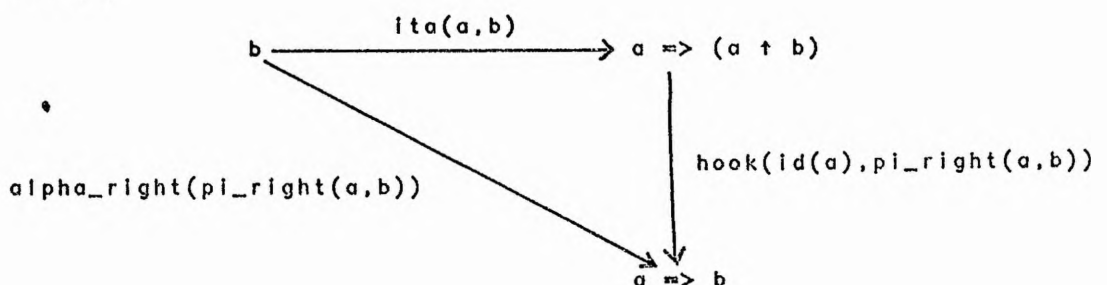
```

which in tree form is :

$$\begin{array}{c}
 \begin{array}{ccc}
 f & & g \\
 a \rightarrow b & & c \rightarrow d \\
 \hline
 a, b \Rightarrow c \rightarrow d \\
 \hline
 a \uparrow (b \Rightarrow c) \rightarrow d
 \end{array}
 \end{array}$$

Example 2

Let a,b be objects of the category X; then the following diagram in  $\text{Foc1}(X)$



commutes .

i.e by the use of the system we prove that

$$\text{alpha\_right}(\text{pi\_right}(a,b)) = \text{comp}(\text{hook}(\text{id}(a),\text{pi\_right}(a,b)),\text{ita}(a,b))$$

as follows :

```
yes
| ?- theory(cartcl).
```

```
yes
| ?- commutative.
```

Insert the first arrow :  $\text{alpha\_right}(\text{pi\_right}(a,b))$ .

The arrow is :

```
=====
alpha_right(pi_right(a,b))
```

Proof is as follows :

```
=====
f9=[id(a),[a]=>[a]]
f7=[th(f9,b),[a,b]=>[a]]
f10=[id(b),[b]=>[b]]
f8=[th(f10,a),[a,b]=>[b]]
f6=[als(f7,f8,a&b),[a,b]=>[a&b]]
f4=[id(b),[b]=>[b]]
f3=[th(f4,a),[a,b]=>[b]]
f2=[ala(f3,a&b),[a&b]=>[b]]
f5=[cut(f6,f2,a&b),[a,b]=>[b]]
f1=[his(f5,a->b),[b]=>[a->b]]
```

Cut\_free proof is :

```
=====
f33=[id(b),[b]=>[b]]
f32=[th(f33,a),[a,b]=>[b]]
f29=[th(f32,b),[a,b,b]=>[b]]
f15=[th(f29,a),[a,b,a,b]=>[b]]
f18=[inc(f15,a,b),[a,a,b,b]=>[b]]
f19=[con(f18,a),[a,b,b]=>[b]]
f5=[con(f19,b),[a,b]=>[b]]
f1=[his(f5,a->b),[b]=>[a->b]]
```

Normal proof is :

```
=====
f33=[id(b),[b]=>[b]]
f5=[th(f33,a),[a,b]=>[b]]
f1=[his(f5,a->b),[b]=>[a->b]]
```

Insert the second arrow :  $\text{comp}(\text{hook}(\text{id}(a),\text{pi\_right}(a,b)),\text{ita}(a,b))$ .

The arrow is :

```
=====
comp(hook(id(a),pi_right(a,b)),ita(a,b))
```

Proof is as follows :

```
=====
f60=[id(a),[a]=>[a]]
f58=[th(f60,b),[a,b]=>[a]]
f61=[id(b),[b]=>[b]]
f59=[th(f61,a),[a,b]=>[b]]
```



```

f57=[ a is (f58, f59, a&b), [a, b] => [a&b]]
f56=[ his (f57, a -> a&b), [b] => [a -> a&b]]
f46=[ id(a), [a] => [a]]
f44=[ th (f46, a -> a&b), [a, a -> a&b] => [a]]
f49=[ id(a), [a] => [a]]
f54=[ id(a), [a] => [a]]
f52=[ th (f54, b), [a, b] => [a]]
f55=[ id(b), [b] => [b]]
f53=[ th (f55, a), [a, b] => [b]]
f51=[ a is (f52, f53, a&b), [a, b] => [a&b]]
f50=[ a is (f51, a&b), [a&b] => [a&b]]
f48=[ his (f49, f50, a -> a&b), [a, a -> a&b] => [a&b]]
f47=[ his (f48, a -> a&b), [a -> a&b] => [a -> a&b]]
f45=[ th (f47, a), [a, a -> a&b] => [a -> a&b]]
f43=[ a is (f44, f45, a&(a -> a&b)), [a, a -> a&b] => [a&(a -> a&b)]]
f37=[ id(a), [a] => [a]]
f40=[ id(b), [b] => [b]]
f39=[ th (f40, a), [a, b] => [b]]
f38=[ a is (f39, a&b), [a&b] => [b]]
f41=[ his (f37, f38, a -> a&b), [a, a -> a&b] => [b]]
f36=[ a is (f41, a&(a -> a&b)), [a&(a -> a&b)] => [b]]
f42=[ cut (f43, f36, a&(a -> a&b)), [a, a -> a&b] => [b]]
f35=[ his (f42, a -> b), [a -> a&b] => [a -> b]]
f34=[ cut (f56, f35, a -> a&b), [b] => [a -> b]]

```

Cut free proof is :

```

f281=[ id(b), [b] => [b]]
f438=[ th (f281, a), [a, b] => [b]]
f437=[ th (f438, b), [a, b, b] => [b]]
f434=[ th (f437, b), [a, b, b, b] => [b]]
f431=[ th (f434, b), [a, b, b, b, b] => [b]]
f428=[ th (f431, a), [a, b, b, a, b, b] => [b]]
f425=[ th (f428, b), [a, b, b, b, a, b, b] => [b]]
f422=[ th (f425, a), [a, b, b, a, b, a, b, b] => [b]]
f419=[ th (f422, b), [a, b, b, a, b, b, a, b, b] => [b]]
f416=[ th (f419, a), [a, b, b, a, b, b, a, b, b] => [b]]
f413=[ inc (f416, a, b), [a, b, b, a, b, a, b, a, b, b] => [b]]
f410=[ inc (f413, a, b), [a, b, b, b, a, b, a, b, a, b, b] => [b]]
f407=[ con (f410, a), [a, b, b, a, b, b, a, b, a, b, b] => [b]]
f404=[ inc (f407, a, b), [a, b, b, a, b, a, b, a, b, a, b, b] => [b]]
f401=[ inc (f404, a, b), [a, b, b, a, b, a, b, a, b, b, a, b, b] => [b]]
f398=[ con (f401, a), [a, b, b, a, b, a, b, a, b, b, a, b, b] => [b]]
f395=[ con (f398, b), [a, b, b, a, b, a, b, a, b, b, a, b, b] => [b]]
f392=[ con (f395, b), [a, b, b, a, b, a, b, a, b, b, a, b, b] => [b]]
f389=[ inc (f392, a, b), [a, b, b, a, b, a, b, a, b, b, a, b, b] => [b]]
f386=[ inc (f389, a, b), [a, b, b, a, b, a, b, a, b, b, a, b, b] => [b]]
f298=[ con (f386, a), [a, b, b, a, b, a, b, a, b, b, a, b, b] => [b]]
f293=[ con (f298, b), [a, b, b, a, b, a, b, a, b, b, a, b, b] => [b]]
f290=[ con (f293, b), [a, b, b, a, b, a, b, a, b, b, a, b, b] => [b]]
f287=[ th (f290, a), [a, b, a, b] => [b]]
f284=[ inc (f287, a, b), [a, a, b, b] => [b]]
f131=[ con (f284, a), [a, b, b] => [b]]
f126=[ con (f131, b), [a, b] => [b]]
f34=[ his (f126, a -> b), [b] => [a -> b]]

```

Normal proof is :

```

f131=[ id(b), [b] => [b]]
f126=[ th (f131, a), [a, b] => [b]]
f34=[ his (f126, a -> b), [b] => [a -> b]]

```

The answer is :

```

alpha_right(pi_right(a,b)) =
comp(hook(id(a), pi_right(a,b)), ita(a,b))

```

because they have the same normal representation .

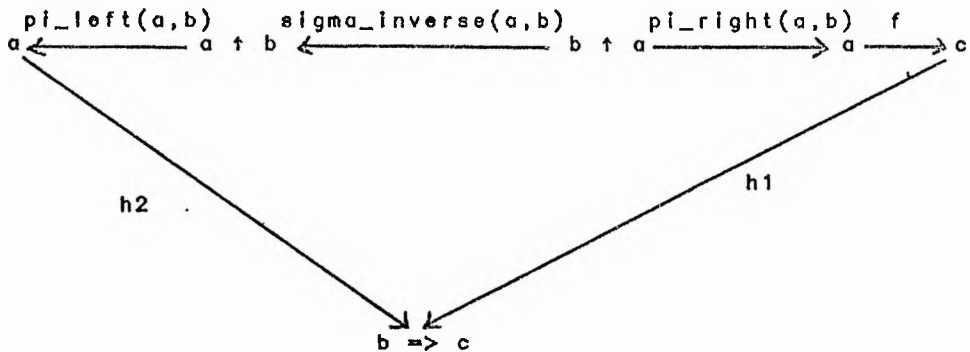
```

yes
| ?-

```

Example 3

Suppose  $a, b, c$  are objects of the category  $X$ ; and that  $f \in X(a, c)$ . Then the following diagram of the free cartesian closed category  $Fcc1(X)$



where

$h1 = \text{alpha\_right}(\text{comp}(f, \text{pi\_right}(b, a))),$

$h2 = \text{alpha\_right}(\text{comp}(f, \text{comp}(\text{pi\_left}(a, b), \text{sigma\_inverse}(a, b))))$

commutes.

i.e. by the use of the system we can prove

$$\begin{aligned}
 & \text{comp}(\text{alpha\_right}(\text{comp}(f, \text{pi\_right}(b, a))), \text{pi\_right}(b, a)) \\
 = & \text{comp}(\text{comp}(\text{alpha\_right}(\text{comp}(\text{comp}(f, \text{pi\_left}(a, b)), \\
 & \text{sigma\_inverse}(a, b))), \\
 & \text{pi\_left}(a, b)), \\
 & \text{sigma\_inverse}(a, b))
 \end{aligned}$$

as follows :

```

yes
| ?- theory(cartcl).

yes
| ?- arrow(a,f,c).

yes
| ?- commutative.
Insert the first arrow : comp(alpha_right(comp(f,pi_right(b,a))),
|: pi_right(b,a)).

```

The arrow is :

=====

$\text{comp}(\text{alpha\_right}(\text{comp}(f, \text{pi\_right}(b, a))), \text{pi\_right}(b, a))$

Proof is as follows :

=====

```

f16= [id(a), [a]=>[a]]
f15= [th(f16,b), [b,a]=>[a]]
f14= [aia(f15,b&a), [b&a]=>[a]]
f12= [id(b), [b]=>[b]]
f10= [th(f12,a), [b,a]=>[b]]
f13= [id(a), [a]=>[a]]
f11= [th(f13,b), [b,a]=>[a]]
f9= [ais(f10,f11,b&a), [b,a]=>[b&a]]
f7= [id(a), [a]=>[a]]

```

```

f6= [th(f7,b),[b,a]=>[a]]
f5= [aia(f6,b&a),[b&a]=>[a]]
f4= [f,[a]=>[c]]
f3= [cut(f5,f4,a),[b&a]=>[c]]
f8= [cut(f9,f3,b&a),[b,a]=>[c]]
f2= [his(f8,b->c),[a]=>[b->c]]
f1= [cut(f14,f2,a),[b&a]=>[b->c]]

```

Cut\_free proof is :

=====

```

f76= [f,[a]=>[c]]
f73= [th(f76,b),[b,a]=>[c]]
f70= [aia(f73,b&a),[b&a]=>[c]]
f70= [aia(f73,b&a),[b&a]=>[c]]
f91= [th(f92,a),[b,a,b&a]=>[c]]
f88= [th(f91,b),[b,b,a,b&a]=>[c]]
f85= [aia(f88,b&a),[b,b&a,b&a]=>[c]]
f82= [th(f85,b),[b,b&a,b,b&a]=>[c]]
f79= [inc(f82,b,b&a),[b,b,b&a,b&a]=>[c]]
f53= [con(f79,b),[b,b&a,b&a]=>[c]]
f48= [con(f53,b&a),[b,b&a]=>[c]]
f1= [his(f48,b->c),[b&a]=>[b->c]]

```

Normal proof is :

=====

```

f76= [f,[a]=>[c]]
f73= [th(f76,b),[b,a]=>[c]]
f53= [aia(f73,b&a),[b&a]=>[c]]
f48= [th(f53,b),[b,b&a]=>[c]]
f1= [his(f48,b->c),[b&a]=>[b->c]]

```

Insert the second arrow :

```

|:      comp(comp(alpha_right(comp(f,pi_left(a,b))),
|:      sigma_inverse(a,b))),
|:      pi_left(a,b)),
|:      sigma_inverse(a,b)).

```

The arrow is :

=====

```

comp(comp(alpha_right(comp(comp(f,pi_left(a,b))),
sigma_inverse(a,b))),
pi_left(a,b)),
sigma_inverse(a,b))

```

Proof is as follows :

=====

```

f121= [id(a),[a]=>[a]]
f120= [th(f121,b),[b,a]=>[a]]
f119= [aia(f120,b&a),[b&a]=>[a]]
f124= [id(b),[b]=>[b]]
f123= [th(f124,a),[b,a]=>[b]]
f122= [aia(f123,b&a),[b&a]=>[b]]
f118= [ais(f119,f122,a&b),[b&a]=>[a&b]]
f117= [id(a),[a]=>[a]]
f116= [th(f117,b),[a,b]=>[a]]
f115= [aia(f116,a&b),[a&b]=>[a]]
f113= [id(b),[b]=>[b]]
f111= [th(f113,a),[b,a]=>[b]]
f114= [id(a),[a]=>[a]]
f112= [th(f114,b),[b,a]=>[a]]
f110= [ais(f111,f112,b&a),[b,a]=>[b&a]]
f105= [id(a),[a]=>[a]]
f104= [th(f105,b),[b,a]=>[a]]
f103= [aia(f104,b&a),[b&a]=>[a]]
f108= [id(b),[b]=>[b]]
f107= [th(f108,a),[b,a]=>[b]]
f106= [aia(f107,b&a),[b&a]=>[b]]
f102= [ais(f103,f106,a&b),[b&a]=>[a&b]]
f101= [id(a),[a]=>[a]]
f100= [th(f101,b),[a,b]=>[a]]
f99= [aia(f100,a&b),[a&b]=>[a]]
f98= [f,[a]=>[c]]
f97= [cut(f99,f98,a),[a&b]=>[c]]

```

```

f96=[cut(f102,f97,a&b),[b&a]=>[c]]
f109=[cut(f110,f96,b&a),[b,a]=>[c]]
f95=[his(f109,b->c),[a]=>[b->c]]
f94=[cut(f115,f95,a),[a&b]=>[b->c]]
f93=[cut(f118,f94,a&b),[b&a]=>[b->c]]

```

Cut\_free proof is :

```

=====
f608=[f,[a]=>[c]]
f605=[th(f608,b),[b,a]=>[c]]
f602=[aia(f605,b&a),[b&a]=>[c]]
f599=[th(f602,b),[b&a,b]=>[c]]
f596=[th(f599,a),[b&a,b,a]=>[c]]
f584=[aia(f596,b&a),[b&a,b&a]=>[c]]
f579=[con(f584,b&a),[b&a]=>[c]]
f647=[th(f579,b),[b,b&a]=>[c]]
f646=[th(f647,a),[b,a,b&a]=>[c]]
f643=[th(f646,b),[b,b,a,b&a]=>[c]]
f640=[aia(f643,b&a),[b,b&a,b&a]=>[c]]
f637=[th(f640,b),[b,b&a,b,b&a]=>[c]]
f634=[th(f637,a),[b,b&a,b,a,b&a]=>[c]]
f622=[aia(f634,b&a),[b,b&a,b&a,b&a]=>[c]]
f617=[con(f622,b&a),[b,b&a,b&a]=>[c]]
f614=[th(f617,b),[b,b&a,b,b&a]=>[c]]
f611=[inc(f614,b,b&a),[b,b,b&a,b&a]=>[c]]
f556=[con(f611,b),[b,b&a,b&a]=>[c]]
f551=[con(f556,b&a),[b,b&a]=>[c]]
f548=[th(f551,b),[b,b&a,b]=>[c]]
f545=[th(f548,a),[b,b&a,b,a]=>[c]]
f542=[th(f545,b),[b,b&a,b,b,a]=>[c]]
f539=[aia(f542,b&a),[b,b&a,b,b&a]=>[c]]
f536=[th(f539,b),[b,b&a,b,b&a,b]=>[c]]
f533=[th(f536,a),[b,b&a,b,b&a,b,a]=>[c]]
f530=[aia(f533,b&a),[b,b&a,b,b&a,b&a]=>[c]]
f527=[con(f530,b&a),[b,b&a,b,b&a]=>[c]]
f524=[th(f527,a),[b,b&a,b,a,b&a]=>[c]]
f521=[th(f524,b),[b,b&a,b,b,a,b&a]=>[c]]
f518=[aia(f521,b&a),[b,b&a,b,b&a,b&a]=>[c]]
f515=[th(f518,b),[b,b&a,b,b&a,b,b&a]=>[c]]
f512=[th(f515,a),[b,b&a,b,b&a,b,a,b&a]=>[c]]
f509=[aia(f512,b&a),[b,b&a,b,b&a,b&a,b&a]=>[c]]
f506=[con(f509,b&a),[b,b&a,b,b&a,b&a]=>[c]]
f503=[th(f506,b),[b,b&a,b,b&a,b,b&a]=>[c]]
f500=[inc(f503,b,b&a),[b,b&a,b,b,b&a,b&a]=>[c]]
f497=[con(f500,b),[b,b&a,b,b&a,b&a]=>[c]]
f494=[con(f497,b&a),[b,b&a,b,b&a]=>[c]]
f491=[inc(f494,b,b&a),[b,b,b&a,b&a]=>[c]]
f386=[con(f491,b),[b,b&a,b&a]=>[c]]
f381=[con(f386,b&a),[b,b&a]=>[c]]
f93=[his(f381,b->c),[b&a]=>[b->c]]

```

Normal proof is :

```

=====
f608=[f,[a]=>[c]]
f605=[th(f608,b),[b,a]=>[c]]
f556=[aia(f605,b&a),[b&a]=>[c]]
f381=[th(f556,b),[b,b&a]=>[c]]
f93=[his(f381,b->c),[b&a]=>[b->c]]

```

The answer is :

```

=====
comp(alpha_right(comp(f,pi_right(b,a))),pi_right(b,a)) =
comp(comp(alpha_right(comp(comp(f,pi_left(a,b)),
sigma_inverse(a,b))),
pi_left(a,b)),
sigma_inverse(a,b))

```

because they have the same normal representation .

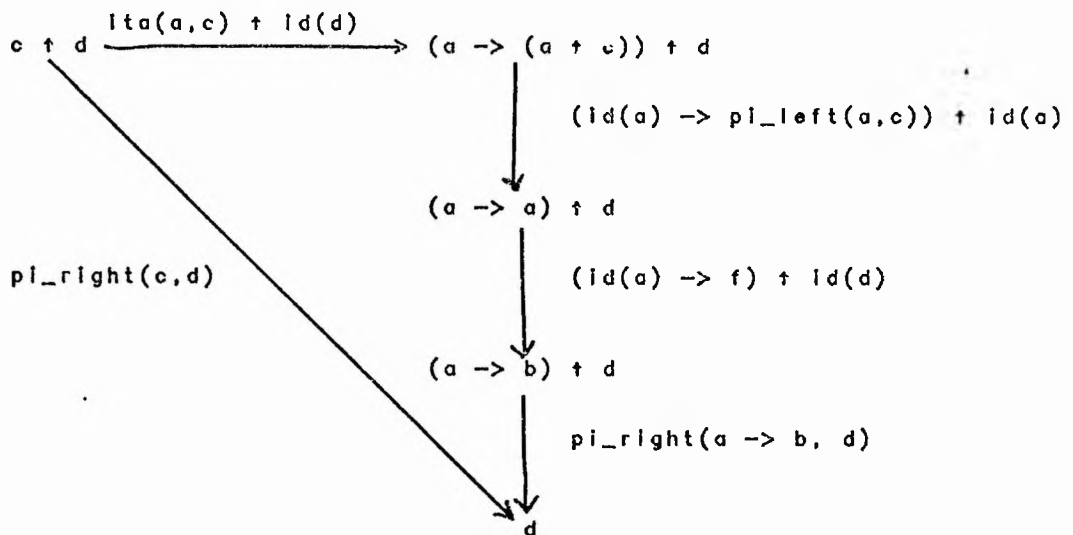
```

yes
| ?-

```

Example 4

For  $f$  in  $X(a,b)$  and  $a,b,c,d$  are objects of the category  $X$ ; then the following diagram of  $\text{Focl}(X)$



commutes.

```
yes
| ?- theory(cartcl).
```

```
yes
| ?- arrow(a,f,b).
```

```
yes
| ?- commutative.
Insert the first arrow : pi_right(c,d).
```

The arrow is :

```
=====
pi_right(c,d)
```

Proof is as follows :

```
=====
f3=[id(d),[d]=>[d]]
f2=[th(f3,c),[c,d]=>[d]]
f1=[aia(f2,c&d),[c&d]=>[d]]
```

Cut\_free proof is :

```
=====
f3=[id(d),[d]=>[d]]
f2=[th(f3,c),[c,d]=>[d]]
f1=[aia(f2,c&d),[c&d]=>[d]]
```

Normal proof is :

```
=====
f3=[id(d),[d]=>[d]]
f2=[th(f3,c),[c,d]=>[d]]
f1=[aia(f2,c&d),[c&d]=>[d]]
```

Insert the second arrow :

```

:      comp(comp(comp(pl_right(a->b,d),
:                  product(hook(id(a),f),id(d))),
:                  product(hook(id(a),pi_left(a,c)),id(d))),
:                  product(ita(a,c),id(d))).

```

The arrow is :

```

=====
comp(comp(comp(pl_right(a->b,d),product(hook(id(a),f),id(d))),
          product(hook(id(a),pi_left(a,c)),id(d))),
      product(ita(a,c),id(d)))

```

Proof is as follows :

```

=====
f86= [ld(c), [c] => [c]]
f85= th(f86, d), [c, d] => [c]
f84= aia(f85, c&d), [c&d] => [c]
f81= [ld(a), [a] => [a]]
f79= th(f81, c), [a, c] => [a]
f82= [ld(c), [c] => [c]]
f80= th(f82, a), [a, c] => [c]
f78= ats(f79, f80, a&c), [a, c] => [a&c]
f77= his(f78, a->a&c), [c] => [a->a&c]
f87= cut(f84, f77, c), [c&d] => [a->a&c]
f90= [ld(d), [d] => [d]]
f89= th(f90, c), [c, d] => [d]
f88= aia(f89, c&d), [c&d] => [d]
f83= [ld(d), [d] => [d]]
f91= cut(f88, f83, d), [c&d] => [d]
f76= ais(f87, f91, (a->a&c)&d), [c&d] => [(a->a&c)&d]
f64= [ld(a), [a] => [a]]
f69= [ld(a), [a] => [a]]
f67= th(f69, c), [a, c] => [a]
f70= [ld(c), [c] => [c]]
f68= th(f70, a), [a, c] => [c]
f66= ais(f67, f68, a&c), [a, c] => [a&c]
f65= aia(f66, a&c), [a&c] => [a&c]
f63= hia(f64, f65, a->a&c), [a, a->a&c] => [a&c]
f62= his(f63, a->a&c), [a->a&c] => [a->a&c]
f61= th(f62, d), [a->a&c, d] => [a->a&c]
f60= aia(f61, (a->a&c)&d), [(a->a&c)&d] => [a->a&c]
f49= [ld(a), [a] => [a]]
f47= th(f49, a->a&c), [a, a->a&c] => [a]
f52= [ld(a), [a] => [a]]
f57= [ld(a), [a] => [a]]
f55= th(f57, c), [a, c] => [a]
f58= [ld(c), [c] => [c]]
f56= th(f58, a), [a, c] => [c]
f54= ais(f55, f56, a&c), [a, c] => [a&c]
f53= aia(f54, a&c), [a&c] => [a&c]
f51= hia(f52, f53, a->a&c), [a, a->a&c] => [a&c]
f50= his(f51, a->a&c), [a->a&c] => [a->a&c]
f48= th(f50, a), [a, a->a&c] => [a->a&c]
f46= ais(f47, f48, a&(a->a&c)), [a, a->a&c] => [a&(a->a&c)]
f40= [ld(a), [a] => [a]]
f43= [ld(a), [a] => [a]]
f42= th(f43, c), [a, c] => [a]
f41= aia(f42, a&c), [a&c] => [a]
f44= hia(f40, f41, a->a&c), [a, a->a&c] => [a]
f39= aia(f44, a&(a->a&c)), [a&(a->a&c)] => [a]
f45= cut(f46, f39, a&(a->a&c)), [a, a->a&c] => [a]
f38= his(f45, a->a), [a->a&c] => [a->a]
f71= cut(f60, f38, a->a&c), [(a->a&c)&d] => [a->a]
f74= [ld(d), [d] => [d]]
f73= th(f74, a->a&c), [a->a&c, d] => [d]
f72= aia(f73, (a->a&c)&d), [(a->a&c)&d] => [d]
f59= [ld(d), [d] => [d]]
f75= cut(f72, f59, d), [(a->a&c)&d] => [d]
f37= ais(f71, f75, (a->a)&d), [(a->a&c)&d] => [(a->a)&d]
f30= [ld(a), [a] => [a]]
f31= [ld(a), [a] => [a]]
f29= hia(f30, f31, a->a), [a, a->a] => [a]
f28= his(f29, a->a), [a->a] => [a->a]
f27= th(f28, d), [a->a, d] => [a->a]
f26= aia(f27, (a->a)&d), [(a->a)&d] => [a->a]
f20= [ld(a), [a] => [a]]
f18= th(f20, a->a), [a, a->a] => [a]

```

```

f23 = [id(a), [a] => [a]]
f24 = [id(a), [a] => [a]]
f22a = [hia(f23, f24, a->a), [a, a->a] => [a]]
f21 = [his(f22, a->a), [a->a] => [a->a]]
f19 = [th(f21, a), [a, a->a] => [a->a]]
f17 = [als(f18, f19, a&(a->a)), [a, a->a] => [a&(a->a)]]
f13 = [id(a), [a] => [a]]
f14 = [f, [a] => [b]]
f15 = [hia(f13, f14, a->a), [a, a->a] => [b]]
f12 = [aia(f15, a&(a->a)), [a&(a->a)] => [b]]
f16 = [cut(f17, f12, a&(a->a)), [a, a->a] => [b]]
f11 = [his(f16, a->b), [a->a] => [a->b]]
f32 = [cut(f26, f11, a->a), [(a->a)&d] => [a->b]]
f35 = [id(d), [d] => [d]]
f34 = [th(f35, a->a), [a->a, d] => [d]]
f33 = [aia(f34, (a->a)&d), [(a->a)&d] => [d]]
f25 = [id(d), [d] => [d]]
f36 = [cut(f33, f25, d), [(a->a)&d] => [d]]
f10 = [als(f32, f36, (a->b)&d), [(a->a)&d] => [(a->b)&d]]
f9 = [id(d), [d] => [d]]
f8 = [th(f9, a->b), [a->b, d] => [d]]
f7 = [aia(f8, (a->b)&d), [(a->b)&d] => [d]]
f6 = [cut(f10, f7, (a->b)&d), [(a->a)&d] => [d]]
f5 = [cut(f37, f6, (a->a)&d), [(a->a)&c] => [d]]
f4 = [cut(f76, f5, (a->a)&c), [c&d] => [d]]

```

Cut\_free proof is :

```

f695 = [id(d), [d] => [d]]
f692 = [th(f695, c), [c, d] => [d]]
f696 = [aia(f692, c&d), [c&d] => [d]]
f684 = [th(f696, c&d), [c&d, c&d] => [d]]
f679 = [con(f684, c&d), [c&d] => [d]]
f676 = [th(f679, d), [c&d, d] => [d]]
f673 = [th(f676, c), [c&d, c, d] => [d]]
f670 = [aia(f673, c&d), [c&d, c&d] => [d]]
f667 = [th(f670, c&d), [c&d, c&d, c&d] => [d]]
f641 = [con(f667, c&d), [c&d, c&d] => [a]]
f636 = [con(f641, c&d), [c&d] => [d]]
f633 = [th(f636, d), [c&d, d] => [d]]
f630 = [th(f633, c), [c&d, c, d] => [d]]
f627 = [aia(f630, c&d), [c&d, c&d] => [d]]
f624 = [th(f627, c&d), [c&d, c&d, c&d] => [d]]
f621 = [con(f624, c&d), [c&d, c&d] => [d]]
f618 = [th(f621, d), [c&d, c&d, d] => [d]]
f615 = [th(f618, c), [c&d, c&d, c, d] => [d]]
f612 = [aia(f615, c&d), [c&d, c&d, c&d] => [d]]
f609 = [th(f612, c&d), [c&d, c&d, c&d, c&d] => [d]]
f570 = [con(f609, c&d), [c&d, c&d, c&d] => [d]]
f570 = [con(f609, c&d), [c&d, c&d, c&d] => [d]]
f4 = [con(f539, c&d), [c&d] => [d]]

```

Normal proof is :

```

f695 = [id(d), [d] => [d]]
f692 = [th(f695, c), [c, d] => [d]]
f4 = [aia(f692, c&d), [c&d] => [d]]

```

The answer is :

```

pi_right(c, d) =
comp(comp(comp(pi_right(a->b, d), product(hook(id(a), f), id(d))),
      product(hook(id(a), pi_left(a, c)), id(d))),
      product(ita(a, c), id(d)))

```

because they have the same normal representation .

```

yes
| ?-

```

Chapter 5  
=====

BICARTESIAN CLOSED CATEGORY THEORY  
=====

In this chapter, we combine all results and implementations of cartesian, bicartesian, distributive bicartesian and cartesian closed categories in order to complete the system to handle bicartesian closed categories. By a bicartesian category is meant a category with an initial and object  $I$ , a terminal object  $T$ , and binary operators  $\dagger$ ,  $\vee$ , and  $\Rightarrow$ . The class of small bicartesian closed categories serves as a model for the proof theory of the full intuitionist propositional calculus.

We have constructed and implemented a proof for corollary (5.5.4), concerning derivations of the form  $\rightarrow a$  where  $a \approx T$  (the terminal object of the free bicartesian closed category generated by  $X$ ,  $Fbcc(X)$  (cf 5.2.4)). To complete the algorithm for computation of normal form, we have constructed and implemented a proof of lemma (5.5.6), stated without proof in SZABO.

(5.1) Definition

A bicartesian closed category is a bicartesian category  $C$  with the following additional structure :

(7) A bifunctor  $(-)\Rightarrow(-) : op(C) \times C \rightarrow C$ .

(8) An adjunction  $\alpha_\lambda$ , where

$$\alpha_\lambda = \{ \alpha_\lambda(A,B,C) : C(A \dagger B, C) \rightarrow C(B, A \Rightarrow C) \in ArEns \mid A,B,C \in ObC \}.$$

(5.2) The Free Bicartesian Closed Category  $Fbccl(X)$

Let  $X$  be a fixed but arbitrary small category. The free bicartesian closed category  $Fbccl(X)$  is constructed as follows :-

(5.2.1) Definition

The class  $bcccl(X)$  of ( bicartesian closed ) formulae over  $X$  is defined to contain all the ( bicartesian ) formulae of the class  $bcl(X)$  and the ( cartesian closed ) formulae of the class  $ccl(X)$ .

(5.2.2) The Labelled Derivations  $LDbcccl(X)$

The class  $LDbcccl(X)$  of ( bicartesian closed ) labelled derivations over  $X$ , and their conclusions, is defined by conditions (i) - (viii) of (1.2.2), (ix) - (xii) of (2.2.2) and (ix) - (x) of (4.2.2).



(5.2.3) Definition

The relation  $\equiv$  is defined as the smallest equivalence relation on  $LD_{bccl}(X)$  satisfying conditions (i) - (viii) of (1.2.3), conditions (ix) - (xii) of (2.2.3) and conditions (ix) - (xvi) of (4.2.3).

(5.2.4) Definition

The free bicartesian closed category is defined to be the category having :

- i) as objects , the bicartesian closed formulae over  $X$  ;
- ii) as arrows , the  $\equiv$ - classes of  $LD_{bccl}(X)$  ;
- iii) - xi) as in the definition (1.2.5) of  $F_c(X)$  ;
- xii) - xvi) as in the definition (2.2.4) of  $F_{bc}(X)$  ;
- xvii) - xviii) as in the definition (4.2.4) of  $F_{ccl}(X)$  .

(5.3) The Unlabelled Derivations  $D_{bccl}(X)$

The class  $D_{bccl}(X)$  of unlabelled ( bicartesian closed ) derivations is defined inductively by the conditions (i) - (xviii'), except (x) [ we omit (x), since  $R_7$  is an admissible rule : If  $p$  is a derivation with conclusion  $G \rightarrow FabE$ , then

$$R_7(p,b,a) = R_1(R_{13}(p,a \vee b), \\ R_{12}( R_5 ( A_1( id(a) ), b), \\ R_5 ( A_1( id(b) ), a), a \vee b), a \vee b ) \quad ]$$

of (3.3) and the conditions (x) - (xi) of (4.3.1) .

(5.4) The Semantics of  $D_{bccl}(X)$

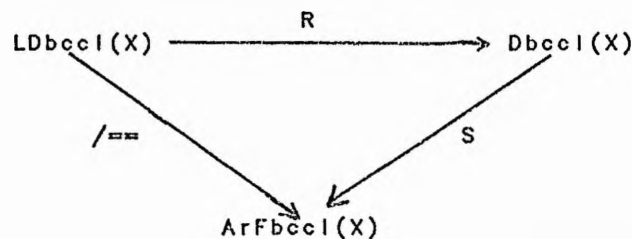
In order to interpret the derivations of  $D_{bccl}(X)$  in  $ArF_{bccl}(X)$ , we combine the interpretations of  $D_c(X)$ ,  $D_{bc}(X)$ ,  $D_{dbc}(X)$  and  $D_{ccl}(X)$  to an interpretation  $S : D_{bccl}(X) \rightarrow ArF_{bccl}(X)$  .

In the following theorem we prove that the class  $D_{bccl}(X)$  is adequate, in the sense that every arrow of  $F_{bccl}(X)$  is representable by means of some derivation of the class  $D_{bccl}(X)$  :

(5.4.1) The Completeness Theorem For  $D_{bccl}(X)$

There is a function

$$R : LD_{bccl}(X) \dashrightarrow D_{bccl}(X) \quad , \quad \text{so that}$$



commutes .

Proof :

We combine the proofs of theorems (1.4.2), (2.4.2), (3.4.2) and (4.4.2) .

###

(5.5) The Syntax of Fbccl(X)

After obtaining a composition-free description of  $ArFbccl(X)$ , in this section we describe the algorithm, to construct cut-free representations for  $ArFbccl(X)$  and to characterize the arrows of  $Fbccl(X)$  by constructing their normal representations.

(5.5.1) The Cut-Elimination Theorem for Dbcccl(X)

For every derivation  $p$  in  $Dbcccl(X)$ , there exists an equivalent cut-free  $q$  in  $Dbcccl(X)$  .

Proof

The proof follows by combining the proofs of theorems (1.5.2), (2.5.2), (3.5.1) and (4.5.1) in addition to the two clauses (C.10) and (C.45) of appendix (2) .

###

(5.5.2) Definition

Let  $L(\perp)$  be the subclass of the class of ( bicartesian closed ) formula  $bcccl(X)$  generated by the conditions :

- 1)  $\perp$  is in  $L(\perp)$ .
- 2) If  $a, b$  are in  $L(\perp)$ , then  $(a \vee b)$  .
- 3) If  $a$  is in  $L(\perp)$  and  $b$  is in  $bcccl(X)$ , then  $(a \dagger b)$ ,  $(b \dagger a)$  are in  $L(\perp)$ .

with the help of  $L(\perp)$ , the subclass  $L(T)$  of the class  $bcccl(X)$  is generated by the conditions :

- 1)  $T$  is in  $L(T)$ .
- 2) If  $a, b$  are in  $L(T)$ , then  $(a \dagger b)$ .
- 3) If  $a$  is in  $L(T)$  and  $b$  is in  $bcccl(X)$ , then  $(b \Rightarrow a)$  is in  $L(T)$ .
- 4) If  $a$  is in  $L(\perp)$  and  $b$  is in  $bcccl(X)$ , then  $(a \Rightarrow b)$  is in  $L(T)$ .
- 5) If  $a$  is in  $L(T)$  and  $b$  is in  $L(\perp)$ , then  $(a \vee b)$ ,  $(b \vee a)$  are in  $L(T)$ .

(5.5.3) Lemma

$a \approx \sim T$  and  $b \approx \sim \perp$  in  $Fbccl(X)$  iff  $a$  is in  $L(T)$  and  $b$  is in  $L(\perp)$ .

(5.5.4) Corollary

If  $b = \sim \perp$ , there exists a unique derivation  $p$  of  $b \rightarrow$  consisting at most of instances of (A3), (R2), (R11), and (R12).

Proof:

Identical to the proof of corollary (3.5.6).

(5.5.5) Corollary

If  $a = \sim T$ , there exists a unique derivation  $p$  of  $\rightarrow a$  consisting at most of instances of (A2), (R2), (R5), (R10), (R13), and (R15).

Proof:

Combining the proofs of corollary (3.5.5) and lemma (4.5.4) and the following :

$a = (c \Rightarrow d)$ ,  $c$  is in  $L(\perp)$  and  $d$  is in  $\text{bccI}(X)$

$$\frac{\frac{c \xrightarrow{h}}{\quad}}{c \rightarrow d} \quad \frac{\quad}{\rightarrow (c \Rightarrow d)}$$

where  $h$  is the unique derivation of  $c \rightarrow$  compatible with the above corollary.

(5.5.6) Lemma

For every cut-free  $p$  in  $\text{DbccI}(X)$  there exists an equivalent cut-free  $q$  in  $\text{DbccI}(X)$  containing no instances of (R6) and (R7), and containing no instances of (R2), (R3), (R4) and (R5) whose active formulae are of the form  $a \uparrow b$  or  $a \vee b$ .

Proof:

In case of (R6) and (R7) the proof follows from lemma (3.5.7) and the additional clauses (E.31) and (E.36) of Appendix(3). The proof in case (R2) and (R5) follows from the proof of lemma (3.5.7). The proof in case (R3) and (R4) is the same as in (4.5.5) but using the more generalized lemma (3.5.7) instead of lemma (1.5.4) and using the priority included in the proof of (5.5.9) below as an extension to that in (4.5.8) and the additional clauses (E.39) and (E.40) of Appendix (3).

###

(5.5.7) Lemma

Every cut-free  $p$  in  $\text{DbccI}(X)$  is equivalent to a cut-free  $q$  in  $\text{DbccI}(X)$  containing no instances of (R14) with the property that the active formulae in their right premisses are isomorphic to  $T$ .

Proof:

Identical to the proof of lemma (4.5.6).

(5.5.8) Definition

A derivation  $p$  in  $\text{Dbccl}(X)$  is normal if it satisfies the following conditions :

- (1) All subderivations of  $p$  not containing instances of (R14) and (R15) are normal in the sense of (3.5.3), with the condition that the active formulae of instances of (R2) and (R5) are atomic replaced by the condition that the active formulae of instances of (R2) and (R5) are either atomic or of the form  $a \Rightarrow b$ .
- (2)  $p$  contains no instances of (R2), (R3), (R4) or (R5) whose active formulae are of the form  $a \uparrow b$  or  $a \vee b$ .
- (3) If  $p$  derives  $G \rightarrow F$ , and if one of the disjunctions of the formulae of  $F$  is isomorphic to  $T$ , then  $p$  is of the form

$$\frac{\begin{array}{c} q \\ \rightarrow F \end{array}}{G \rightarrow F} \quad (K)$$

where  $q$  is the unique derivation of  $\rightarrow F$  compatible with corollary (5.5.5), and where (K) consists of the unique steps required to derive  $G \rightarrow F$  from  $\rightarrow F$ .

- (4) If  $p$  derives  $G \rightarrow F$ , and if one of the conjunctions of the formulae of  $G$  is isomorphic to  $\rightarrow$ , and if no disjunction of the formulae of  $F$  is isomorphic to  $T$ , then  $p$  is of the form

$$\frac{\begin{array}{c} q \\ G \rightarrow \end{array}}{G \rightarrow F} \quad (K)$$

where  $q$  is the unique derivation of  $G \rightarrow$  compatible with corollary (5.5.4), and where (K) consists of the unique steps required to derive  $G \rightarrow F$  from  $G \rightarrow$ .

- (5)  $p$  contains no instance of (R14) whose active formula in the right-hand premiss is isomorphic to  $T$ .
- (6) Using the relation  $\Leftarrow$  as defined in (1.5.4); for normality in  $\text{Dbccl}(X)$  it is required that

$$\begin{aligned} (R5) \Leftarrow (R13) \Leftarrow (R10) \Leftarrow (R12) \Leftarrow (R14) \Leftarrow (R3) \Leftarrow (R11) \Leftarrow (R2) \Leftarrow (R4), \\ (R15) \Leftarrow (R12) \Leftarrow (R14) \Leftarrow (R3) \Leftarrow (R11) \Leftarrow (R2) \Leftarrow (R4). \end{aligned}$$

(5.5.9) The Normalization Theorem For Dbcc1(X)

For every  $p$  in  $\text{Dbcc1}(X)$ , there exists an equivalent normal  $q$  in  $\text{Dbcc1}(X)$ .

Proof :

The theorem follows from theorems (3.5.9) and (4.5.8) and the following additional clauses of Appendix (4) :

(D.17), (D.20), (D.21), (D.25), (D.30), (D.31), (D.39), (D.60), (D.61), (D.63), (D.74), (D.76), and (D.78).

###

The following examples are sessions of input and output with the system for derivations of  $\text{Dbcc1}(X)$  and their equivalents of cut-free and normal form, in order to illustrate the algorithms in (5.5.1), (5.5.6) and (5.5.9) and satisfy the definition in (5.5.8).

Example 1

Suppose  $a, b, c, d$  are objects of the category  $X$ ; and that  $f$  in  $X(a, c)$ ,  $g$  in  $X(d, a)$ ,  $h$  in  $X(d, b)$ ,  $i$  in  $X(d, c)$  with  $\text{comp}(f, g) = i$ . Then the derivation

$$\begin{array}{c}
 \frac{a \rightarrow a \quad b \rightarrow b}{a, b \rightarrow a \quad a, b \rightarrow b} \quad \frac{g \quad h}{d \rightarrow a \quad d \rightarrow b} \quad \frac{h}{d \rightarrow b} \\
 \frac{a, b \rightarrow a \quad a, b \rightarrow b}{a, b \rightarrow a \uparrow b} \quad \frac{f \quad d \rightarrow a \quad d \rightarrow b}{a \rightarrow c \quad d \rightarrow a \uparrow b} \quad \frac{a, d \rightarrow b}{a, b, d \rightarrow b} \quad \frac{f}{a \rightarrow c} \\
 \frac{a, b \rightarrow a \uparrow b}{a \uparrow b \rightarrow a \uparrow b} \quad \frac{a, c \Rightarrow d \rightarrow a \uparrow b}{a, c \Rightarrow d \rightarrow a \uparrow b} \quad \frac{a \uparrow b, d \rightarrow b}{a \uparrow b \rightarrow d \Rightarrow b} \quad \frac{a \uparrow b \rightarrow c}{a \uparrow b \rightarrow c} \\
 \frac{a \uparrow b \rightarrow a \uparrow b}{a \uparrow b \rightarrow c, a \uparrow b} \quad \frac{a, c \Rightarrow d \rightarrow c, a \uparrow b}{a, c \Rightarrow d \rightarrow c, a \uparrow b} \quad \frac{a \uparrow b \rightarrow (d \Rightarrow b) \uparrow c}{a \uparrow b \rightarrow (d \Rightarrow b) \uparrow c} \\
 \frac{a \uparrow b \rightarrow c, a \uparrow b}{(a \uparrow b) \vee (a \uparrow (c \Rightarrow d)) \rightarrow c, a \uparrow b} \quad \frac{a \uparrow (c \Rightarrow d) \rightarrow c, a \uparrow b}{a \uparrow (c \Rightarrow d) \rightarrow c, a \uparrow b} \quad \frac{a \uparrow b, d \rightarrow (d \Rightarrow b) \uparrow c}{a \uparrow b, d \rightarrow (d \Rightarrow b) \uparrow c} \\
 \frac{(a \uparrow b) \vee (a \uparrow (c \Rightarrow d)) \rightarrow c, a \uparrow b}{(a \uparrow b) \vee (a \uparrow (c \Rightarrow d)), d \rightarrow c, (d \Rightarrow b) \uparrow c, e} \quad \frac{a \uparrow b, d \rightarrow (d \Rightarrow b) \uparrow c, e}{a \uparrow b, d \rightarrow (d \Rightarrow b) \uparrow c, e}
 \end{array}$$

is represented, and then transformed to its normal form, as follows

```

yes
| ?- theory(bicartcl).
yes
| ?- composite(f,g,i).
yes
| ?- prove(s1 = [id(a), [a] => [a]]).
yes
| ?- prove(s2 = [th(s1,b), [a,b] => [a]]).
yes
| ?- prove(s3 = [id(b), [b] => [b]]).
yes
| ?- prove(s4 = [th(s3,a), [a,b] => [b]]).
yes
| ?- prove(s5 = [ais(s2,s4,a & b), [a,b] => [a & b]]).
yes
| ?- prove(s6 = [aia(s5 ,a&b), [a&b] => [a&b]]).
yes
| ?- prove(s7 = [th(s6,c), [a&b] => [c,a&b]]).
yes
| ?- prove(s8 = [f, [a] => [c]]).
yes
| ?- prove(s9 = [g, [d] => [a]]).
yes
| ?- prove(s10 = [h, [d] => [b]]).
yes

```

```

| ?- prove(s11 = [ais(s9,s10,a&b), [d] => [a&b]]).
yes
| ?- prove(s12 = [hia(s8,s11,c->d), [a,c->d] => [a&b]]).
yes
| ?- prove(s13 = [th(s12,c), [a,c->d] => [c,a&b]]).
yes
| ?- prove(s14 = [aia(s13,a&(c->d)), [a&(c->d)] => [c,a&b]]).
yes
| ?- prove(s15 = [oia(s7,s14,(a&b)#(a&(c->d))),
                [(a&b)#(a&(c->d))] => [c,a&b]]).
yes
| ?- prove(s16 = [th(s10,a), [a,d] => [b]]).
yes
| ?- prove(s17 = [th(s16,b), [a,b,d] => [b]]).
yes
| ?- prove(s18 = [aia(s17,a&b), [a&b,d] => [b]]).
yes
| ?- prove(s19 = [his(s18,d->b), [a&b] => [d->b]]).
yes
| ?- prove(s20 = [th(s8,b), [a,b] => [c]]).
yes
| ?- prove(s21 = [aia(s20,a&b), [a&b] => [c]]).
yes
| ?- prove(s22 = [ais(s19,s21,(d->b)&c), [a&b] => [(d->b)&c]]).
yes
| ?- prove(s23 = [th(s22,d), [a&b,d] => [(d->b)&c]]).
yes
| ?- prove(s24 = [th(s23,e), [a&b,d] => [(d->b)&c,e]]).
yes
| ?- prove(s = [cut(s15,s24,a&b), [(a&b)#(a&(c->d)),d] =>
                [c,(d->b)&c,e]]).
yes
| ?- normal(s).

```

Proof is as follows :

```

s1=[id(a),[a]=>[a]]
s2=[th(s1,b),[a,b]=>[a]]
s3=[id(b),[b]=>[b]]
s4=[th(s3,a),[a,b]=>[b]]
s5=[ais(s2,s4,a&b),[a,b]=>[a&b]]
s6=[aia(s5,a&b),[a&b]=>[a&b]]
s7=[th(s6,c),[a&b]=>[c,a&b]]
f6=[f,[a]=>[c]]
s9=[g,[d]=>[a]]
f4=[h,[d]=>[b]]
s11=[ais(s9,f4,a&b),[d]=>[a&b]]
s12=[hia(f6,s11,c->d),[a,c->d]=>[a&b]]
s13=[th(s12,c),[a,c->d]=>[c,a&b]]
s14=[aia(s13,a&(c->d)),[a&(c->d)]=>[c,a&b]]
s15=[oia(s7,s14,(a&b)#a&(c->d)),[(a&b)#a&(c->d)]=>[c,a&b]]
f5=[h,[d]=>[b]]
s16=[th(f5,a),[a,d]=>[b]]
s17=[th(s16,b),[a,b,d]=>[b]]
s18=[aia(s17,a&b),[a&b,d]=>[b]]
s19=[his(s18,d->b),[a&b]=>[d->b]]
f7=[f,[a]=>[c]]
s20=[th(f7,b),[a,b]=>[c]]
s21=[aia(s20,a&b),[a&b]=>[c]]
s22=[ais(s19,s21,(d->b)&c),[a&b]=>[(d->b)&c]]
s23=[th(s22,d),[a&b,d]=>[(d->b)&c]]
s24=[th(s23,e),[a&b,d]=>[(d->b)&c,e]]
s=[cut(s15,s24,a&b),[(a&b)#a&(c->d),d]=>[c,(d->b)&c,e]]

```

Cut\_free proof is :

```

f58=[h,[d]=>[b]]
f57=[th(f58,a),[a,d]=>[b]]
f54=[th(f57,b),[a,b,d]=>[b]]
f51=[th(f54,b),[a,b,b,d]=>[b]]
f39=[th(f51,a),[a,b,a,b,d]=>[b]]
f42=[inc(f39,a,b),[a,c,b,b,d]=>[b]]
f43=[con(f42,a),[a,b,b,d]=>[b]]
f34=[con(f43,b),[a,b,d]=>[b]]
f31=[aia(f34,a&b),[a&b,d]=>[b]]
f28=[his(f31,d->b),[a&b]=>[d->b]]
f22=[th(f28,c),[a&b]=>[c,d->b]]
f71=[f,[a]=>[c]]
f83=[h,[d]=>[b]]
f82=[th(f83,d),[d,d]=>[b]]

```

```

f76= th(f82, d), [d, d, d] => [b]]
f70= con(f76, d), [d, d] => [b]]
f67= hia(f71, f70, c->d), [a, c->d, d] => [b]]
f64= his(f67, d->b), [a, c->d] => [d->b]]
f61= th(f64, c), [a, c->d] => [c, d->b]]
f25= aia(f61, a&(c->d)), [a&(c->d)] => [c, d->b]]
f16= oia(f22, f25, (a&b)#a&(c->d)), [(a&b)#a&(c->d)] => [c, d->b]]
f118= f, [a] => [c]]
f115= th(f118, b), [a, b] => [c]]
f112= th(f115, b), [a, b, b] => [c]]
f100= th(f112, a), [a, b, a, b] => [c]]
f103= inc(f100, a, b), [a, a, b, b] => [c]]
f104= con(f103, a), [a, b, b] => [c]]
f95= con(f104, b), [a, b] => [c]]
f92= aia(f95, a&b), [a&b] => [c]]
f86= th(f92, c), [a&b] => [c, c]]
f128= f, [a] => [c]]
f139= i, [d] => [c]]
f133= th(f139, d), [d, d] => [c]]
f127= con(f133, d), [d] => [c]]
f124= hia(f128, f127, c->d), [a, c->d] => [c]]
f121= th(f124, c), [a, c->d] => [c, c]]
f89= oia(f121, a&(c->d)), [a&(c->d)] => [c, c]]
f19= oia(f86, f89, (a&b)#a&(c->d)), [(a&b)#a&(c->d)] => [c, c]]
f13= ais(f16, f19, (d->b)&c), [(a&b)#a&(c->d)] => [c, (d->b)&c]]
f10= th(f13, d), [(a&b)#a&(c->d), d] => [c, (d->b)&c]]
s= [th(f10, e), [(a&b)#a&(c->d), d] => [c, (d->b)&c, e]]

```

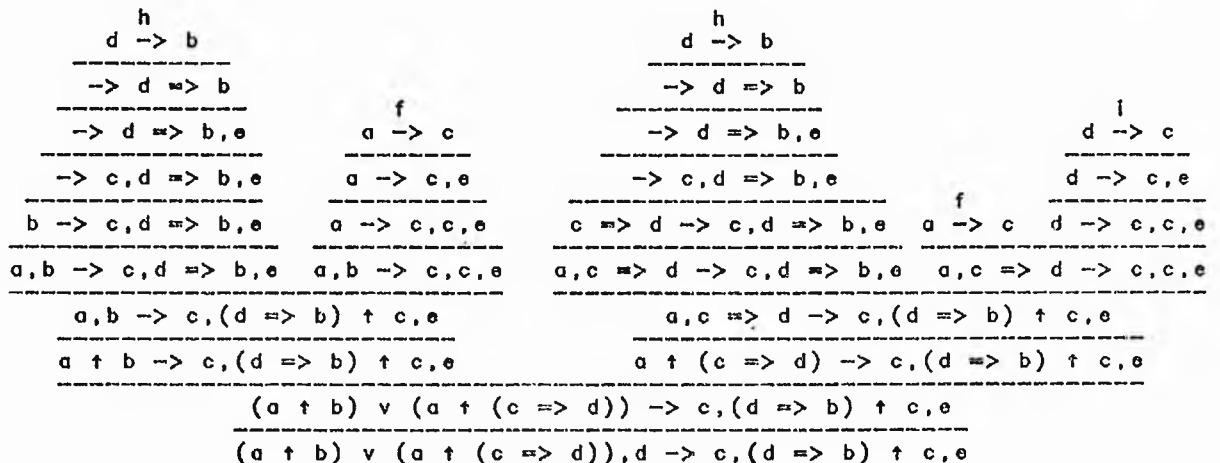
Normal proof is :

```

f58= [h, [d] => [b]]
f57= his(f58, d->b), [ ] => [d->b]]
f34= th(f57, e), [ ] => [d->b, e]]
f31= th(f34, c), [ ] => [c, d->b, e]]
f28= th(f31, b), [b] => [c, d->b, e]]
f22= th(f28, a), [a, b] => [c, d->b, e]]
f118= f, [a] => [c]]
f95= th(f118, e), [a] => [c, e]]
f92= th(f95, c), [a] => [c, c, e]]
f142= th(f92, b), [a, b] => [c, c, e]]
f16= aia(f22, f142, (d->b)&c), [a, b] => [c, (d->b)&c, e]]
f13= oia(f16, a&b), [a&b] => [c, (d->b)&c, e]]
f83= h, [d] => [b]]
f70= his(f83, d->b), [ ] => [d->b]]
f67= th(f70, e), [ ] => [d->b, e]]
f71= th(f67, c), [ ] => [c, d->b, e]]
f61= th(f71, c->d), [c->d] => [c, d->b, e]]
f25= th(f61, a), [a, c->d] => [c, d->b, e]]
f128= f, [a] => [c]]
f127= i, [d] => [c]]
f124= th(f127, e), [d] => [c, e]]
f121= th(f124, c), [d] => [c, c, e]]
f141= hia(f128, f121, c->d), [a, c->d] => [c, c, e]]
f19= ais(f25, f141, (d->b)&c), [a, c->d] => [c, (d->b)&c, e]]
f140= aia(f19, a&(c->d)), [a&(c->d)] => [c, (d->b)&c, e]]
f10= oia(f13, f140, (a&b)#a&(c->d)), [(a&b)#a&(c->d)] => [c, (d->b)&c, e]]
s= [th(f10, d), [(a&b)#a&(c->d), d] => [c, (d->b)&c, e]]

```

which in tree form is







Cut\_free proof is :

=====

```

f1=[f,[a]=>[b]]
s2=[th(f1,c),[a,c]=>[b]]
s3=[th(s2,b),[a,b,c]=>[b]]
s4=[id(b),[b]=>[b]]
s5=[th(s4,c),[c,b]=>[b]]
s6=[th(s5,c),[c,b,c]=>[b]]
s7=[oia(s3,s6,a#c),[a#c,b,c]=>[b]]
f2=[f,[a]=>[b]]
s8=[th(f2,a),[a,a]=>[b]]
s9=[g,[c]=>[b]]
s10=[th(s9,a),[c,a]=>[b]]
s11=[oia(s8,s10,a#c),[a#c,a]=>[b]]
s12=[th(s11,d),[a#c,d,a]=>[b]]
s13=[th(s12,e),[a#c,e,d,a]=>[b]]
s14=[hia(s7,s13,b->c),[a#c,a#c,b,c,b->e,d,a]=>[b]]
s=[con(s14,a#c),[a#c,b,c,b->e,d,a]=>[b]]

```

Normal proof is :

=====

```

f2=[f,[a]=>[b]]
s9=[g,[c]=>[b]]
s8=[oia(f2,s9,a#c),[a#c]=>[b]]
s11=[th(s8,a),[a#c,a]=>[b]]
s12=[th(s11,d),[a#c,d,a]=>[b]]
f4=[th(s12,b->e),[a#c,b->e,d,a]=>[b]]
s14=[th(f4,c),[a#c,c,b->e,d,a]=>[b]]
s=[th(s14,b),[a#c,b,c,b->e,d,a]=>[b]]

```

yes  
| ?-

which in tree form is

$$\begin{array}{c}
 \begin{array}{cc}
 f & g \\
 a \rightarrow b & c \rightarrow b
 \end{array} \\
 \hline
 a \# c \rightarrow b \\
 \hline
 a \vee c, a \rightarrow b \\
 \hline
 a \vee c, d, a \rightarrow b \\
 \hline
 a \vee c, b \Rightarrow e, d, a \rightarrow b \\
 \hline
 a \vee c, c, b \Rightarrow e, d, a \rightarrow b \\
 \hline
 a \vee c, b, c, b \Rightarrow e, d, a \rightarrow b
 \end{array}$$

#### (5.5.10) The CHURCH-ROSSER Theorem for Dbcl(X)

If  $p \approx q$ , then there exists a normal  $r$  in Dbcl(X) such that  $p \succ r$  and  $q \succ r$ .

#### (5.6) Application To Category Theory

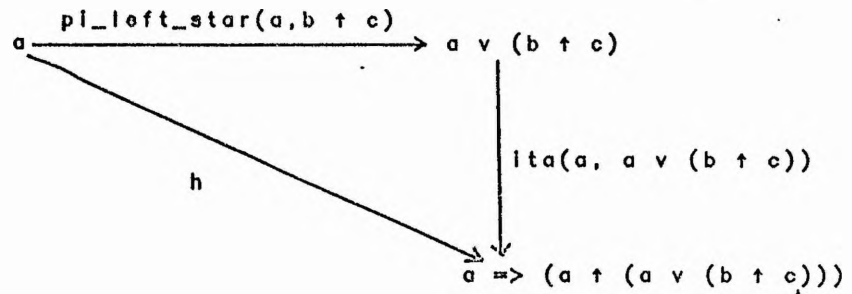
In this section, we use the system to produce the normal representations for some arrows of Fbcl(X) and test the commutativity of some diagrams in Fbcl(X).

##### Example 1

For objects  $a, b, c$  of the category  $X$ , the normal representation of the arrow

$$h = \text{comp}(\text{ita}(a, a \vee (b \dagger c)), \text{pi\_left\_star}(a, b \dagger c))$$

In the diagram



is obtained as follows :

```
yes
| ?- theory(bicartcl).
```

```
yes
| ?- norm_rep(comp(ita(a, a#(b&c)), pi_left_star(a, b&c))).
```

The arrow is :

=====

```
comp(ita(a, a#b&c), pi_left_star(a, b&c))
```

Proof is as follows :

=====

```
f40=[id(a), [a]=>[a]]
f39=[th(f40, b&c), [a]=>[a, b&c]]
f38=[ois(f39, a#b&c), [a]=>[a#b&c]]
f26=[id(a), [a]=>[a]]
f24=[th(f26, a#b&c), [a, a#b&c]=>[a]]
f31=[id(a), [a]=>[a]]
f29=[th(f31, b&c), [a]=>[a, b&c]]
f36=[id(b), [b]=>[b]]
f34=[th(f36, c), [b, c]=>[b]]
f37=[id(c), [c]=>[c]]
f35=[th(f37, b), [b, c]=>[c]]
f33=[ais(f34, f35, b&c), [b, c]=>[b&c]]
f32=[ala(f33, b&c), [b&c]=>[b&c]]
f30=[th(f32, a), [b&c]=>[a, b&c]]
f28=[ola(f29, f30, a#b&c), [a#b&c]=>[a, b&c]]
f27=[ois(f28, a#b&c), [a#b&c]=>[a#b&c]]
f25=[th(f27, a), [a, a#b&c]=>[a#b&c]]
f23=[ais(f24, f25, a&a#b&c), [a, a#b&c]=>[a&a#b&c]]
f22=[his(f23, a->a&a#b&c), [a#b&c]=>[a->a&a#b&c]]
f21=[cut(f38, f22, a#b&c), [a]=>[a->a&a#b&c]]
```

Cut\_free proof is :

=====

```
f68=[id(a), [a]=>[a]]
f67=[th(f68, a), [a]=>[a, a]]
f64=[th(f67, a), [a, a]=>[a, a]]
f56=[th(f64, a), [a, a, a]=>[a, a]]
f57=[con(f56, a), [a, a, a]=>[a]]
f46=[con(f57, a), [a, a]=>[a]]
f90=[id(a), [a]=>[a]]
f122=[th(f90, b&c), [a]=>[a, b&c]]
f125=[th(f122, a), [a]=>[a, a, b&c]]
f126=[th(f125, b), [a]=>[a, a, b, b&c]]
f123=[th(f126, b&c), [a]=>[a, a, b, b&c, b&c]]
f124=[th(f123, a), [a, a]=>[a, a, b, b&c, b&c]]
f90=[id(a), [a]=>[a]]
f128=[th(f90, b&c), [a]=>[a, b&c]]
f131=[th(f128, a), [a]=>[a, a, b&c]]
f132=[th(f131, c), [a]=>[a, a, c, b&c]]
f129=[th(f132, b&c), [a]=>[a, a, c, b&c, b&c]]
f130=[th(f129, a), [a, a]=>[a, a, c, b&c, b&c]]
f100=[ais(f124, f130, b&c), [a, a]=>[a, a, b&c, b&c, b&c]]
f101=[con(f100, b&c), [a, a]=>[a, a, b&c, b&c]]
f102=[con(f101, a), [a, a]=>[a, b&c, b&c]]
f93=[con(f102, a), [a]=>[a, b&c, b&c]]
f79=[th(f93, a), [a]=>[a, b&c, a, b&c]]
f82=[inc(f79, a, b&c), [a]=>[a, a, b&c, b&c]]
```

```
f83=[con(f82,a),[a]=>[a,b&c,b&c]]
f80=[con(f83,b&c),[a]=>[a,b&c]]
f71=[ois(f80,a#b&c),[a]=>[a#b&c]]
f49=[th(f71,a),[a,a]=>[a#b&c]]
f43=[ois(f46,f49,a&a#b&c),[a,a]=>[a&a#b&c]]
f21=[his(f43,a->a&a#b&c),[a]=>[a->a&a#b&c]]
```

Normal proof is :

```
f64=[id(a),[a]=>[a]]
f90=[id(a),[a]=>[a]]
f134=[th(f90,b),[a]=>[a,b]]
f136=[id(a),[a]=>[a]]
f135=[th(f136,c),[a]=>[a,c]]
f80=[ois(f134,f135,b&c),[a]=>[a,b&c]]
f71=[ois(f80,a#b&c),[a]=>[a#b&c]]
f46=[ois(f64,f71,a&a#b&c),[a]=>[a&a#b&c]]
f43=[his(f46,a->a&a#b&c),[a]=>[a->a&a#b&c]]
f21=[th(f43,a),[a]=>[a->a&a#b&c]]
```

yes  
| ?-

which in tree form is

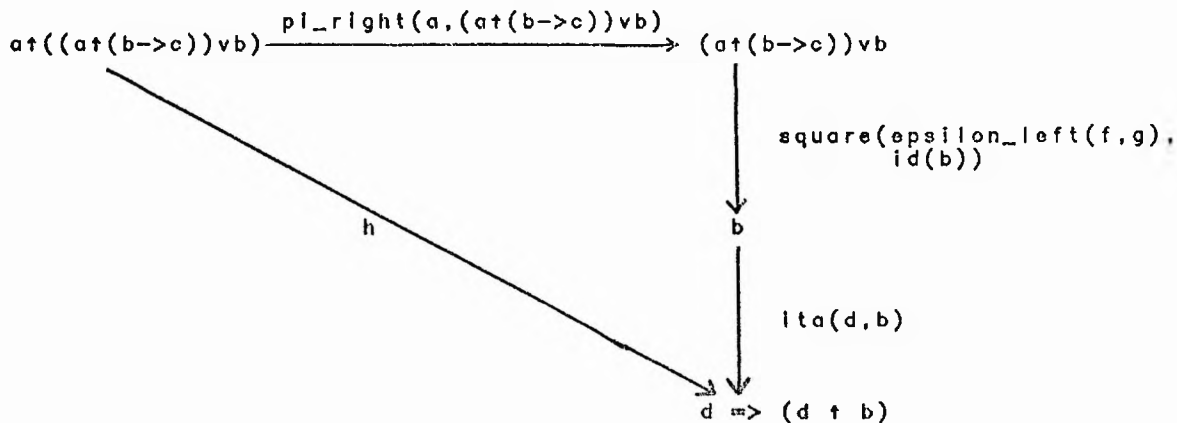
$$\begin{array}{c}
 \frac{a \rightarrow a \quad a \rightarrow a}{a \rightarrow a, b} \quad \frac{a \rightarrow a \quad a \rightarrow a}{a \rightarrow a, c} \\
 \hline
 a \rightarrow a, b \uparrow c \\
 \hline
 a \rightarrow a \quad a \rightarrow a \vee (b \uparrow c) \\
 \hline
 d \rightarrow a \uparrow (a \vee (b \uparrow c)) \\
 \hline
 \rightarrow a \Rightarrow (a \uparrow (a \vee (b \uparrow c))) \\
 \hline
 a \rightarrow a \Rightarrow (a \uparrow (a \vee (b \uparrow c)))
 \end{array}$$

Example 2

For arrows  $f$  in  $X(a,b)$  and  $g$  in  $X(c,b)$  for some category  $X$ , the normal representation of the arrow

$$h = \text{comp}(\text{comp}(\text{ita}(d,b), \text{square}(\text{epsilon\_left}(f,g), \text{id}(b))), \text{pi\_right}(a, (a \uparrow (b \rightarrow c)) \vee b))$$

In the diagram



is obtained as follows :

```

yes
| ?- theory(bicartcl).

yes
| ?- arrow(a, f, b).

yes
| ?- arrow(c, g, b).

yes
| ?- norm_rep(comp(comp(ita(d, b), square(epsilon_left(f, g),
|                                     id(b))),
|                                     pi_right(a, (a&(b->c))#b))).

```

The arrow is :

```

=====
comp(comp(ita(d, b), square(epsilon_left(f, g), id(b))),
      pi_right(a, (a&(b->c))#b))

```

Proof is as follows :

```

=====
f26= id(a), [a] => [a]
f24= th(f26, b->c), [a, b->c] => [a]
f29= id(b), [b] => [b]
f30= id(c), [c] => [c]
f28= h1a(f29, f30, b->c), [b, b->c] => [c]
f27= h1s(f28, b->c), [b->c] => [b->c]
f25= th(f27, a), [a, b->c] => [b->c]
f23= als(f24, f25, a&(b->c)), [a, b->c] => [a&(b->c)]
f21= a1a(f23, a&(b->c)), [a&(b->c)] => [a&(b->c)]
f19= th(f21, b), [a&(b->c)] => [a&(b->c), b]
f22= id(b), [b] => [b]
f20= th(f22, a&(b->c)), [b] => [a&(b->c), b]
f18= o1a(f19, f20, (a&(b->c))#b), [(a&(b->c))#b] => [a&(b->c), b]
f17= o1s(f18, (a&(b->c))#b), [(a&(b->c))#b] => [(a&(b->c))#b]
f16= th(f17, a), [a, (a&(b->c))#b] => [(a&(b->c))#b]
f15= a1a(f16, a&(a&(b->c))#b), [a&(a&(b->c))#b] => [(a&(b->c))#b]
f11= f, [a] => [b]
f12= g, [c] => [b]
f13= h1a(f11, f12, b->c), [a, b->c] => [b]
f10= a1a(f13, a&(b->c)), [a&(b->c)] => [b]
f14= id(b), [b] => [b]
f9= o1a(f10, f14, (a&(b->c))#b), [(a&(b->c))#b] => [b]
f7= id(d), [d] => [d]
f5= th(f7, b), [d, b] => [d]
f8= id(b), [b] => [b]
f6= th(f8, d), [d, b] => [b]
f4= als(f5, f6, d&b), [d, b] => [d&b]
f3= h1s(f4, d->d&b), [b] => [d->d&b]
f2= cut(f9, f3, b), [(a&(b->c))#b] => [d->d&b]
f1= cut(f15, f2, (a&(b->c))#b), [a&(a&(b->c))#b] => [d->d&b]

```

Cut\_free proof is :

```

=====
f141= f, [a] => [b]
f137= th(f141, b->c), [a, b->c] => [b]
f131= id(d), [d] => [d]
f138= th(f131, c), [d, c] => [d]
f134= h1a(f137, f138, b->c), [d, a, b->c, b->c] => [d]
f109= th(f134, a), [d, a, b->c, a, b->c] => [d]
f112= inc(f109, a, b->c), [d, a, a, b->c, b->c] => [d]
f113= con(f112, a), [d, a, b->c, b->c] => [d]
f104= con(f113, b->c), [d, a, b->c] => [d]
f176= a1a(f104, a&(b->c)), [d, a&(b->c)] => [d]
f177= th(f176, d), [d, a&(b->c)] => [d, d]
f172= th(f177, d), [d, d, a&(b->c)] => [d, d]
f216= id(d), [d] => [d]
f215= th(f216, b), [d, b] => [d]
f212= th(f215, d), [d, b] => [d, d]
f209= th(f212, d), [d, d, b] => [d, d]
f206= th(f209, b), [d, d, b, b] => [d, d]
f203= th(f206, d), [d, d, b, b] => [d, d, d]
f186= th(f203, d), [d, d, d, b, b] => [d, d, d]
f187= con(f186, d), [d, d, d, b, b] => [d, d]

```

```

f181=[con(f187,d),[d,d,b,b]=>[d,d]]
f175=[con(f181,b),[d,d,b]=>[d,d]]
f90=[oia(f172,f175,(a&(b->c))#b),[d,d,(a&(b->c))#b]=>[d,d]]
f91=[con(f90,d),[d,d,(a&(b->c))#b]=>[d]]
f85=[con(f91,d),[d,(a&(b->c))#b]=>[d]]
f82=[th(f85,a),[d,a,(a&(b->c))#b]=>[d]]
f76=[oia(f82,a&(a&(b->c))#b),[d,a&(a&(b->c))#b]=>[d]]
f280=[f,[a]=>[b]]
f276=[th(f280,b->c),[a,b->c]=>[b]]
f277=[g,[c]=>[b]]
f273=[hia(f276,f277,b->c),[a,b->c,b->c]=>[b]]
f249=[th(f273,a),[a,b->c,a,b->c]=>[b]]
f252=[inc(f249,a,b->c),[a,a,b->c,b->c]=>[b]]
f253=[con(f252,a),[a,b->c,b->c]=>[b]]
f244=[con(f253,b->c),[a,b->c]=>[b]]
f310=[oia(f244,a&(b->c)),[a&(b->c)]=>[b]]
f306=[th(f310,b),[a&(b->c)]=>[b,b]]
f301=[id(b),[b]=>[b]]
f302=[th(f301,b),[b]=>[b,b]]
f317=[th(f302,b),[b,b]=>[b,b]]
f316=[th(f317,b),[b,b]=>[b,b,b]]
f313=[con(f316,b),[b,b]=>[b,b]]
f309=[con(f313,b),[b]=>[b,b]]
f230=[oia(f306,f309,(a&(b->c))#b),[(a&(b->c))#b]=>[b,b]]
f231=[con(f230,b),[(a&(b->c))#b]=>[b]]
f222=[th(f231,a),[a,(a&(b->c))#b]=>[b]]
f219=[oia(f222,a&(a&(b->c))#b),[a&(a&(b->c))#b]=>[b]]
f79=[th(f219,d),[d,a&(a&(b->c))#b]=>[b]]
f73=[ois(f76,f79,d&b),[d,a&(a&(b->c))#b]=>[d&b]]
f1=[his(f73,d->d&b),[a&(a&(b->c))#b]=>[d->d&b]]
    
```

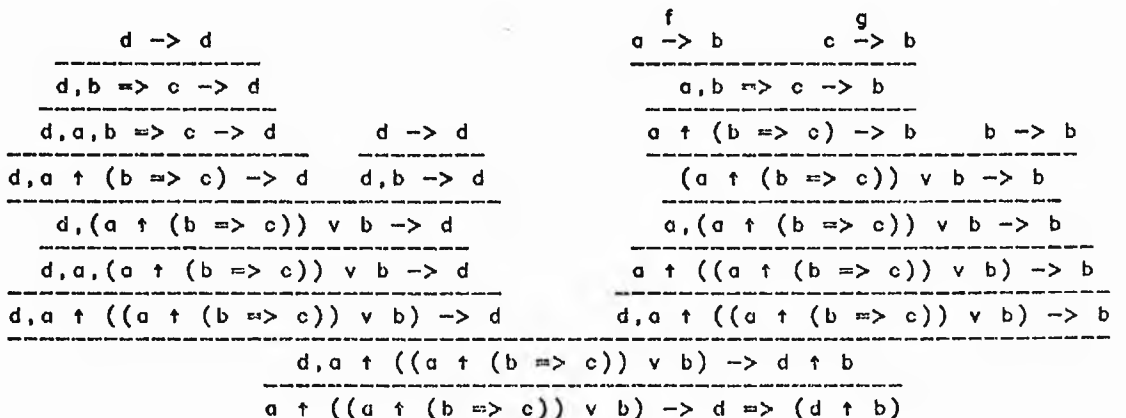
Normal proof is :

```

f131=[id(d),[d]=>[d]]
f141=[th(f131,b->c),[d,b->c]=>[d]]
f104=[th(f141,a),[d,a,b->c]=>[d]]
f91=[oia(f104,a&(b->c)),[d,a&(b->c)]=>[d]]
f216=[id(d),[d]=>[d]]
f319=[th(f216,b),[d,b]=>[d]]
f85=[oia(f91,f319,(a&(b->c))#b),[d,(a&(b->c))#b]=>[d]]
f82=[th(f85,a),[d,a,(a&(b->c))#b]=>[d]]
f76=[oia(f82,a&(a&(b->c))#b),[d,a&(a&(b->c))#b]=>[d]]
f280=[f,[a]=>[b]]
f277=[g,[c]=>[b]]
f244=[hia(f280,f277,b->c),[a,b->c]=>[b]]
f230=[oia(f244,a&(b->c)),[a&(b->c)]=>[b]]
f320=[id(b),[b]=>[b]]
f231=[oia(f230,f320,(a&(b->c))#b),[(a&(b->c))#b]=>[b]]
f222=[th(f231,a),[a,(a&(b->c))#b]=>[b]]
f219=[oia(f222,a&(a&(b->c))#b),[a&(a&(b->c))#b]=>[b]]
f79=[th(f219,d),[d,a&(a&(b->c))#b]=>[b]]
f73=[ois(f76,f79,d&b),[d,a&(a&(b->c))#b]=>[d&b]]
f1=[his(f73,d->d&b),[a&(a&(b->c))#b]=>[d->d&b]]
    
```

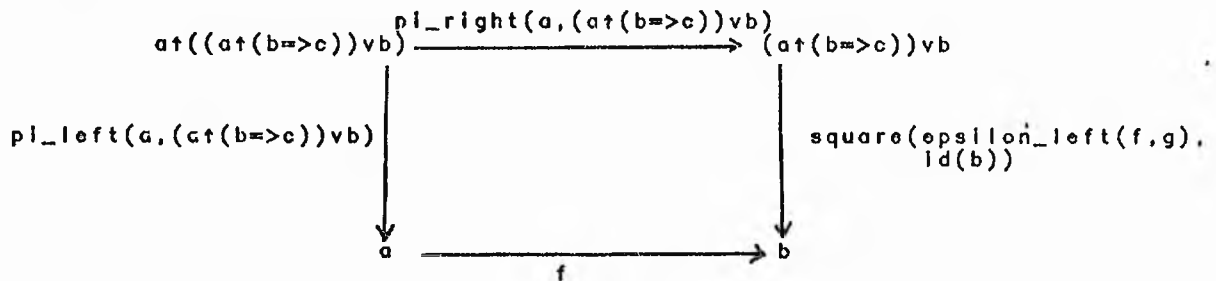
yes  
| ?-

which in tree form is



### Example 3

For  $f$  in  $X(a,b)$  and  $g$  in  $X(c,b)$  for some category  $X$ , the following diagram in  $\text{Fbcat}(X)$



does not commute.

```
yes
| ?- theory(bicartcl).
```

```
yes
| ?- arrow(a,f,b).
```

```
yes
| ?- arrow(c,g,b).
```

```
yes
| ?- commutative.
```

```
Insert the first arrow : comp(square(epsilon_left(f,g),id(b)),
|: pi_right(a,(a&(b->c))#b)).
```

The arrow is :

```
comp(square(epsilon_left(f,g),id(b)),pi_right(a,(a&(b->c))#b))
```

Proof is as follows :

```
f19= [id(a), [a]=>[a]]
f17= [th(f19, b->c), [a, b->c]=>[a]]
f22= [id(b), [b]=>[b]]
f23= [id(c), [c]=>[c]]
f21= [hia(f22, f23, b->c), [b, b->c]=>[c]]
f20= [his(f21, b->c), [b->c]=>[b->c]]
f18= [th(f20, a), [a, b->c]=>[b->c]]
f16= [ais(f17, f18, a&(b->c)), [a, b->c]=>[a&(b->c)]]
f14= [aia(f16, a&(b->c)), [a&(b->c)]=>[a&(b->c)]]
f12= [th(f14, b), [a&(b->c)]=>[a&(b->c), b]]
f15= [id(b), [b]=>[b]]
f13= [th(f15, a&(b->c)), [b]=>[a&(b->c), b]]
f11= [oia(f12, f13, (a&(b->c))#b), [(a&(b->c))#b]=>[a&(b->c), b]]
f10= [ois(f11, (a&(b->c))#b), [(a&(b->c))#b]=>[(a&(b->c))#b]]
f9= [th(f10, a), [a, (a&(b->c))#b]=>[(a&(b->c))#b]]
f8= [aia(f9, a&(a&(b->c))#b), [a&(a&(b->c))#b]=>[(a&(b->c))#b]]
f4= [f, [a]=>[b]]
f5= [g, [c]=>[b]]
f6= [hia(f4, f5, b->c), [a, b->c]=>[b]]
f3= [aia(f6, a&(b->c)), [a&(b->c)]=>[b]]
f7= [id(b), [b]=>[b]]
f2= [oia(f3, f7, (a&(b->c))#b), [(a&(b->c))#b]=>[b]]
f1= [cut(f8, f2, (a&(b->c))#b), [a&(a&(b->c))#b]=>[b]]
```

Cut\_free proof is :

```
f84= [f, [a]=>[b]]
f80= [th(f84, b->c), [a, b->c]=>[b]]
f81= [g, [c]=>[b]]
f77= [hia(f80, f81, b->c), [a, b->c, b->c]=>[b]]
```

```

f53=[th(f77,a),[a,b->c,a,b->c]=>[b]]
f56=[inc(f53,a,b->c),[a,a,b->c,b->c]=>[b]]
f57=[con(f56,a),[a,b->c,b->c]=>[b]]
f48=[con(f57,b->c),[a,b->c]=>[b]]
f114=[aia(f48,a&(b->c)),[a&(b->c)]=>[b]]
f110=[th(f114,b),[a&(b->c)]=>[b,b]]
f105=[id(b),[b]=>[b]]
f106=[th(f105,b),[b]=>[b,b]]
f121=[th(f106,b),[b,b]=>[b,b]]
f120=[th(f121,b),[b,b]=>[b,b,b]]
f117=[con(f120,b),[b,b]=>[b,b]]
f113=[con(f117,b),[b]=>[b,b]]
f34=[oia(f110,f113,(a&(b->c))#b),[(a&(b->c))#b]=>[b,b]]
f35=[con(f34,b),[(a&(b->c))#b]=>[b]]
f26=[th(f35,a),[a,(a&(b->c))#b]=>[b]]
f1=[aia(f26,a&(a&(b->c))#b),[a&(a&(b->c))#b]=>[b]]

```

Normal proof is :

```

f84=[f,[a]=>[b]]
f81=[g,[c]=>[b]]
f48=[hia(f84,f81,b->c),[a,b->c]=>[b]]
f34=[aia(f48,a&(b->c)),[a&(b->c)]=>[b]]
f122=[id(b),[b]=>[b]]
f35=[oia(f34,f122,(a&(b->c))#b),[(a&(b->c))#b]=>[b]]
f26=[th(f35,a),[a,(a&(b->c))#b]=>[b]]
f1=[aia(f26,a&(a&(b->c))#b),[a&(a&(b->c))#b]=>[b]]

```

Insert the second arrow : comp(f,pi\_left(a,(a&(b->c))#b)).

The arrow is :

```
comp(f,pi_left(a,(a&(b->c))#b))
```

Proof is as follows :

```

f127=[id(a),[a]=>[a]]
f126=[th(f127,(a&(b->c))#b),[a,(a&(b->c))#b]=>[a]]
f125=[aia(f126,a&(a&(b->c))#b),[a&(a&(b->c))#b]=>[a]]
f124=[f,[a]=>[b]]
f123=[cut(f125,f124,a),[a&(a&(b->c))#b]=>[b]]

```

Cut\_free proof is :

```

f133=[f,[a]=>[b]]
f130=[th(f133,(a&(b->c))#b),[a,(a&(b->c))#b]=>[b]]
f123=[aia(f130,a&(a&(b->c))#b),[a&(a&(b->c))#b]=>[b]]

```

Normal proof is :

```

f133=[f,[a]=>[b]]
f138=[th(f133,b->c),[a,b->c]=>[b]]
f137=[aia(f138,a&(b->c)),[a&(b->c)]=>[b]]
f134=[th(f137,a),[a,a&(b->c)]=>[b]]
f136=[f,[a]=>[b]]
f135=[th(f136,b),[a,b]=>[b]]
f130=[oia(f134,f135,(a&(b->c))#b),[a,(a&(b->c))#b]=>[b]]
f123=[aia(f130,a&(a&(b->c))#b),[a&(a&(b->c))#b]=>[b]]

```

The answer is :

```

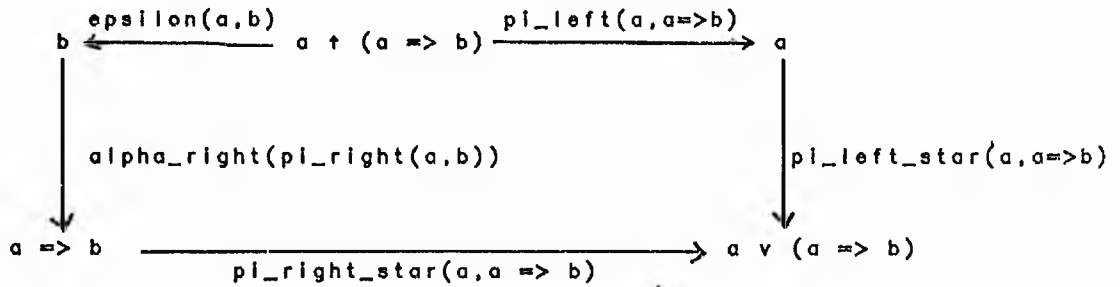
comp(square(epsilon_left(f,g),id(b)),pi_right(a,(a&(b->c))#b))
is not equal to
comp(f,pi_left(a,(a&(b->c))#b))

```

yes  
| ?-

Example 4

For  $a, b$  in  $Ob(X)$  for some category  $X$ , the following diagram in  $Fbcat(X)$



does not commute.

```
yes
| ?- theory(bicartcl).
```

```
yes
| ?- commutative.
```

```
Insert the first arrow : comp(pi_left_star(a, a->b),
                             pi_left(a, a->b)).
```

The arrow is :

```
comp(pi_left_star(a, a->b), pi_left(a, a->b))
```

Proof is as follows :

```
f7=[id(a), [a]=>[a]]
f6=[th(f7, a->b), [a, a->b]=>[a]]
f5=[aia(f6, a&(a->b)), [a&(a->b)]=>[a]]
f4=[id(a), [a]=>[a]]
f3=[th(f4, a->b), [a]=>[a, a->b]]
f2=[ois(f3, a#(a->b)), [a]=>[a#(a->b)]]
f1=[cut(f5, f2, a), [a&(a->b)]=>[a#(a->b)]]
```

Cut\_free proof is :

```
f19=[id(a), [a]=>[a]]
f16=[th(f19, a->b), [a, a->b]=>[a]]
f13=[aia(f16, a&(a->b)), [a&(a->b)]=>[a]]
f10=[th(f13, a->b), [a, a->b]=>[a, a->b]]
f1=[ois(f10, a#(a->b)), [a&(a->b)]=>[a#(a->b)]]
```

Normal proof is :

```
f19=[id(a), [a]=>[a]]
f16=[th(f19, a->b), [a]=>[a, a->b]]
f13=[ois(f16, a#(a->b)), [a]=>[a#(a->b)]]
f10=[th(f13, a->b), [a, a->b]=>[a#(a->b)]]
f1=[aia(f10, a&(a->b)), [a&(a->b)]=>[a#(a->b)]]
```

```
Insert the second arrow : comp(comp(pi_right_star(a, a->b),
|:, alpha_right(pi_right(a,b))),
|:, epsilon(a,b)).
```

The arrow is :

```
comp(comp(pi_right_star(a, a->b), alpha_right(pi_right(a,b))),
      epsilon(a,b))
```



Proof is as follows :

```

f40= [id(a), [a] => [a]]
f41= [id(b), [b] => [b]]
f39= [hia(f40, f41, a->b), [a, a->b] => [b]]
f38= [aia(f39, a&(a->b)), [a&(a->b)] => [b]]
f36= [id(a), [a] => [a]]
f34= [th(f36, b), [a, b] => [a]]
f37= [id(b), [b] => [b]]
f35= [th(f37, a), [a, b] => [b]]
f33= [ais(f34, f35, a&b), [a, b] => [a&b]]
f31= [id(b), [b] => [b]]
f30= [th(f31, a), [a, b] => [b]]
f29= [aia(f30, a&b), [a&b] => [b]]
f32= [cut(f33, f29, a&b), [a, b] => [b]]
f28= [his(f32, a->b), [b] => [a->b]]
f26= [id(a), [a] => [a]]
f27= [id(b), [b] => [b]]
f25= [hia(f26, f27, a->b), [a, a->b] => [b]]
f24= [his(f25, a->b), [a->b] => [a->b]]
f23= [th(f24, a), [a->b] => [a, a->b]]
f22= [ois(f23, a#(a->b)), [a->b] => [a#(a->b)]]
f21= [cut(f28, f22, a->b), [b] => [a#(a->b)]]
f20= [cut(f38, f21, b), [a&(a->b)] => [a#(a->b)]]

```

Cut\_free proof is :

```

f168= [id(a), [a] => [a]]
f152= [id(a), [a] => [a]]
f151= [id(b), [b] => [b]]
f148= [hia(f152, f151, a->b), [a, a->b] => [b]]
f145= [aia(f148, a&(a->b)), [a&(a->b)] => [b]]
f169= [th(f145, a), [a, a&(a->b)] => [b]]
f167= [th(f169, b), [a, b, a&(a->b)] => [b]]
f164= [hia(f168, f167, a->b), [a, a, a->b, a&(a->b)] => [b]]
f161= [aia(f164, a&(a->b)), [a, a&(a->b), a&(a->b)] => [b]]
f158= [th(f161, a), [a, a&(a->b), a, a&(a->b)] => [b]]
f155= [inc(f158, a, a&(a->b)), [a, a, a&(a->b), a&(a->b)] => [b]]
f128= [con(f155, a), [a, a&(a->b), a&(a->b)] => [b]]
f123= [con(f128, a&(a->b)), [a, a&(a->b)] => [b]]
f120= [his(f123, a->b), [a&(a->b)] => [a->b]]
f117= [th(f120, a), [a&(a->b)] => [a, a->b]]
f20= [ois(f117, a#(a->b)), [a&(a->b)] => [a#(a->b)]]

```

Normal proof is :

```

f152= [id(a), [a] => [a]]
f151= [id(b), [b] => [b]]
f148= [hia(f152, f151, a->b), [a, a->b] => [b]]
f128= [aia(f148, a&(a->b)), [a&(a->b)] => [b]]
f123= [th(f128, a), [a, a&(a->b)] => [b]]
f120= [his(f123, a->b), [a&(a->b)] => [a->b]]
f117= [th(f120, a), [a&(a->b)] => [a, a->b]]
f20= [ois(f117, a#(a->b)), [a&(a->b)] => [a#(a->b)]]

```

The answer is :

```

comp(pi_left_star(a, a->b), pi_left(a, a->b))
  is not equal to
comp(comp(pi_right_star(a, a->b), alpha_right(pi_right(a, b))),
  epsilon(a, b))

```

```

yes
| ?-

```

## Chapter 6

=====

### Notes on the Implementation

=====

This chapter describes how the algorithms have been implemented and how the implementation works. The implementation is coded in C-Prolog and is divided into the following main parts, corresponding to the theoretical work described in chapters 1-5 :-

#### (6.1) Proof Checker Procedure

=====

A sequent calculus derivation is represented in our system by a 'step-collection' of proof-steps. The syntax of proof-steps is given in the user manual (appendix 5), where a proof-step is represented by a name followed by a list of information about the axiom or the applied rule, the antecedent and the succedent of the sequent. Because of the nature of the cut-elimination and the normalisation algorithms, this representation of the derivation is chosen in order to simplify the calling, retracting, replacing and manipulating of parts of the stored proof without any logical changes to the other parts of the proof.

The axioms and rules of the unlabelled derivation systems have been implemented in the general case as in chapter 5 with some restrictions, where appropriate, for the specific cases of the deductive systems in chapters 1-5 (e.g. the 'and-introduction-succedent' rule is implemented in its general case

$$\frac{G \rightarrow FaE \quad G \rightarrow FbE}{G \rightarrow F a\&b E}$$

where we restrict the two lists F and E to be empty in the case of the theories of [cartesian, bicartesian, cartesian closed] categories.

The predicate "rule(X)" where X is an axiom or the conclusion of a rule, is used to construct proofs in the deductive systems. (e.g. for the 'and-introduction-succedent' rule X is

$$L = [ais(M,N,A \& B), G \Rightarrow T].$$

where

$$M = [_, G \Rightarrow T1],$$

$$N = [_, G \Rightarrow T2],$$

and

$$T = E(A\&B)F, \quad T1 = EAF, \quad T2 = EBF ).$$

The proof is checked one step at a time from bottom up using the predicate "prove(X)", where X represents a proof-step, then it will take this X and try to satisfy the subgoal "rule(X)" which by turn tries to match one of the implemented axioms and rules. If the step is correct, it will be kept in the database for further use.

The predicate "proof" is used, as another way to check the correctness and assert derivations without using the formal syntax for proof-step. In order to check a proof-tree, call the predicate "proof", the system repeatedly waits to read either the name of the rule ( or axiom) or the word 'end'. If the name is of an axiom or rule, the system asks for the name to be given to the proof-step, the name ( or names ) of the previous step ( or steps ) , if any , and the antecedent and the succedent formulae; then the system uses this information to construct a proof-step. The above predicate "prove(X)", where X is the constructed proof-step, is used to check the correctness of this step. The system repeats this process, until the word 'end', and then returns control to top-level.

## (6.2) Interpretation Procedure

=====

In this section, we discuss the code of the interpretation function S which interprets sequent calculus derivations as arrows of the free [cartesian,.....] categories. All the details in chapters 1-5 concerning the interpretation of the axioms and rules have been implemented using the predicate "interpr(Z,H)", where Z is the name of the last step of a derivation, and H is then instantiated as the interpretation of this derivation.

In order to implement the interpretation function S as defined in chapters 1-5, we must implement all the special cases for each rule ( e.g the thinning rule is of the form

$$\frac{GD \rightarrow F}{G\&D \rightarrow F} \quad :$$

in order to implement the interpretation of this rule, we have the following three possible cases which have three different interpretations:

- i) G is empty.
- ii) D is empty.
- iii) G and D are not empty. )

We can divide the code of the interpretation function S into two parts according to the axioms and rules. In case of axioms A1, A2 and A3, the arrow H providing the interpretation is the arrow in A1,  $id(tr)$  and  $id(bo)$  respectively. In the case of the rules, to interpret the conclusion of the rule, the system goes recursively to compute first the interpretation of the premisses.

In order to interpret a proof as an arrow, we use the predicate "interpr(Z,H)", where Z is the name of the last step in the proof. The system reads the proof-step with name Z, then matches this step with one of the implemented axioms and rules and continues recursively to reach the leaves of the proof. H is then instantiated to the arrow representing Z.

### (6.3) Cut-Elimination Procedure

This section discusses the implementation of the cut elimination algorithm described in Appendix 2. The clauses therein have been implemented in the same order as in that Appendix; using the predicate "cut\_elim(Z = [cut(Z1,Z2,C), G => T])", where Z1 and Z2 depend on the clause being implemented ( e.g. when

$$\begin{aligned} Z1 &= [als(Z3,Z4,A \& B), G1 \Rightarrow T1], \\ Z2 &= [th(Z5,A \& B), G2 \Rightarrow T2], \end{aligned}$$

this is the implementation of the clause (C.3) in Appendix 2 ).

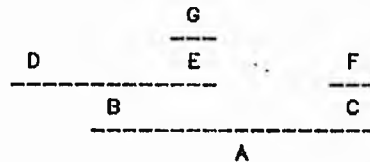
In order to eliminate the cuts from a proof, we can use the predicate "cut\_free(Z)", where Z represents the name of the last proof-step; the system will start by satisfying the subgoal "one\_step" and keep satisfying this goal using the "repeat" technique in Prolog until there are no instances of cuts. In order to satisfy the goal "one\_step", the system looks for an instance of the cut rule; if there is one, it will use the pattern-matching facilities of Prolog to match this instance with one of the "cut\_elim" clauses. Then, satisfying the predicates

"expcon", "expinc" and "expth" which are designed to solve the problem which arises in some clauses ( e.g (C.3), (C.4), (C.20.3), ... ) by producing proof-steps ( with lists of contraction , thinning and interchange formulae ), so these proofs must be expanded in the normal way, where the active formulae must be single and not lists. After satisfying the goals of "expcon", "expinc" and "expth", the system will purge; in other words it will retract all the unwanted proof-steps and put the proof in the right order (By this is meant, ordering the proof-tree in the database using the Traversal Postorder technique defined by:

```

    traverse left subtree in postorder
    traverse right subtree in postorder
    visit root
  
```

e.g. the following proof-tree



is ordered in the database according to the postorder technique as follows :

D G E B F C A

So the system will be able to pick up the nearest instance of the 'cut' rule to the leaves.

The system will repeat the procedure of "one\_step" until there are no more instances of 'cuts' ; the system then prints the proof, using the predicate "printall(Z)", where Z is the name of the last step in the original proof.

#### (6.4) Normalisation Procedure

This section discusses the implementation of the normalisation algorithms in appendices 3 & 4. The normalisation procedure is divided into two procedures

- i) for elimination of contractions & interchanges.
- ii) for ordering the application of rules according to the priority in the definition of the normal form.

#### (6.4.1) Contraction and Interchange-Elimination Procedure

This subsection concerns the clauses of Appendix 3 which are used to eliminate the instances of the contraction and interchange rules in cut-free proofs, for the theories of cartesian, bicartesian and distributive bicartesian categories. As for the theories of cartesian closed and bicartesian closed, there are some clauses to remove all the instances of the contraction and interchange rules with active formulae of the form  $a \uparrow b$  and  $a \vee b$ . That appendix contains some conditions on the thinning formulae. All these clauses have been implemented using the predicate "norm1(A)", where A represents the last step in each clause.

In order to produce a contraction & interchange-free proof, we use the predicate "con\_inc\_free(X)", where X represents the name of the last step in a cut-free proof. The system will first satisfy the subgoal "normal1", which will be satisfied by repeatedly satisfying the goal "do1\_step" until no more reductions can be carried out on the proof. The procedure "do1\_step" is satisfied by satisfying a number of subgoals as follows :

- 1) The predicate "put\_in\_list(X,L1)", where X represents the name of the last proof-step, copies the proof into a list L1.
- 2) The predicate "assert(proved(mark))" is used to assert the clause "proved(mark)" at the end of the proof in the database.
- 3) The predicate "norm1(proved(Z))" is used to match one proof-step at a time using the pattern-matching facilities, with one of the implemented clauses.
- 4) The system will continue the above step for the whole proof, when Z in the predicate "norm1(proved(Z))" is instantiated for 'mark', the system then copies the result proof in a list "L2" using the predicate "put\_in\_list", abolishes all the proof-steps from the database and reasserts "L2" again in the database in the right order.
- 5) Finally, the system compares the two lists L1 ( the proof before the reduction processes ) and L2 ( the proof after the reduction processes ). If they are not the same, the predicate "do1\_step" is repeated many times until the two lists are the same, this means that there are no more reductions. The system then prints the proof, using the predicate "printall".

### (6.4.2) Priority Procedure

In this subsection, we discuss the implementation of the clauses in appendix 4. These are used to complete the normalisation of a proof by ordering the instances of the rules applied, according to the priority in the definition of 'normal form'.

After the procedure of the predicate "normal1" in the above subsection has been carried out, the system will continue to perform the priority procedure by satisfying the predicate "normal2". The code of the procedure "normal2" is similar to the code of the procedure "normal1", by replacing the predicate "do1\_step" in "normal1" by the predicate "do2\_step". The two procedures "do1\_step" and "do2\_step" differs only in one predicate; the first has the predicate "norm1(Z)" ( which implements the clauses of the Appendix 3 ) and the second has the predicate "norm2(Z)" (which implements the clauses in the Appendix 4 with Z representing the conclusion of each clause ). This procedure ends by printing out the normal form proof.

#### Note

We constructed one procedure "normal" as a combination of three procedures altogether: "cut\_free", "normal1" and "normal2", in addition to other little procedures such as "hook\_terminal" and "term\_inti" to characterise the normal proofs for the initial and terminal arrows. This "normal" procedure is used to produce normal form proofs.

### (6.5) Representation Procedure

This section discusses the implementation of the completeness theorems mentioned in chapters 1-5, for the representation of arrows of the free [ -cartesian, ... ] categories by means of sequent calculus proofs. The arrow "Y" is represented by a proof by using the procedure "rep\_of(Y)". To satisfy the goal "rep\_of(Y)", the system generates an atom, using the predicate "gensym", to represent the name of the last step in the result representation. Since each of the arrows in chapters 1 - 5 is written as a term with argument inside ( e.g. the arrow 'comp(f,g)' ), the system appends the generated name to the arguments of the arrow ( e.g. in the above arrow, the result is 'comp(f,g,n)', where n is the name generated by "gensym" ). The system matches this with one of implemented arrows ( e.g. we may find the representation F of pi\_left(A,B) by satisfying the goal pi\_left(A,B,F) ).

Note .

By combining the two procedures "rep\_of(Y)" and "normal", we constructed a procedure "norm\_rep(Y)", which produces the normal representation for an arrow 'Y'.

(6.6) Commutativity Procedure  
 =====

This procedure has been constructed to help deciding the equality between two arrows of a category. In order to do this we use the predicate "commutative": to satisfy the goal "commutative" the system satisfies the following subgoals:

- 1) The system waits to read the first arrow Y.
- 2) The predicate "norm\_rep(Y)" is satisfied by producing the normal representation for the arrow Y.
- 3) The predicate "tree(Z,L1)" is constructed in order to save the normal representation of the first arrow as a data structure in a list L1. This must be done, since the database needs to be cleaned from the representation of the first arrow, in order to insert the second order. As an example, the following proof:

$$\begin{aligned}
 l &= [f, [a] \Rightarrow [b]], \\
 m &= [th(l,c), [a,c] \Rightarrow [b]], \\
 n &= [aia(m,a \& c), [a \& c] \Rightarrow [b]].
 \end{aligned}$$

is saved as a data structure in a list as follows:

$$[f : [a] \Rightarrow [b], th(c) : [a,c] \Rightarrow [b], aia(a\&c) : [a\&c] \Rightarrow [b]].$$

This data structure is ordered in the list from right to left in the same order for the proof in the database, e.g. in the above example the proof-step representing the conclusion of applying thinning comes directly after the proof-step representing the premiss of the thinning rule.

- 4) Then the predicate "clean" cleans all trace of the first proof from the database.
- 5) The predicate "read" causes the system to read the second arrow.
- 6) The same processes, as in the steps 2 - 4 above, for the second



arrow to produce a list L2 with the data structure for the normal representation of the second arrow.

- 7) Finally, the system compares the two lists L1 and L2; if they are the same then the two arrows are equal, and if they are not then the two arrows are not equal.

#### (6.7) Other subprocedures

There are other procedures for manipulating lists such as

- i) `append(L1,L2,L)` : list L1 prepended before list L2 is list L.
- ii) `islist(A)` : to check if A is a list or not.
- iii) `gensym(f,F)` : to generate a new name F using the atom f.
- iv) `member(H,L)` : to check if H is a member of the list L.
- v) `sub(X,C,L,L1)` : this is to substitute C for X in the list L ;  
the result list is L1.
- vi) `clean` : it is used after each session of use, in order to use the system using the same theory and the same arrows, if any, of the underlying category for another session.
- vii) `new` : Cleans up the database from all the processes of the last session and puts the system again to the top level.

## Conclusion

=====

In this thesis, we have implemented, in Prolog, the cut-elimination and the normalisation algorithms for a variant of Gentzen's sequent calculus, in order to provide a mechanised version of the decision procedure for the word problem for various kinds of category. These algorithms were outlined in SZABO, for both the cartesian and monoidal kinds of category: we have restricted our work to the cartesian kind. Other algorithms have been implemented to interpret sequent calculus derivations as arrows in the [ cartesian, ... ] categories, and to represent arrows of these categories by means of sequent calculus derivations. Some of the details in SZABO's algorithm have been modified or expanded to allow implementation.

The pattern-matching and debugging facilities of Prolog were found to be well-suited for this type of work. But the algorithms are both space and time expensive, and our implementation in turn takes a lot of space and time. Our representation of the sequent calculus derivations in Prolog is too complicated; this seems to be the main reason behind the inefficiency of the system.

We do not propose to extend our implementation with other transformation rules to handle predicate logic, or other categories with more complicated structure such as limits and colimits. Another representation for sequent calculus derivations should be adopted, perhaps as in the type theoretic work of the PRL team (CONSTABLE [1985]).

In conclusion, this implementation has been useful as an educational exercise in formalising a piece of logic and category theory. Proof theoretic studies of the sequent calculus are of continuing importance for theorem-proving in predicate logic and type theory; the present approach to category theory via the sequent calculus seems, in view of the recent work of CURIEN [1985], no longer to be the best approach. Nevertheless, as noted by HUET [1985a], proof theoretic techniques may yet be useful in proving, for example, termination properties of the rewriting systems studied by CURIEN.

References

- CLOCKSIN, W. F. and MELLISH, C. S.  
 [1981] Programming in Prolog ( Springer-Verlag, Berlin ).
- COLMERAUER, A., KANOUI, H. and VAN CANEGHEM, M.  
 [1983] Reports and Documents, Prolog, Theoretical principles and Current trends, Technology and Science of Informatics, vol. 2, no. 4, 255 - 292.
- CONSTABLE, R.  
 [1985] Implementing mathematics. The NuPRL book, draft, Cornell Univ.
- CURIEN, P. -L.  
 [1985] Categorical combinators, sequential algorithms and functional programming; (these d'etat) ( CNRS - University Paris VII, LITP, No 85-26).
- EILENBERG, S., HARRISON, D. K., MACLANE, S. and ROHRL, H.(editors)  
 [1966] Proceedings of the conference on categorical algebra, La Jolla 1965 ( Springer, New York ).
- EILENBERG, S. and KELLY, G. M.  
 [1966] Closed Categories, in : EILENBERG et al. [1966] pp. 421 - 562.
- FEFERMAN, S.  
 [1968] Lectures on proof theory, Lecture Notes in Mathematics, vol. 70, pp. 1 - 107.
- FENSTAD, J. (editor)  
 [1971] Proceedings of the Second Scandinavian Logic Colloquium (North-Holland, Amsterdam ).
- FITTING, M. C.  
 [1969] Intuitionist Logic, Model Theory and Forcing (North-Holland, Amsterdam ).
- HERRLICH, H. and STRECKER, G. E.  
 [1973] Category Theory ( Allyn and Bacon, Boston ).

HILTON, P. ( editor )

[1969a] Category Theory, Homology Theory and their Applications I, Lecture Notes in Mathematics, vol. 86 ( Springer, Berlin ).

[1969b] Category Theory, Homology Theory and their Applications II, Lecture Notes in Mathematics, vol. 92 ( Springer, Berlin ).

HUET, G.

[1985a] Equational systems for category theory and intuitionistic logic; talk at Rewriting Techniques & Applications Conference, Dijon.

[1985b] Cartesian Closed Categories and Lambda-Calculus, preprint CMU and INRIA.

HUET, G. and OPPEN, D.

[1980] Equations and rewrite rules : A survey, in "Formal Language: Perspectives and Open Problems " ( R. BOOK Ed. ), Academic Press, New York.

KELLY, G. M.

[1964] On MACLANE's conditions for coherence of natural associativities, etc., Journal of Algebra 1, 397 - 402.

KELLY, G. M. and MACLANE, S.

[1971] Coherence in closed categories, Journal of Pure and Applied Algebra 1, 97 - 140.

KREISEL, G.

[1968] A survey of proof theory I, Journal of Symbolic Logic 33, 321 - 388.

[1971] A survey of proof theory II, in : FENSTAD [1971] pp. 109-170.

LAMBEK, J.

[1968] Deductive systems and categories I, Mathematical Systems Theory 2, 287 - 318.

[1969] Deductive systems and categories II, Lecture Notes in Mathematics, volume 86, pp. 76 - 122.

[1972] Deductive systems and categories III, Lecture Notes in Mathematics, volume 274, pp. 57 - 82.

- [1974] Functional completeness of cartesian categories, *Annals Mathematical Logic* 6, 259 - 292.

LAMBEK, J. and SCOTT, P. J.

- [1984] Cartesian closed categories and lambda-calculus, provisional version of part I of a book on Higher Order Categorical Logic ( to appear, Cambridge University Press ).

LAPLAZA, M.

- [1972] Coherence for distributivity, *Lecture Notes in Mathematics*, vol. 281, pp. 29 - 65.

LAWVERE, F. W.

- [1969] Diagonal arguments and cartesian closed categories, *Lecture Notes in Mathematics*, vol. 86, pp. 134 - 145.

MACLANE, S.

- [1963] Natural associativity and commutativity, *Rice University studies* 49, 28 - 46.

- [1971] *Categories for the Working Mathematician* (Springer, Berlin).

- [1972] ( editor ) *Coherence in Categories*, *Lecture Notes in Mathematics*, vol. 281 ( Springer Verlag, Berlin ).

MANN, C.

- [1975] The connection between equivalences of proofs and cartesian closed categories, *Proceedings of the London Mathematical Society, Series 3* 31, 289 - 310.

NAUR, P. et al.

- [1960] Report on the algorithmic language Algol 60 (1960), comm. *ACM* 3:5, 299 - 314, revised in comm. *ACM* 6 : 1, 1 - 17.

PEREIRA, F.

- [1984] *C-Prolog User's Manual*, version 1.5, Edinburgh Computer Aided Architectural Design.

POTTINGER, G.

- [1979] The Church-Rosser Theorem for the typed lambda-calculus with Extensional Pairing, preprint, Carnegie-Mellon University, Pittsburgh.

PRAWITZ, D.

[1965] Natural Deduction ( Almqvist & Wiksell, Stockholm ).

[1971] Ideas and results in proof theory, In : FENSTAD [1971]  
pp. 235 - 307.

ROSE, H. E. and SHEPHERDSON, J. C. ( editors )

[1975] Logic Colloquium 73 ( North-Holland, Amsterdam ).

SZABO, M. E.

[1969] (editor) The collected papers of Gerhard Gentzen ( North-  
Holland, Amsterdam ).

[1971] Gentzen methods applied to category theory I : Free symmetric  
monoidal closed categories, Journal of Symbolic Logic 36, 705.

[1972] Gentzen methods applied to category theory II : Free cartesian  
closed multicategories, Journal of Symbolic Logic 37, 434.

[1973] Multinomial coefficients in closed categories, Notice of the  
American Mathematical Society 73T, A190.

[1974] A categorical equivalence of proofs, Notre Dame Journal of  
Formal Logic 15, 177 - 191.

[1975] A counter-example to coherence in cartesian closed categories,  
Canadian Mathematical Bulletin 18, 111 - 114.

[1976] An addendum to my paper " A categorical equivalence of proofs",  
Notre Dame Journal of Formal Logic 17, 78.

[1977] The logic of closed categories, Notre Dame Journal of Formal  
Logic 18, 441 - 457.

[1978] Algebra of Proofs ( North-Holland, Amsterdam ).









Appendix 2

Cut Elimination Algorithm

In this appendix, we collect all the clauses used to prove the cut-elimination theorems. These are implemented in our system, in order to produce an equivalent cut-free representation for every arrow of each of the free categories defined in chapters 1 - 5.

(A1,A1) For all A,B,C in Ob(X) , all f in X(A,B) , and all g in X(B,C),

$$\frac{A \xrightarrow{f} B \quad B \xrightarrow{g} C}{A \rightarrow C} > A \xrightarrow{\text{comp}(g,f)} C \quad (C.1)$$

(A2,R2) For all A,B in Ob(X) , and all f in X(A,B) .

$$\frac{A \xrightarrow{f} B \quad \frac{GD \xrightarrow{g} F}{GBD \rightarrow F}}{GAD \rightarrow F} > \frac{GD \xrightarrow{g} F}{GAD \rightarrow F} \quad (C.2.1)$$

(A3,R2)

$$\frac{\rightarrow T \quad \frac{GD \xrightarrow{f} F}{GTD \rightarrow F}}{GD \rightarrow F} > GD \xrightarrow{f} F \quad (C.2.3)$$

(R10,R2)

$$\frac{\frac{G \xrightarrow{f} FaE \quad G \xrightarrow{g} FbE}{G \rightarrow Fa \uparrow bE} \quad \frac{DL \xrightarrow{h} H}{DabL \rightarrow H}}{DGL \rightarrow FHE} > \frac{\frac{G \xrightarrow{f} FaE \quad \frac{GD \xrightarrow{g} FbE \quad DabL \rightarrow H}{DaGL \rightarrow FHE}}{DGGL \rightarrow FFHE}}{DGGL \rightarrow FHE}}{DGL \rightarrow FHE} \quad (C.3)$$

(R13,R2)

$$\frac{\frac{G \xrightarrow{f} FabE}{G \rightarrow Fa \vee bE} \quad \frac{DL \xrightarrow{g} H}{Da \vee bL \rightarrow H}}{DGL \rightarrow FHE} > \frac{\frac{G \xrightarrow{f} FabE \quad \frac{DL \xrightarrow{g} H}{DbL \rightarrow H}}{DGL \rightarrow FaHE} \quad \frac{DL \xrightarrow{g} H}{DaL \rightarrow H}}{DDGLL \rightarrow FHHE}}{DDGLL \rightarrow FHE}}{DDGL \rightarrow FHE}}{DGL \rightarrow FHE} \quad (C.4)$$

(R15,R2)

$$\frac{\frac{G \overset{f}{a} D \rightarrow b}{GD \rightarrow a \Rightarrow b} \quad \frac{LZ \overset{g}{\rightarrow} H}{La \Rightarrow bZ \rightarrow H}}{LGDZ \rightarrow H} > \frac{\frac{LZ \overset{g}{\rightarrow} H}{LGZ \rightarrow H}}{LGDZ \rightarrow H} \quad (C.5)$$

(R5,A1)

$$\frac{\frac{G \overset{f}{\rightarrow} FE}{G \rightarrow FAE} \quad A \overset{g}{\rightarrow} B}{G \rightarrow FBE} > \frac{\frac{G \overset{f}{\rightarrow} FE}{G \rightarrow FBE}}{G \rightarrow FBE} \quad (C.7.1)$$

(R5,A4)

$$\frac{\frac{G \overset{f}{\rightarrow} FE}{G \rightarrow FE} \quad L \rightarrow}{G \rightarrow FE} > \frac{G \overset{f}{\rightarrow} FE}{G \rightarrow FE} \quad (C.7.2)$$

(R5,R11)

$$\frac{\frac{G \overset{f}{\rightarrow} FE}{G \rightarrow Fa \uparrow bE} \quad \frac{DabL \overset{g}{\rightarrow} H}{Da \uparrow bL \rightarrow H}}{DGL \rightarrow FHE} > \frac{\frac{G \overset{f}{\rightarrow} FE}{G \rightarrow FbE} \quad \frac{G \overset{f}{\rightarrow} FE}{G \rightarrow FaE} \quad \frac{DabL \overset{g}{\rightarrow} H}{DaGL \rightarrow FHE}}{DDGL \rightarrow FFHE}$$

$$\frac{DGGL \rightarrow FFHE}{DGGL \rightarrow FHE}}{DGL \rightarrow FHE} \quad (C.8)$$

(R5,R12)

$$\frac{\frac{G \overset{f}{\rightarrow} FE}{G \rightarrow Fa \vee bE} \quad \frac{DaL \overset{g}{\rightarrow} H}{Da \vee bL \rightarrow H} \quad \frac{DbL \overset{h}{\rightarrow} H}{DbL \rightarrow H}}{DGL \rightarrow FHE} > \frac{\frac{G \overset{f}{\rightarrow} FE}{G \rightarrow FbE} \quad \frac{G \overset{f}{\rightarrow} FE}{G \rightarrow FabE} \quad \frac{DaL \overset{g}{\rightarrow} H}{DGL \rightarrow FHbE} \quad \frac{DbL \overset{h}{\rightarrow} H}{DbL \rightarrow H}}{DDGLL \rightarrow FHHE}$$

$$\frac{DDGLL \rightarrow FHE}{DDGL \rightarrow FHE}}{DGL \rightarrow FHE} \quad (C.9)$$

(R5,R14)

$$\frac{\frac{G \overset{f}{\rightarrow} FE}{G \rightarrow Fa \Rightarrow bE} \quad \frac{D \overset{g}{\rightarrow} a}{LDa \Rightarrow bZ \rightarrow H} \quad \frac{LbZ \overset{h}{\rightarrow} H}{LbZ \rightarrow H}}{LDGZ \rightarrow FHE} > \frac{\frac{D \overset{g}{\rightarrow} a}{DG \rightarrow FE} \quad \frac{G \overset{f}{\rightarrow} FE}{aG \rightarrow FE} \quad \frac{LbZ \overset{h}{\rightarrow} H}{LbZ \rightarrow H}}{LDGZ \rightarrow FHE} \quad (C.10)$$

(R5,R2)

$$\begin{array}{c}
 \begin{array}{c}
 \frac{G \xrightarrow{f} FE}{G \xrightarrow{f} FaE} \\
 \frac{DL \xrightarrow{g} H}{DaL \xrightarrow{g} H} \\
 \hline
 DGL \xrightarrow{f} FHE
 \end{array}
 >
 \begin{array}{c}
 \frac{G \xrightarrow{f} FE}{G \xrightarrow{f} FHE} \\
 \frac{GL \xrightarrow{g} FHE}{DGL \xrightarrow{g} FHE}
 \end{array}
 \end{array}
 \quad (C.12)$$

(R10,R11)

$$\begin{array}{c}
 \begin{array}{c}
 \frac{G \xrightarrow{f} FaE}{G \xrightarrow{f} Fa \uparrow bE} \\
 \frac{G \xrightarrow{g} FbE}{Da \uparrow bL \xrightarrow{g} H} \\
 \frac{DabL \xrightarrow{h} H}{DatbL \xrightarrow{h} H} \\
 \hline
 DGL \xrightarrow{f} FHE
 \end{array}
 >
 \begin{array}{c}
 \frac{G \xrightarrow{f} FaE}{DGGL \xrightarrow{f} FFHEE} \\
 \frac{DaGL \xrightarrow{g} FHE}{DGGL \xrightarrow{g} FFHE} \\
 \frac{DabL \xrightarrow{h} H}{DGGL \xrightarrow{h} FHE} \\
 \hline
 DGL \xrightarrow{f} FHE
 \end{array}
 \end{array}
 \quad (C.14)$$

(R13,R12)

$$\begin{array}{c}
 \begin{array}{c}
 \frac{G \xrightarrow{f} FabE}{G \xrightarrow{f} Fa \vee bE} \\
 \frac{DaL \xrightarrow{g} H}{Da \vee bL \xrightarrow{g} H} \\
 \frac{DbL \xrightarrow{h} H}{DbL \xrightarrow{h} H} \\
 \hline
 DGL \xrightarrow{f} FHE
 \end{array}
 >
 \begin{array}{c}
 \frac{G \xrightarrow{f} FabE}{DGL \xrightarrow{f} FHbE} \\
 \frac{DaL \xrightarrow{g} H}{DbL \xrightarrow{h} H} \\
 \frac{DGL \xrightarrow{g} FHE}{DDGLL \xrightarrow{g} FHHE} \\
 \frac{DGL \xrightarrow{h} H}{DDGLL \xrightarrow{h} FHE} \\
 \hline
 DGL \xrightarrow{f} FHE
 \end{array}
 \end{array}
 \quad (C.15)$$

(R15,R14)

$$\begin{array}{c}
 \begin{array}{c}
 \frac{GaD \xrightarrow{f} b}{GD \xrightarrow{f} a \Rightarrow b} \\
 \frac{L \xrightarrow{g} a}{ZLa \Rightarrow bS \xrightarrow{g} H} \\
 \frac{ZbS \xrightarrow{h} H}{ZbS \xrightarrow{h} H} \\
 \hline
 ZLGDS \xrightarrow{f} H
 \end{array}
 >
 \begin{array}{c}
 \frac{L \xrightarrow{g} a}{GLD \xrightarrow{g} b} \\
 \frac{GaD \xrightarrow{f} b}{ZGLDS \xrightarrow{f} H} \\
 \frac{ZbS \xrightarrow{h} H}{ZLGDS \xrightarrow{h} H} \\
 \hline
 ZLGDS \xrightarrow{f} H
 \end{array}
 \end{array}
 \quad (C.16)$$

(-,R2)

$$\begin{array}{c}
 \begin{array}{c}
 \frac{G \xrightarrow{f} FcE}{DGLaZ \xrightarrow{f} FHE} \\
 \frac{DcLZ \xrightarrow{g} H}{DcLaZ \xrightarrow{g} H} \\
 \hline
 DGLaZ \xrightarrow{f} FHE
 \end{array}
 >
 \begin{array}{c}
 \frac{G \xrightarrow{f} FcE}{DGLZ \xrightarrow{f} FHE} \\
 \frac{DcLZ \xrightarrow{g} H}{DGLaZ \xrightarrow{g} FHE} \\
 \hline
 DGLaZ \xrightarrow{f} FHE
 \end{array}
 \end{array}
 \quad (C.18.1)$$

$$\begin{array}{c}
 \begin{array}{c}
 \frac{G \xrightarrow{f} FcE}{DaLGZ \xrightarrow{f} FHE} \\
 \frac{DLcZ \xrightarrow{g} H}{DaLcZ \xrightarrow{g} H} \\
 \hline
 DaLGZ \xrightarrow{f} FHE
 \end{array}
 >
 \begin{array}{c}
 \frac{G \xrightarrow{f} FcE}{DLGZ \xrightarrow{f} FHE} \\
 \frac{DLcZ \xrightarrow{g} H}{DaLGZ \xrightarrow{g} FHE} \\
 \hline
 DaLGZ \xrightarrow{f} FHE
 \end{array}
 \end{array}
 \quad (C.18.2)$$

(-,R3)

$$\begin{array}{c}
 \begin{array}{c}
 \frac{G \xrightarrow{f} FcE}{DGLaZ \xrightarrow{f} FHE} \\
 \frac{DcLaaZ \xrightarrow{g} H}{DcLaZ \xrightarrow{g} H} \\
 \hline
 DGLaZ \xrightarrow{f} FHE
 \end{array}
 >
 \begin{array}{c}
 \frac{G \xrightarrow{f} FcE}{DGLaaZ \xrightarrow{f} FHE} \\
 \frac{DcLaaZ \xrightarrow{g} H}{DGLaZ \xrightarrow{g} FHE} \\
 \hline
 DGLaZ \xrightarrow{f} FHE
 \end{array}
 \end{array}
 \quad (C.19.1)$$

$$\begin{array}{c}
 \begin{array}{c}
 \frac{G \xrightarrow{f} FcE}{DaLGZ \xrightarrow{f} FHE} \\
 \frac{DaaLcZ \xrightarrow{g} H}{DaLcZ \xrightarrow{g} H} \\
 \hline
 DaLGZ \xrightarrow{f} FHE
 \end{array}
 >
 \begin{array}{c}
 \frac{G \xrightarrow{f} FcE}{DaaLGZ \xrightarrow{f} FHE} \\
 \frac{DaaLcZ \xrightarrow{g} H}{DaLGZ \xrightarrow{g} FHE} \\
 \hline
 DaLGZ \xrightarrow{f} FHE
 \end{array}
 \end{array}
 \quad (C.19.2)$$

$$\begin{array}{c}
 \begin{array}{c}
 \frac{G \xrightarrow{f} FcE \quad \frac{DccL \xrightarrow{g} H}{DcL \rightarrow H}}{DGL \rightarrow FHE} \\
 \end{array}
 >
 \begin{array}{c}
 \frac{G \xrightarrow{f} FcE \quad \frac{G \xrightarrow{f} FcE \quad \frac{DccL \xrightarrow{g} H}{DcGL \rightarrow FHE}}{DcGL \rightarrow FHE}}{DGGL \rightarrow FFHEE} \\
 \frac{DGGL \rightarrow FFHEE}{DGGL \rightarrow FFHE} \\
 \frac{DGGL \rightarrow FFHE}{DGGL \rightarrow FHE} \\
 \frac{DGGL \rightarrow FHE}{DGL \rightarrow FHE}
 \end{array}
 \end{array}
 \tag{C.19.3}$$

(-,R4)

$$\begin{array}{c}
 \begin{array}{c}
 \frac{G \xrightarrow{f} FcE \quad \frac{DcLabZ \xrightarrow{g} H}{DcLbaZ \rightarrow H}}{DGLbaZ \rightarrow FHE} \\
 \end{array}
 >
 \begin{array}{c}
 \frac{G \xrightarrow{f} FcE \quad \frac{DcLabZ \xrightarrow{g} H}{DGLabZ \rightarrow FHE}}{DGLbaZ \rightarrow FHE} \\
 \frac{DGLabZ \rightarrow FHE}{DGLbaZ \rightarrow FHE}
 \end{array}
 \end{array}
 \tag{C.20.1}$$

$$\begin{array}{c}
 \begin{array}{c}
 \frac{G \xrightarrow{f} FcE \quad \frac{DabLcZ \xrightarrow{g} H}{DbaLcZ \rightarrow H}}{DbaLGZ \rightarrow FHE} \\
 \end{array}
 >
 \begin{array}{c}
 \frac{G \xrightarrow{f} FcE \quad \frac{GabLcZ \xrightarrow{g} H}{DabLGZ \rightarrow FHE}}{DbaLGZ \rightarrow FHE} \\
 \frac{DabLGZ \rightarrow FHE}{DbaLGZ \rightarrow FHE}
 \end{array}
 \end{array}
 \tag{C.20.2}$$

$$\begin{array}{c}
 \begin{array}{c}
 \frac{G \xrightarrow{f} FcE \quad \frac{DdcL \xrightarrow{g} H}{DcdL \rightarrow H}}{DGdL \rightarrow FHE} \\
 \end{array}
 >
 \begin{array}{c}
 \frac{G \xrightarrow{f} FcE \quad \frac{DdcL \xrightarrow{g} H}{DdGL \rightarrow FHE}}{DGdL \rightarrow FHE} \\
 \frac{DdGL \rightarrow FHE}{DGdL \rightarrow FHE}
 \end{array}
 \end{array}
 \tag{C.20.3}$$

$$\begin{array}{c}
 \begin{array}{c}
 \frac{G \xrightarrow{f} FcE \quad \frac{DcdL \xrightarrow{g} H}{DdcL \rightarrow H}}{DdGL \rightarrow FHE} \\
 \end{array}
 >
 \begin{array}{c}
 \frac{G \xrightarrow{f} FcE \quad \frac{DcdL \xrightarrow{g} H}{DGdL \rightarrow FHE}}{DdGL \rightarrow FHE} \\
 \frac{DGdL \rightarrow FHE}{DdGL \rightarrow FHE}
 \end{array}
 \end{array}
 \tag{C.20.4}$$

(-,R5)

$$\begin{array}{c}
 \begin{array}{c}
 \frac{G \xrightarrow{f} FcE \quad \frac{DcL \xrightarrow{g} H_0}{DcL \rightarrow H_0}}{DGL \rightarrow FH_0OE} \\
 \end{array}
 >
 \begin{array}{c}
 \frac{G \xrightarrow{f} FcE \quad \frac{DcL \xrightarrow{g} H_0}{DGL \rightarrow FH_0OE}}{DGL \rightarrow FH_0OE} \\
 \frac{DGL \rightarrow FH_0OE}{DGL \rightarrow FH_0OE}
 \end{array}
 \end{array}
 \tag{C.21}$$

(-,R6)

$$\begin{array}{c}
 \begin{array}{c}
 \frac{G \xrightarrow{f} FcE \quad \frac{DcL \xrightarrow{g} H_{aa}O}{DcL \rightarrow H_{aO}}}{DGL \rightarrow FH_{aO}E} \\
 \end{array}
 >
 \begin{array}{c}
 \frac{G \xrightarrow{f} FcE \quad \frac{DcL \xrightarrow{g} H_{aa}O}{DGL \rightarrow FH_{aa}OE}}{DGL \rightarrow FH_{aO}E} \\
 \frac{DGL \rightarrow FH_{aa}OE}{DGL \rightarrow FH_{aO}E}
 \end{array}
 \end{array}
 \tag{C.22}$$

(-,R7)

$$\begin{array}{c}
 \begin{array}{c}
 \frac{G \xrightarrow{f} FcE \quad \frac{DcL \xrightarrow{g} H_{ab}O}{DcL \rightarrow H_{ba}O}}{DGL \rightarrow FH_{ba}OE} \\
 \end{array}
 >
 \begin{array}{c}
 \frac{G \xrightarrow{f} FcE \quad \frac{DcL \xrightarrow{g} H_{ab}O}{DGL \rightarrow FH_{ab}OE}}{DGL \rightarrow FH_{ba}OE} \\
 \frac{DGL \rightarrow FH_{ab}OE}{DGL \rightarrow FH_{ba}OE}
 \end{array}
 \end{array}
 \tag{C.23}$$

(-,R10)

$$\begin{array}{c}
 \begin{array}{c}
 \frac{G \xrightarrow{f} FcE \quad \frac{DcL \xrightarrow{g} H_{aO} \quad DcL \xrightarrow{h} H_{bO}}{DcL \rightarrow H_{a \uparrow bO}}}{DGL \rightarrow FH_{a \uparrow bOE}} \\
 \end{array}
 >
 \begin{array}{c}
 \frac{G \xrightarrow{f} FcE \quad \frac{DcL \xrightarrow{g} H_{aO} \quad G \xrightarrow{f} FcE \quad DcL \xrightarrow{h} H_{bO}}{DGL \rightarrow FH_{aO}E} \quad \frac{G \xrightarrow{f} FcE \quad DcL \xrightarrow{h} H_{bO}}{DGL \rightarrow FH_{bOE}}}{DGL \rightarrow FH_{a \uparrow bOE}} \\
 \frac{DGL \rightarrow FH_{aO}E \quad DGL \rightarrow FH_{bOE}}{DGL \rightarrow FH_{a \uparrow bOE}}
 \end{array}
 \end{array}
 \tag{C.26}$$

(-,R11)

$$\frac{\frac{G \xrightarrow{f} FcE \quad \frac{DcLa \uparrow bZ \xrightarrow{g} H}{DcLabZ \xrightarrow{g} H}}{DGLa \uparrow bZ \xrightarrow{f} FHE}}{\quad} > \frac{\frac{G \xrightarrow{f} FcE \quad \frac{DcLabZ \xrightarrow{g} H}{DGLabZ \xrightarrow{f} FHE}}{DGLa \uparrow bZ \xrightarrow{g} FHE}}{\quad} \quad (C.27.1)$$

$$\frac{\frac{G \xrightarrow{f} FcE \quad \frac{Da \uparrow bLcZ \xrightarrow{g} H}{DabLcZ \xrightarrow{g} H}}{Da \uparrow bLGZ \xrightarrow{f} FHE}}{\quad} > \frac{\frac{G \xrightarrow{f} FcE \quad \frac{DabLcZ \xrightarrow{g} H}{DabLGZ \xrightarrow{f} FHE}}{Da \uparrow bLGZ \xrightarrow{g} FHE}}{\quad} \quad (C.27.2)$$

(-,R12)

$$\frac{\frac{G \xrightarrow{f} FcE \quad \frac{DcLa \vee bZ \xrightarrow{g} H \quad DcLbZ \xrightarrow{h} H}{DcLa \vee bZ \xrightarrow{g} H}}{DGLa \vee bZ \xrightarrow{f} FHE}}{\quad} > \frac{\frac{G \xrightarrow{f} FcE \quad \frac{DcLaZ \xrightarrow{g} H}{DGLaZ \xrightarrow{f} FHE} \quad \frac{G \xrightarrow{f} FcE \quad \frac{DcLbZ \xrightarrow{h} H}{DGLbZ \xrightarrow{f} FHE}}{DGLa \vee bZ \xrightarrow{h} FHE}}{DGLa \vee bZ \xrightarrow{g} FHE}}{\quad} \quad (C.28.1)$$

$$\frac{\frac{G \xrightarrow{f} FcE \quad \frac{Da \vee bLcZ \xrightarrow{g} H \quad DbLcZ \xrightarrow{h} H}{Da \vee bLcZ \xrightarrow{g} H}}{Da \vee bLGZ \xrightarrow{f} FHE}}{\quad} > \frac{\frac{G \xrightarrow{f} FcE \quad \frac{DaLcZ \xrightarrow{g} H}{DaLGZ \xrightarrow{f} FHE} \quad \frac{G \xrightarrow{f} FcE \quad \frac{DbLcZ \xrightarrow{h} H}{DbLGZ \xrightarrow{f} FHE}}{Da \vee bLGZ \xrightarrow{h} FHE}}{Da \vee bLGZ \xrightarrow{g} FHE}}{\quad} \quad (C.28.2)$$

(-,R13)

$$\frac{\frac{G \xrightarrow{f} FcE \quad \frac{DcL \xrightarrow{g} HabO \quad DcL \xrightarrow{h} Ha \vee bO}{DcL \xrightarrow{g} HabO}}{DGL \xrightarrow{f} FHa \vee bOE}}{\quad} > \frac{\frac{G \xrightarrow{f} FcE \quad \frac{DcL \xrightarrow{g} HabO}{DGL \xrightarrow{f} FHabOE}}{DGL \xrightarrow{g} FHa \vee bOE}}{\quad} \quad (C.29)$$

(-,R14)

$$\frac{\frac{G \xrightarrow{f} c \quad \frac{DcL \xrightarrow{g} a \quad ZbP \xrightarrow{h} F}{ZDcLa \Rightarrow bP \rightarrow F}}{ZDGLa \Rightarrow bP \rightarrow F}}{\quad} > \frac{\frac{G \xrightarrow{f} c \quad \frac{DcL \xrightarrow{g} a}{DGL \xrightarrow{f} a} \quad \frac{ZbP \xrightarrow{h} F}{ZbP \xrightarrow{h} F}}{ZDGLa \Rightarrow bP \rightarrow F}}{\quad} \quad (C.30.1)$$

$$\frac{\frac{G \xrightarrow{f} c \quad \frac{D \xrightarrow{g} a \quad LcZbP \xrightarrow{h} F}{LcZDa \Rightarrow bP \rightarrow F}}{LGZDa \Rightarrow bP \rightarrow F}}{\quad} > \frac{\frac{D \xrightarrow{g} a \quad \frac{G \xrightarrow{f} c \quad LcZbP \xrightarrow{h} F}{LGZbP \rightarrow F}}{LGZDa \Rightarrow bP \rightarrow F}}{\quad} \quad (C.30.2)$$

$$\frac{\frac{G \xrightarrow{f} c \quad \frac{D \xrightarrow{g} a \quad LbZcP \xrightarrow{h} F}{LDa \Rightarrow bZcP \rightarrow F}}{LDa \Rightarrow bZGP \rightarrow F}}{\quad} > \frac{\frac{D \xrightarrow{g} a \quad \frac{G \xrightarrow{f} c \quad LbZcP \xrightarrow{h} F}{LbZGP \rightarrow F}}{LDa \Rightarrow bZGP \rightarrow F}}{\quad} \quad (C.30.3)$$

(-,R15)

$$\frac{\frac{G \xrightarrow{f} c \quad \frac{DcLaZ \xrightarrow{g} b \quad DcLZ \rightarrow (a \Rightarrow b)}{DGLZ \rightarrow (a \Rightarrow b)}}{\quad} > \frac{\frac{G \xrightarrow{f} c \quad \frac{DcLaZ \xrightarrow{g} b}{DGLaZ \rightarrow b}}{DGLZ \rightarrow (a \Rightarrow b)}}{\quad} \quad (C.31.1)$$

$$\frac{\frac{G \xrightarrow{f} c \quad \frac{DaLcZ \xrightarrow{g} b}{DLcZ \rightarrow (a \Rightarrow b)}}{DLGZ \rightarrow (a \Rightarrow b)}}{DLGZ \rightarrow (a \Rightarrow b)} > \frac{\frac{G \xrightarrow{f} c \quad DaLcZ \xrightarrow{g} b}{DaLGZ \rightarrow b}}{DLGZ \rightarrow (a \Rightarrow b)} \quad (C.31.2)$$

(R2,-)

$$\frac{\frac{GD \xrightarrow{f} FcE \quad LcZ \xrightarrow{g} H}{LGaDZ \rightarrow FHE}}{LGaDZ \rightarrow FHE} > \frac{\frac{GD \xrightarrow{f} FcE \quad LcZ \xrightarrow{g} H}{LGDZ \rightarrow FHE}}{LGaDZ \rightarrow FHE} \quad (C.34)$$

(R3,-)

$$\frac{\frac{GaaD \xrightarrow{f} FcE \quad LcZ \xrightarrow{g} H}{LGaDZ \rightarrow FHE}}{LGaDZ \rightarrow FHE} > \frac{\frac{GaaD \xrightarrow{f} FcE \quad LcZ \xrightarrow{g} H}{LGaaDZ \rightarrow FHE}}{LGaDZ \rightarrow FHE} \quad (C.35)$$

(R4,-)

$$\frac{\frac{GabD \xrightarrow{f} FcE \quad LcZ \xrightarrow{g} H}{LGbaDZ \rightarrow FHE}}{LGbaDZ \rightarrow FHE} > \frac{\frac{GabD \xrightarrow{f} FcE \quad LcZ \xrightarrow{g} H}{LGabDZ \rightarrow FHE}}{LGbaDZ \rightarrow FHE} \quad (C.36)$$

(R5,-)

$$\frac{\frac{G \xrightarrow{f} FcEH \quad DcL \xrightarrow{g} O}{DGL \rightarrow FOEaH}}{DGL \rightarrow FOEaH} > \frac{\frac{G \xrightarrow{f} FcEH \quad DcL \xrightarrow{g} O}{DGL \rightarrow FOEH}}{DGL \rightarrow FOEaH} \quad (C.37.1)$$

$$\frac{\frac{G \xrightarrow{f} FEcH \quad DcL \xrightarrow{g} O}{DGL \rightarrow FaEOH}}{DGL \rightarrow FaEOH} > \frac{\frac{G \xrightarrow{f} FEcH \quad DcL \xrightarrow{g} O}{DGL \rightarrow FEOH}}{DGL \rightarrow FaEOH} \quad (C.37.2)$$

(R6,-)

$$\frac{\frac{G \xrightarrow{f} FcEaaH \quad DcL \xrightarrow{g} O}{DGL \rightarrow FOEaH}}{DGL \rightarrow FOEaH} > \frac{\frac{G \xrightarrow{f} FcEaaH \quad DcL \xrightarrow{g} O}{DGL \rightarrow FOEaaH}}{DGL \rightarrow FOEaH} \quad (C.38.1)$$

$$\frac{\frac{G \xrightarrow{f} FaaEcH \quad DcL \xrightarrow{g} O}{DGL \rightarrow FaEOH}}{DGL \rightarrow FaEOH} > \frac{\frac{G \xrightarrow{f} FaaEcH \quad DcL \xrightarrow{g} O}{DGL \rightarrow FaaEOH}}{DGL \rightarrow FaEOH} \quad (C.38.2)$$

$$\frac{\frac{G \xrightarrow{f} FccE \quad DcL \xrightarrow{g} H}{DGL \rightarrow FHE}}{DGL \rightarrow FHE} > \frac{\frac{G \xrightarrow{f} FccE \quad DcL \xrightarrow{g} H}{DGL \rightarrow FcHE} \quad DcL \xrightarrow{g} H}{\frac{DDGLL \rightarrow FHHE}{DDGLL \rightarrow FHE}}{DGL \rightarrow FHE} \quad (C.38.3)$$

(R7,-)

$$\frac{\begin{array}{l} G \xrightarrow{f} FcEabH \\ G \xrightarrow{g} FcEbH \quad DcL \xrightarrow{g} 0 \\ \hline DGL \rightarrow FOEbaH \end{array}}{>} \frac{\begin{array}{l} G \xrightarrow{f} FcEabH \quad DcL \xrightarrow{g} 0 \\ \hline DGL \rightarrow FOEabH \\ \hline DGL \rightarrow FOEbaH \end{array}}{(C.39.1)}$$

$$\frac{\begin{array}{l} G \xrightarrow{f} FabEcH \\ G \xrightarrow{g} FbaEcH \quad DcL \xrightarrow{g} 0 \\ \hline DGL \rightarrow FbaEOH \end{array}}{>} \frac{\begin{array}{l} G \xrightarrow{f} FabEcH \quad DcL \xrightarrow{g} 0 \\ \hline DGL \rightarrow FabEOH \\ \hline DGL \rightarrow FbaEOH \end{array}}{(C.39.2)}$$

$$\frac{\begin{array}{l} G \xrightarrow{f} FdcE \\ G \xrightarrow{g} FcdE \quad DcL \xrightarrow{g} H \\ \hline DGL \rightarrow FHdE \end{array}}{>} \frac{\begin{array}{l} G \xrightarrow{f} FdcE \quad DcL \xrightarrow{g} H \\ \hline DGL \rightarrow FdHE \\ \hline DGL \rightarrow FHdE \end{array}}{(C.39.3)}$$

$$\frac{\begin{array}{l} G \xrightarrow{f} FcdE \\ G \xrightarrow{g} FdcE \quad DcL \xrightarrow{g} H \\ \hline DGL \rightarrow FdHE \end{array}}{>} \frac{\begin{array}{l} G \xrightarrow{f} FcdE \quad DcL \xrightarrow{g} H \\ \hline DGL \rightarrow FHdE \\ \hline DGL \rightarrow FdHE \end{array}}{(C.39.4)}$$

(R10,-)

$$\frac{\begin{array}{l} G \xrightarrow{f} FcEaH \quad G \xrightarrow{g} FcEbH \\ G \xrightarrow{h} FcEa \uparrow bH \quad DcL \xrightarrow{h} 0 \\ \hline DGL \rightarrow FOEa \uparrow bH \end{array}}{>} \frac{\begin{array}{l} G \xrightarrow{f} FcEaH \quad DcL \xrightarrow{h} 0 \quad G \xrightarrow{g} FcEbH \quad DcL \xrightarrow{h} 0 \\ \hline DGL \rightarrow FOEaH \quad DGL \rightarrow FOEbH \\ \hline DGL \rightarrow FOEa \uparrow bH \end{array}}{(C.41.1)}$$

$$\frac{\begin{array}{l} G \xrightarrow{f} FaEcH \quad G \xrightarrow{g} FbEcH \\ G \xrightarrow{h} Fa \uparrow bEcH \quad DcL \xrightarrow{h} 0 \\ \hline DGL \rightarrow Fa \uparrow bEOH \end{array}}{>} \frac{\begin{array}{l} G \xrightarrow{f} FaEcH \quad DcL \xrightarrow{h} 0 \quad G \xrightarrow{g} FbEcH \quad DcL \xrightarrow{h} 0 \\ \hline DGL \rightarrow FaEOH \quad DGL \rightarrow FbEOH \\ \hline DGL \rightarrow Fa \uparrow bEOH \end{array}}{(C.41.2)}$$

(R11,-)

$$\frac{\begin{array}{l} GabD \xrightarrow{f} FcE \\ Ga \uparrow bD \xrightarrow{g} FcE \quad LcZ \xrightarrow{g} H \\ \hline LGa \uparrow bDZ \rightarrow FHE \end{array}}{>} \frac{\begin{array}{l} GabD \xrightarrow{f} FcE \quad LcZ \xrightarrow{g} H \\ \hline LGabDZ \rightarrow FHE \\ \hline LGa \uparrow bDZ \rightarrow FHE \end{array}}{(C.42)}$$

(R12,-)

$$\frac{\begin{array}{l} GaD \xrightarrow{f} FcE \quad GbD \xrightarrow{g} FcE \\ Ga \vee bD \xrightarrow{h} FcE \quad LcZ \xrightarrow{h} H \\ \hline LGa \vee bDZ \rightarrow FHE \end{array}}{>} \frac{\begin{array}{l} GaD \xrightarrow{f} FcE \quad LcZ \xrightarrow{h} H \quad GbD \xrightarrow{g} FcE \quad LcZ \xrightarrow{h} H \\ \hline LGaDZ \rightarrow FHE \quad LGbDZ \rightarrow FHE \\ \hline LGa \vee bDZ \rightarrow FHE \end{array}}{(C.43)}$$

(R13,-)

$$\frac{\begin{array}{l} G \xrightarrow{f} FcEabH \\ G \xrightarrow{g} FcEa \vee bH \quad DcL \xrightarrow{g} 0 \\ \hline DGL \rightarrow FOEa \vee bH \end{array}}{>} \frac{\begin{array}{l} G \xrightarrow{f} FcEabH \quad DcL \xrightarrow{g} 0 \\ \hline DGL \rightarrow FOEabH \\ \hline DGL \rightarrow FOEa \vee bH \end{array}}{(C.44.1)}$$



$$\begin{array}{c}
 \frac{G \xrightarrow{f} FabEcH}{G \xrightarrow{f} Fa \vee bEcH} \\
 \hline
 DGL \rightarrow Fa \vee bEOH
 \end{array}
 >
 \begin{array}{c}
 \frac{G \xrightarrow{f} FabEcH \quad DcL \xrightarrow{g} O}{DGL \rightarrow FabEOH} \\
 \hline
 DGL \rightarrow Fa \vee bEOH
 \end{array}
 \quad (C.44.2)$$

(R14,-)

$$\begin{array}{c}
 \frac{G \xrightarrow{f} a \quad DbL \xrightarrow{g} FcE}{DGa \Rightarrow bL \rightarrow FcE} \\
 \hline
 ZDGa \Rightarrow bLP \rightarrow FHE
 \end{array}
 >
 \begin{array}{c}
 \frac{G \xrightarrow{f} a \quad DbL \xrightarrow{g} FcE \quad ZcP \xrightarrow{h} H}{ZDbLP \rightarrow FHE} \\
 \hline
 ZDGa \Rightarrow bLP \rightarrow FHE
 \end{array}
 \quad (C.45)$$

Appendix 3

Contraction & Interchange Elimination Algorithm

In this appendix, we extend the relation  $>$  of appendix 2 to the global reducibility relation  $>=$  which is implemented in the system. The clauses of this appendix prove that for every cut-free derivation  $p$  which represents an arrow of one of the free categories in chapters 1-3, and which contains an instance of contraction or interchange, there exists a contraction-and-interchange-free  $q$  deriving the same sequent. As for chapters 4 and 5, it will remove all the instances of contraction and interchange with active formulae of the form  $a \dagger b$ . The algorithm is constructed to prove some lemmas stated without proofs in SZABO, in order to complete the description of the normal form.

(R2,R2)

$$\frac{\frac{\text{GDL} \xrightarrow{f} F}{\text{GaDL} \xrightarrow{f} F}}{\text{GaDbL} \xrightarrow{f} F} \quad >= \quad \frac{\frac{\text{GDL} \xrightarrow{f} F}{\text{GDbL} \xrightarrow{f} F}}{\text{GaDbL} \xrightarrow{f} F} \quad (\text{E.1})$$

(R2,R3)

$$\frac{\frac{\text{GaD} \xrightarrow{f} F}{\text{GaaD} \xrightarrow{f} F}}{\text{GaD} \xrightarrow{f} F} \quad >= \quad \text{GaD} \xrightarrow{f} F \quad (\text{E.2.1})$$

$$\frac{\frac{\frac{\text{GDaaL} \xrightarrow{f} F}{\text{GbDaaL} \xrightarrow{f} F}}{\text{GbDaL} \xrightarrow{f} F}}{\text{GbDaL} \xrightarrow{f} F} \quad >= \quad \frac{\frac{\text{GDaaL} \xrightarrow{f} F}{\text{GDaL} \xrightarrow{f} F}}{\text{GbDaL} \xrightarrow{f} F} \quad (\text{E.2.2})$$

$$\frac{\frac{\frac{\text{GaaDL} \xrightarrow{f} F}{\text{GaaDbL} \xrightarrow{f} F}}{\text{GaDbL} \xrightarrow{f} F}}{\text{GaDbL} \xrightarrow{f} F} \quad >= \quad \frac{\frac{\text{GaaDL} \xrightarrow{f} F}{\text{GaDL} \xrightarrow{f} F}}{\text{GaDbL} \xrightarrow{f} F} \quad (\text{E.2.3})$$

(R2,R4)

$$\frac{\frac{\text{GaD} \xrightarrow{f} F}{\text{GabD} \xrightarrow{f} F}}{\text{GbaD} \xrightarrow{f} F} \quad >= \quad \frac{\text{GaD} \xrightarrow{f} F}{\text{GbaD} \xrightarrow{f} F} \quad (\text{E.3.1})$$

$$\frac{\frac{\text{GaD} \xrightarrow{f} F}{\text{GbaD} \xrightarrow{f} F}}{\text{GabD} \xrightarrow{f} F} \quad >= \quad \frac{\text{GaD} \xrightarrow{f} F}{\text{GabD} \xrightarrow{f} F} \quad (\text{E.3.2})$$

$$\begin{array}{c}
 \text{GDabL} \xrightarrow{f} \text{F} \\
 \hline
 \text{GcDabL} \xrightarrow{f} \text{F} \\
 \hline
 \text{GcDbal} \xrightarrow{f} \text{F}
 \end{array}
 \quad \Rightarrow \quad
 \begin{array}{c}
 \text{GDabL} \xrightarrow{f} \text{F} \\
 \hline
 \text{GDbaL} \xrightarrow{f} \text{F} \\
 \hline
 \text{GcDbal} \xrightarrow{f} \text{F}
 \end{array}
 \quad (\text{E.3.3})$$

$$\begin{array}{c}
 \text{GabDL} \xrightarrow{f} \text{F} \\
 \hline
 \text{GabDcL} \xrightarrow{f} \text{F} \\
 \hline
 \text{GbaDcL} \xrightarrow{f} \text{F}
 \end{array}
 \quad \Rightarrow \quad
 \begin{array}{c}
 \text{GabDL} \xrightarrow{f} \text{F} \\
 \hline
 \text{GbaDL} \xrightarrow{f} \text{F} \\
 \hline
 \text{GbaDcL} \xrightarrow{f} \text{F}
 \end{array}
 \quad (\text{E.3.4})$$

(R2,R6)

$$\begin{array}{c}
 \text{GD} \xrightarrow{f} \text{FaaE} \\
 \hline
 \text{GbD} \xrightarrow{f} \text{FaaE} \\
 \hline
 \text{GbD} \xrightarrow{f} \text{FaE}
 \end{array}
 \quad \Rightarrow \quad
 \begin{array}{c}
 \text{GD} \xrightarrow{f} \text{FaaE} \\
 \hline
 \text{GD} \xrightarrow{f} \text{FaE} \\
 \hline
 \text{GbD} \xrightarrow{f} \text{FaE}
 \end{array}
 \quad (\text{E.4})$$

(R2,R7)

$$\begin{array}{c}
 \text{GD} \xrightarrow{f} \text{FabE} \\
 \hline
 \text{GcD} \xrightarrow{f} \text{FabE} \\
 \hline
 \text{GcD} \xrightarrow{f} \text{FbaE}
 \end{array}
 \quad \Rightarrow \quad
 \begin{array}{c}
 \text{GD} \xrightarrow{f} \text{FabE} \\
 \hline
 \text{GD} \xrightarrow{f} \text{FbaE} \\
 \hline
 \text{GcD} \xrightarrow{f} \text{FbaE}
 \end{array}
 \quad (\text{E.5})$$

(R3,R4)

$$\begin{array}{c}
 \text{GaaDbcL} \xrightarrow{f} \text{F} \\
 \hline
 \text{GaDbcL} \xrightarrow{f} \text{F} \\
 \hline
 \text{GaDcbL} \xrightarrow{f} \text{F}
 \end{array}
 \quad \Rightarrow \quad
 \begin{array}{c}
 \text{GaaDbcL} \xrightarrow{f} \text{F} \\
 \hline
 \text{GaaDcbL} \xrightarrow{f} \text{F} \\
 \hline
 \text{GaDcbL} \xrightarrow{f} \text{F}
 \end{array}
 \quad (\text{E.6.1})$$

$$\begin{array}{c}
 \text{GbcDaal} \xrightarrow{f} \text{F} \\
 \hline
 \text{GbcDaL} \xrightarrow{f} \text{F} \\
 \hline
 \text{GcbDaL} \xrightarrow{f} \text{F}
 \end{array}
 \quad \Rightarrow \quad
 \begin{array}{c}
 \text{GbcDaal} \xrightarrow{f} \text{F} \\
 \hline
 \text{GcbDaal} \xrightarrow{f} \text{F} \\
 \hline
 \text{GcbDaL} \xrightarrow{f} \text{F}
 \end{array}
 \quad (\text{E.6.2})$$

$$\begin{array}{c}
 \text{Gaabl} \xrightarrow{f} \text{F} \\
 \hline
 \text{Gabl} \xrightarrow{f} \text{F} \\
 \hline
 \text{GbaL} \xrightarrow{f} \text{F}
 \end{array}
 \quad \Rightarrow \quad
 \begin{array}{c}
 \text{Gaabl} \xrightarrow{f} \text{F} \\
 \hline
 \text{GabaL} \xrightarrow{f} \text{F} \\
 \hline
 \text{Gbaal} \xrightarrow{f} \text{F} \\
 \hline
 \text{GbaL} \xrightarrow{f} \text{F}
 \end{array}
 \quad (\text{E.6.3})$$

$$\begin{array}{c}
 \text{Gbaal} \xrightarrow{f} \text{F} \\
 \hline
 \text{GbaL} \xrightarrow{f} \text{F} \\
 \hline
 \text{Gabl} \xrightarrow{f} \text{F}
 \end{array}
 \quad \Rightarrow \quad
 \begin{array}{c}
 \text{Gbaal} \xrightarrow{f} \text{F} \\
 \hline
 \text{GabaL} \xrightarrow{f} \text{F} \\
 \hline
 \text{Gaabl} \xrightarrow{f} \text{F} \\
 \hline
 \text{Gabl} \xrightarrow{f} \text{F}
 \end{array}
 \quad (\text{E.6.4})$$

(R3,R5)

$$\begin{array}{c}
 \text{GaaD} \xrightarrow{f} \text{FE} \\
 \hline
 \text{GaaD} \xrightarrow{f} \text{FbE} \\
 \hline
 \text{GaD} \xrightarrow{f} \text{FbE}
 \end{array}
 \quad \Rightarrow \quad
 \begin{array}{c}
 \text{GaaD} \xrightarrow{f} \text{FE} \\
 \hline
 \text{GaD} \xrightarrow{f} \text{FE} \\
 \hline
 \text{GaD} \xrightarrow{f} \text{FbE}
 \end{array}
 \quad (\text{E.7})$$



$$\frac{\frac{\frac{GccDaL \xrightarrow{f} F \quad GccDbL \xrightarrow{g} F}{GccDa \vee bL \rightarrow F}}{GcDa \vee bL \rightarrow F}}{\geq} \frac{\frac{\frac{GccDaL \xrightarrow{f} F \quad GccDbL \xrightarrow{g} F}{GcDaL \rightarrow F \quad GcDbL \rightarrow F}}{GcDa \vee bL \rightarrow F}}{(E.11.2)}$$

$$\frac{\frac{\frac{\frac{GaD \xrightarrow{p} F \quad GbD \xrightarrow{q} F}{GaaD \rightarrow F \quad GbaD \rightarrow F}}{Ga \vee baD \rightarrow F} \quad \frac{\frac{GaD \xrightarrow{r} F \quad GbD \xrightarrow{s} F}{GabD \rightarrow F \quad GbbD \rightarrow F}}{Ga \vee bbD \rightarrow F}}{Ga \vee ba \vee bD \rightarrow F}}{Ga \vee bD \rightarrow F}}{\geq} \frac{\frac{GaD \xrightarrow{p} F \quad GbD \xrightarrow{s} F}{Ga \vee bD \rightarrow F}}{(E.11.3)}$$

$$\frac{\frac{\frac{\frac{GaD \xrightarrow{p} F \quad GbD \xrightarrow{q} F}{GaaD \rightarrow F \quad GabD \rightarrow F}}{Gaa \vee bD \rightarrow F} \quad \frac{\frac{GaD \xrightarrow{r} F \quad GbD \xrightarrow{s} F}{GbaD \rightarrow F \quad GbbD \rightarrow F}}{Gba \vee bD \rightarrow F}}{Ga \vee ba \vee bD \rightarrow F}}{Ga \vee bD \rightarrow F}}{\geq} \frac{\frac{GaD \xrightarrow{p} F \quad GbD \xrightarrow{s} F}{Ga \vee bD \rightarrow F}}{(E.11.4)}$$

(R3,R13)

$$\frac{\frac{\frac{GccD \xrightarrow{f} FabE}{GccD \xrightarrow{f} Fa \vee bE}}{GcD \xrightarrow{f} Fa \vee bE}}{\geq} \frac{\frac{GccD \xrightarrow{f} FabE}{GcD \xrightarrow{f} FabE}}{GcD \xrightarrow{f} Fa \vee bE}}{(E.12)}$$

(R3,R14)

$$\frac{\frac{\frac{GccD \rightarrow a \quad LbS \xrightarrow{g} e}{LGccDa \Rightarrow bS \rightarrow e}}{LGcDa \Rightarrow bS \rightarrow e}}{\geq} \frac{\frac{GccD \rightarrow a}{GcD \rightarrow a \quad LbS \xrightarrow{g} e}}{LGcDa \Rightarrow bS \rightarrow e}}{(E.13.1)}$$

$$\frac{\frac{\frac{G \xrightarrow{h} a \quad DccLbS \xrightarrow{k} e}{DccLGa \Rightarrow bS \rightarrow e}}{DcLGa \Rightarrow bS \rightarrow e}}{\geq} \frac{\frac{G \xrightarrow{h} a \quad DccLbS \xrightarrow{k} e}{DcLbS \rightarrow e}}{DcLGa \Rightarrow bS \rightarrow e}}{(E.13.2)}$$

$$\frac{\frac{\frac{G \xrightarrow{h} a \quad DbLccS \xrightarrow{j} e}{DGa \Rightarrow bLccS \rightarrow e}}{DGa \Rightarrow bLcS \rightarrow e}}{\geq} \frac{\frac{G \xrightarrow{h} a \quad DbLccS \xrightarrow{j} e}{DbLcS \rightarrow e}}{DGa \Rightarrow bLcS \rightarrow e}}{(E.13.3)}$$

(R3,R15)

$$\frac{\frac{\frac{GccDaL \rightarrow b}{GccDL \rightarrow (a \Rightarrow b)}}{GcDL \rightarrow (a \Rightarrow b)}}{\geq} \frac{\frac{GccDaL \xrightarrow{f} b}{GcDaL \rightarrow b}}{GcDL \rightarrow (a \Rightarrow b)}}{(E.14.1)}$$

$$\frac{\frac{\frac{GaDccL \rightarrow b}{GDccL \rightarrow (a \Rightarrow b)}}{GDcL \rightarrow (a \Rightarrow b)}}{\geq} \frac{\frac{GaDccL \xrightarrow{f} b}{GaDcL \rightarrow b}}{GDcL \rightarrow (a \Rightarrow b)}}{(E.14.2)}$$

$$\begin{array}{c}
 \frac{GcacD \xrightarrow{f} b}{GccD \rightarrow (a \Rightarrow b)} \\
 \hline
 GcD \rightarrow (a \Rightarrow b)
 \end{array}
 \quad \geq \quad
 \begin{array}{c}
 \frac{GcacD \xrightarrow{f} b}{GccD \rightarrow b} \\
 \hline
 GcaD \rightarrow b \\
 \hline
 GcD \rightarrow (a \Rightarrow b)
 \end{array}
 \quad (E.14.3)$$

(R4,R5)

$$\begin{array}{c}
 \frac{GabD \xrightarrow{f} FE}{GabD \xrightarrow{\quad} FcE} \\
 \hline
 GbaD \xrightarrow{\quad} FcE
 \end{array}
 \quad \geq \quad
 \begin{array}{c}
 \frac{GabD \xrightarrow{f} FE}{GbaD \xrightarrow{\quad} FE} \\
 \hline
 GbaD \xrightarrow{\quad} FcE
 \end{array}
 \quad (E.15)$$

(R4,R6)

$$\begin{array}{c}
 \frac{GabD \xrightarrow{f} FccE}{GabD \xrightarrow{\quad} FcE} \\
 \hline
 GbaD \xrightarrow{\quad} FcE
 \end{array}
 \quad \geq \quad
 \begin{array}{c}
 \frac{GabD \xrightarrow{f} FccE}{GbaD \xrightarrow{\quad} FccE} \\
 \hline
 GbaD \xrightarrow{\quad} FcE
 \end{array}
 \quad (E.16)$$

(R4,R10)

$$\begin{array}{c}
 \frac{GcdD \xrightarrow{f} FaE \quad GcdD \xrightarrow{g} FbE}{GcdD \rightarrow Fa \uparrow bE} \\
 \hline
 GdcD \rightarrow Fa \uparrow bE
 \end{array}
 \quad \geq \quad
 \begin{array}{c}
 \frac{GcdD \xrightarrow{f} FaE \quad GcdD \xrightarrow{g} FbE}{GdcD \rightarrow FaE \quad GdcD \rightarrow FbE} \\
 \hline
 GdcD \rightarrow Fa \uparrow bE
 \end{array}
 \quad (E.17)$$

(R4,R11)

$$\begin{array}{c}
 \frac{GcdDabL \xrightarrow{f} F}{GctdDabL \rightarrow F} \\
 \hline
 GctdDbal \rightarrow F
 \end{array}
 \quad \geq \quad
 \begin{array}{c}
 \frac{GcdDabL \xrightarrow{f} F}{GcdDbal \rightarrow F} \\
 \hline
 GctdDbal \rightarrow F
 \end{array}
 \quad (E.18.1)$$

$$\begin{array}{c}
 \frac{GabDcdL \xrightarrow{f} F}{GabDctdL \rightarrow F} \\
 \hline
 GbaDctdL \rightarrow F
 \end{array}
 \quad \geq \quad
 \begin{array}{c}
 \frac{GabDcdL \xrightarrow{f} F}{GbaDcdL \rightarrow F} \\
 \hline
 GbaDctdL \rightarrow F
 \end{array}
 \quad (E.18.2)$$

$$\begin{array}{c}
 \frac{GabcD \xrightarrow{f} F}{Ga \uparrow bcD \rightarrow F} \\
 \hline
 Gca \uparrow bD \rightarrow F
 \end{array}
 \quad \geq \quad
 \begin{array}{c}
 \frac{GabcD \xrightarrow{f} F}{GacbD \rightarrow F} \\
 \hline
 GcabD \rightarrow F \\
 \hline
 Gca \uparrow bD \rightarrow F
 \end{array}
 \quad (E.18.3)$$

$$\begin{array}{c}
 \frac{GcabD \xrightarrow{f} F}{Gca \uparrow bD \rightarrow F} \\
 \hline
 Ga \uparrow bcD \rightarrow F
 \end{array}
 \quad \geq \quad
 \begin{array}{c}
 \frac{GcabD \xrightarrow{f} F}{GacbD \rightarrow F} \\
 \hline
 GabcD \rightarrow F \\
 \hline
 Ga \uparrow bcD \rightarrow F
 \end{array}
 \quad (E.18.4)$$

(R4,R12)

$$\begin{array}{c}
 \frac{GcdDaL \xrightarrow{f} F \quad GcdDbL \xrightarrow{g} F}{GcdDa \vee bL \rightarrow F} \\
 \hline
 GdcDa \vee bL \rightarrow F
 \end{array}
 \quad \geq \quad
 \begin{array}{c}
 \frac{GcdDaL \xrightarrow{f} F \quad GcdDbL \xrightarrow{g} F}{GdcDaL \rightarrow F \quad GdcDbL \rightarrow F} \\
 \hline
 GdcDa \vee bL \rightarrow F
 \end{array}
 \quad (E.19.1)$$

$$\frac{\frac{GaDcdL \xrightarrow{f} F \quad GbDcdL \xrightarrow{g} F}{Ga \vee bDcdL \rightarrow F}}{Ga \vee bDdcL \rightarrow F} \geq \frac{\frac{GaDcdL \xrightarrow{f} F \quad GbDcdL \xrightarrow{g} F}{GaDdcL \rightarrow F} \quad \frac{GbDcdL \xrightarrow{g} F}{GbDdcL \rightarrow F}}{Ga \vee bDdcL \rightarrow F} \quad (E.19.2)$$

$$\frac{\frac{GacD \xrightarrow{f} F \quad GbcD \xrightarrow{g} F}{Ga \vee bcD \rightarrow F}}{Gca \vee bD \rightarrow F} \geq \frac{\frac{GacD \xrightarrow{f} F}{GcaD \rightarrow F} \quad \frac{GbcD \xrightarrow{g} F}{GcbD \rightarrow F}}{Gca \vee bD \rightarrow F} \quad (E.19.3)$$

$$\frac{\frac{GcaD \xrightarrow{f} F \quad GcbD \xrightarrow{g} F}{Gca \vee bD \rightarrow F}}{Ga \vee bcD \rightarrow F} \geq \frac{\frac{GcaD \xrightarrow{f} F}{GacD \rightarrow F} \quad \frac{GcbD \xrightarrow{g} F}{GbcD \rightarrow F}}{Ga \vee bcD \rightarrow F} \quad (E.19.4)$$

(R4,R13)

$$\frac{\frac{GcdD \xrightarrow{f} FabE}{GcdD \rightarrow Fa \vee bE}}{GdcD \rightarrow Fa \vee bE} \geq \frac{\frac{GcdD \xrightarrow{f} FabE}{GdcD \rightarrow FabE}}{GdcD \rightarrow Fa \vee bE} \quad (E.20)$$

(R4,R14)

$$\frac{\frac{GcdD \xrightarrow{f} a \quad LbS \xrightarrow{g} e}{LGcdDa \Rightarrow bS \rightarrow e}}{LGdcDa \Rightarrow bS \rightarrow e} \geq \frac{\frac{GcdD \xrightarrow{f} a}{GdcD \rightarrow a} \quad LbS \xrightarrow{g} e}{LGdcDa \Rightarrow bS \rightarrow e} \quad (E.21.1)$$

$$\frac{\frac{G \xrightarrow{h} a \quad DcdLbS \xrightarrow{k} e}{DcdLGa \Rightarrow bS \rightarrow e}}{DdcLGa \Rightarrow bS \rightarrow e} \geq \frac{\frac{G \xrightarrow{h} a \quad DcdLbS \xrightarrow{k} e}{DdcLbS \rightarrow e}}{DdcLGa \Rightarrow bS \rightarrow e} \quad (E.21.2)$$

$$\frac{\frac{G \xrightarrow{h} a \quad DbLcdS \xrightarrow{j} e}{DGa \Rightarrow bLcdS \rightarrow e}}{DGa \Rightarrow bLdcS \rightarrow e} \geq \frac{\frac{G \xrightarrow{h} a \quad DbLcdS \xrightarrow{j} e}{DbLcdS \rightarrow e}}{DGa \Rightarrow bLdcS \rightarrow e} \quad (E.21.3)$$

(R4,R15)

$$\frac{\frac{GcdDaL \xrightarrow{f} b}{GcdDL \rightarrow (a \Rightarrow b)}}{GdcDL \rightarrow (a \Rightarrow b)} \geq \frac{\frac{GcdDaL \xrightarrow{f} b}{GdcDaL \rightarrow b}}{GdcDL \rightarrow (a \Rightarrow b)} \quad (E.22.1)$$

$$\frac{\frac{GaDcdL \xrightarrow{f} b}{GDcdL \rightarrow (a \Rightarrow b)}}{GDdcL \rightarrow (a \Rightarrow b)} \geq \frac{\frac{GaDcdL \xrightarrow{f} b}{GaDdcL \rightarrow b}}{GDdcL \rightarrow (a \Rightarrow b)} \quad (E.22.2)$$

(R5,R5)

$$\frac{\frac{G \xrightarrow{f} FEH}{G \rightarrow FaEH}}{G \rightarrow FaEbH} \geq \frac{\frac{G \xrightarrow{f} FEH}{G \rightarrow FEbH}}{G \rightarrow FaEbH} \quad (E.23)$$

(R5, R6)

$$\frac{\frac{G \rightarrow FaE}{G \rightarrow FaaE}}{G \rightarrow FaE} \quad \supset= \quad \frac{G \xrightarrow{f} FaE}{G \xrightarrow{f} FaE} \quad (E.24.1)$$

$$\frac{\frac{\frac{G \rightarrow FaaEH}{G \rightarrow FaaEbH}}{G \rightarrow FaEbH}}{G \rightarrow FaEbH} \quad \supset= \quad \frac{\frac{G \xrightarrow{f} FaaEH}{G \xrightarrow{f} FaEH}}{G \xrightarrow{f} FaEbH}}{G \xrightarrow{f} FaEbH} \quad (E.24.2)$$

$$\frac{\frac{\frac{G \rightarrow FEaaH}{G \rightarrow FbEaaH}}{G \rightarrow FbEaH}}{G \rightarrow FbEaH} \quad \supset= \quad \frac{\frac{G \xrightarrow{f} FEaaH}{G \xrightarrow{f} FEaH}}{G \xrightarrow{f} FbEaH}}{G \xrightarrow{f} FbEaH} \quad (E.24.3)$$

(R5, R7)

$$\frac{\frac{\frac{G \xrightarrow{f} FEabH}{G \rightarrow FcEabH}}{G \rightarrow FcEbaH}}{G \rightarrow FcEbaH} \quad \supset= \quad \frac{\frac{\frac{G \xrightarrow{f} FEabH}{G \rightarrow FEbaH}}{G \rightarrow FcEbaH}}{G \rightarrow FcEbaH}}{G \rightarrow FcEbaH} \quad (E.25.1)$$

$$\frac{\frac{\frac{G \xrightarrow{f} FabEH}{G \rightarrow FabEcH}}{G \rightarrow FbaEcH}}{G \rightarrow FbaEcH} \quad \supset= \quad \frac{\frac{\frac{G \xrightarrow{f} FabEH}{G \rightarrow FbaEH}}{G \rightarrow FbaEcH}}{G \rightarrow FbaEcH}}{G \rightarrow FbaEcH} \quad (E.25.2)$$

$$\frac{\frac{\frac{G \xrightarrow{f} FaE}{G \rightarrow FabE}}{G \rightarrow FbaE}}{G \rightarrow FbaE} \quad \supset= \quad \frac{\frac{G \xrightarrow{f} FaE}{G \rightarrow FbaE}}{G \rightarrow FbaE}}{G \rightarrow FbaE} \quad (E.25.3)$$

$$\frac{\frac{\frac{G \xrightarrow{f} Fae}{G \rightarrow FbaE}}{G \rightarrow FabE}}{G \rightarrow FabE} \quad \supset= \quad \frac{\frac{G \xrightarrow{f} FaE}{G \rightarrow FabE}}{G \rightarrow FabE}}{G \rightarrow FabE} \quad (E.25.4)$$

(R6, R7)

$$\frac{\frac{\frac{G \xrightarrow{f} FccEabH}{G \rightarrow FcEabH}}{G \rightarrow FcEbaH}}{G \rightarrow FcEbaH} \quad \supset= \quad \frac{\frac{\frac{G \xrightarrow{f} FccEabH}{G \rightarrow FccEbaH}}{G \rightarrow FcEbaH}}{G \rightarrow FcEbaH}}{G \rightarrow FcEbaH} \quad (E.26.1)$$

$$\frac{\frac{\frac{G \xrightarrow{f} FabEccH}{G \rightarrow FabEcH}}{G \rightarrow FbaEcH}}{G \rightarrow FbaEcH} \quad \supset= \quad \frac{\frac{\frac{G \xrightarrow{f} FabEccH}{G \rightarrow FbaEccH}}{G \rightarrow FbaEcH}}{G \rightarrow FbaEcH}}{G \rightarrow FbaEcH} \quad (E.26.2)$$

$$\frac{\frac{\frac{G \xrightarrow{f} FaabE}{G \rightarrow FabE}}{G \rightarrow FbaE}}{G \rightarrow FbaE} \quad \supset= \quad \frac{\frac{\frac{G \xrightarrow{f} FaabE}{G \rightarrow FabaE}}{G \rightarrow FbaaE}}{G \rightarrow FbaE}}{G \rightarrow FbaE}}{G \rightarrow FbaE} \quad (E.26.3)$$









$$\begin{array}{c}
 \frac{G \xrightarrow{f} FcaE \quad G \xrightarrow{g} FcbE}{G \rightarrow Fca \uparrow bE} \\
 \hline
 G \rightarrow Fa \uparrow bcE
 \end{array}
 \quad \Rightarrow \quad
 \begin{array}{c}
 \frac{G \xrightarrow{f} FcaE \quad G \xrightarrow{g} FcbE}{G \rightarrow FcaE \quad G \rightarrow FcbE} \\
 \hline
 G \rightarrow Fa \uparrow bcE
 \end{array}
 \quad (E.32.4)$$

(R7,R11)

$$\begin{array}{c}
 \frac{G \xrightarrow{f} FcdE}{Ga \uparrow bD \rightarrow FcdE} \\
 \hline
 Ga \uparrow bD \rightarrow FdcE
 \end{array}
 \quad \Rightarrow \quad
 \begin{array}{c}
 \frac{G \xrightarrow{f} FcdE}{G \xrightarrow{f} FcdE} \\
 \hline
 Ga \uparrow bD \rightarrow FdcE
 \end{array}
 \quad (E.33)$$

(R7,R12)

$$\begin{array}{c}
 \frac{GaD \xrightarrow{f} FcdE \quad GbD \xrightarrow{g} FcdE}{Ga \vee bD \rightarrow FcdE} \\
 \hline
 Ga \vee bD \rightarrow FdcE
 \end{array}
 \quad \Rightarrow \quad
 \begin{array}{c}
 \frac{GaD \xrightarrow{f} FcdE \quad GbD \xrightarrow{g} FcdE}{GaD \rightarrow FdcE \quad GbD \rightarrow FdcE} \\
 \hline
 Ga \vee bD \rightarrow FdcE
 \end{array}
 \quad (E.34)$$

(R7,R13)

$$\begin{array}{c}
 \frac{G \xrightarrow{f} FabEcdH}{G \rightarrow Fa \vee bEcdH} \\
 \hline
 G \rightarrow Fa \vee bEdcH
 \end{array}
 \quad \Rightarrow \quad
 \begin{array}{c}
 \frac{G \xrightarrow{f} FabEcdH}{G \rightarrow FabEcdH} \\
 \hline
 G \rightarrow Fa \vee bEdcH
 \end{array}
 \quad (E.35.1)$$

$$\begin{array}{c}
 \frac{G \xrightarrow{f} FcdEabH}{G \rightarrow FcdEa \vee bH} \\
 \hline
 G \rightarrow FdcEa \vee bH
 \end{array}
 \quad \Rightarrow \quad
 \begin{array}{c}
 \frac{G \xrightarrow{f} FcdEabH}{G \rightarrow FcdEabH} \\
 \hline
 G \rightarrow FdcEa \vee bH
 \end{array}
 \quad (E.35.2)$$

$$\begin{array}{c}
 \frac{G \xrightarrow{f} FabcH}{G \rightarrow Fa \vee bcH} \\
 \hline
 G \rightarrow Fca \vee bH
 \end{array}
 \quad \Rightarrow \quad
 \begin{array}{c}
 \frac{G \xrightarrow{f} FabcH}{G \rightarrow FacbH} \\
 \hline
 G \rightarrow FcabH \\
 \hline
 G \rightarrow Fca \vee bH
 \end{array}
 \quad (E.35.3)$$

$$\begin{array}{c}
 \frac{G \xrightarrow{f} FcabH}{G \rightarrow Fca \vee bH} \\
 \hline
 G \rightarrow Fa \vee bcH
 \end{array}
 \quad \Rightarrow \quad
 \begin{array}{c}
 \frac{G \xrightarrow{f} FcabH}{G \rightarrow FacbH} \\
 \hline
 G \rightarrow FcabH \\
 \hline
 G \rightarrow Fa \vee bcH
 \end{array}
 \quad (E.35.4)$$

(R7,R14)

$$\begin{array}{c}
 \frac{G \xrightarrow{f} a \quad DbL \xrightarrow{g} FcdE}{DGa \Rightarrow bL \rightarrow FcdE} \\
 \hline
 DGa \Rightarrow bL \rightarrow FdcE
 \end{array}
 \quad \Rightarrow \quad
 \begin{array}{c}
 \frac{G \xrightarrow{f} a \quad DbL \xrightarrow{g} FcdE}{G \rightarrow a \quad DbL \rightarrow FcdE} \\
 \hline
 DGa \Rightarrow bL \rightarrow FdcE
 \end{array}
 \quad (E.36)$$

(R2,R3,R11)

$$\begin{array}{c}
 \frac{Ga \uparrow bL \xrightarrow{f} F}{Ga \uparrow bbL \rightarrow F} \\
 \hline
 \frac{Ga \uparrow babL \rightarrow F}{Ga \uparrow ba \uparrow bL \rightarrow F} \\
 \hline
 Ga \uparrow bL \rightarrow F
 \end{array}
 \quad \Rightarrow \quad
 \frac{Ga \uparrow bL \xrightarrow{f} F}{Ga \uparrow bL \rightarrow F}
 \quad (E.37.1)$$



$$\begin{array}{c}
 \text{f} \\
 \hline
 G \rightarrow FabbE \\
 \hline
 G \rightarrow Fa \vee bbE \\
 \hline
 G \rightarrow Fa \vee babE \\
 \hline
 G \rightarrow Fa \vee ba \vee bE \\
 \hline
 G \rightarrow Fa \vee bE
 \end{array}
 \quad \Rightarrow \quad
 \begin{array}{c}
 G \rightarrow FabbE \\
 \hline
 G \rightarrow FabE \\
 \hline
 G \rightarrow Fa \vee bE
 \end{array}
 \quad (E.38.3)$$

$$\begin{array}{c}
 \text{f} \\
 \hline
 G \rightarrow FbabE \\
 \hline
 G \rightarrow Fba \vee bE \\
 \hline
 G \rightarrow Faba \vee bE \\
 \hline
 G \rightarrow Fa \vee ba \vee bE \\
 \hline
 G \rightarrow Fa \vee bE
 \end{array}
 \quad \Rightarrow \quad
 \begin{array}{c}
 G \rightarrow FbabE \\
 \hline
 G \rightarrow FabbE \\
 \hline
 G \rightarrow FabE \\
 \hline
 G \rightarrow Fa \vee bE
 \end{array}
 \quad (E.38.4)$$

$$\begin{array}{c}
 \text{f} \\
 \hline
 G \rightarrow FabaE \\
 \hline
 G \rightarrow Fa \vee baE \\
 \hline
 G \rightarrow Fa \vee babE \\
 \hline
 G \rightarrow Fa \vee ba \vee bE \\
 \hline
 G \rightarrow Fa \vee bE
 \end{array}
 \quad \Rightarrow \quad
 \begin{array}{c}
 G \rightarrow FabaE \\
 \hline
 G \rightarrow FaabE \\
 \hline
 G \rightarrow FabE \\
 \hline
 G \rightarrow Fa \vee bE
 \end{array}
 \quad (E.38.5)$$

$$\begin{array}{c}
 \text{f} \\
 \hline
 G \rightarrow FaabE \\
 \hline
 G \rightarrow Faa \vee bE \\
 \hline
 G \rightarrow Faba \vee bE \\
 \hline
 G \rightarrow Fa \vee ba \vee bE \\
 \hline
 G \rightarrow Fa \vee bE
 \end{array}
 \quad \Rightarrow \quad
 \begin{array}{c}
 G \rightarrow FaabE \\
 \hline
 G \rightarrow FabE \\
 \hline
 G \rightarrow Fa \vee bE
 \end{array}
 \quad (E.38.6)$$

(R3, R12, R14)

$$\begin{array}{c}
 \begin{array}{c}
 \text{f} \quad \text{g} \\
 \hline
 aG \rightarrow c \quad bG \rightarrow c \\
 \hline
 a \vee bG \rightarrow c
 \end{array}
 \quad
 \begin{array}{c}
 \text{h} \quad \text{k} \\
 \hline
 DadL \rightarrow F \quad DbdL \rightarrow F \\
 \hline
 Da \vee bdL \rightarrow F
 \end{array} \\
 \hline
 \begin{array}{c}
 Da \vee b a \vee bGc \Rightarrow dL \rightarrow F \\
 \hline
 Da \vee bGc \Rightarrow dL \rightarrow F
 \end{array} \\
 \\
 \begin{array}{c}
 \text{f} \quad \text{h} \quad \text{g} \quad \text{k} \\
 \hline
 aG \rightarrow c \quad DadL \rightarrow F \quad bG \rightarrow c \quad DbdL \rightarrow F \\
 \hline
 DaaGc \Rightarrow dL \rightarrow F \quad DbbGc \Rightarrow dL \rightarrow F \\
 \hline
 DaGc \Rightarrow dL \rightarrow F \quad DbGc \Rightarrow dL \rightarrow F \\
 \hline
 Da \vee bGc \Rightarrow dL \rightarrow F
 \end{array}
 \quad \Rightarrow \quad
 \end{array}
 \quad (E.39)$$

(R4, R12, R14)

$$\begin{array}{c}
 \begin{array}{c}
 \text{f} \quad \text{g} \\
 \hline
 aG \rightarrow c \quad bG \rightarrow c \\
 \hline
 a \vee bG \rightarrow c
 \end{array}
 \quad
 \begin{array}{c}
 \text{k} \\
 \hline
 DedL \rightarrow F \\
 \hline
 DedL \rightarrow F
 \end{array} \\
 \hline
 \begin{array}{c}
 Daa \vee bGc \Rightarrow dL \rightarrow F \\
 \hline
 Da \vee beGc \Rightarrow dL \rightarrow F
 \end{array}
 \quad \Rightarrow \quad
 \begin{array}{c}
 \text{f} \quad \text{k} \quad \text{g} \quad \text{k} \\
 \hline
 aG \rightarrow c \quad DedL \rightarrow F \quad bG \rightarrow c \quad DedL \rightarrow F \\
 \hline
 DeaGc \Rightarrow dL \rightarrow F \quad DebGc \Rightarrow dL \rightarrow F \\
 \hline
 DaeGc \Rightarrow dL \rightarrow F \quad DbeGc \Rightarrow dL \rightarrow F \\
 \hline
 Da \vee beGc \Rightarrow dL \rightarrow F
 \end{array}
 \quad (E.40.1)$$



Appendix 4  
=====

Normalization Algorithm  
=====

In this appendix, we collect from appendix (D) in SZABO all those clauses which determine the choice and order of application of Rules (R2)-(R15) in case of the categories considered above. This completes the description of the normal form in chapters 1-5.

(R2,R2)

$$\frac{\frac{\text{GDL} \xrightarrow{f} \text{F}}{\text{GaDL} \xrightarrow{f} \text{F}}}{\text{GaDbL} \xrightarrow{f} \text{F}} \quad \supset= \quad \frac{\frac{\text{GDL} \xrightarrow{f} \text{F}}{\text{GDbL} \xrightarrow{f} \text{F}}}{\text{GaDbL} \xrightarrow{f} \text{F}} \quad (\text{D.1})$$

(R2,R3)

$$\frac{\frac{\text{GaaDL} \xrightarrow{f} \text{F}}{\text{GaaDbL} \xrightarrow{f} \text{F}}}{\text{GaDbL} \xrightarrow{f} \text{F}} \quad \supset= \quad \frac{\frac{\text{GaaDL} \xrightarrow{f} \text{F}}{\text{GaDL} \xrightarrow{f} \text{F}}}{\text{GaDbL} \xrightarrow{f} \text{F}} \quad (\text{D.2.1})$$

$$\frac{\frac{\text{GDaaL} \xrightarrow{f} \text{F}}{\text{GbDaaL} \xrightarrow{f} \text{F}}}{\text{GbDaL} \xrightarrow{f} \text{F}} \quad \supset= \quad \frac{\frac{\text{GDaaL} \xrightarrow{f} \text{F}}{\text{GDaL} \xrightarrow{f} \text{F}}}{\text{GbDaL} \xrightarrow{f} \text{F}} \quad (\text{D.2.2})$$

$$\frac{\frac{\text{GaD} \xrightarrow{f} \text{F}}{\text{GabD} \xrightarrow{f} \text{F}}}{\text{GaD} \xrightarrow{f} \text{F}} \quad \supset= \quad \text{GaD} \xrightarrow{f} \text{F}, \text{ with } a = b. \quad (\text{D.2.3})$$

$$\frac{\frac{\text{GaD} \xrightarrow{f} \text{F}}{\text{GbaD} \xrightarrow{f} \text{F}}}{\text{GaD} \xrightarrow{f} \text{F}} \quad \supset= \quad \text{GaD} \xrightarrow{f} \text{F}, \text{ with } a = b. \quad (\text{D.2.4})$$

(R2,R4)

$$\frac{\frac{\text{GabDL} \xrightarrow{f} \text{F}}{\text{GbaDL} \xrightarrow{f} \text{F}}}{\text{GbaDcL} \xrightarrow{f} \text{F}} \quad \supset= \quad \frac{\frac{\text{GabDL} \xrightarrow{f} \text{F}}{\text{GabDcL} \xrightarrow{f} \text{F}}}{\text{GbaDcL} \xrightarrow{f} \text{F}} \quad (\text{D.3.1})$$

$$\frac{\frac{\text{GDabL} \xrightarrow{f} \text{F}}{\text{GDbal} \xrightarrow{f} \text{F}}}{\text{GcDbal} \xrightarrow{f} \text{F}} \quad \supset= \quad \frac{\frac{\text{GDabL} \xrightarrow{f} \text{F}}{\text{GcDabL} \xrightarrow{f} \text{F}}}{\text{GcDbal} \xrightarrow{f} \text{F}} \quad (\text{D.3.2})$$

$$\frac{\frac{\text{GaD} \xrightarrow{f} \text{F}}{\text{GabD} \xrightarrow{f} \text{F}}}{\text{GbaD} \xrightarrow{f} \text{F}} \quad \supset= \quad \frac{\text{GaD} \xrightarrow{f} \text{F}}{\text{GbaD} \xrightarrow{f} \text{F}} \quad (\text{D.3.3})$$



$$\frac{\frac{\frac{GaD \xrightarrow{f} F}{GbaD \rightarrow F}}{GabD \rightarrow F}}{GaD \xrightarrow{f} F} \quad \gg \quad \frac{GaD \xrightarrow{f} F}{GabD \rightarrow F} \quad (D.3.4)$$

(R2,R5)

$$\frac{\frac{\frac{GD \xrightarrow{f} FE}{GaD \rightarrow FE}}{GaD \rightarrow FbE}}{GD \xrightarrow{f} FE} \quad \gg \quad \frac{GD \xrightarrow{f} FE}{GaD \rightarrow FbE} \quad (D.4)$$

(R2,R10)

$$\frac{\frac{\frac{GD \xrightarrow{f} FaE}{GcD \rightarrow FaE} \quad \frac{GD \xrightarrow{g} FbE}{GcD \rightarrow FbE}}{GcD \rightarrow Fa \uparrow bE}}{GD \xrightarrow{f} FaE} \quad \frac{GD \xrightarrow{g} FbE}{GcD \rightarrow Fa \uparrow bE} \quad \gg \quad \frac{GD \xrightarrow{f} FaE} \quad \frac{GD \xrightarrow{g} FbE}{GcD \rightarrow Fa \uparrow bE} \quad (D.7)$$

(R2,R11)

$$\frac{\frac{\frac{GDabL \xrightarrow{f} F}{GcDabL \rightarrow F}}{GcDa \uparrow bL \rightarrow F}}{GDabL \xrightarrow{f} F} \quad \gg \quad \frac{GDabL \xrightarrow{f} F}{GcDa \uparrow bL \rightarrow F} \quad (D.8.1)$$

$$\frac{\frac{\frac{GabDL \xrightarrow{g} F}{GabDcL \rightarrow F}}{Ga \uparrow bDcL \rightarrow F}}{GabDL \xrightarrow{g} F} \quad \gg \quad \frac{GabDL \xrightarrow{g} F}{Ga \uparrow bDcL \rightarrow F} \quad (D.8.2)$$

(R2,R12)

$$\frac{\frac{\frac{GDaL \xrightarrow{f} F}{GcDaL \rightarrow F} \quad \frac{GDbL \xrightarrow{g} F}{GcDbL \rightarrow F}}{GcDa \vee bL \rightarrow F}}{GDaL \xrightarrow{f} F} \quad \frac{GDbL \xrightarrow{g} F}{GcDa \vee bL \rightarrow F} \quad \gg \quad \frac{GDaL \xrightarrow{f} F} \quad \frac{GDbL \xrightarrow{g} F}{GcDa \vee bL \rightarrow F} \quad (D.9.1)$$

$$\frac{\frac{\frac{GaDL \xrightarrow{f} F}{GaDcL \rightarrow F} \quad \frac{GbDL \xrightarrow{g} F}{GbDcL \rightarrow F}}{Ga \vee bDcL \rightarrow F}}{GaDL \xrightarrow{f} F} \quad \frac{GbDL \xrightarrow{g} F}{Ga \vee bDcL \rightarrow F} \quad \gg \quad \frac{GaDL \xrightarrow{f} F} \quad \frac{GbDL \xrightarrow{g} F}{Ga \vee bDcL \rightarrow F} \quad (D.9.2)$$

(R2,R13)

$$\frac{\frac{\frac{GD \xrightarrow{f} FabE}{GcD \rightarrow FabE}}{GcD \rightarrow Fa \vee bE}}{GD \xrightarrow{f} FabE} \quad \gg \quad \frac{GD \xrightarrow{f} FabE}{GcD \rightarrow Fa \vee bE} \quad (D.10)$$

(R2,R14)

$$\frac{\frac{\frac{GD \xrightarrow{f} a}{GcD \rightarrow a} \quad \frac{LbZ \xrightarrow{g} F}{LbZ \rightarrow F}}{LGcDa \Rightarrow bZ \rightarrow F}}{GD \xrightarrow{f} a} \quad \frac{LbZ \xrightarrow{g} F}{LbZ \rightarrow F} \quad \gg \quad \frac{GD \xrightarrow{f} a} \quad \frac{LbZ \xrightarrow{g} F}{LbZ \rightarrow F} \quad (D.11.1)$$

$$\frac{\frac{\frac{G \xrightarrow{h} a}{DcLGa \Rightarrow bZ \rightarrow F} \quad \frac{DLbZ \xrightarrow{k} F}{DcLbZ \rightarrow F}}{DcLGa \Rightarrow bZ \rightarrow F}}{G \xrightarrow{h} a} \quad \frac{DLbZ \xrightarrow{k} F}{DcLbZ \rightarrow F} \quad \gg \quad \frac{G \xrightarrow{h} a} \quad \frac{DLbZ \xrightarrow{k} F}{DcLbZ \rightarrow F} \quad (D.11.2)$$

$$\frac{\frac{G \xrightarrow{h} a \quad \frac{DbLZ \xrightarrow{m} F}{DbLcZ \xrightarrow{m} F}}{DGa \Rightarrow bLcZ \xrightarrow{m} F}}{DGa \Rightarrow bLcZ \xrightarrow{m} F} \succ \equiv \frac{\frac{G \xrightarrow{h} a \quad \frac{DbLZ \xrightarrow{m} F}{DGa \Rightarrow bLZ \xrightarrow{m} F}}{DGa \Rightarrow bLcZ \xrightarrow{m} F}}{DGa \Rightarrow bLcZ \xrightarrow{m} F} \quad (D.11.3)$$

$$\frac{\frac{G \xrightarrow{h} a \quad \frac{DL \xrightarrow{p} F}{DbL \xrightarrow{p} F}}{DGa \Rightarrow bL \xrightarrow{p} F}}{DGa \Rightarrow bL \xrightarrow{p} F} \succ \equiv \frac{\frac{DL \xrightarrow{p} F}{Da \Rightarrow bL \xrightarrow{p} F}}{DGa \Rightarrow bL \xrightarrow{p} F} \quad (D.11.4)$$

(R2, R15)

$$\frac{\frac{GaDL \xrightarrow{f} b}{GaDcL \xrightarrow{f} b}}{GDcL \xrightarrow{f} (a \Rightarrow b)} \succ \equiv \frac{\frac{GaDL \xrightarrow{f} b}{GDL \xrightarrow{f} (a \Rightarrow b)}}{GDcL \xrightarrow{f} (a \Rightarrow b)} \quad (D.12.1)$$

$$\frac{\frac{GDaL \xrightarrow{f} b}{GcDaL \xrightarrow{f} b}}{GcDL \xrightarrow{f} (a \Rightarrow b)} \succ \equiv \frac{\frac{GDaL \xrightarrow{f} b}{GDL \xrightarrow{f} (a \Rightarrow b)}}{GcDL \xrightarrow{f} (a \Rightarrow b)} \quad (D.12.2)$$

(R3, R3)

$$\frac{\frac{GacDbbL \xrightarrow{f} F}{GaDbbL \xrightarrow{f} F}}{GaDbL \xrightarrow{f} F} \succ \equiv \frac{\frac{GaaDbbL \xrightarrow{f} F}{GaaDbL \xrightarrow{f} F}}{GaDbL \xrightarrow{f} F} \quad (D.15.1)$$

(R3, R4)

$$\frac{\frac{GaaDbcL \xrightarrow{f} F}{GaaDcbL \xrightarrow{f} F}}{GaDcbL \xrightarrow{f} F} \succ \equiv \frac{\frac{GaaDbcL \xrightarrow{f} F}{GaDbcL \xrightarrow{f} F}}{GaDcbL \xrightarrow{f} F} \quad (D.16.1)$$

$$\frac{\frac{GbcDaaL \xrightarrow{g} F}{GcbDaaL \xrightarrow{g} F}}{GcbDaL \xrightarrow{g} F} \succ \equiv \frac{\frac{GbcDaaL \xrightarrow{g} F}{GbcDaL \xrightarrow{g} F}}{GcbDaL \xrightarrow{g} F} \quad (D.16.2)$$

$$\frac{\frac{GabD \xrightarrow{h} F}{GbaD \xrightarrow{h} F}}{GbD \xrightarrow{h} F} \succ \equiv \frac{\frac{GabD \xrightarrow{h} F}{GbD \xrightarrow{h} F}}{GbD \xrightarrow{h} F} \quad (D.16.3)$$

(R3, R5)

$$\frac{\frac{GaaD \xrightarrow{f} FE}{GaD \xrightarrow{f} FE}}{GaD \xrightarrow{f} FbE} \succ \equiv \frac{\frac{GaaD \xrightarrow{f} FE}{GaaD \xrightarrow{f} FbE}}{GaD \xrightarrow{f} FbE} \quad (D.17)$$

(R3, R10)

$$\frac{\frac{GccD \xrightarrow{f} FaE \quad GccD \xrightarrow{g} FbE}{GcD \xrightarrow{f} FaE \quad GcD \xrightarrow{g} FbE}}{GcD \xrightarrow{f} Fa \uparrow bE} \succ \equiv \frac{\frac{GccD \xrightarrow{f} FaE \quad GccD \xrightarrow{g} FbE}{GccD \xrightarrow{f} Fa \uparrow bE}}{GcD \xrightarrow{f} Fa \uparrow bE} \quad (D.18)$$

(R3,R11)

$$\frac{\frac{\frac{GaaDbcL \xrightarrow{f} F}{GaaDb \uparrow cL \rightarrow F}}{GaDb \uparrow cL \rightarrow F} \Rightarrow \frac{\frac{GaaDbcL \xrightarrow{f} F}{GaDbcL \rightarrow F}}{GaDb \uparrow cL \rightarrow F}}{(D.19.1)}$$

$$\frac{\frac{\frac{GbcDaaL \xrightarrow{g} F}{Gb \uparrow cDaaL \rightarrow F}}{Gb \uparrow cDaL \rightarrow F} \Rightarrow \frac{\frac{GbcDaaL \xrightarrow{g} F}{GbcDaL \rightarrow F}}{Gb \uparrow cDaL \rightarrow F}}{(D.19.2)}$$

$$\frac{\frac{\frac{\frac{GaaaaD \xrightarrow{h} F}{Gaa(a \uparrow a)D \rightarrow F}}{G(a \uparrow a)(a \uparrow a)D \rightarrow F}}{Ga \uparrow aD \rightarrow F} \Rightarrow \frac{\frac{\frac{Gaa.aa.D \xrightarrow{h} F}{G.aa..a.D \rightarrow F}}{GaaD \rightarrow F}}{Ga \uparrow aD \rightarrow F}}{(D.19.3)}$$

(R3,R12)

$$\frac{\frac{\frac{GccDaL \xrightarrow{f} F}{GcDaL \rightarrow F} \quad \frac{GccDbL \xrightarrow{g} F}{GcDbL \rightarrow F}}{GcDa \vee bL \rightarrow F} \Rightarrow \frac{\frac{GccDaL \xrightarrow{f} F \quad GccDbL \xrightarrow{g} F}{GccDa \vee bL \rightarrow F}}{GcDa \vee bL \rightarrow F}}{(D.20.1)}$$

$$\frac{\frac{\frac{GaDccL \xrightarrow{f} F}{GaDcL \rightarrow F} \quad \frac{GbDccL \xrightarrow{g} F}{GbDcL \rightarrow F}}{Ga \vee bDcL \rightarrow F} \Rightarrow \frac{\frac{GaDccL \xrightarrow{f} F \quad GbDccL \xrightarrow{g} F}{Ga \vee bDccL \rightarrow F}}{Ga \vee bDcL \rightarrow F}}{(D.20.2)}$$

(R3,R13)

$$\frac{\frac{\frac{GccD \xrightarrow{f} FabE}{GcD \rightarrow FabE}}{GcD \rightarrow Fa \vee bE} \Rightarrow \frac{\frac{GccD \xrightarrow{f} FabE}{GccD \rightarrow Fa \vee bE}}{GcD \rightarrow Fa \vee bE}}{(D.21)}$$

(R3,R14)

$$\frac{\frac{\frac{GccD \xrightarrow{f} a}{GcD \rightarrow a} \quad \frac{LbZ \xrightarrow{g} F}{LbZ \rightarrow F}}{LGcDa \Rightarrow bZ \rightarrow F} \Rightarrow \frac{\frac{GccD \xrightarrow{f} a \quad LbZ \xrightarrow{g} F}{LGcDa \Rightarrow bZ \rightarrow F}}{LGcDa \Rightarrow bZ \rightarrow F}}{(D.22.1)}$$

$$\frac{\frac{\frac{DccLbZ \xrightarrow{k} F}{DcLbZ \rightarrow F}}{DcLGa \Rightarrow bZ \rightarrow F} \Rightarrow \frac{\frac{G \xrightarrow{h} a \quad DccLbZ \xrightarrow{k} F}{DccLGa \Rightarrow bZ \rightarrow F}}{DcLGa \Rightarrow bZ \rightarrow F}}{(D.22.2)}$$

$$\frac{\frac{\frac{DbLccZ \xrightarrow{m} F}{DbLcZ \rightarrow F}}{DGa \Rightarrow bLcZ \rightarrow F} \Rightarrow \frac{\frac{G \xrightarrow{h} a \quad DbLccZ \xrightarrow{m} F}{DGa \Rightarrow bLccZ \rightarrow F}}{DGa \Rightarrow bLcZ \rightarrow F}}{(D.22.3)}$$

(R3,R15)

$$\frac{\frac{\frac{GccDaL \xrightarrow{f} b}{GcDaL \rightarrow b}}{GcDL \rightarrow (a \Rightarrow b)} \Rightarrow \frac{\frac{GccDaL \xrightarrow{f} b}{GccDL \rightarrow (a \Rightarrow b)}}{GcDL \rightarrow (a \Rightarrow b)}}{(D.23.1)}$$

$$\frac{\frac{\frac{GaDccL \xrightarrow{f} b}{GaDcL \rightarrow b}}{GDcL \rightarrow (a \Rightarrow b)} \Rightarrow \frac{\frac{\frac{GaDccL \xrightarrow{f} b}{GDccL \rightarrow (a \Rightarrow b)}}{GDcL \rightarrow (a \Rightarrow b)}}{(D.23.2)}$$

(R4,R5)

$$\frac{\frac{\frac{GabD \xrightarrow{f} FE}{GbaD \rightarrow FE}}{GbaD \rightarrow FcE} \Rightarrow \frac{\frac{\frac{GabD \xrightarrow{f} FE}{GabD \rightarrow FcE}}{GbaD \rightarrow FcE}}{(D.25)}$$

(R4,R10)

$$\frac{\frac{\frac{GcdD \xrightarrow{f} FaE}{GdcD \rightarrow FaE} \quad \frac{GcdD \xrightarrow{g} FbE}{GdcD \rightarrow FbE}}{GdcD \rightarrow Fa \uparrow bE} \Rightarrow \frac{\frac{\frac{GcdD \xrightarrow{f} FaE} \quad \frac{GcdD \xrightarrow{g} FbE}}{GcdD \rightarrow Fa \uparrow bE}}{GdcD \rightarrow Fa \uparrow bE}}{(D.28)}$$

(R4,R11)

$$\frac{\frac{\frac{\frac{GabDcdL \xrightarrow{f} F}{GabDc \uparrow dL \rightarrow F}}{GbaDc \uparrow dL \rightarrow F} \Rightarrow \frac{\frac{\frac{GabDcdL \xrightarrow{f} F}{GbaDcdL \rightarrow F}}{GbaDc \uparrow dL \rightarrow F}}{(D.29.1)}$$

$$\frac{\frac{\frac{\frac{GabDcdL \xrightarrow{f} F}{Ga \uparrow bDcdL \rightarrow F}}{Ga \uparrow bDdcL \rightarrow F} \Rightarrow \frac{\frac{\frac{GabDcdL \xrightarrow{f} F}{GabDdcL \rightarrow F}}{Ga \uparrow bDdcL \rightarrow F}}{(D.29.2)}$$

$$\frac{\frac{\frac{\frac{GabcD \xrightarrow{g} F}{G(a \uparrow b)cD \rightarrow F}}{Gc(a \uparrow b)D \rightarrow F} \Rightarrow \frac{\frac{\frac{\frac{GabcD \xrightarrow{g} F}{GacbD \rightarrow F}}{GcabD \rightarrow F}}{Gc(a \uparrow b)D \rightarrow F}}{(D.29.3)}$$

$$\frac{\frac{\frac{\frac{GabcD \xrightarrow{g} F}{Ga(b \uparrow c)D \rightarrow F}}{G(b \uparrow c)aD \rightarrow F} \Rightarrow \frac{\frac{\frac{\frac{GabcD \xrightarrow{g} F}{GbacD \rightarrow F}}{GbcaD \rightarrow F}}{G(b \uparrow c)aD \rightarrow F}}{(D.29.4)}$$

(R4,R12)

$$\frac{\frac{\frac{\frac{GcdDaL \xrightarrow{f} F}{GdcDaL \rightarrow F} \quad \frac{GcdDbL \xrightarrow{g} F}{GdcDbL \rightarrow F}}{GdcDa \vee bL \rightarrow F} \Rightarrow \frac{\frac{\frac{\frac{GcdDaL \xrightarrow{f} F}{GcdDa \vee bL \rightarrow F}}{GdcDa \vee bL \rightarrow F}}{(D.30.1)}$$

$$\frac{\frac{\frac{\frac{GaDcdL \xrightarrow{h} F}{GaDdcL \rightarrow F} \quad \frac{GbDcdL \xrightarrow{k} F}{GbDdcL \rightarrow F}}{Ga \vee bDdcL \rightarrow F} \Rightarrow \frac{\frac{\frac{\frac{GaDcdL \xrightarrow{h} F}{GbDcdL \xrightarrow{k} F}}{Ga \vee bDcdL \rightarrow F}}{Ga \vee bDdcL \rightarrow F}}{(D.30.2)}$$

(R4,R13)

$$\frac{\frac{\frac{GcdD \xrightarrow{f} FabE}{GdcD \xrightarrow{f} FabE}}{GcdD \xrightarrow{f} Fa \vee bE}}{GcdD \xrightarrow{f} Fa \vee bE}}{GcdD \xrightarrow{f} FabE} \supset_{=} \frac{\frac{\frac{GcdD \xrightarrow{f} FabE}{GcdD \xrightarrow{f} Fa \vee bE}}{GcdD \xrightarrow{f} Fa \vee bE}}{GcdD \xrightarrow{f} Fa \vee bE}}{GcdD \xrightarrow{f} FabE} \quad (D.31)$$

(R4,R14)

$$\frac{\frac{\frac{GcdD \xrightarrow{f} a}{GdcD \xrightarrow{f} a} \quad \frac{LbZ \xrightarrow{g} F}{LGdcDa \Rightarrow bZ \rightarrow F}}{LGdcDa \Rightarrow bZ \rightarrow F}}{LGdcDa \Rightarrow bZ \rightarrow F} \supset_{=} \frac{\frac{\frac{GcdD \xrightarrow{f} a}{LGcdDa \Rightarrow bZ \rightarrow F} \quad \frac{LbZ \xrightarrow{g} F}{LGdcDa \Rightarrow bZ \rightarrow F}}{LGcdDa \Rightarrow bZ \rightarrow F}}{LGcdDa \Rightarrow bZ \rightarrow F} \quad (D.32.1)$$

$$\frac{\frac{\frac{G \xrightarrow{h} a}{DdcLbZ \xrightarrow{k} F} \quad \frac{DcdLbZ \xrightarrow{k} F}{DdcLbZ \xrightarrow{k} F}}{DdcLbZ \xrightarrow{k} F}}{DdcLbZ \xrightarrow{k} F} \supset_{=} \frac{\frac{\frac{G \xrightarrow{h} a}{DcdLGa \Rightarrow bZ \rightarrow F} \quad \frac{DcdLbZ \xrightarrow{k} F}{DcdLGa \Rightarrow bZ \rightarrow F}}{DcdLGa \Rightarrow bZ \rightarrow F}}{DcdLGa \Rightarrow bZ \rightarrow F} \quad (D.32.2)$$

$$\frac{\frac{\frac{G \xrightarrow{h} a}{DbLcdZ \xrightarrow{m} F} \quad \frac{DbLdcZ \xrightarrow{m} F}{DGa \Rightarrow bLdcZ \rightarrow F}}{DGa \Rightarrow bLdcZ \rightarrow F}}{DGa \Rightarrow bLdcZ \rightarrow F} \supset_{=} \frac{\frac{\frac{G \xrightarrow{h} a}{DGa \Rightarrow bLcdZ \rightarrow F} \quad \frac{DbLcdZ \xrightarrow{m} F}{DGa \Rightarrow bLcdZ \rightarrow F}}{DGa \Rightarrow bLcdZ \rightarrow F}}{DGa \Rightarrow bLcdZ \rightarrow F} \quad (D.32.3)$$

$$\frac{\frac{\frac{G \xrightarrow{h} a}{DcbL \xrightarrow{p} F} \quad \frac{DcbL \xrightarrow{p} F}{DG(a \Rightarrow b)cL \rightarrow F}}{DG(a \Rightarrow b)cL \rightarrow F}}{DG(a \Rightarrow b)cL \rightarrow F} \supset_{=} \frac{\frac{\frac{G \xrightarrow{h} a}{DcGa \Rightarrow bL \rightarrow F} \quad \frac{DcbL \xrightarrow{p} F}{DcGa \Rightarrow bL \rightarrow F}}{DcGa \Rightarrow bL \rightarrow F}}{DcGa \Rightarrow bL \rightarrow F} \quad (D.32.4)$$

$$\frac{\frac{\frac{G \xrightarrow{h} a}{DbcL \xrightarrow{q} F} \quad \frac{DbcL \xrightarrow{q} F}{DcG(a \Rightarrow b)L \rightarrow F}}{DcG(a \Rightarrow b)L \rightarrow F}}{DcG(a \Rightarrow b)L \rightarrow F} \supset_{=} \frac{\frac{\frac{G \xrightarrow{h} a}{DG(a \Rightarrow b)cL \rightarrow F} \quad \frac{DbcL \xrightarrow{q} F}{DG(a \Rightarrow b)cL \rightarrow F}}{DG(a \Rightarrow b)cL \rightarrow F}}{DG(a \Rightarrow b)cL \rightarrow F} \quad (D.32.5)$$

(R4,R15)

$$\frac{\frac{\frac{GaDcdL \xrightarrow{f} b}{GaDdcL \xrightarrow{f} b}}{GDdcL \rightarrow (a \Rightarrow b)}}{GDdcL \rightarrow (a \Rightarrow b)} \supset_{=} \frac{\frac{\frac{GaDcdL \xrightarrow{f} b}{GDcdL \rightarrow (a \Rightarrow b)}}{GDdcL \rightarrow (a \Rightarrow b)}}{GDdcL \rightarrow (a \Rightarrow b)} \quad (D.33.1)$$

$$\frac{\frac{\frac{GcdDaL \xrightarrow{g} b}{GdcDaL \xrightarrow{g} b}}{GdcDL \rightarrow (a \Rightarrow b)}}{GdcDL \rightarrow (a \Rightarrow b)} \supset_{=} \frac{\frac{\frac{GcdDaL \xrightarrow{g} b}{GcdDL \rightarrow (a \Rightarrow b)}}{GdcDL \rightarrow (a \Rightarrow b)}}{GdcDL \rightarrow (a \Rightarrow b)} \quad (D.33.2)$$

$$\frac{\frac{\frac{GcaD \xrightarrow{h} b}{GacD \xrightarrow{h} b}}{GcD \rightarrow a \Rightarrow b}}{GcD \rightarrow a \Rightarrow b} \supset_{=} \frac{\frac{\frac{GcaD \xrightarrow{h} b}{GcD \rightarrow a \Rightarrow b}}{GcD \rightarrow a \Rightarrow b}}{GcD \rightarrow a \Rightarrow b} \quad (D.33.3)$$

$$\frac{\frac{\frac{GacD \xrightarrow{k} b}{GcaD \xrightarrow{k} b}}{GcD \rightarrow a \Rightarrow b}}{GcD \rightarrow a \Rightarrow b} \supset_{=} \frac{\frac{\frac{GacD \xrightarrow{k} b}{GcD \rightarrow a \Rightarrow b}}{GcD \rightarrow a \Rightarrow b}}{GcD \rightarrow a \Rightarrow b} \quad (D.33.4)$$

(R5,R5)

$$\frac{\frac{G \xrightarrow{f} FEH}{G \rightarrow FaEH}}{G \rightarrow FaEbH} \geq \frac{\frac{G \xrightarrow{f} FEH}{G \rightarrow FEbH}}{G \rightarrow FaEbH} \quad (D.34)$$

(R5,R10)

$$\frac{\frac{G \xrightarrow{f} FEaH}{G \rightarrow FEa \uparrow bH} \quad G \xrightarrow{g} FEbH}{G \rightarrow FcEa \uparrow bH} \geq \frac{\frac{G \xrightarrow{f} FEaH}{G \rightarrow FcEaH} \quad G \xrightarrow{g} FEbH}{G \rightarrow FcEa \uparrow bH} \quad (D.35.1)$$

$$\frac{\frac{G \xrightarrow{f} FaEH}{G \rightarrow Fa \uparrow bEH} \quad G \xrightarrow{g} FbEH}{G \rightarrow Fa \uparrow bEcH} \geq \frac{\frac{G \xrightarrow{f} FaEH}{G \rightarrow FaEcH} \quad G \xrightarrow{g} FbEH}{G \rightarrow Fa \uparrow bEcH} \quad (D.35.2)$$

$$\frac{\frac{G \xrightarrow{f} Fa}{G \rightarrow Fab} \quad \frac{G \xrightarrow{g} Fc}{G \rightarrow Fac}}{G \rightarrow Fa(b \uparrow c)} \geq \frac{\frac{G \xrightarrow{f} Fa}{G \rightarrow Fab} \quad \frac{G \xrightarrow{f} Fa}{G \rightarrow Fac}}{G \rightarrow Fa(b \uparrow c)} \quad (D.35.3)$$

$$\frac{\frac{G \xrightarrow{f} dE}{G \rightarrow bdE} \quad \frac{G \xrightarrow{g} cE}{G \rightarrow cdE}}{G \rightarrow (b \uparrow c)dE} \geq \frac{\frac{G \xrightarrow{f} dE}{G \rightarrow bdE} \quad \frac{G \xrightarrow{f} dE}{G \rightarrow cdE}}{G \rightarrow (b \uparrow c)dE} \quad (D.35.3)$$

$$\frac{\frac{G \xrightarrow{f} FaE}{G \rightarrow FadE} \quad \frac{G \xrightarrow{h} FdE}{G \rightarrow FadE}}{G \rightarrow Fa(b \uparrow c)dE} \geq \frac{\frac{G \xrightarrow{f} FaE}{G \rightarrow FadE} \quad \frac{G \xrightarrow{f} FaE}{G \rightarrow FadE}}{G \rightarrow Fa(b \uparrow c)dE} \quad (D.35.5)$$

$$\frac{\frac{G \xrightarrow{f} FaE}{G \rightarrow FadE} \quad \frac{G \xrightarrow{h} FcE}{G \rightarrow FcdE}}{G \rightarrow Fa(b \uparrow c)dE} \geq \frac{\frac{G \xrightarrow{f} FaE}{G \rightarrow FadE} \quad \frac{G \xrightarrow{f} FaE}{G \rightarrow FadE}}{G \rightarrow Fa(b \uparrow c)dE} \quad (D.35.6)$$

$$\frac{\frac{G \xrightarrow{f} FaE}{G \rightarrow FadE} \quad \frac{G \xrightarrow{k} FdE}{G \rightarrow FcdE}}{G \rightarrow Fa(b \uparrow c)dE} \geq \frac{\frac{G \xrightarrow{k} FdE}{G \rightarrow FbdE} \quad \frac{G \xrightarrow{k} FdE}{G \rightarrow FcdE}}{G \rightarrow Fa(b \uparrow c)dE} \quad (D.35.7)$$

$$\frac{\frac{G \xrightarrow{m} FbE}{G \rightarrow FbdE} \quad \frac{G \xrightarrow{f} FaE}{G \rightarrow FadE}}{G \rightarrow Fa(b \uparrow c)dE} \geq \frac{\frac{G \xrightarrow{f} FaE}{G \rightarrow FadE} \quad \frac{G \xrightarrow{f} FaE}{G \rightarrow FadE}}{G \rightarrow Fa(b \uparrow c)dE} \quad (D.35.8)$$



(R10,R10)

$$\begin{array}{c}
 \begin{array}{ccc}
 \frac{G \xrightarrow{f} FaEaH}{G \rightarrow Fa \uparrow aEaH} & \frac{G \xrightarrow{g} FaEaH}{G \rightarrow Fa \uparrow aEaH} & \frac{G \xrightarrow{h} FaEaH}{G \rightarrow Fa \uparrow aEaH} \quad \frac{G \xrightarrow{k} FaEaH}{G \rightarrow Fa \uparrow aEaH} \\
 \hline
 G \rightarrow Fa \uparrow aEa \uparrow aH
 \end{array} \\
 \\
 \begin{array}{ccc}
 \frac{G \xrightarrow{f} FaEaH}{G \rightarrow FaEa \uparrow aH} & \frac{G \xrightarrow{g} FaEaH}{G \rightarrow FaEa \uparrow aH} & \frac{G \xrightarrow{h} FaEaH}{G \rightarrow FaEa \uparrow aH} \quad \frac{G \xrightarrow{k} FaEaH}{G \rightarrow FaEa \uparrow aH} \\
 \hline
 G \rightarrow Fa \uparrow aEa \uparrow aH
 \end{array} \\
 \Rightarrow
 \end{array}
 \tag{D.48}$$

(R10,R11)

$$\begin{array}{c}
 \begin{array}{ccc}
 \frac{GabD \xrightarrow{f} FcE}{Ga \uparrow bD \rightarrow FcE} & \frac{GabD \xrightarrow{g} FdE}{Ga \uparrow bD \rightarrow FdE} & \frac{GabD \xrightarrow{f} FcE}{GabD \rightarrow Fc \uparrow dE} \quad \frac{GabD \xrightarrow{g} FdE}{Ga \uparrow bD \rightarrow Fc \uparrow dE} \\
 \hline
 Ga \uparrow bD \rightarrow Fc \uparrow dE
 \end{array} \\
 \Rightarrow
 \end{array}
 \tag{D.49}$$

(R10,R12)

$$\begin{array}{c}
 \begin{array}{ccc}
 \frac{GaD \xrightarrow{f} FcE}{Ga \vee bD \rightarrow FcE} & \frac{GbD \xrightarrow{g} FcE}{Ga \vee bD \rightarrow FcE} & \frac{GaD \xrightarrow{h} FdE}{Ga \vee bD \rightarrow FdE} \quad \frac{GbD \xrightarrow{k} FdE}{Ga \vee bD \rightarrow FdE} \\
 \hline
 Ga \vee bD \rightarrow Fc \uparrow dE
 \end{array} \\
 \\
 \begin{array}{ccc}
 \frac{GaD \xrightarrow{f} FcE}{GaD \rightarrow Fc \uparrow dE} & \frac{GaD \xrightarrow{h} FdE}{GaD \rightarrow Fc \uparrow dE} & \frac{GbD \xrightarrow{g} FcE}{GbD \rightarrow Fc \uparrow dE} \quad \frac{GbD \xrightarrow{k} FdE}{GbD \rightarrow Fc \uparrow dE} \\
 \hline
 Ga \vee bD \rightarrow Fc \uparrow dE
 \end{array} \\
 \Rightarrow
 \end{array}
 \tag{D.50}$$

(R10,R13)

$$\begin{array}{c}
 \begin{array}{ccc}
 \frac{G \xrightarrow{f} FcdEaH}{G \rightarrow Fc \vee dEa \uparrow bH} & \frac{G \xrightarrow{g} FcdEbH}{G \rightarrow Fc \vee dEbH} & \frac{G \xrightarrow{f} FcdEaH}{G \rightarrow Fc \vee dEa \uparrow bH} \quad \frac{G \xrightarrow{g} FcdEbH}{G \rightarrow Fc \vee dEbH} \\
 \hline
 G \rightarrow Fc \vee dEa \uparrow bH
 \end{array} \\
 \Rightarrow
 \end{array}
 \tag{D.51.1}$$

$$\begin{array}{c}
 \begin{array}{ccc}
 \frac{G \xrightarrow{h} FaEcdH}{G \rightarrow Fa \uparrow bEcdH} & \frac{G \xrightarrow{k} FbEcdH}{G \rightarrow FbEc \vee dH} & \frac{G \xrightarrow{h} FaEcdH}{G \rightarrow Fa \uparrow bEc \vee dH} \quad \frac{G \xrightarrow{k} FbEcdH}{G \rightarrow FbEc \vee dH} \\
 \hline
 G \rightarrow Fa \uparrow bEc \vee dH
 \end{array} \\
 \Rightarrow
 \end{array}
 \tag{D.51.2}$$

(R10,R14)

$$\begin{array}{c}
 \begin{array}{ccc}
 \frac{G \xrightarrow{f} a \quad DbL \xrightarrow{g} FcE}{DGa \Rightarrow bL \rightarrow FcE} & \frac{G \xrightarrow{f} a \quad DbL \xrightarrow{h} FdE}{DGa \Rightarrow bL \rightarrow FdE} & \frac{DbL \xrightarrow{g} FcE \quad DbL \xrightarrow{h} FdE}{DbL \rightarrow Fc \uparrow dE} \\
 \hline
 DGa \Rightarrow bL \rightarrow Fc \uparrow dE
 \end{array} \\
 \Rightarrow
 \end{array}
 \tag{D.52}$$

(R11,R11)

$$\begin{array}{c}
 \begin{array}{ccc}
 \frac{GabDcdL \xrightarrow{f} F}{Ga \uparrow bDcdL \rightarrow F} & \frac{GabDcdL \xrightarrow{f} F}{GabDc \uparrow dL \rightarrow F} & \frac{GabDcdL \xrightarrow{f} F}{Ga \uparrow bDc \uparrow dL \rightarrow F} \\
 \hline
 Ga \uparrow bDc \uparrow dL \rightarrow F
 \end{array} \\
 \Rightarrow
 \end{array}
 \tag{D.53}$$



(R11,R12)

$$\frac{\frac{\frac{GcdDaL \xrightarrow{f} F}{Gc \uparrow dDaL \rightarrow F} \quad \frac{GcdDbL \xrightarrow{g} F}{Gc \uparrow dDbL \rightarrow F}}{Gc \uparrow dDa \vee bL \rightarrow F} \Rightarrow \quad \frac{\frac{GcdDaL \xrightarrow{f} F}{Gc \uparrow dDa \vee bL \rightarrow F} \quad \frac{GcdDbL \xrightarrow{g} F}{Gc \uparrow dDa \vee bL \rightarrow F}}{Gc \uparrow dDa \vee bL \rightarrow F} \quad (D.54.1)$$

$$\frac{\frac{\frac{GaDcdL \xrightarrow{h} F}{GaDc \uparrow dL \rightarrow F} \quad \frac{GbDcdL \xrightarrow{k} F}{GbDc \uparrow dL \rightarrow F}}{Ga \vee bDc \uparrow dL \rightarrow F} \Rightarrow \quad \frac{\frac{GaDcdL \xrightarrow{h} F}{Ga \vee bDcdL \rightarrow F} \quad \frac{GbDcdL \xrightarrow{k} F}{Ga \vee bDc \uparrow dL \rightarrow F}}{Ga \vee bDc \uparrow dL \rightarrow F} \quad (D.54.2)$$

(R11,R13)

$$\frac{\frac{\frac{GabD \xrightarrow{f} FcdE}{Ga \uparrow bD \rightarrow FcdE}}{Ga \uparrow bD \rightarrow Fc \vee dE} \Rightarrow \quad \frac{\frac{GabD \xrightarrow{f} FcdE}{GabD \rightarrow Fc \vee dE}}{Ga \uparrow bD \rightarrow Fc \vee dE} \quad (D.55)$$

(R11,R14)

$$\frac{\frac{\frac{GcdD \xrightarrow{f} a}{Gc \uparrow dD \rightarrow a} \quad \frac{LbZ \xrightarrow{g} F}{LbZ \xrightarrow{g} F}}{LGc \uparrow dDa \Rightarrow bZ \rightarrow F} \Rightarrow \quad \frac{\frac{GcdD \xrightarrow{f} a}{LGcdDa \Rightarrow bZ \rightarrow F} \quad \frac{LbZ \xrightarrow{g} F}{LGc \uparrow dDa \Rightarrow bZ \rightarrow F}}{LGc \uparrow dDa \Rightarrow bZ \rightarrow F} \quad (D.56.1)$$

$$\frac{\frac{\frac{G \xrightarrow{h} a}{Dc \uparrow dLg \Rightarrow bZ \rightarrow F} \quad \frac{DcdLbZ \xrightarrow{k} F}{Dc \uparrow dLbZ \rightarrow F}}{Dc \uparrow dLg \Rightarrow bZ \rightarrow F} \Rightarrow \quad \frac{\frac{G \xrightarrow{h} a}{DcdLGa \Rightarrow bZ \rightarrow F} \quad \frac{DcdLbZ \xrightarrow{k} F}{Dc \uparrow dLGa \Rightarrow bZ \rightarrow F}}{Dc \uparrow dLGa \Rightarrow bZ \rightarrow F} \quad (D.56.2)$$

$$\frac{\frac{\frac{G \xrightarrow{h} a}{DGa \Rightarrow bLc \uparrow dZ \rightarrow F} \quad \frac{DbLcdZ \xrightarrow{m} F}{DbLc \uparrow dZ \rightarrow F}}{DGa \Rightarrow bLc \uparrow dZ \rightarrow F} \Rightarrow \quad \frac{\frac{G \xrightarrow{h} a}{DGa \Rightarrow bLcdZ \rightarrow F} \quad \frac{DbLcdZ \xrightarrow{m} F}{DbLc \uparrow dZ \rightarrow F}}{DGa \Rightarrow bLc \uparrow dZ \rightarrow F} \quad (D.56.3)$$

(R11,R15)

$$\frac{\frac{\frac{GaDcdL \xrightarrow{f} b}{GaDc \uparrow dL \rightarrow b}}{GDc \uparrow dL \rightarrow (a \Rightarrow b)} \Rightarrow \quad \frac{\frac{GaDcdL \xrightarrow{f} b}{GDcdL \rightarrow (a \Rightarrow b)}}{GDc \uparrow dL \rightarrow (a \Rightarrow b)} \quad (D.57.1)$$

$$\frac{\frac{\frac{GcdDaL \xrightarrow{g} b}{Gc \uparrow dDaL \rightarrow b}}{Gc \uparrow dDL \rightarrow (a \Rightarrow b)} \Rightarrow \quad \frac{\frac{GcdDaL \xrightarrow{g} b}{GcdDL \rightarrow (a \Rightarrow b)}}{Gc \uparrow dDL \rightarrow (a \Rightarrow b)} \quad (D.57.2)$$

$$\frac{\frac{\frac{\frac{GabDcdL \xrightarrow{h} e}{GabDc \uparrow dL \rightarrow e}}{Ga \uparrow bDc \uparrow dL \rightarrow e}}{GDc \uparrow dL \rightarrow (a \uparrow b) \Rightarrow e} \Rightarrow \quad \frac{\frac{\frac{GabDcdL \xrightarrow{h} e}{Ga \uparrow bDcdL \rightarrow e}}{GDcdL \rightarrow (a \uparrow b) \Rightarrow e}}{GDc \uparrow dL \rightarrow (a \uparrow b) \Rightarrow e} \quad (D.57.3)$$

(R12,R12)

$$\begin{array}{c}
 \frac{\frac{\text{GaDaL} \xrightarrow{f} F \quad \text{GaDaL} \xrightarrow{g} F}{\text{Ga v aDaL} \rightarrow F} \quad \frac{\text{GaDaL} \xrightarrow{h} F \quad \text{GaDaL} \xrightarrow{k} F}{\text{Ga v aDaL} \rightarrow F}}{\text{Ga v aDa v aL} \rightarrow F} \\
 \\
 \frac{\frac{\frac{\text{GaDaL} \xrightarrow{f} F \quad \text{GaDaL} \xrightarrow{h} F}{\text{GaDa v aL} \rightarrow F} \quad \frac{\text{GaDaL} \xrightarrow{g} F \quad \text{GaDaL} \xrightarrow{k} F}{\text{GaDa v aL} \rightarrow F}}{\text{Ga v aDa v aL} \rightarrow F} \\
 \Rightarrow \hspace{15em} \text{(D.58)}
 \end{array}$$

(R12,R13)

$$\begin{array}{c}
 \frac{\frac{\text{GaD} \xrightarrow{f} \text{FcdE} \quad \text{GbD} \xrightarrow{g} \text{FcdE}}{\text{Ga v bD} \rightarrow \text{FcdE}}}{\text{Ga v bD} \rightarrow \text{Fc v dE}} \quad \Rightarrow \quad \frac{\frac{\text{GaD} \xrightarrow{f} \text{FcdE} \quad \text{GbD} \xrightarrow{g} \text{FcdE}}{\text{GaD} \rightarrow \text{Fc v dE}} \quad \frac{\text{GbD} \xrightarrow{g} \text{FcdE}}{\text{GbD} \rightarrow \text{Fc v dE}}}{\text{Ga v bD} \rightarrow \text{Fc v dE}} \\
 \hspace{15em} \text{(D.59)}
 \end{array}$$

(R12,R14)

$$\begin{array}{c}
 \frac{\frac{\text{G} \xrightarrow{f} a \quad \text{DbLcZ} \xrightarrow{g} F}{\text{DGa} \Rightarrow \text{bLcZ} \rightarrow F} \quad \frac{\text{G} \xrightarrow{f} a \quad \text{DbLdZ} \xrightarrow{h} F}{\text{DGa} \Rightarrow \text{bLdZ} \rightarrow F}}{\text{DGa} \Rightarrow \text{bLc v dZ} \rightarrow F} \quad \Rightarrow \quad \frac{\frac{\text{DbLcZ} \xrightarrow{g} F \quad \text{DbLdZ} \xrightarrow{h} F}{\text{G} \xrightarrow{f} a \quad \text{DbLc v dZ} \rightarrow F}}{\text{DGa} \Rightarrow \text{bLc v dZ} \rightarrow F} \\
 \hspace{15em} \text{(D.60.1)}
 \end{array}$$

$$\begin{array}{c}
 \frac{\frac{\text{G} \xrightarrow{f} a \quad \text{DcLbZ} \xrightarrow{g} F}{\text{DcLGa} \Rightarrow \text{bZ} \rightarrow F} \quad \frac{\text{G} \xrightarrow{f} a \quad \text{DdLbL} \xrightarrow{h} F}{\text{DdLGa} \Rightarrow \text{bZ} \rightarrow F}}{\text{Dc v dLGa} \Rightarrow \text{bZ} \rightarrow F} \quad \Rightarrow \quad \frac{\frac{\text{DcLbL} \xrightarrow{g} F \quad \text{DdLbL} \xrightarrow{h} F}{\text{G} \xrightarrow{f} a \quad \text{Dc v dLbZ} \rightarrow F}}{\text{Dc v dLGa} \Rightarrow \text{bZ} \rightarrow F} \\
 \hspace{15em} \text{(D.60.2)}
 \end{array}$$

(R12,R15)

$$\begin{array}{c}
 \frac{\frac{\text{GcDaL} \xrightarrow{f} d \quad \text{GcDbL} \xrightarrow{g} d}{\text{GcDa v bL} \rightarrow d}}{\text{GDa v bL} \rightarrow c \Rightarrow d} \quad \Rightarrow \quad \frac{\frac{\text{GcDaL} \xrightarrow{f} d \quad \text{GcDbL} \xrightarrow{g} d}{\text{GDaL} \rightarrow c \Rightarrow d} \quad \frac{\text{GcDbL} \xrightarrow{g} d}{\text{GDBL} \rightarrow c \Rightarrow d}}{\text{GDa v bL} \rightarrow c \Rightarrow d} \\
 \hspace{15em} \text{(D.61.1)}
 \end{array}$$

$$\begin{array}{c}
 \frac{\frac{\text{GaDcL} \xrightarrow{h} d \quad \text{GbDcL} \xrightarrow{k} d}{\text{Ga v bDcL} \rightarrow d}}{\text{Ga v bDL} \rightarrow c \Rightarrow d} \quad \Rightarrow \quad \frac{\frac{\text{GaDcL} \xrightarrow{h} d \quad \text{GbDcL} \xrightarrow{k} d}{\text{GaDL} \rightarrow c \Rightarrow d} \quad \frac{\text{GbDcL} \xrightarrow{k} d}{\text{GbDL} \rightarrow c \Rightarrow d}}{\text{Ga v bDL} \rightarrow c \Rightarrow d} \\
 \hspace{15em} \text{(D.61.2)}
 \end{array}$$

(R13,R13)

$$\begin{array}{c}
 \frac{\frac{\text{G} \xrightarrow{f} \text{FabEcdH}}{\text{G} \rightarrow \text{Fa v bEcdH}}}{\text{G} \rightarrow \text{Fa v bEc v dH}} \quad \Rightarrow \quad \frac{\frac{\text{G} \xrightarrow{f} \text{FabEcdH}}{\text{G} \rightarrow \text{FabEc v dH}}}{\text{G} \rightarrow \text{Fa v bEc v dH}} \\
 \hspace{15em} \text{(D.62)}
 \end{array}$$

(R13,R14)

$$\begin{array}{c}
 \frac{\frac{\text{G} \xrightarrow{f} a \quad \text{DbL} \xrightarrow{g} \text{FcdE}}{\text{DGa} \Rightarrow \text{bL} \rightarrow \text{FcdE}}}{\text{DGa} \Rightarrow \text{bL} \rightarrow \text{Fc v dE}} \quad \Rightarrow \quad \frac{\frac{\text{DbL} \xrightarrow{g} \text{FcdE}}{\text{G} \xrightarrow{f} a \quad \text{DbL} \rightarrow \text{Fc v dE}}}{\text{DGa} \Rightarrow \text{bL} \rightarrow \text{Fc v dE}} \\
 \hspace{15em} \text{(D.63)}
 \end{array}$$

(R14,R14)

$$\begin{array}{c}
 \frac{\frac{D \xrightarrow{g} a \quad LbZdP \xrightarrow{h} F}{G \xrightarrow{f} c \quad LDa \Rightarrow bZdP \rightarrow F}}{LDa \Rightarrow bZGc \Rightarrow dP \rightarrow F} >_{\equiv} \frac{\frac{G \xrightarrow{f} c \quad LbZdP \xrightarrow{h} F}{D \xrightarrow{g} a \quad LbZGc \Rightarrow dP \rightarrow F}}{LDa \Rightarrow bZGc \Rightarrow dP \rightarrow F} \\
 \hline
 \hline
 \end{array} \quad (D.64)$$

(R14,R15)

$$\begin{array}{c}
 \frac{\frac{G \xrightarrow{f} a \quad DcLbZ \xrightarrow{g} d}{DcLGa \Rightarrow bZ \rightarrow d}}{DLGa \Rightarrow bZ \rightarrow c \Rightarrow d} >_{\equiv} \frac{\frac{DcLbZ \xrightarrow{g} d}{G \xrightarrow{f} a \quad DLbZ \rightarrow c \Rightarrow d}}{DLGa \Rightarrow bZ \rightarrow c \Rightarrow d} \\
 \hline
 \hline
 \end{array} \quad (D.65.1)$$

$$\begin{array}{c}
 \frac{\frac{G \xrightarrow{f} a \quad DbLcZ \xrightarrow{g} d}{DGa \Rightarrow bLcZ \rightarrow d}}{DGa \Rightarrow bLZ \rightarrow c \Rightarrow d} >_{\equiv} \frac{\frac{DbLcZ \xrightarrow{g} d}{G \xrightarrow{f} a \quad DbLZ \rightarrow c \Rightarrow d}}{DGa \Rightarrow bLZ \rightarrow c \Rightarrow d} \\
 \hline
 \hline
 \end{array} \quad (D.65.2)$$

$$\begin{array}{c}
 \frac{\frac{\frac{G \xrightarrow{f} c \quad LbZdP \xrightarrow{h} e}{D \xrightarrow{g} a \quad LbZGc \Rightarrow dP \rightarrow e}}{LDa \Rightarrow bZGc \Rightarrow dP \rightarrow e}}{LDZGc \Rightarrow dP \rightarrow (a \Rightarrow b) \Rightarrow e} >_{\equiv} \frac{\frac{\frac{D \xrightarrow{g} a \quad LbZdP \xrightarrow{h} e}{LDa \Rightarrow bZdP \rightarrow e}}{G \xrightarrow{f} c \quad LDZdP \rightarrow (a \Rightarrow b) \Rightarrow e}}{LDZGc \Rightarrow dP \rightarrow (a \Rightarrow b) \Rightarrow e} \\
 \hline
 \hline
 \end{array} \quad (D.65.3)$$

$$\begin{array}{c}
 \frac{\frac{\frac{G \xrightarrow{m} a \quad DcdLbZ \xrightarrow{n} e}{DcdLGa \Rightarrow bZ \rightarrow e}}{Dc \uparrow dLGa \Rightarrow bZ \rightarrow e}}{DLGa \Rightarrow bZ \rightarrow (c \uparrow d) \Rightarrow e} >_{\equiv} \frac{\frac{\frac{DcdLbZ \xrightarrow{n} e}{Dc \uparrow dLbZ \rightarrow e}}{G \xrightarrow{m} a \quad DLbZ \rightarrow (c \uparrow d) \Rightarrow e}}{DLGa \Rightarrow bZ \rightarrow (c \uparrow d) \Rightarrow e} \\
 \hline
 \hline
 \end{array} \quad (D.65.4)$$

$$\begin{array}{c}
 \frac{\frac{\frac{G \xrightarrow{m} a \quad DbLcdZ \xrightarrow{r} e}{DGa \Rightarrow bLcdZ \rightarrow e}}{DGa \Rightarrow bLc \uparrow dZ \rightarrow e}}{DGa \Rightarrow bLZ \rightarrow (c \uparrow d) \Rightarrow e} >_{\equiv} \frac{\frac{\frac{DbLcdZ \xrightarrow{r} e}{DbLc \uparrow dZ \rightarrow e}}{G \xrightarrow{m} a \quad DbLZ \rightarrow (c \uparrow d) \Rightarrow e}}{DGa \Rightarrow bLZ \rightarrow (c \uparrow d) \Rightarrow e} \\
 \hline
 \hline
 \end{array} \quad (D.65.5)$$

$$\begin{array}{c}
 \frac{\frac{\frac{G \xrightarrow{m} a \quad DbLZ \xrightarrow{s} e}{DGa \Rightarrow bLZ \rightarrow e}}{DGa \Rightarrow bLcZ \rightarrow e}}{DGa \Rightarrow bLZ \rightarrow c \Rightarrow e} >_{\equiv} \frac{\frac{\frac{DbLZ \xrightarrow{s} e}{DbLcZ \rightarrow e}}{G \xrightarrow{m} a \quad DbLZ \rightarrow c \Rightarrow e}}{DGa \Rightarrow bLZ \rightarrow c \Rightarrow e} \\
 \hline
 \hline
 \end{array} \quad (D.65.6)$$

$$\begin{array}{c}
 \frac{\frac{\frac{G \xrightarrow{m} a \quad DLbZ \xrightarrow{t} e}{DLGa \Rightarrow bZ \rightarrow e}}{DcLGa \Rightarrow bZ \rightarrow e}}{DLGa \Rightarrow bZ \rightarrow c \Rightarrow e} >_{\equiv} \frac{\frac{\frac{DLbZ \xrightarrow{t} e}{DcLbZ \rightarrow e}}{G \xrightarrow{m} a \quad DLbZ \rightarrow c \Rightarrow e}}{DLGa \Rightarrow bZ \rightarrow c \Rightarrow e} \\
 \hline
 \hline
 \end{array} \quad (D.65.7)$$



$$\frac{\frac{GaDccL \xrightarrow{p} F}{GaDcL \rightarrow F} \quad \frac{GbDcL \xrightarrow{q} F}{GbDcL \rightarrow F}}{Ga \vee bDcL \rightarrow F} \quad \text{>H}$$

$$\frac{\frac{GaDccL \xrightarrow{p} F}{Ga \vee bDccL \rightarrow F} \quad \frac{GbDcL \xrightarrow{q} F}{GbDccL \rightarrow F}}{Ga \vee bDcL \rightarrow F} \quad \text{(D.74.4)}$$

(R2, R4, R10)

$$\frac{\frac{GcD \xrightarrow{f} FaE}{GdcD \rightarrow FaE} \quad \frac{GcdD \xrightarrow{g} FbE}{GcdD \rightarrow FbE}}{GdcD \rightarrow Fa \uparrow bE} \quad \text{>H}$$

$$\frac{\frac{GcD \xrightarrow{f} FaE}{GcdD \rightarrow FaE} \quad \frac{GcdD \xrightarrow{g} FbE}{GcdD \rightarrow Fa \uparrow bE}}{GdcD \rightarrow Fa \uparrow bE} \quad \text{(D.75.1)}$$

$$\frac{\frac{GdD \xrightarrow{h} FaE}{GdcD \rightarrow FaE} \quad \frac{GcdD \xrightarrow{g} FbE}{GcdD \rightarrow FbE}}{GdcD \rightarrow Fa \uparrow bE} \quad \text{>H}$$

$$\frac{\frac{GdD \xrightarrow{h} FaE}{GcdD \rightarrow FaE} \quad \frac{GcdD \xrightarrow{g} FbE}{GcdD \rightarrow Fa \uparrow bE}}{GdcD \rightarrow Fa \uparrow bE} \quad \text{(D.75.2)}$$

$$\frac{\frac{GcdD \xrightarrow{g} FbE}{GdcD \rightarrow FbE} \quad \frac{GcD \xrightarrow{f} FaE}{GcdD \rightarrow FaE}}{GdcD \rightarrow Fb \uparrow aE} \quad \text{>H}$$

$$\frac{\frac{GcdD \xrightarrow{g} FbE}{GcdD \rightarrow Fb \uparrow aE} \quad \frac{GcD \xrightarrow{f} FaE}{GcdD \rightarrow FaE}}{GdcD \rightarrow Fb \uparrow aE} \quad \text{(D.75.3)}$$

$$\frac{\frac{GcdD \xrightarrow{g} FbE}{GdcD \rightarrow FbE} \quad \frac{GdD \xrightarrow{h} FaE}{GcdD \rightarrow FaE}}{GdcD \rightarrow Fb \uparrow aE} \quad \text{>H}$$

$$\frac{\frac{GcdD \xrightarrow{g} FbE}{GcdD \rightarrow Fb \uparrow aE} \quad \frac{GdD \xrightarrow{h} FaE}{GcdD \rightarrow FaE}}{GdcD \rightarrow Fb \uparrow aE} \quad \text{(D.75.4)}$$

(R2, R4, R12)

$$\frac{\frac{GcDaL \xrightarrow{f} F}{GdcDaL \rightarrow F} \quad \frac{GcdDbL \xrightarrow{g} F}{GcdDbL \rightarrow F}}{GdcDa \vee bL \rightarrow F} \quad \text{>H}$$

$$\frac{\frac{GcDaL \xrightarrow{f} F}{GcdDaL \rightarrow F} \quad \frac{GcdDbL \xrightarrow{g} F}{GcdDbL \rightarrow F}}{GdcDa \vee bL \rightarrow F} \quad \text{(D.76.1)}$$

$$\frac{\frac{GdDaL \xrightarrow{h} F}{GdcDaL \rightarrow F} \quad \frac{GcdDbL \xrightarrow{g} F}{GcdDbL \rightarrow F}}{GdcDa \vee bL \rightarrow F} \quad \text{>H}$$

$$\frac{\frac{GdDaL \xrightarrow{h} F}{GcdDaL \rightarrow F} \quad \frac{GcdDbL \xrightarrow{g} F}{GcdDbL \rightarrow F}}{GdcDa \vee bL \rightarrow F} \quad \text{(D.76.2)}$$

$$\frac{\frac{GcdDbL \xrightarrow{g} F}{GdcDbL \rightarrow F} \quad \frac{GcDaL \xrightarrow{f} F}{GcdDaL \rightarrow F}}{GdcDb \vee aL \rightarrow F} \quad \text{>H}$$

$$\frac{\frac{GcdDbL \xrightarrow{g} F}{GcdDb \vee aL \rightarrow F} \quad \frac{GcDaL \xrightarrow{f} F}{GcdDaL \rightarrow F}}{GdcDb \vee aL \rightarrow F} \quad \text{(D.76.3)}$$

$$\begin{array}{c}
 \frac{\text{GcdDbL} \xrightarrow{g} F \quad \text{GdDaL} \xrightarrow{h} F}{\text{GdcDbL} \xrightarrow{g} F \quad \text{GdcDaL} \xrightarrow{h} F} \\
 \hline
 \text{GdcDb} \vee \text{aL} \rightarrow F \quad \text{>H}
 \end{array}
 \qquad
 \begin{array}{c}
 \frac{\text{GcdDbL} \xrightarrow{g} F \quad \frac{\text{GdDaL} \xrightarrow{h} F}{\text{GcdDaL} \xrightarrow{h} F}}{\text{GcdDb} \vee \text{aL} \xrightarrow{g} F} \\
 \hline
 \text{GdcDb} \vee \text{aL} \rightarrow F \quad \text{(D.76.4)}
 \end{array}$$

$$\begin{array}{c}
 \frac{\text{GaDcL} \xrightarrow{k} F \quad \text{GbDcdL} \xrightarrow{m} F}{\text{GaDdcL} \xrightarrow{k} F \quad \text{GbDdcL} \xrightarrow{m} F} \\
 \hline
 \text{Ga} \vee \text{bDdcL} \rightarrow F \quad \text{>H}
 \end{array}
 \qquad
 \begin{array}{c}
 \frac{\text{GaDcL} \xrightarrow{k} F}{\text{GaDcdL} \xrightarrow{k} F} \quad \text{GbDcdL} \xrightarrow{m} F \\
 \hline
 \text{Ga} \vee \text{bDcdL} \rightarrow F \quad \text{(D.76.5)}
 \end{array}$$

$$\begin{array}{c}
 \frac{\text{GaDdL} \xrightarrow{n} F \quad \text{GbDcdL} \xrightarrow{m} F}{\text{GaDdcL} \xrightarrow{n} F \quad \text{GbDdcL} \xrightarrow{m} F} \\
 \hline
 \text{Ga} \vee \text{bDdcL} \rightarrow F \quad \text{>H}
 \end{array}
 \qquad
 \begin{array}{c}
 \frac{\text{GaDdL} \xrightarrow{n} F}{\text{GaDcdL} \xrightarrow{n} F} \quad \text{GbDcdL} \xrightarrow{m} F \\
 \hline
 \text{Ga} \vee \text{bDcdL} \rightarrow F \quad \text{(D.76.6)}
 \end{array}$$

$$\begin{array}{c}
 \frac{\text{GbDcdL} \xrightarrow{m} F \quad \text{GaDcL} \xrightarrow{k} F}{\text{GbDdcL} \xrightarrow{m} F \quad \text{GaDdcL} \xrightarrow{k} F} \\
 \hline
 \text{Gb} \vee \text{aDdcL} \rightarrow F \quad \text{>H}
 \end{array}
 \qquad
 \begin{array}{c}
 \text{GbDcdL} \xrightarrow{m} F \quad \frac{\text{GaDcL} \xrightarrow{k} F}{\text{GaDcdL} \xrightarrow{k} F} \\
 \hline
 \text{Gb} \vee \text{aDcdL} \rightarrow F \quad \text{(D.76.7)}
 \end{array}$$

$$\begin{array}{c}
 \frac{\text{GbDcdL} \xrightarrow{m} F \quad \text{GaDdL} \xrightarrow{n} F}{\text{GbDdcL} \xrightarrow{m} F \quad \text{GaDdcL} \xrightarrow{n} F} \\
 \hline
 \text{Gb} \vee \text{aDdcL} \rightarrow F \quad \text{>H}
 \end{array}
 \qquad
 \begin{array}{c}
 \text{GbDcdL} \xrightarrow{m} F \quad \frac{\text{GaDdL} \xrightarrow{n} F}{\text{GaDcdL} \xrightarrow{n} F} \\
 \hline
 \text{Gb} \vee \text{aDcdL} \rightarrow F \quad \text{(D.76.8)}
 \end{array}$$

$$\begin{array}{c}
 \frac{\text{GaD} \xrightarrow{f} F \quad \text{GaD} \xrightarrow{g} F}{\text{GbaD} \xrightarrow{f} F \quad \text{GcaD} \xrightarrow{g} F} \\
 \hline
 \text{G}(b \vee c)\text{aD} \rightarrow F \\
 \hline
 \text{Ga}(b \vee c)\text{D} \rightarrow F \quad \text{>H}
 \end{array}
 \qquad
 \begin{array}{c}
 \frac{\text{GaD} \xrightarrow{f} F \quad \text{GaD} \xrightarrow{g} F}{\text{GabD} \xrightarrow{f} F \quad \text{GacD} \xrightarrow{g} F} \\
 \hline
 \text{Ga}(b \vee c)\text{D} \rightarrow F \quad \text{(D.76.9)}
 \end{array}$$

$$\begin{array}{c}
 \frac{\text{GbD} \xrightarrow{h} F \quad \text{GaD} \xrightarrow{g} F}{\text{GbaD} \xrightarrow{h} F \quad \text{GcaD} \xrightarrow{g} F} \\
 \hline
 \text{G}(b \vee c)\text{aD} \rightarrow F \\
 \hline
 \text{Ga}(b \vee c)\text{D} \rightarrow F \quad \text{>H}
 \end{array}
 \qquad
 \begin{array}{c}
 \frac{\text{GbD} \xrightarrow{h} F \quad \text{GaD} \xrightarrow{g} F}{\text{GabD} \xrightarrow{h} F \quad \text{GacD} \xrightarrow{g} F} \\
 \hline
 \text{Ga}(b \vee c)\text{D} \rightarrow F \quad \text{(D.76.10)}
 \end{array}$$

$$\begin{array}{c}
 \frac{\text{GaD} \xrightarrow{f} F \quad \text{GcD} \xrightarrow{k} F}{\text{GbaD} \xrightarrow{f} F \quad \text{GcaD} \xrightarrow{k} F} \\
 \hline
 \text{G}(b \vee c)\text{aD} \rightarrow F \\
 \hline
 \text{Ga}(b \vee c)\text{D} \rightarrow F \quad \text{>H}
 \end{array}
 \qquad
 \begin{array}{c}
 \frac{\text{GaD} \xrightarrow{f} F \quad \text{GcD} \xrightarrow{k} F}{\text{GabD} \xrightarrow{f} F \quad \text{GacD} \xrightarrow{k} F} \\
 \hline
 \text{Ga}(b \vee c)\text{D} \rightarrow F \quad \text{(D.76.11)}
 \end{array}$$

$$\frac{\frac{GbD \xrightarrow{h} F}{GbaD \xrightarrow{h} F} \quad \frac{GcD \xrightarrow{k} F}{GcaD \xrightarrow{k} F}}{G(b \vee c)aD \xrightarrow{h} F} \quad \frac{G(b \vee c)aD \xrightarrow{h} F}{Ga(b \vee c)D \xrightarrow{h} F} \quad \Rightarrow \quad \frac{\frac{GbD \xrightarrow{h} F}{GbaD \xrightarrow{h} F} \quad \frac{GcD \xrightarrow{k} F}{GcaD \xrightarrow{k} F}}{Ga(b \vee c)D \xrightarrow{h} F} \quad (D.76.12)$$

$$\frac{\frac{GaD \xrightarrow{f} F}{GabD \xrightarrow{f} F} \quad \frac{GaD \xrightarrow{g} F}{GacD \xrightarrow{g} F}}{Ga(b \vee c)D \xrightarrow{f} F} \quad \frac{Ga(b \vee c)D \xrightarrow{f} F}{G(b \vee c)aD \xrightarrow{f} F} \quad \Rightarrow \quad \frac{\frac{GaD \xrightarrow{f} F}{GabD \xrightarrow{f} F} \quad \frac{GaD \xrightarrow{g} F}{GacD \xrightarrow{g} F}}{G(b \vee c)aD \xrightarrow{f} F} \quad (D.76.13)$$

$$\frac{\frac{GbD \xrightarrow{h} F}{GabD \xrightarrow{h} F} \quad \frac{GaD \xrightarrow{g} F}{GacD \xrightarrow{g} F}}{Ga(b \vee c)D \xrightarrow{h} F} \quad \frac{Ga(b \vee c)D \xrightarrow{h} F}{G(b \vee c)aD \xrightarrow{h} F} \quad \Rightarrow \quad \frac{\frac{GbD \xrightarrow{h} F}{GabD \xrightarrow{h} F} \quad \frac{GaD \xrightarrow{g} F}{GacD \xrightarrow{g} F}}{G(b \vee c)aD \xrightarrow{h} F} \quad (D.76.14)$$

$$\frac{\frac{GaD \xrightarrow{f} F}{GabD \xrightarrow{f} F} \quad \frac{GcD \xrightarrow{k} F}{GacD \xrightarrow{k} F}}{Ga(b \vee c)D \xrightarrow{f} F} \quad \frac{Ga(b \vee c)D \xrightarrow{f} F}{G(b \vee c)aD \xrightarrow{f} F} \quad \Rightarrow \quad \frac{\frac{GaD \xrightarrow{f} F}{GabD \xrightarrow{f} F} \quad \frac{GcD \xrightarrow{k} F}{GacD \xrightarrow{k} F}}{G(b \vee c)aD \xrightarrow{f} F} \quad (D.76.15)$$

$$\frac{\frac{GbD \xrightarrow{h} F}{GabD \xrightarrow{h} F} \quad \frac{GcD \xrightarrow{k} F}{GacD \xrightarrow{k} F}}{Ga(b \vee c)D \xrightarrow{h} F} \quad \frac{Ga(b \vee c)D \xrightarrow{h} F}{G(b \vee c)aD \xrightarrow{h} F} \quad \Rightarrow \quad \frac{\frac{GbD \xrightarrow{h} F}{GabD \xrightarrow{h} F} \quad \frac{GcD \xrightarrow{k} F}{GacD \xrightarrow{k} F}}{G(b \vee c)aD \xrightarrow{h} F} \quad (D.76.16)$$

(R3, R3, R10)

$$\frac{\frac{GcddD \xrightarrow{f} FaE}{GcdD \xrightarrow{f} FaE} \quad \frac{GcdddD \xrightarrow{g} FbE}{GcdD \xrightarrow{g} FbE}}{GcdD \xrightarrow{f} Fa \uparrow bE} \quad \Rightarrow \quad \frac{\frac{GcddD \xrightarrow{f} FaE}{GcddD \xrightarrow{f} Fa \uparrow bE} \quad \frac{GcdddD \xrightarrow{g} FbE}{GcdddD \xrightarrow{g} FbE}}{GcdD \xrightarrow{f} Fa \uparrow bE} \quad (D.77.1)$$

$$\frac{\frac{GcdddD \xrightarrow{h} FaE}{GcdD \xrightarrow{h} FaE} \quad \frac{GcddD \xrightarrow{k} FbE}{GcdD \xrightarrow{k} FbE}}{GcdD \xrightarrow{h} Fa \uparrow bE} \quad \Rightarrow \quad \frac{\frac{GcdddD \xrightarrow{h} FaE}{GcdddD \xrightarrow{h} Fa \uparrow bE} \quad \frac{GcddD \xrightarrow{k} FbE}{GcdddD \xrightarrow{k} FbE}}{GcdD \xrightarrow{h} Fa \uparrow bE} \quad (D.77.2)$$

(R3, R3, R12)

$$\frac{\frac{GcddDaL \xrightarrow{f} F}{GcdDaL \xrightarrow{f} F} \quad \frac{GcddDbL \xrightarrow{g} F}{GcdDbL \xrightarrow{g} F}}{GcdDa \vee bL \xrightarrow{f} F} \quad \Rightarrow \quad \frac{\frac{GcddDaL \xrightarrow{f} F}{GcddDa \vee bL \xrightarrow{f} F} \quad \frac{GcddDbL \xrightarrow{g} F}{GcddDbL \xrightarrow{g} F}}{GcdDa \vee bL \xrightarrow{f} F} \quad (D.78.1)$$





Appendix (5)

User Manual

\$1. Introduction

szabo is an implementation, running under prolog, to automate some reasoning about cartesian, bicartesian, distributive bicartesian, cartesian closed and bicartesian closed categories.

The implementation is based on the algorithms outlined in SZABO [1978], to interpret derivations from the sequent propositional calculus as arrows of the corresponding free category and to represent the arrows of this free category by means of derivations with cuts. Also it includes procedures to eliminate the cut from the derivations and to produce the normal form, according to the algorithms of cut-elimination and normalization due to SZABO. All the work behind this implementation is discussed in detail in chapters 1-5.

\$2. 'szabo' Program Syntax

We define here the syntax, using BNF (NAUR [1960]), of the formal language processed by our implementation. Note the use of lower case letters in the object language, since prolog restricts upper case letters to be names of meta-variables.

Each proof as input to the program is a list of named steps, such that each step is either a representation of an axiom or derived from another (or others) by using a rule of inference. The conclusion of the original proof will be represented by the last step in the input proof. The syntax of each step will be described in BNF as follows :-

```

PROOF_STEP      ::= STEP_NAME " = " STEP
STEP_NAME       ::= LOWER_CASE_WORD | NUMBERED_LOWER_CASE_WORD
STEP            ::= "[" JUSTIFICATION "," ANTECEDENT "=>" <SUCCEDENT "]"
ANTECEDENT      ::= "[" FORMULA_SEQUENCE "]" | "[" "]"
SUCCEDENT       ::= "[" FORMULA_SEQUENCE "]" | "[" "]"
FORMULA_SEQUENCE ::= FORMULA | FORMULA_SEQUENCE "," FORMULA
FORMULA         ::= ATOMIC_FORMULA | FORMULA CONNECTIVE ATOMIC_FORMULA
ATOMIC_FORMULA  ::= PROPOSITIONAL_SYMBOL | bo | tr | id
PROPOSITIONAL_SYMBOL ::= LOWER_CASE_WORD
                  [ other than one of {bo, tr, CURRENT_NUMBERED_f}]
NUMBERED_LOWER_CASE_WORD ::= LOWER_CASE_WORD INTEGER

```



b) Operational rules

program syntax : : : : : : : : : : :	Sequent Calculus syntax : : : : : : : : : : :
ala	$\uparrow I \rightarrow$ (and introduction antecedent)
als	$\rightarrow \uparrow I$ (and introduction succedent)
ola	$\vee I \rightarrow$ (or introduction antecedent)
ols	$\rightarrow \vee I$ (or introduction succedent)
hla	$\Rightarrow I \rightarrow$ (hook introduction antecedent)
his	$\rightarrow \Rightarrow I$ (hook introduction succedent)

We summarize the syntax to be used in the implementation corresponding to the traditional propositional calculus syntax for the connectives as follows :-

Connective : : : : : : : : : : :	Prop. Calculus syntax : : : : : : : : : : :	Program syntax : : : : : : : : : : :
conjunction	$a \uparrow b$	$a \& b$
disjunction	$a \vee b$	$a \# b$
implication	$a \Rightarrow b$	$a \rightarrow b$
true	$\top$	tr
bottom	$\perp$	bo

NB .  
===

Here we remark on the syntax for the rules of inference defined in §2. Any rule of inference name must be followed by a number of arguments, and these arguments must be in the order as shown in §2. We explain this order for some of them which are easily confused :

1) Inc(l,a,b)

This refers to the instance of the interchange rule which, operating on a step 'l' having adjacent formulae 'b,a' ( in that order) in antecedent or succedent, produces a new step with these two formulae in order ' a,b '.

2) cut(l,m,c)

This means that the cut rule is applied to the steps which have names 'l', 'm', with the cut formula c; 'l' is the name of the step which has 'c' in its succedent, and 'm' is the name of the step which has 'c' in its antecedent.

3) als(l,m,a & b),

'l' is the name of a step with 'a' in its succedent; 'm' has 'b' in its succedent : this construct performs 'and-introduction' on the succedents.

4) oia(l,m,a # b)

'l' is the name of a step with 'a' in its antecedent; 'm' has 'b' in its antecedent : this construct performs 'or-introduction' on the antecedents.

5) hla(l,m,a -> b)

'l' is the name of a step with 'a' in its succedent; 'm' has 'b' in its antecedent : this construct performs 'hook-introduction' on the antecedents.

#### \$4. Canonical arrows in category theory

In this section we give some canonical arrows from different free categories ( These particular arrows, and their representations by means of derivations of the sequent propositional calculus, are built into the system for the user's convenience ) .

##### (4.1) Canonical arrows in Fc(X) ( The free cartesian category )

Let X be a category and Fc(X) is the free cartesian category generated by X and a terminal object T and connective  $\dagger$  ; the following arrows are in ArFc(X) :

- i)  $\pi_{left}(a,b) : a \dagger b \rightarrow a$  , for  $a,b$  in  $ObFc(X)$  ;
- ii)  $\pi_{right}(a,b) : a \dagger b \rightarrow b$  , for  $a,b$  in  $ObFc(X)$  ;
- iii)  $\delta(a) : a \rightarrow a \dagger a$  , for  $a$  in  $ObFc(X)$  ;
- iv)  $\alpha(a,b,c) : a \dagger (b \dagger c) \rightarrow (a \dagger b) \dagger c$ , for  $a,b,c$  in  $ObFc(X)$ ;

- v)  $\alpha\_inverse(a,b,c) : (a \uparrow b) \uparrow c \rightarrow a \uparrow (b \uparrow c),$   
for  $a,b,c$  in  $ObFc(X);$
- vi) for  $f : a \rightarrow b$  and  $g : c \rightarrow d$  in  $ArFc(X),$   
 $product(f,g) : a \uparrow c \rightarrow b \uparrow d$  ;
- vii)  $\sigma(a,b) : a \uparrow b \rightarrow b \uparrow a$  , for  $a,b$  in  $ObFc(X)$  ;
- viii)  $\sigma\_inverse(a,b) : b \uparrow a \rightarrow a \uparrow b$  , for  $a,b$  in  $ObFc(X)$  ;
- ix) for  $f : a \rightarrow b$  and  $g : b \rightarrow c$  in  $ArFc(X),$   
 $comp(g,f) : a \rightarrow c$  ;
- x) for  $f : a \rightarrow b$  and  $g : a \rightarrow c$  in  $ArFc(X),$   
 $angle(f,g) : a \rightarrow b \uparrow c$  .

#### (4.2) Canonical arrows in Fbc(X) ( The free bicartesian category)

We extend the canonical arrows in  $Fc(X)$  by the following :

- xi)  $\pi\_left\_star(a,b) : a \rightarrow a \vee b$  , for  $a,b$  in  $ObFbc(X)$  ;
- xii)  $\pi\_right\_star(a,b) : b \rightarrow a \vee b$  , for  $a,b$  in  $ObFbc(X)$  ;
- xiii)  $\delta\_star(a) : a \vee a \rightarrow a$  , for  $a$  in  $ObFbc(X),$  ;
- xiv)  $\alpha\_star(a,b,c) : a \vee (b \vee c) \rightarrow (a \vee b) \vee c,$  for  $a,b,c$  in  $ObFbc(X)$  ;
- xv)  $\alpha\_star\_inverse(a,b,c) : (a \vee b) \vee c \rightarrow a \vee (b \vee c),$   
for  $a,b,c$  in  $ObFbc(X)$  ;
- xvi)  $\sigma\_star(a,b) : a \vee b \rightarrow b \vee a$  , for  $a,b$  in  $ObFbc(X)$  ;
- xvii)  $\sigma\_star\_inverse(a,b) : b \vee a \rightarrow a \vee b,$  for  $a,b$  in  $ObFbc(X);$
- xviii) for  $f : a \rightarrow b$  and  $g : c \rightarrow d$  in  $ArFbc(X),$   
 $union(f,g) : a \vee c \rightarrow b \vee d$  ;
- xix) for  $f : a \rightarrow c$  and  $g : b \rightarrow c$  in  $ArFbc(X),$   
 $square(f,g) : a \vee b \rightarrow c$  .

#### (4.3) Canonical arrows in Fdbc(X)(free distributive bicartesian category)

We extend the canonical arrows in  $Fbc(X)$  by the following :

- xx)  $\delta\_left(a,b,c) : a \uparrow (b \vee c) \rightarrow (a \uparrow b) \vee (a \uparrow c)$  ;
- xxi)  $\delta\_right(a,b,c) : (a \vee b) \uparrow c \rightarrow (a \uparrow c) \vee (b \uparrow c)$  ;

- xxii)  $\text{delta\_left\_star}(a,b,c) : (a \vee b) \wedge (a \vee c) \rightarrow a \vee (b \wedge c) ;$   
 xxiii)  $\text{delta\_right\_star}(a,b,c) : (a \vee c) \wedge (b \vee c) \rightarrow (a \wedge b) \vee c ;$   
 xxiv)  $\text{kapa}(a,b,c,d,e) : (a \wedge ((b \vee c) \vee d)) \wedge e \rightarrow (b \vee ((a \wedge c) \wedge e)) \vee d ;$

#### (4.4) Canonical arrows in Fccl(X) (The free cartesian closed category)

We extend the canonical arrows in  $\text{Fc}(X)$  by the following :

- xi) for  $f : a \rightarrow b$  and  $g : c \rightarrow d$  in  $\text{ArFccl}(X)$ ,  
 $\text{epsilon\_left}(f,g) : a \wedge (b \Rightarrow c) \rightarrow d ;$   
 xii) for  $f : a \wedge b \rightarrow c$  in  $\text{ArFccl}(X)$ ,  
 $\text{alpha\_right}(f) : b \rightarrow (a \Rightarrow c) ;$   
 xiii)  $\text{epsilon}(a,b) : a \wedge (a \Rightarrow b) \rightarrow b$  , for  $a,b$  in  $\text{ObFccl}(X)$ ;  
 xiv)  $\text{ita}(a,b) : b \rightarrow a \Rightarrow (a \wedge b)$  , for  $a,b$  in  $\text{ObFccl}(X)$ ;  
 xv) for  $f : a \rightarrow b$  and  $g : c \rightarrow d$  in  $\text{ArFccl}(X)$ ,  
 $\text{hook}(f,g) : (b \Rightarrow c) \rightarrow (a \Rightarrow d)$

#### (4.5) Canonical arrows in Fbccl(X) (The free bicartesian closed category)

The canonical arrows in  $\text{Fbccl}(X)$  are combination of all canonical arrows in  $\text{Fc}(X)$ ,  $\text{Fbc}(X)$ ,  $\text{Fdbc}(X)$  and  $\text{Fccl}(X)$ .

### \$5. Predicates ( Commands)

In order to use this system there are two types of command ( we call them predicates, because of prolog convention ) :

- I) Proof theory predicates.
- II) Category theory predicates.

#### I) Proof theory predicates ( commands )

The following predicates are used to manipulate derivations from the classes of sequent propositional calculus discussed in chapters 1-5; the derivations must be asserted to the system according to the syntax described in \$2.

- 1-  $\text{theory}(X)$  : Tells the system that the theory we intend to work with is  $X$ , where  $X$  represents one of the following abbreviations cart (cartesian), bicart (bicartesian), dbicart (distributive bicartesian), cartcl (cartesian closed) or bicartcl (bicartesian closed).

- 2- `composite(F,G,H)` : This predicate is used, when arrows F,G of the underlying category X with composition H are used in the construction of the derivation in operation.
- 3- `prove(X)` : Checks the correctness of the current instance of a proof-step X; if it is correct, it will be added to the database for further use. We keep using this command until the proof is complete.
- 4- `proof` : This command is used, as another way to check the correctness and assert derivations without using the formal syntax for proof-step in §2 and the predicate ' `prove(X)` ' above. It will switch on the system to communicate with the user to assert the derivation in informal way.
- 5- `interpr(X,K)` : After asserting a derivation, this command is used to compute the interpretation of this derivation as an arrow in the free category. X represents the name of the last proof-step and H will be instantiated as the arrow computed to interpret the derivation.
- 6- `cut_free(X)` : Tells the system to start the processes of transforming the derivation in question to its equivalent cut-free one. X represents the name of the last proof-step.
- 7- `con_inc_free(X)` : where X represents the name of the last step of a cut-free proof. This predicate cause the system to start the processes of transforming the cut-free proof into its equivalent contraction & interchange-free one. This is used only if we dealing with the theories of cart, bicart or dbicart.
- 8- `normal(X)` : This predicate is a combination of three processes : cut-elimination, contraction & interchange-elimination and normalization; so that this general command is used in place of `cut_free` and `con_inc_free` to produce the equivalent normal form for a derivation. X represents the name of the last step of the derivation.

## II) Category theory predicates ( Commands )

This subsection describes predicates for manipulating arrows of the free cartesian, bicartesian, distributive bicartesian, cartesian closed and bicartesian closed categories. These arrows will be represented

by means of derivations from the sequent propositional calculus. By the processes of cut-elimination and normalization, these representations will be transformed to their equivalent normal form. The system also is used to test whether diagrams, from the theories mentioned above, dose commute or not. The following are the predicates of the system to operate the above methods.

- 1- arrow(A,F,B) : Insert an arrow , of the underlying category X, being used in the construction of the arrow of the free category in question. F represents an arrow from A to B (  $F : A \rightarrow B$  ).
- 2- rep\_of(X) : Will produce a representation of an arrow X as a derivation.
- 3- norm\_rep(X) : If X represents an arrow of some free category, this predicate causes the system to produce a representation of X by means of a derivation, then calling the processes of cut-elimination and normalization to produce the corresponding normal representation.
- 4- commutative : Tests whether two arrows are equal or not, in order to test the commutativity of the diagrams in category theory. After reading this command, the system waits to read the LHS arrow of the equality in question and responds by computing its normal representation. The system waits again to read the RHS arrow of the equality and responds by computing its normal representation. Finally it compares these two normal representations and responds with the conclusion.
- 5- inter\_norm\_rep(X): This is a combination of two processes; one for normal representation of the arrow X, the other is to interpret this normal representation to an arrow of the same theory of the arrow X.

The following are general commands to be used :

- 1- help : This displays the commands available for the system.
- 2- clean : This command is to be used after each session of use, in order to use the system using the same theory and the same arrows, if any, of the underlying category for another session.



- 3- new : Cleans up the database from all the processes of the last session and puts the system again to the top level.

## \$6. How To Use The System ?

To use this system for both proof theory and category theory there are two ways :

- 1) Direct interaction with the system.
- 2) Using a text editor to create predicates in a file, then telling the system to be directed by the file.

### 1) Direct interaction with the system :-

Type the UNIX command ' szabo ' at your terminal ; this will prompt by ' : ?- ' , and will then wait for you to start your session, using the predicates described in \$5 above.

### 2) Indirectly from a file.

In this way there two options :

#### a) Printing the results on the screen

In this option create a session of input in a file using a text editor. After you have the session of input to the system in a file (which has a name 'file\_name') you can type the command " szabo file\_name " at your terminal, it will respond by printing out the answer you are looking for.

#### b) Collecting the results in a new file :-

Using the redirection facility of UNIX, the output sent to the screen as in (a) above can be sent to a file . The command " szabo file\_name > newfile\_name " will do the process.

Example :-

Suppose the following proof has been created in a file ' ex ' as follows :

```
theory(cartel).
prove(l = [ld(a), [a] => [a]]).
prove(m = [th(l, b), [a, b] => [a]]).
prove(n = [id(b), [b] => [b]]).
prove(o = [th(n, a), [a, b] => [b]]).
prove(p = [als(m, o, a&b), [a, b] => [a&b]]).
prove(t = [h, [a] => [c]]).
prove(u = [th(t, b), [a, b] => [c]]).
prove(q = [ala(u, a&b), [a&b] => [c]]).
prove(r = [cut(p, q, a&b), [a, b] => [c]]).
prove(s = [his(r, a->c), [b] => [a->c]]).

normal(s).
```

we then use the command;

```
$ szabo ex > ex.lis
```

will read the predicates from ' ex ' and output the proof and its equivalent normal form to file ' ex.lis '.

## \$7. Input & Output Sessions with the System

In this section, we give some examples to show the user how to use the system for both proof theory and category theory.

### (7.1) Proof theory Examples

In the following example we use the predicate ' proof ' to check the correctness and assert a derivation to the system in an informal way without using the syntax of \$2 above.

#### Example 1

For  $f$  in  $X(a,b)$ ,  $g$  in  $X(a,c)$  and  $h$  in  $X(e,d)$ , for some category  $X$ , the following derivation in the class of derivations of the theory of cartesian categories :

$$\begin{array}{c}
 \begin{array}{ccc}
 a & \xrightarrow{f} & b \\
 \hline
 a & \xrightarrow{\quad} & b \uparrow c
 \end{array}
 \quad
 \begin{array}{ccc}
 a & \xrightarrow{g} & c \\
 \hline
 a, e & \xrightarrow{\quad} & d
 \end{array}
 \end{array}
 \quad
 \begin{array}{ccc}
 e & \xrightarrow{h} & d \\
 \hline
 b \uparrow c, e & \xrightarrow{\quad} & d
 \end{array}$$

can be checked and asserted in the system as follows :

```

yes
| ?- theory(cart).
yes
| ?- proof.
Enter name of the applied rule, axiom, id(a) or write end if
the proof has been finished : axiom.

Enter name of proof step : l.
Enter name of the arrow : f.
Enter the antecedent included in a list : [a].
Enter the succedent included in a list : [b].

next step
Enter name of the applied rule, axiom, id(a) or write end if
the proof has been finished : axiom.

Enter name of proof step : m.
Enter name of the arrow : g.
Enter the antecedent included in a list : [a].
Enter the succedent included in a list : [c].

```

next step  
 Enter name of the applied rule, axiom, id(a) or write end if  
 the proof has been finished : ais.

Enter name of proof step : n.

Enter name of LHS of previous steps : l.

Enter name of RHS of previous steps : m.

Enter name of ais formula : b & c.

Enter the antecedent included in a list : [a].

Enter the succedent included in a list : [b & c].

next step  
 Enter name of the applied rule, axiom, id(a) or write end if  
 the proof has been finished : axiom.

Enter name of proof step : o.

Enter name of the arrow : h.

Enter the antecedent included in a list : [e].

Enter the succedent included in a list : [d].

next step  
 Enter name of the applied rule, axiom, id(a) or write end if  
 the proof has been finished : th.

Enter name of proof step : p.

Enter name of previous step : o.

Enter name of th formula : b & c.

Enter the antecedent included in a list : [b & c, e].

Enter the succedent included in a list : [d].

next step  
 Enter name of the applied rule, axiom, id(a) or write end if  
 the proof has been finished : cut.

Enter name of proof step : q.

Enter name of LHS of previous steps : n.

Enter name of RHS of previous steps : p.

Enter name of cut formula : b & c.

Enter the antecedent included in a list : [a, e].

Enter the succedent included in a list : [d].

next step  
 Enter name of the applied rule, axiom, id(a) or write end if  
 the proof has been finished : end.

the inserted proof has been checked and is correct .

yes  
 | ?--

In the following two examples, we use the syntax of the proof-step  
 in §2 and the predicate ' prove(X) ' to check and assert derivations  
 for further use :

Example 2

Let  $h$  be an arrow in  $X(a,c)$  and  $a,b,c$  be objects of a category  $X$ , the following derivation of the class of derivations in the free cartesian closed category over  $X$  :

$$\begin{array}{c}
 \begin{array}{c}
 \text{Id}(a) \\
 \frac{A \longrightarrow A}{A, B \longrightarrow A} \\
 \hline
 A, B \longrightarrow A \uparrow B
 \end{array}
 \qquad
 \begin{array}{c}
 \text{Id}(b) \\
 \frac{B \longrightarrow B}{A, B \longrightarrow B} \\
 \hline
 A, B \longrightarrow B
 \end{array}
 \qquad
 \begin{array}{c}
 h \\
 \frac{A \longrightarrow C}{A, B \longrightarrow C} \\
 \hline
 A \uparrow B \longrightarrow C
 \end{array} \\
 \hline
 \frac{A, B \longrightarrow C}{B \longrightarrow (A \Rightarrow C)}
 \end{array}$$

is interpreted by means of an arrow of the cartesian closed category as follows :

```

yes
| ?- theory(cartcl).

yes
| ?- prove(s1 = [id(a), [a] => [a]]).

yes
| ?- prove(s2 = [th(s1,b), [a,b] => [a]]).

yes
| ?- prove(s3 = [id(b), [b] => [b]]).

yes
| ?- prove(s4 = [th(s3,a), [a,b] => [b]]).

yes
| ?- prove(s5 = [dis(s2,s4,a&b), [a,b] => [a & b]]).

yes
| ?- prove(s6 = [h, [a] => [c]]).

yes
| ?- prove(s7 = [th(s6,b), [a,b] => [c]]).

yes
| ?- prove(s8 = [ala(s7,a & b), [a & b] => [c]]).

yes
| ?- prove(s9 = [cut(s5,s8,a & b), [a,b] => [c]]).

yes
| ?- prove(s = [his(s9,a -> c), [b] => [a -> c]]).

yes
| ?- intorpr(s,K).

K = comp(hook(id(a),comp(comp(h,pi_left(a,b)),
                        comp(product(pi_left(a,b),
                                   pi_right(a,b)),
                                   delta(a#b))))),
        ita(a,b))

yes
| ?-

```

Example 3

For arrows  $f$  in  $X(x,a)$ ,  $g$  in  $X(a,c)$ ,  $h$  in  $X(a,d)$ ,  $i$  in  $X(b,c)$  and  $j$  in  $X(b,d)$  and the composition of  $g,f$  is  $k_1$  and the composition of  $h,f$  is  $k_2$  in a category  $X$ , the following derivation from the theory of bicartesian categories :

$$\begin{array}{c}
 \begin{array}{c}
 \frac{x \xrightarrow{f} a}{x,y \xrightarrow{\quad} a} \\
 \frac{x,y \xrightarrow{\quad} a,b}{x,y \xrightarrow{\quad} a,b,v} \\
 \frac{x,y \xrightarrow{\quad} a,b,v}{x,y \xrightarrow{\quad} a \vee b,v}
 \end{array}
 \qquad
 \begin{array}{c}
 \frac{a \xrightarrow{g} c \quad a \xrightarrow{h} d}{a \xrightarrow{\quad} c \uparrow d} \\
 \frac{a \xrightarrow{\quad} c \uparrow d}{a \vee b \xrightarrow{\quad} c \uparrow d}
 \end{array}
 \qquad
 \begin{array}{c}
 \frac{b \xrightarrow{i} c \quad b \xrightarrow{j} d}{b \xrightarrow{\quad} c \uparrow d} \\
 \frac{b \xrightarrow{\quad} c \uparrow d}{a \vee b \xrightarrow{\quad} c \uparrow d}
 \end{array} \\
 \hline
 x,y \xrightarrow{\quad} c \uparrow d, v
 \end{array}$$

is transformed to its equivalent normal form as follows :

```

yes
| ?- theory(bicart).
yes
| ?- composite(g,f,k1).
yes
| ?- composite(h,f,k2).
yes
| ?- prove(l1 = [f, [x] => [a]]).
yes
| ?- prove(l2 = [th(l1,y), [x,y] => [a]]).
yes
| ?- prove(l3 = [th(l2,b), [x,y] => [a,b]]).
yes
| ?- prove(l4 = [th(l3,v), [x,y] => [a,b,v]]).
yes
| ?- prove(l5 = [ois(l4,a#b), [x,y] => [a#b,v]]).
yes
| ?- prove(l6 = [g, [a] => [c]]).
yes
| ?- prove(l7 = [h, [a] => [d]]).
yes
| ?- prove(l8 = [ais(l6,l7,c & d), [a] => [c & d]]).
yes
| ?- prove(l9 = [i, [b] => [c]]).
yes
| ?- prove(l10 = [j, [b] => [d]]).
yes
| ?- prove(l11 = [ais(l9,l10,c & d), [b] => [c & d]]).
yes
| ?- prove(l12 = [ois(l8,l11,a#b), [a#b] => [c&d]]).
yes
| ?- prove(l13 = [cut(l5,l12,a#b), [x,y] => [c&d,v]]).
yes
| ?- normal(l13).

```

Proof is as follows :

```

=====
l1=[f,[x]=>[a]]
l2=[th(l1,y),[x,y]=>[a]]
l3=[th(l2,b),[x,y]=>[a,b]]
l4=[th(l3,v),[x,y]=>[a,b,v]]
l5=[ois(l4,a#b),[x,y]=>[a#b,v]]
l6=[g,[a]=>[c]]
l7=[h,[a]=>[d]]
l8=[ais(l6,l7,c&d),[a]=>[c&d]]
l9=[i,[b]=>[c]]

```

```

l10=[ ], [b] => [d]
l11=[ a!s(19, l10, c&d), [b] => [c&d] ]
l12=[ o!a(18, l11, a#b), [a#b] => [c&d] ]
l13=[ cut(15, l12, a#b), [x, y] => [c&d, v] ]

```

Cut\_free proof is :

```

=====
f22=[ k1, [x] => [c] ]
f47=[ th(f22, y), [x, y] => [c] ]
f46=[ th(f47, c), [x, y] => [c, c] ]
f40=[ th(f46, v), [x, y] => [c, c, v] ]
f31=[ k2, [x] => [d] ]
f51=[ th(f31, y), [x, y] => [d] ]
f50=[ th(f51, c), [x, y] => [d, c] ]
f43=[ th(f50, v), [x, y] => [d, c, v] ]
f34=[ a!s(f40, f43, c&d), [x, y] => [c&d, c, v] ]
f22=[ k1, [x] => [c] ]
f61=[ th(f22, y), [x, y] => [c] ]
f60=[ th(f61, d), [x, y] => [c, d] ]
f54=[ th(f60, v), [x, y] => [c, d, v] ]
f31=[ k2, [x] => [d] ]
f65=[ th(f31, y), [x, y] => [d] ]
f64=[ th(f65, d), [x, y] => [d, d] ]
f57=[ th(f64, v), [x, y] => [d, d, v] ]
f37=[ a!s(f54, f57, c&d), [x, y] => [c&d, d, v] ]
f5=[ a!s(f34, f37, c&d), [x, y] => [c&d, c&d, v] ]
l13=[ con(f5, c&d), [x, y] => [c&d, v] ]

```

Contraction and interchange free proof is :

```

=====
f22=[ k1, [x] => [c] ]
f47=[ th(f22, y), [x, y] => [c] ]
f46=[ th(f47, v), [x, y] => [c, v] ]
f31=[ k2, [x] => [d] ]
f65=[ th(f31, y), [x, y] => [d] ]
f64=[ th(f65, v), [x, y] => [d, v] ]
l13=[ a!s(f46, f64, c&d), [x, y] => [c&d, v] ]

```

Normal proof is :

```

=====
f22=[ k1, [x] => [c] ]
f47=[ th(f22, v), [x] => [c, v] ]
f31=[ k2, [x] => [d] ]
f65=[ th(f31, v), [x] => [d, v] ]
f46=[ a!s(f47, f65, c&d), [x] => [c&d, v] ]
l13=[ th(f46, y), [x, y] => [c&d, v] ]

```

```

yes
| ?-

```

which in tree form is :

$$\begin{array}{c}
 \begin{array}{ccc}
 & k1 & \\
 x & \longrightarrow & c \\
 \hline
 x & \longrightarrow & c, v
 \end{array}
 \qquad
 \begin{array}{ccc}
 & k2 & \\
 x & \longrightarrow & d \\
 \hline
 x & \longrightarrow & d, v
 \end{array} \\
 \hline
 x & \longrightarrow & c \uparrow d, v \\
 \hline
 x, y & \longrightarrow & c \uparrow d, v
 \end{array}$$

## (7.2) Category theory examples

In this subsection, we give some examples; in order to use the system to automate reasoning about category theory.

Example 1

For  $f$  in  $X(a,b)$ , for some category  $X$ , the normal representation of the arrow :

$$h = \text{comp}(\text{comp}(\text{pi\_right\_star}(a,b),f),\text{pi\_left}(a,b))$$

In the following diagram :

$$a \uparrow b \xrightarrow{\text{pi\_left}(a,b)} a \xrightarrow{f} b \xrightarrow{\text{pi\_right\_star}(a,b)} a \vee b$$

can be obtained as follows :

```
yes
| ?- theory(bicart).
```

```
yes
| ?- arrow(a,f,b).
```

```
yes
| ?- norm_rep(comp(comp(pi_right_star(a,b),f),
                    pi_left(a,b))).
```

The arrow is :

```
=====
comp(comp(pi_right_star(a,b),f),pi_left(a,b))
```

Proof is as follows :

```
=====
f9=[id(a),[a]=>[a]]
f8=[th(f9,b),[a,b]=>[a]]
f7=[aia(f8,a&b),[a&b]=>[a]]
f6=[f,[a]=>[b]]
f5=[id(b),[b]=>[b]]
f4=[th(f5,a),[b]=>[a,b]]
f3=[ois(f4,a#b),[b]=>[a#b]]
f2=[cut(f6,f3,b),[a]=>[a#b]]
f1=[cut(f7,f2,a),[a&b]=>[a#b]]
```

Cut\_free proof is :

```
=====
f27=[f,[a]=>[b]]
f24=[th(f27,b),[a,b]=>[b]]
f21=[aia(f24,a&b),[a&b]=>[b]]
f18=[th(f21,a),[a&b]=>[a,b]]
f1=[ois(f18,a#b),[a&b]=>[a#b]]
```

Contraction and interchange free proof is :

```
=====
f27=[f,[a]=>[b]]
f24=[th(f27,b),[a,b]=>[b]]
f21=[aia(f24,a&b),[a&b]=>[b]]
f18=[th(f21,a),[a&b]=>[a,b]]
f1=[ois(f18,a#b),[a&b]=>[a#b]]
```

Normal proof is :

```
=====
f27=[f,[a]=>[b]]
f24=[th(f27,a),[a]=>[a,b]]
f21=[ois(f24,a#b),[a]=>[a#b]]
f18=[th(f21,b),[a,b]=>[a#b]]
f1=[aia(f18,a&b),[a&b]=>[a#b]]
```

```
yes
| ?-
```





Proof is as follows :

```

=====
f379= [id(a), [a]=>[a]]
f378= [th(f379, b), [a, b]=>[a]]
f377= [aia(f378, a&b), [a&b]=>[a]]
f395= [id(b), [b]=>[b]]
f394= [th(f395, a), [a, b]=>[b]]
f393= [aia(f394, a&b), [a&b]=>[b]]
f391= [id(a), [a]=>[a]]
f389= [th(f391, b), [a, b]=>[a]]
f392= [id(b), [b]=>[b]]
f390= [th(f392, a), [a, b]=>[b]]
f388= [aia(f389, f390, a&b), [a, b]=>[a&b]]
f386= [id(a), [a]=>[a]]
f385= [th(f386, b), [a, b]=>[a]]
f384= [aia(f385, a&b), [a&b]=>[a]]
f383= [f, [a]=>[c]]
f382= [cut(f384, f383, a), [a&b]=>[c]]
f387= [cut(f388, f382, a&b), [a, b]=>[c]]
f381= [his(f387, a->c), [b]=>[a->c]]
f380= [cut(f393, f381, b), [a&b]=>[a->c]]
f376= [aia(f377, f380, a&(a->c)), [a&b]=>[a&(a->c)]]
f374= [id(a), [a]=>[a]]
f375= [id(c), [c]=>[c]]
f373= [hia(f374, f375, a->c), [a, a->c]=>[c]]
f372= [aia(f373, a&(a->c)), [a&(a->c)]=>[c]]
f371= [cut(f376, f372, a&(a->c)), [a&b]=>[c]]

```

Cut\_free proof is :

```

=====
f582= [f, [a]=>[c]]
f579= [th(f582, b), [a, b]=>[c]]
f576= [aia(f579, a&b), [a&b]=>[c]]
f573= [th(f576, b), [a&b, b]=>[c]]
f570= [th(f573, a), [a&b, a, b]=>[c]]
f567= [aia(f570, a&b), [a&b, a&b]=>[c]]
f564= [th(f567, b), [a&b, b, a&b]=>[c]]
f561= [th(f564, a), [a&b, a, b, a&b]=>[c]]
f558= [aia(f561, a&b), [a&b, a&b, a&b]=>[c]]
f555= [th(f558, a), [a&b, a&b, a, a&b]=>[c]]
f552= [th(f555, b), [a&b, a&b, a, b, a&b]=>[c]]
f537= [aia(f552, a&b), [a&b, a&b, a&b, a&b]=>[c]]
f532= [con(f537, a&b), [a&b, a&b, a&b]=>[c]]
f474= [con(f532, a&b), [a&b, a&b]=>[c]]
f371= [con(f474, a&b), [a&b]=>[c]]

```

Normal proof is :

```

=====
f582= [f, [a]=>[c]]
f579= [th(f582, b), [a, b]=>[c]]
f371= [aia(f579, a&b), [a&b]=>[c]]

```

Insert the second arrow : comp(f, pi\_left(a, b)).

The arrow is :

```

=====
comp(f, pi_left(a, b))

```

Proof is as follows :

```

=====
f587= [id(a), [a]=>[a]]
f586= [th(f587, b), [a, b]=>[a]]
f585= [aia(f586, a&b), [a&b]=>[a]]
f584= [f, [a]=>[c]]
f583= [cut(f585, f584, a), [a&b]=>[c]]

```

Cut\_free proof is :

```

=====
f593= [f, [a]=>[c]]
f590= [th(f593, b), [a, b]=>[c]]
f583= [aia(f590, a&b), [a&b]=>[c]]

```

Normal proof is :

=====

```
f593=[f,[a]=>[c]]
f590=[th(f593,b),[a,b]=>[c]]
f583=[aia(f590,a&b),[a&b]=>[c]]
```

The answer is :

=====

```
comp(epsilon(a,c),
      angle(pi_left(a,b),
            comp(alpha_right(comp(f,pi_left(a,b))),
                  pi_right(a,b))))
= comp(f,pi_left(a,b))
```

because they have the same normal representation .

yes  
| ?-

## Appendix 6

=====

## Predicate Index

=====

```

6606  add_alla(N,S,F,Z1,T)
6634  add_alla(N,S,F,K,E)
  850  alpha(A,B,C,Z)
  854  alpha_inverse(A,B,C,Z)
1027  alpha_right(U,Z)
  929  alpha_star(A,B,C,Z)
  934  alpha_star_inverse(A,B,C,Z)
  836  angle(U,V,Z)
6710  append([M|L1],L2,[M|L3])
6712  append3(X,Y,Z,U)
6715  append4(X,Y,Z,V,U)
6718  append5(X,Y,Z,U,V,W)
  765  ar(F,Z)
  761  arrow(A,F,B)
4190  ass_list([X|Y])
  162  binar(X,X1)
6671  binary(X, L, M)
6761  bra_binar(Z,Z1,Z2)
6747  bra_nul(Z)
6750  bra_unar(Z,Z1)
6742  branch(Z)
6429  break_up(E1,L1,L2,E,S,F)
6534  break_up_con(E1,L,E,S,F)
6628  break_up_th(E,L,E1,S,F)
  745  clean
  665  commutative
  828  comp(V,U,Z)
  763  composite(G,F,H)
2458  con_inc_free(X)
6724  cop
1136  cut_elm(proved(Z = [cut(Z1,Z2,B), G => T]))
1095  cut_free(Z)
  843  delta(A,Z)
  981  delta_left(A,B,C,Z)
  988  delta_left_star(A,B,C,Z)
  984  delta_right(A,B,C,Z)
  998  delta_right_star(A,B,C,Z)
  900  delta_star(A,Z)
2474  do1_step
4175  do2_step
1044  epsilon(A,B,Z)
1017  epsilon_left(U,V,Z)
6461  expcon
6478  expcona
6334  expinc
6375  expinca
6413  expincs
2436  expinti(X1,Y)
2364  expterm(X1, Y)
6572  expth
6589  exptha
6613  expths
6519  expcons
6685  gensym(Root,Atom)
6695  get_num(Root,Num)
   9  help
   71  help_arrow
  102  help_clean
   86  help_commutative
   28  help_composite
   56  help_con_inc_free
   51  help_cut_free
   45  help_interpr
   97  help_inter_norm_rep
  107  help_new
   64  help_normal
   79  help_norm_rep
   38  help_proof
   33  help_prove
   76  help_rep_of
   19  help_theory
1067  hook(U,V,Z)

```

```

2285 hook_termin(Z = [hia(Z1,Z2,A -> B), G => T])
2278 hook_terminal
768 ld(K,X)
140 inc(X,X1)
2399 initial(X = [L, G => T])
6700 integer_name(Int,List)
6701 integer_name(I,Sofar,[C|Sofar])
278 inter_norm_rep(X)
285 interpr(Z,H)
6723 islist([A|B])
2418 isom_bo(X)
1053 ita(A,B,Z)
2343 isom_tr(X)
1003 kappa(A,B,C,D,E,Z)
6679 member(H,[_|L])
6393 move_over(N,M,S,F,Z1,T)
6495 move_over_all(L,N,S,F,Z1,T)
6545 move_over_all(L,N,S,F,Z1,E)
741 new
725 norm_rep(Y)
2486 norm1(proved(Z))
4194 norm2(proved(Z))
1077 normal(Z)
2465 normal1
4167 normal2
135 nul(X,X1)
6659 nullary(X)
1109 one_stop
1069 part(X,Z)
638 pi(Z,L)
813 pi_left(A,B,X)
886 pi_left_star(A,B,X)
820 pi_right(A,B,X)
893 pi_right_star(A,B,X)
6645 prinlist(X,L)
6647 prinlist(X, Sofar, K)
6654 printall(X)
860 product(U,V,Z)
114 proof
178 prove(X)
1120 purge
6643 put_in_list(X,L)
1128 reassert_all(X)
748 rep_of(Y)
660 rmcomp(H1,H)
948 round(U,V,Z)
181 rule(X)
873 sigma(A,B,Z)
876 sigma_inverse(A,B,Z)
939 sigma_star(A,B,Z)
942 sigma_star_inverse(A,B,Z)
6404 singlemove(N,[A|M1],S,F,Z1,T)
6508 singlemove(L1,A,L2,[A|M],S,F,Z1,T)
6449 singlemoves(N,[A|M1],S,F,K,E)
6558 singlemoves(L1,A,L2,[A|M],S,F,K,E)
908 square(U,V,Z)
6681 sub(X,C,[X|L],[C|Li])
808 tau(A,X)
881 tau_star(A,X)
2312 term_inti
2319 terminal(X = [L, G => T])
176 theory(X)
2307 thinning(Z2,B,Delta,Lambda)
687 tree(X,L)
689 tree(X, Sofar, K)
6662 unary(X,L)
649 un(Z,L)
151 unar(X,X1)
916 union(U,V,Z)

```

Appendix 7Maintenance Information

In this appendix, we give some information to help in maintaining the system. The system can be run by typing the command " szabo ", where szabo is a executable consisting of the line Prolog szabo.sav (together with instruction setting the heap size etc ). In order to added or change some rules in the system, edit the text " szabo.pro " using . Then invoke Prolog and compile the program using the directive:

```
|?- ['szabo.pro'].
```

which will instruct the Prolog-interpretor to consult the program. Then save the compiled form of the program into the file " szabo.sav ", by performing the command :

```
|?- save('szabo.sav').
```

Then exit from Prolog, the " szabo " command can now be used as before.

## Appendix 8

## Program Listing

```

1 :- op(30, fx, '\').
2 :- op(100, xfy, '#').
3 :- op(100, xfy, '&').
4 :- op(150, xfy, '->').
5 :- op(200, xfy, '=>').
6 :- op(150, xfy, '<->').
7 :- op(250, xfy, ':').

8                                     /* doc */

9 help :-
10  write('Commands'),nl,
11  write('====='),nl,nl,
12  write('1- theory          2- composite 3- prove'),nl,
13  write('4- proof          5- interpr  6- cut_free'),nl,
14  write('7- con_inc_free    8- normal  9- arrow '),nl,
15  write('10- rep_of        11- norm_rep 12- commutative'),nl,
16  write('13- inter_norm_rep 14- clean  15- new '),nl,nl,
17  write('Help on any of these commands may be had by typing,').
18  nl,write('help_* where * is one of the above commands.').

19 help_theory :-
20  write('theory(X) : Tells the system that the theory we intend'),
21  nl, write(' to work with is X, where X represents one of the '),
22  nl, write(' following abbreviations cart (cartesian), bicart
23  _____ '),
24  nl, write(' (bicartesian), dbicart (distributive bicartesian),
25  _____ '),nl,
26  write(' cartcl (cartesian closed) or bicartcl (bicartesian closed).
27  _____ '),nl.

28 help_composita :-
29  write('composite(F,G,H) : This predicate is used, when arrows'),
30  nl, write(' F,G of the underlying category X with composition'),
31  nl, write(' H are used in the construction of the derivation '),
32  nl, write(' in operation. '),nl.

33 help_prove :-
34  write('prove(X) : Checks the correctness of the current instance'),
35  nl, write(' of a proof-step X; if it is correct, it will be added'),
36  nl, write(' to the database for further use. We keep using this'),
37  nl, write(' command until the proof is complete. '),nl.

38 help_proof :-
39  write('proof : This command is used, as another way to check the'),
40  nl, write(' correctness and assert derivations without using the'),
41  nl, write(' formal syntax for proof-step in $2 and the predicate'),
42  nl, write(' " prove(X) " above. It will switch on the system to'),
43  nl, write(' communicate with the user to assert the derivation'),
44  nl, write(' in informal way. '),nl.

45 help_interpr :-
46  write('interpr(X,K) : After asserting a derivation, this command'),
47  nl,write(' is used to compute the interpretation of this derivation'),
48  nl,write(' as an arrow in the free category. X represents the name'),
49  nl,write(' of the last proof-step and M will be instantiated as'),
50  nl,write(' the arrow computed to interpret the derivation. '),nl.

51 help_cut_free :-
52  write('cut_free(X) : Tells the system to start the processes'),
53  nl, write(' of transforming the derivation in question to its'),
54  nl, write(' equivalent cut-free one. X represents the name'),nl,
55  write(' of the last proof-step. '),nl.

```

```

56 help_con_inc_free :-
57   write('con_inc_free(X) : where X represents the name of the'),
58   nl, write('last stop of a cut-free proof. This predicate cause'),
59   nl, write('the system to start the processes of transforming '),
60   nl, write('the cut-free proof into its equivalent contraction &'),
61   nl, write('interchange-free one. This is used only if we dealing'),
62   nl, write('with the theories of cart, bicart or dbicart.
63   ----- '),nl.

64 help_normal :-
65   write('normal(X) :This predicate is a combination of three'),nl,
66   write('processes: cut-elimination, contraction & interchange-'),nl,
67   write('elimination and normalization; so that this general'),nl,
68   write('command is used in place of cut_free and con_inc_free'),nl,
69   write('to produce the equivalent normal form for a derivation. '),nl,
70   write('X represents the name of the last stop of the derivation. '),nl.

71 help_arrow :-
72   write('arrow(A,F,B) : Insert all arrows, of the underling category'),
73   nl, write('X, being used in the construction of the arrow of '),
74   nl, write('the free category in question. F represents an '),
75   nl, write('arrow from A to B ( F : A -> B ). '),nl.

76 help_rep_of :-
77   write('rep_of(X) : Will produce a representation of an arrow'),
78   nl, write('X as a derivation. '),nl.

79 help_norm_rep :-
80   write('norm_rep(X) : If X represents an arrow of some free'),nl,
81   write('category, this predicate causes the system to produce '),
82   nl, write('a representation of X by means of a derivation. '),nl,
83   write(' then calling the processes of cut-elimination and '),nl,
84   write(' normalization to produce the corresponding normal '),nl,
85   write(' representation. '),nl.

86 help_commutative :-
87   write('commutative : Tests whether two arrows are equal or not. '),
88   nl, write('In order to test the commutativity of the diagrams '),
89   nl, write('in category theory. After reading this command, the '),
90   nl, write('system waits to read the LHS arrow of the equality '),
91   nl, write('in question and responds by computing its normal '),nl,
92   write('representation. The system waits again to read the '),nl,
93   write('RHS arrow of the equality and responds by computing '),nl,
94   write('its normal representation. Finally it compares these '),nl,
95   write('two normal representations and responds with the '),nl,
96   write('conclusion. '),nl.

97 help_inter_norm_rep :-
98   write('inter_norm_rep(X): This is a combination of two processes;'),
99   nl, write('one for normal representation of the arrow X, the other'),
100  nl, write('is to interpret this normal representation to an arrow'),
101  nl, write('of the same theory of the arrow X. '),nl.

102 help_clean :-
103   write('clean : This command is to be used after each session of'),
104   nl, write('use, in order to use the system using the same theory'),
105   nl, write('and the same arrows, if any, of the underling category'),
106   nl, write('for another session. '),nl.

107 help_new :-
108   write('new : Cleans up the database from all the processes of'),
109   nl, write('the last session and puts the system again to the top
110   level. '),nl.

111                                     /* Proof Checker Program */

112 /* This program is to help the user to insert the proof in more
113    detail rather than in the list form */

114 proof :-
115   write('Enter name of the applied rule, axiom, id(a) or write'),
116   nl, write('end if the proof has been finished : '),
117   read(X),
118   ((X = end, nl,

```

```

119     write('the inserted proof has been checked and is correct. '),nl)
120 ;(write('Enter name of proof step : '),
121     read(Y),
122     ((X =.. [id,A],
123     prove(Y = [id(A), [A] => [A]]) )
124 ;((nul(X,X1)
125     ;unar(X,X1)
126     ;inc(X,X1)
127     ;binar(X,X1) ),
128     write('Enter the antecedent included in a list : '),
129     read(A),
130     write('Enter the succedent included in a list : '),
131     read(S),
132     prove(Y = [X1, A => S]) ) ) ).
133 nl,nl, write('next step'),nl,
134 proof) ).

```

```

135 nul(X,X1) :-
136     X = axiom,
137     write('Enter name of the arrow : '),
138     read(Z),
139     X1 = Z.

```

```

140 inc(X,X1) :-
141     X = inc,
142     write('Enter name of previous step : '),
143     read(Y),
144     write('Enter name of the first formula of the two
145     interchangeable formulae as appear in this step : '),
146     read(L),
147     write('Enter name of the second formula of the two
148     interchangeable formulae as appear in this step : '),
149     read(M),
150     X1 = inc(Y,L,M).

```

```

151 unar(X,X1) :-
152     (X = th
153     ;X = con
154     ;X = aia
155     ;X = ois
156     ;X = his),
157     write('Enter name of previous step : '),
158     read(Y),
159     write('Enter name of '), write(X), write(' formula : '),
160     read(C),
161     X1 =.. [X,Y,C].

```

```

162 binar(X,X1) :-
163     (X = cut
164     ;X = ais
165     ;X = oia
166     ;X = hia),
167     write('Enter name of LHS of previous steps : '),
168     read(Z1),
169     write('Enter name of RHS of previous steps : '),
170     read(Z2),
171     write('Enter name of '), write(X), write(' formula : '),
172     read(C),
173     X1 =.. [X,Z1,Z2,C].

```

```

174 /* This is a proof checker program using the syntax of the
175 proof-step */
176 theory(X) :-
177     assert(thry(X)).

```

```

178 prove(X) :- X = ( Y = Z ),
179     rule(X),
180     assert(proved(X)).

```

```

181 rule(X) :- X = ( L = [Y,[A] => [B]] ),
182     (Y =.. [_] ; Y =.. [id,_]),
183     !.

```

```

184 rule(X) :- X = ( L = [true, [] => [tr]] ),
185     !.

```



```

186 rule(X) :- X = ( L = [bottom, [bo] => []]),
187     !.

188 rule(X) :- X = ( L = [ th(M,A), G => T]),
189     !,
190     ((proved(M = [_ , G1 => T]),
191     append(Gamma, [A|Delta], G),
192     append(Gamma, Delta, G1))
193     ;(proved(M = [_ , G => T1]),
194     append(Phi, [A|Psi], T),
195     append(Phi, Psi, T1),
196     (thry(bicart); thry(dbicart); thry(bicartcl)))).

197 rule(X) :- X = ( L = [con(M,A), G => T]),
198     !,
199     ((proved(M = [_ , G1 => T]),
200     append(Gamma, [A|[A|Delta]], G1),
201     append(Gamma, [A|Delta], G))
202     ;(proved(M = [_ , G => T1]),
203     append(Phi, [A|[A|Psi]], T1),
204     append(Phi, [A|Psi], T),
205     (thry(bicart); thry(dbicart); thry(bicartcl)))).

206 rule(X) :- X = ( L = [inc(M,B,A), G => T]),
207     !,
208     ((proved(M = [_ , G1 => T]),
209     append(Gamma, [B|[A|Delta]], G),
210     append(Gamma, [A|[B|Delta]], G1))
211     ;(proved(M = [_ , G => T1]),
212     append(Phi, [B|[A|Psi]], T),
213     append(Phi, [A|[B|Psi]], T1),
214     (thry(bicart); thry(dbicart); thry(bicartcl)))).

215 rule(X) :- X = ( L = [cut(M,N,A), E => F]),
216     !,
217     proved(M = [_ , Gamma => T1]),
218     proved(N = [_ , G1 => Theta]),
219     append(Delta, [A|Lambda], G1),
220     append3(Delta, Gamma, Lambda, G),
221     append(Phi, [A|Psi], T1),
222     append3(Phi, Theta, Psi, T),
223     ((thry(cart) ; thry(cartcl) ; thry(bicart)),
224     Phi = [], Psi = [])
225     ;(thry(bicart), Delta = [], Lambda = [])
226     ;thry(dbicart)
227     ;thry(bicartcl)).

228 rule(X) :- X = ( L = [ais(M,N,A & B),G => T]),
229     !,
230     proved( M = [_ , G => T1]),
231     proved( N = [_ , G => T2]),
232     append(Phi, [A|Psi], T1),
233     append(Phi, [B|Psi], T2),
234     append(Phi, [A & B|Psi], T),
235     ((thry(cart); thry(bicart); thry(cartcl)),
236     Phi = [], Psi = [])
237     ;thry(dbicart)
238     ;thry(bicartcl)).

239 rule(X) :- X = ( L = [aia(M,A & B),G => T]),
240     !,
241     proved(M = [_ ,G1 => T]),
242     append(Gamma, [A & B|Delta], G),
243     append(Gamma, [A|[B|Delta]], G1).

244 rule(X) :- X = ( L = [oia(M,N,A # B), G => T]),
245     !,
246     proved( M = [_ ,G1 => T]),
247     proved( N = [_ ,G2 => T]),
248     append(Gamma, [A # B|Delta], G),
249     append(Gamma, [A|Delta], G1),
250     append(Gamma, [B|Delta], G2),
251     ((thry(bicart), Gamma = [], Delta = [])
252     ;thry(dbicart)
253     ;thry(bicartcl)).

```

```

254 rule(X) :- X = ( L = [ois(M,A # B), G => T]),
255     |,
256     proved(M = [_, G => T1]),
257     append(Phi, [A # B|Psi], T),
258     append(Phi, [A|[B|Psi]], T1),
259     (thry(bicart) ; thry(dbicart) ; thry(bicartcl)).

260 rule(X) :- X = ( L = [his(M,A -> B), G => [A -> B]]),
261     |,
262     proved( M = [_, G1 => [B]]),
263     append(Gamma, [A|Delta], G1),
264     append(Gamma, Delta, G),
265     (thry(cartcl) ; thry(bicartcl)).

266 rule(X) :- X = ( L = [hia(M,N,A -> B), G => T]),
267     |,
268     proved( M = [_, Gamma => [A]]),
269     proved( N = [_, G1 => T]),
270     append(Gamma, [A -> B], G2),
271     append(Delta, [B|Lambda], G1),
272     append3(Delta, G2, Lambda, G),
273     ((thry(cartcl), T = [_])
274     ; thry(bicartcl)).

275                                     /* interpr */

276 /* this part is to interpret sequent calculus proofs as
277    arrows of category theory */

278 inter_norm_rep(X) :-
279     norm_rep(X),
280     main(Y),
281     interpr(Y,H),
282     write('The irreducible arrow is :'), nl,
283     write('===== '), nl, nl,
284     write(H).

285 interpr(Z,H) :-
286     proved(Z = [F, [A] => [B]]),
287     (F == [_] ; F == [id,A]),
288     |,
289     H = F.

290 interpr(Z,H) :-
291     proved(Z = [true, [] => [tr]]),
292     |,
293     H = Id(tr).

294 interpr(Z,H) :-
295     proved(Z = [cut(Z1,Z2,C), G => T]),
296     proved(Z1 = [_, Gamma => T1]),
297     proved(Z2 = [_, G1 => Theta]),
298     append(Delta, [C|Lambda], G1),
299     append3(Delta,Gamma,Lambda,G),
300     append(Phi, [C|Psi], T1),
301     append3(Phi, Theta, Psi, T),
302     |,
303     interpr(Z1,H1),
304     interpr(Z2,H2),
305     pi(Gamma, Gamma1),
306     pi(Delta, Delta1),
307     pi(Lambda, Lambda1),
308     un(Phi, Phi1),
309     un(Psi, Psi1),
310     un(Theta, Theta1),
311     ((Phi = [], Psi = []),
312     (Delta = [], Lambda = [],
313     H3 = comp(H2,H1) )
314     ;(Delta == [], Lambda \== [],
315     H3 = comp(H2, product(H1, id(Lambda1))) )
316     ;(Lambda == [], Delta \== [],
317     H3 = comp(H2, product(id(Delta1), H1)) )
318     ;(Delta \== [], Lambda \== [],
319     H3 = comp(H2, product(product(id(Delta1),H1),id(Lambda1))))))
320     ;(Delta = [], Lambda = [],

```

```

321     (Phi = [], Psi \== [],
322     H3 = comp(union(H2, id(Psi1)), H1) )
323 ;(Psi = [], Phi \== [],
324     H3 = comp(union(id(Phi1), H2), H1) )
325 ;(Phi \== [], Psi \== [],
326     H3 = comp(union(union(id(Phi1), H2), id(Psi1)), H1) )
327 ;(Phi = [], Delta = [], Psi \== [], Lambda \== [],
328     H3 = comp(comp(union(H2, pi_left(Psi1, Lambda1)),
329     delta_right(C, Psi1, Lambda1)),
330     product(H1, id(Lambda1))) )
331 ;(Phi = [], Lambda = [], Psi \== [], Delta \== [],
332     H3 = comp(comp(union(H2, pi_right(Delta1, Psi1)),
333     delta_left(Delta1, C, Psi1)),
334     product(id(Delta1), H1)) )
335 ;(Psi = [], Delta = [], Phi \== [], Lambda \== [],
336     H3 = comp(comp(union(pi_left(Phi1, Lambda1), H2),
337     delta_right(Phi1, C, Lambda1)),
338     product(H1, id(Lambda1))) )
339 ;(Psi = [], Lambda = [], Phi \== [], Delta \== [],
340     H3 = comp(comp(union(pi_right(Delta1, Phi1), H2),
341     delta_left(Delta1, Phi1, C)),
342     product(id(Delta1), H1)) )
343 ;(Phi = [], Psi \== [], Delta \== [], Lambda \== [],
344     H3 = comp(comp(comp(comp(union(id(Theta1), pi_right(Delta1, Psi1)),
345     union(H2, pi_left(Delta1&Psi1, Lambda1))),
346     delta_right(Delta1&C, Delta1&Psi1, Lambda1)),
347     product(delta_left(Delta1, C, Psi1), id(Lambda1))),
348     product(product(id(Delta1), H1), id(Lambda1))) )
349 ;(Psi = [], Phi \== [], Delta \== [], Lambda \== [],
350     H3 = comp(comp(comp(comp(union(pi_right(Delta1, Phi1), id(Theta1)),
351     union(pi_left(Delta1&Phi1, Lambda1), H2))),
352     delta_right(Delta1 & Phi1, Delta1&C, Lambda1)),
353     product(delta_left(Delta1, Phi1, C), id(Lambda1))),
354     product(product(id(Delta1), H1), id(Lambda1))) )
355 ;(Delta = [], Phi \== [], Psi \== [], Lambda \== [],
356     H3 = comp(comp(comp(union(union(pi_left(Phi1, Lambda1), H2), id(Psi1)),
357     union(delta_right(Phi1, C, Lambda1),
358     pi_left(Psi1, Lambda1))),
359     delta_right(Phi1 & C, Psi1, Lambda1)),
360     product(H1, id(Lambda1))) )
361 ;(Lambda = [], Phi \== [], Psi \== [], Delta \== [],
362     H3 = comp(comp(comp(union(union(pi_right(Delta1, Phi1), H2), id(Psi1)),
363     union(delta_left(Delta1, Phi1, C),
364     pi_right(Delta1, Psi1))),
365     delta_left(Delta1, Phi1 & C, Psi1)),
366     product(id(Delta1), H1)) )
367 ;(Phi \== [], Psi \== [], Delta \== [], Lambda \== [],
368     H3 = comp(comp(union(union(id(Phi1), H2), id(Psi1)),
369     kappa(Delta1, Phi1, C, Psi1, Lambda1)),
370     product(product(id(Delta1), H1), id(Lambda1))))),
371     rmcomp(H3, H).

```

```

372 Interpr(Z, H) :-
373     proved(Z = [th(Z1, A), G => T]),
374     proved(Z1 = [_, G1 => T]),
375     append(Gamma, [A|Delta], G),
376     append(Gamma, Delta, G1).
377
378     !,
379     Interpr(Z1, H1),
380     pi(Gamma, Gamma1),
381     pi(Delta, Delta1),
382     ((Gamma = [],
383     H2 = comp(H1, pi_right(A, Delta1)))
384 ;(Delta = [],
385     H2 = comp(H1, pi_left(Gamma1, A)))
386 ; H2 = comp(H1, product(pi_left(Gamma1, A), id(Delta1))))),
387     rmcomp(H2, H).

```

```

388 Interpr(Z, H) :-
389     proved(Z = [als(Z1, Z2, A & B), G => T]),
390     proved(Z1 = [_, G => T1]),
391     proved(Z2 = [_, G => T2]),
392     append(Phi, [A & B|Psi], T),
393     append(Phi, [A|Psi], T1),
394     append(Phi, [B|Psi], T2),
395
396     !,
397     Interpr(Z1, H1),
398     Interpr(Z2, H2),
399     un(Phi, Phi1),
400     un(Psi, Psi1),
401     un(G, G1).

```

```

400 ((Phi = [],
401 Psi = []),
402 H3 = comp(product(H1,H2),delta(G1)))
403 ;(Phi = [],
404 H3 = comp(comp(delta_right_star(A,B,Psi1),product(H1,H2)),
405 delta(G1)))
406 ;(Psi = [],
407 H3 = comp(comp(delta_left_star(Phi1,A,B),product(H1,H2)),
408 delta(G1)))
409 ; H3 = comp(comp(comp(product(delta_left_star(Phi1,A,B),id(Psi1)),
410 delta_right_star(Phi1 # A, Phi1 # B, Psi1)),
411 product(H1,H2)),
412 delta(G1))),
413 rmcomp(H3,H).

414 interpr(Z,H) :-
415 proved(Z = [con(Z1,A), G => T]),
416 proved(Z1 = [_, G1 => T]),
417 append(Gamma, [A|[A|Delta]], G1),
418 append(Gamma, [A|Delta], G), .
419 |,
420 interpr(Z1,H1),
421 pi(Gamma, Gamma1),
422 pi(Delta, Delta1),
423 ((Gamma = [], Delta = [],
424 H2 = comp(H1, delta(A)) )
425 ;(Gamma = [],
426 H2 = comp(H1, product(delta(A), id(Delta1))) )
427 ;(Delta = [],
428 H2 = comp(comp(H1, alpha(Gamma1, A, A)),
429 product(id(Gamma1), delta(A))) )
430 ; H2=comp(comp(H1, product(alpha(Gamma1,A,A),id(Delta1))),
431 product(product(id(Gamma1),delta(A)),id(Delta1))))),
432 rmcomp(H2,H).

433 interpr(Z,H) :-
434 proved(Z = [aia(Z1,A & B), G => T]),
435 proved(Z1 = [_, G1 => T]),
436 append(Gamma, [A & B|Delta], G),
437 append(Gamma, [A|[B|Delta]], G1),
438 |,
439 interpr(Z1,H1),
440 pi(Gamma, Gamma1),
441 pi(Delta, Delta1),
442 ((Gamma = [],
443 Delta = [],
444 H = H1)
445 ;(Gamma = [],
446 H2 = comp(H, alpha_inverse(A,B,Delta1)))
447 ;(Delta = [],
448 H2 = comp(H1,alpha(Gamma1,A,B)))
449 ; H2 = comp(H1,product(alpha(Gamma1,A,B),id(Delta1))))),
450 rmcomp(H2,H).

451 interpr(Z,H) :-
452 proved(Z = [bottom, [bo] => []]),
453 |,
454 H = id(bo).

455 interpr(Z,H) :-
456 proved(Z = [th(Z1,A), G => T]),
457 proved(Z1 = [_, G => T1]),
458 append(Psi, [A|Psi], T),
459 append(Psi, Psi, T1),
460 |,
461 interpr(Z1,H1),
462 un(Psi, Psi1),
463 un(Psi, Psi1),
464 ((Phi = [],
465 H2 = comp(pi_right_star(A,Psi1), H1))
466 ;(Psi = [],
467 H2 = comp(pi_left_star(Phi1,A), H1))
468 ; H2 = comp(union(pi_left_star(Phi1,A), id(Psi1)), H1)),
469 rmcomp(H2,H).

470 interpr(Z,H) :-
471 proved(Z = [con(Z1,A), G => T]),
472 proved(Z1 = [_, G => T1]),

```

```

473     append(Phi, [A|[A|Psi]], T1),
474     append(Phi, [A|Psi], T),
475   |,
476   |,
477   |,
478   |,
479   |,
480   |,
481   |,
482   |,
483   |,
484   |,
485   |,
486   |,
487   |,
488   |,
489   |,
490   |,
491   |,
492   |,

```

```

493 Interpr(Z,H) :-
494   proved(Z = [ois(Z1,Z2,A # B), G => T]),
495   proved(Z1 = [-, G1 => T]),
496   proved(Z2 = [-, G2 => T]),
497   append(Gamma, [A # B|Delta], G),
498   append(Gamma, [A|Delta], G1),
499   append(Gamma, [B|Delta], G2),
500   |,
501   |,
502   |,
503   |,
504   |,
505   |,
506   |,
507   |,
508   |,
509   |,
510   |,
511   |,
512   |,
513   |,
514   |,
515   |,
516   |,
517   |,
518   |,

```

```

519 Interpr(Z,H) :-
520   proved(Z = [ois(Z1,A # B), G => T]),
521   proved(Z1 = [-, G => T]),
522   append(Phi, [A # B|Psi], T),
523   append(Phi, [A|[B|Psi]], T1),
524   |,
525   |,
526   |,
527   |,
528   |,
529   |,
530   |,
531   |,
532   |,
533   |,
534   |,
535   |,
536   |,

```

```

537 Interpr(Z,H) :-
538   proved(Z = [inc(Z1,B,A), G => T]),
539   proved(Z1 = [-, G1 => T]),
540   append(Gamma, [B|[A|Delta]], G),
541   append(Gamma, [A|[B|Delta]], G1),
542   |,
543   |,
544   |,
545   |,
546   |,
547   |,
548   |,
549   |,

```

```

550   H2 = comp(H1, product(sigma_inverse(A,B), id(Delta1)))
551   ; (Delta = []).
552   H2 = comp(comp(H1, alpha(Gamma1, A, B)),
553             product(id(Gamma1), sigma_inverse(A, B))),
554             alpha_inverse(Gamma1, B, A))
555   ; H2 = comp(comp(H1, product(alpha(Gamma1, A, B), id(Delta1))),
556             product(product(id(Gamma1), sigma_inverse(A, B)),
557                     id(Delta1))),
558             product(alpha_inverse(Gamma1, B, A), id(Delta1))),
559   rmcomp(H2, H).

```

```

560   Interpr(Z, H) :-
561     proved(Z = [inc(Z1, B, A), G => T]),
562     proved(Z1 = [_, G => T1]),
563     append(Phi, [B | [A | Psi]], T),
564     append(Phi, [A | [B | Psi]], T1).
565   |
566     Interpr(Z1, H1),
567     un(Phi, Phi1),
568     un(Psi, Psi1),
569     ((Phi = [], Psi = []),
570      H2 = comp(sigma_star(A, B), H1) )
571   ; (Phi = []).
572   H2 = comp(union(sigma_star(A, B), id(Psi1)), H1) )
573   ; (Psi = []).
574   H2 = comp(comp(comp(alpha_star(Phi1, B, A),
575                     union(id(Phi1), sigma_star(A, B))),
576                     alpha_star_inverse(Phi1, A, B)),
577             H1) )
578   ; H2 = comp(comp(comp(union(alpha_star(Phi1, B, A), id(Psi1)),
579                     union(union(id(Phi1), sigma_star(A, B)),
580                             id(Psi1))),
581                     union(alpha_star_inverse(Phi1, A, B), id(Psi1))),
582             H1) ).
583   rmcomp(H2, H).

```

```

584   Interpr(Z, H) :-
585     proved(Z = [h1a(Z1, Z2, A -> B), G => T]),
586     proved(Z1 = [_, Gamma => [A]]),
587     proved(Z2 = [_, G1 => T]),
588     append(Gamma, [A -> B], M),
589     append(Delta, [B | Lambda], G1),
590     append3(Delta, M, Lambda, G),
591   |
592     Interpr(Z1, H1),
593     Interpr(Z2, H2),
594     pi(Gamma, Gamma1),
595     pi(Delta, Delta1),
596     pi(Lambda, Lambda1),
597     ((Delta = [],
598      Lambda = []),
599      H3 = comp(comp(H2, epsilon(A, B)),
600                product(H1, id(A -> B))))
601   ; (Delta = []).
602   H3 = comp(comp(H2, product(epsilon(A, B), id(Lambda1))),
603             product(product(H1, id(A -> B)), id(Lambda1)))
604   ; (Lambda = []).
605   H3 = comp(comp(comp(H2, product(id(Delta1), epsilon(A, B))),
606                 product(id(Delta1), product(H1, id(A -> B)))),
607             alpha_inverse(Delta1, Gamma1, A -> B))
608   ; H3 = comp(comp(comp(H2, product(product(id(Delta1), epsilon(A, B)),
609                                     id(Lambda1))),
610                 product(product(product(id(Delta1), H1), id(A -> B)),
611                             id(Lambda1))),
612                 product(alpha_inverse(Delta1, Gamma1, A -> B), id(Lambda1))).
613   rmcomp(H3, H).

```

```

614   Interpr(Z, H) :-
615     proved(Z = [his(Z1, A -> B), G => [A -> B]]),
616     proved(Z1 = [_, G1 => [B]]),
617     append(Gamma, [A | Delta], G1),
618     append(Gamma, Delta, G),
619   |
620     Interpr(Z1, H1),
621     pi(Gamma, Gamma1),
622     pi(Delta, Delta1),
623     ((Gamma = [],
624      H2 = comp(hook(id(A), H1), ita(A, Delta1)))
625   ; (Delta = []).
626   H2 = comp(comp(hook(id(A), H1), hook(id(A), sigma(A, Gamma1))),

```

```

627         Ita(A,Gamma1)))
628     ; H2 = comp(comp(comp(hook(id(A),H1),
629         hook(id(A),product(sigma(A,Gamma1),
630             id(Delta1))))),
631         hook(id(A),alpha(A,Gamma1,Delta1))),
632         Ita(A,Gamma1 & Delta1)),
633     rmcomp(H2,H).

```

```

634 /* pi(Z,H) means translating the list of elements Z into just
635 one element using &, for example [a,b,c,d,e] translates
636 to a&(b&(c&(d&e))) */

```

```

637 pi([A],A).

```

```

638 pi(Z,L) :-
639     append(Z1,[A],Z),
640     Z1 \== [],
641     pi(Z1,L1),
642     ((L1 == [-],
643     L = L1 & A)
644     ;(L1 == [-|_],
645     L = (L1) & A)).

```

```

646 pi([],[]).

```

```

647 /* un the same like pi but using # instead of & */

```

```

648 un([A],A).

```

```

649 un(Z,L) :-
650     append(Z1,[A],Z),
651     Z1 \== [],
652     un(Z1,L1),
653     ((L1 == [-],
654     L = L1 # A)
655     ;(L1 == [-|_],
656     L = (L1) # A)).

```

```

657 un([],[]).

```

```

658 /* rmcomp(H1,H) means that if H1 = comp(id(a),f), we put H = f,
659 or, if H1 = comp(g,id(a)), we put H = g, otherwise H = H1 */

```

```

660 rmcomp(H1,H) :-
661     ((H1 = comp(id(A),F), H = F)
662     ;(H1 = comp(G,id(B)), H = G)
663     ;H = H1).

```

```

664 /* This command to test the equality of arrows */

```

```

665 commutative :-
666     write('Insert the first arrow : '),
667     read(X),nl,nl,nl,
668     norm_rep(X),
669     main(Z1),
670     tree(Z1,L1),
671     clean,
672     write('Insert the second arrow : '),
673     read(Y),nl,nl,nl,
674     norm_rep(Y),
675     main(Z2),
676     tree(Z2,L2),
677     write('The answer is :'),nl,
678     write('===== '),nl,nl,
679     ((L1 == L2,
680     write(X), write(' = '), nl,
681     write(Y), nl, nl,
682     write('because they have the same normal representation.'),
683     nl,nl)
684     ;(write(X) , nl,
685     write('
686     write(Y),nl,nl)).

```

```

687 tree(X,L) :-
688     tree(X, [], L).

```

```

689 tree(X, Sofar, K) :-
690     proved(X = [Y,Z]).
691     (((Y =.. [_] ; Y =.. [id,_]).
692     append(Sofar, [Y:Z], K))
693     ; (Y =.. [th,L1,A],
694     tree(L1, Sofar, K1),
695     append(K1, [th(A) : Z], K))
696     ; (Y =.. [con,L1,A],
697     tree(L1, Sofar, K1),
698     append(K1, [con(A) : Z], K))
699     ; (Y =.. [aia,L1,A],
700     tree(L1, Sofar, K1),
701     append(K1, [aia(A) : Z], K))
702     ; (Y =.. [ois,L1,A],
703     tree(L1, Sofar, K1),
704     append(K1, [ois(A) : Z], K))
705     ; (Y =.. [his,L1,A],
706     tree(L1, Sofar, K1),
707     append(K1, [his(A) : Z], K))
708     ; (Y =.. [inc,L1,B,A],
709     tree(L1, Sofar, K1),
710     append(K1, [inc(B,A) : Z], K))
711     ; (Y =.. [ais,L1,L2,A],
712     tree(L1, Sofar, K1),
713     tree(L2, K1, K2),
714     append(K2, [ais(A) : Z], K))
715     ; (Y =.. [oia,L1,L2,A],
716     tree(L1, Sofar, K1),
717     tree(L2, K1, K2),
718     append(K2, [oia(A) : Z], K))
719     ; (Y =.. [hia,L1,L2,A],
720     tree(L1, Sofar, K1),
721     tree(L2, K1, K2),
722     append(K2, [hia(A) : Z], K))).

```

```

723 /* This is a program to produce representations of arrows of the
724 free (cartesian) categories by means of Gentezen derivations */

```

```

725 norm_rep(Y) :-
726     write('The arrow is :'), nl,
727     write('====='), nl, nl,
728     assert(inserted(Y)), /* this needed only to tell the
729                             system when the processes of the
730                             normal start that we are working
731                             on category theory and don't satisfy
732                             'cop', since there is no need */
733     write(Y), nl, nl,
734     gensym(f,Z),
735     ((Y =.. [X], Y1 = [ar,X])
736     ; Y =.. Y1),
737     append(Y1, [Z], L1),
738     L =.. L1,
739     L,
740     normal(Z).

```

```

741 new :- clean,
742         abolish(arr,3),
743         abolish(composition,3),
744         abolish(thry,1).

```

```

745 clean :- abolish(proved,1),
746          abolish(main,1),
747          abolish(inserted,1).

```

```

748 rep_of(Y) :-
749     write('The arrow is :'), nl,
750     write('====='), nl, nl,
751     write(Y), nl, nl,
752     gensym(f,Z),
753     Y =.. Y1,
754     append(Y1, [Z], L1),
755     L =.. L1,
756     L,
757     write('Representation :-'), nl,
758     write('====='), nl, nl,
759     printall(Z), nl.

```



```

760 /* arrow(A,F,S) means F : A -----> S */
761 arrow(A,F,B) :-
762     assert(arr(A,F,B)).

763 composite(G,F,H) :-
764     assert(composition(G,F,H)).

765 ar(F,Z) :- arr(A,F,B),
766     assert(proved(Z = [F, [A] => [B]))).

767 /* id(A) : A -----> A */
768 id(K,X) :-
769     ((K =.. [true], assert(proved(X = [true, []] => [tr]))))
770     ;(K =.. [bottom], assert(proved(X = [bottom, [bo]] => [ ])))
771     ;(K =.. [A],
772     assert(proved(X = [id(A), [A] => [A]])))
773     ;(K =.. [&,A,B],
774     gensym(f,Z1),
775     assert(proved(X = [aia(Z1,A & B), [A & B] => [A & B]])),
776     gensym(f,Z2),
777     gensym(f,Z3),
778     assert(proved(Z1 = [ais(Z2,Z3,A & B), [A,B] => [A & B]])),
779     gensym(f,Z4),
780     assert(proved(Z2 = [th(Z4,B), [A,B] => [A]])),
781     gensym(f,Z5),
782     assert(proved(Z3 = [th(Z5,A), [A,B] => [B]])),
783     id(A,Z4),
784     id(B,Z5))
785     ;(K =.. [# ,A,B],
786     gensym(f,Z1),
787     assert(proved(X = [ois(Z1,A # B), [A # B] => [A # B]])),
788     gensym(f,Z2),
789     gensym(f,Z3),
790     assert(proved(Z1 = [oia(Z2,Z3,A # B), [A # B] => [A,B]])),
791     gensym(f,Z4),
792     assert(proved(Z2 = [th(Z4,B), [A] => [A,B]])),
793     gensym(f,Z5),
794     assert(proved(Z3 = [th(Z5,A), [B] => [A,B]])),
795     id(A,Z4),
796     id(B,Z5))
797     ;(K =.. [->,A,B],
798     gensym(f,Z1),
799     assert(proved(X = [his(Z1,A -> B), [A -> B] => [A -> B]])),
800     gensym(f,Z2),
801     gensym(f,Z3),
802     assert(proved(Z1 = [hia(Z2,Z3,A -> B), [A,A -> B] => [B]])),
803     id(A,Z2),
804     id(B,Z3)).

805 /* Canonical Arrows for Fc(X) */
806 /* ===== */

807 /* tau(A) : A -----> T where T is true */
808 tau(A,X) :-
809     gensym(f,F1),
810     assert(proved(F1 = [true, []] => [tr])),
811     assert(proved(X = [th(F1,A), [A] => [tr]))).

812 /* pi_left(A,B) : A & B -----> A */
813 pi_left(A,B,X) :-
814     gensym(f,F1),
815     assert(proved(X = [aia(F1,A & B), [A & B] => [A]])),
816     gensym(f,F2),
817     assert(proved(F1 = [th(F2,B), [A,B] => [A]])),
818     id(A,F2).

819 /* pi_right(A,B) : A & B -----> B */
820 pi_right(A,B,X) :-
821     gensym(f,F1),
822     assert(proved(X = [aia(F1,A & B), [A & B] => [B]])),
823     gensym(f,F2),
824     assert(proved(F1 = [th(F2,A), [A,B] => [B]])),
825     id(B,F2).

```

```

826 /* comp(V,U) : A -----> D where
827 U : A -----> B , V : B -----> D */

828 comp(V,U,Z) :-
829     part(V,Z2),
830     part(U,Z1),
831     proved(Z1 = [-, A => [C]]),
832     proved(Z2 = [-, [C] => S]),
833     assert(proved(Z = [cut(Z1,Z2,C), A => S])).

834 /* angle(U,V) = <U,V> : A -----> B & C where
835 U : A -----> B , V : A -----> C */

836 angle(U,V,Z) :-
837     part(U,Z1),
838     part(V,Z2),
839     proved(Z1 = [-, A => [B]]),
840     proved(Z2 = [-, A => [C]]),
841     assert(proved(Z = [dis(Z1,Z2,B & C), A => [B & C]])).

842 /* delta(A) : A -----> A & A */

843 delta(A,Z) :-
844     gensym(f,F1),
845     gensym(f,F2),
846     assert(proved(Z = [dis(F1,F2,A & A), [A] => [A & A]])),
847     id(A,F1),
848     id(A,F2).

849 /* alpha(A,B,C) : A & (B & C) -----> (A & B) & C */
850 alpha(A,B,C,Z) :-
851     angle(angle(pi_left(A,B&C),comp(pi_left(B,C),pi_right(A,B&C))),
852           comp(pi_right(B,C),pi_right(A,B&C)),Z).

853 /* alpha_inverse(A,B,C) : (A & B) & C -----> A & (B & C) */
854 alpha_inverse(A,B,C,Z) :-
855     angle(comp(pi_left(A,B),pi_left(A&B,C)),
856           angle(comp(pi_right(A,B),pi_left(A&B,C)),pi_right(A&B,C)),
857           Z).

858 /* product(U,V) = U & V : A & B -----> C & D where
859 U : A -----> C , V : B -----> D */

860 product(U,V,Z) :-
861     part(U,Z1),
862     part(V,Z2),
863     proved(Z1 = [-, [A] => [C]]),
864     proved(Z2 = [-, [B] => [D]]),
865     part(pi_left(A,B),Z3),
866     gensym(f,Z4),
867     assert(proved(Z4 = [cut(Z3,Z1,A), [A & B] => [C]])),
868     part(pi_right(A,B),Z5),
869     gensym(f,Z6),
870     assert(proved(Z6 = [cut(Z5,Z2,B), [A & B] => [D]])),
871     assert(proved(Z = [dis(Z4,Z6,C & D), [A & B] => [C & D]])).

872 /* sigma(A,B) : A & B -----> B & A */
873 sigma(A,B,Z) :-
874     angle(pi_right(A,B),pi_left(A,B), Z).

875 /* sigma_inverse(A,B) : B & A -----> A & B */
876 sigma_inverse(A,B,Z) :-
877     angle(pi_right(B,A),pi_left(B,A), Z).

878 /* Canonical Arrows for Fbc(X) */
879 /* ===== */

880 /* tau_star(A) : bottom -----> A */
881 tau_star(A,X) :-
882     gensym(f,F1),
883     assert(proved(F1 = [bottom, [bo] => [ ]])),
884     assert(proved(X = [th(F1,A), [bo] => [A]])).

```

```

885 /* pi_left_star(A,B) : A -----> A # B */
886 pi_left_star(A,B,X) :-
887     gensym(f,F1),
888     assert(proved(X = [ois(F1,A # B), [A] => [A # B]])),
889     gensym(f,F2),
890     assert(proved(F1 = [th(F2,B), [A] => [A,B]])),
891     id(A,F2).

892 /* pi_right_star(A,B) : B -----> A # B */
893 pi_right_star(A,B,X) :-
894     gensym(f,F1),
895     assert(proved(X = [ois(F1,A # B), [B] => [A # B]])),
896     gensym(f,F2),
897     assert(proved(F1 = [th(F2,A), [B] => [A,B]])),
898     id(B,F2).

899 /* delta_star(A) : A # A -----> A */
900 delta_star(A,Z) :-
901     gensym(f,F1),
902     gensym(f,F2),
903     assert(proved(Z = [oia(F1,F2,A # A), [A # A] => [A]])),
904     id(A,F1),
905     id(A,F2).

906 /* square(U,V) = [U,V] : A # B -----> C where
907     U : A -----> C , V : B -----> C */
908 square(U,V,Z) :-
909     part(U,Z1),
910     part(V,Z2),
911     proved(Z1 = [-, [A] => S]),
912     proved(Z2 = [-, [B] => S]),
913     assert(proved(Z = [oia(Z1,Z2,A # B), [A # B] => S])).

914 /* union(U,V) = U # V : A # C -----> B # D where
915     U : A -----> B , V : C -----> D */
916 union(U,V,Z) :-
917     part(U,Z1),
918     part(V,Z2),
919     proved(Z1 = [-, [A] => [B]]),
920     proved(Z2 = [-, [C] => [D]]),
921     part(pi_left_star(B,D),Z3),
922     gensym(f,Z4),
923     assert(proved(Z4 = [cut(Z1,Z3,B), [A] => [B # D]])),
924     part(pi_right_star(B,D),Z5),
925     gensym(f,Z6),
926     assert(proved(Z6 = [cut(Z2,Z5,D), [C] => [B # D]])),
927     assert(proved(Z = [oia(Z4,Z6,A # C), [A # C] => [B # D]])).

928 /* alpha_star(A,B,C) : A # (B # C) -----> (A # B) # C */
929 alpha_star(A,B,C,Z) :-
930     square(comp(pi_left_star(A#B,C),pi_left_star(A,B)),
931     square(comp(pi_left_star(A#B,C),pi_right_star(A,B)),
932     pi_right_star(A#B,C)), Z).

933 /* alpha_star_inverse(A,B,C) : (A # B) # C -> A # (B # C) */
934 alpha_star_inverse(A,B,C,Z) :-
935     square(square(pi_left_star(A,B#C),comp(pi_right_star(A,B#C),
936     pi_left_star(B,C))),
937     comp(pi_right_star(A,B#C),pi_right_star(B,C)), Z).

938 /* sigma_star(A,B) : A # B -----> B # A */
939 sigma_star(A,B,Z) :-
940     square(pi_right_star(B,A),pi_left_star(B,A), Z).

941 /* sigma_star_inverse(A,B) : B # A -----> A # B */
942 sigma_star_inverse(A,B,Z) :-
943     square(pi_right_star(A,B),pi_left_star(A,B), Z).

```

```

944 /* Canonical Arrows for Fdbc(X) */
945 /* ===== */

946 /* round(U,V) = (U,V) : A & (B # C) -----> D      where
947    U : A & B -----> D , V : A & C -----> D      */
948 round(U,V,Z) :-
949     part(U,Z1),
950     part(V,Z2),
951     proved(Z1 = [_, [A & B] => S]),
952     proved(Z2 = [_, [A & C] => S]),
953     gensym(f,F1),
954     assert(proved(Z = [dia(F1,A & (B # C)), [A & (B # C)] => 'S'))),
955     gensym(f,F2),
956     gensym(f,F3),
957     assert(proved(F1 = [oia(F2,F3,B # C), [A,B # C] => S))),
958     gensym(f,F4),
959     assert(proved(F2 = [cut(F4,Z1,A & B), [A,B] => S))),
960     gensym(f,F5),
961     assert(proved(F3 = [cut(F5,Z2,A & C), [A,C] => S))),
962     gensym(f,F6),
963     gensym(f,F7),
964     assert(proved(F4 = [ais(F6,F7,A & B), [A,B] => [A & B]])),
965     gensym(f,F8),
966     gensym(f,F9),
967     assert(proved(F5 = [ais(F8,F9,A & C), [A,C] => [A & C]])),
968     gensym(f,F10),
969     assert(proved(F6 = [th(F10,B), [A,B] => [A]])),
970     gensym(f,F11),
971     assert(proved(F7 = [th(F11,A), [A,B] => [B]])),
972     gensym(f,F12),
973     assert(proved(F8 = [th(F12,C), [A,C] => [A]])),
974     gensym(f,F13),
975     assert(proved(F9 = [th(F13,A), [A,C] => [C]])),
976     id(A,F10),
977     id(B,F11),
978     id(A,F12),
979     id(C,F13).

980 /* delta_left(A,B,C) : A & (B # C) -> (A & B) # (A & C) */
981 delta_left(A,B,C,Z) :-
982     round(pi_left_star(A&B,A&C),pi_right_star(A&B,A&C),Z).

983 /* delta_right(A,B,C) : (A # B) & C -> (A & C) # (B & C) */
984 delta_right(A,B,C,Z) :-
985     comp(union(sigma(C,A),sigma(C,B)),comp(delta_left(C,A,B),
986     sigma(A#B)),Z).

987 /* delta_left_star(A,B,C) : (A # B) & (A # C) -> A # (B & C) */
988 delta_left_star(A,B,C,Z) :-
989     comp(comp(comp(comp(comp(delta_star(A#(B&C))),
990     union(id(A#(B&C)),union(pi_right(B,A),
991     id(B&C)))),
992     union(id(A#(B&C)),delta_left(B,A,C))),
993     union(pi_left_star(A,B&C),id(B&(A#C))),
994     union(pi_left(A,A#C),id(B&(A#C))),
995     delta_right(A,B,A#C),
996     Z).

997 /* delta_right_star(A,B,C) : (A # C) & (B # C) -> (A & B) # C */
998 delta_right_star(A,B,C,Z) :-
999     comp(comp(sigma_star(C,A & B), delta_left_star(C,A,B)),
1000     product(sigma_star(A,C),sigma_star(B,C)),
1001     Z).

1002 /* kappa(A,B,C,D,E) : (A&((B # C) # D))&E -> (B#((A&C)&E))#D */
1003 kappa(A,B,C,D,E,Z) :-
1004     comp(comp(comp(comp(union(union(pi_right(A,B),id((A&C)&E)),
1005     pi_right(A,D)),
1006     union(union(pi_left(A&B,E),id((A&C)&E)),
1007     pi_left(A&D,E))),
1008     union(delta_right(A&B,A&C,E),id((A&D)&E))),
1009     delta_right((A&B) # (A&C),A&D,E)),
1010     product(union(delta_left(A,B,C),id(A & D)),id(E)),
1011     product(delta_left(A,B # C,D),id(E)),
1012     Z).

```

```

1013 /* Canonical arrows in Fccl(X) :- */
1014 /* ===== */

1015 /* epsilon_left(U,V) : A & (B -> C) -----> D   where
1016    U : A -----> B ,   V : C -----> D   */
1017 epsilon_left(U,V,Z) :-
1018     part(U,Z1),
1019     part(V,Z2),
1020     proved(Z1 = [-, [A] => [B]]),
1021     proved(Z2 = [-, [C] => S]),
1022     gensym(f,F1),
1023     assert(proved(Z = [aia(F1,A&(B -> C)), [A&(B -> C)] => S))),
1024     assert(proved(F1 = [hia(Z1,Z2,B -> C), [A,B -> C] => S))).

1025 /* alpha_right(U) : B -----> (A -> C)   where
1026    U : A & B -----> C   */
1027 alpha_right(U,Z) :-
1028     part(U,Z1),
1029     proved(Z1 = [-, [A & B] => [C]]),
1030     gensym(f,F1),
1031     assert(proved(Z = [hia(F1,A -> C), [B] => [A -> C]])),
1032     gensym(f,F2),
1033     assert(proved(F1 = [cut(F2,Z1,A & B), [A,B] => [C]])),
1034     gensym(f,F3),
1035     gensym(f,F4),
1036     assert(proved(F2 = [ais(F3,F4,A & B), [A,B] => [A & B]])),
1037     gensym(f,F5),
1038     assert(proved(F3 = [th(F5,B), [A,B] => [A]])),
1039     gensym(f,F6),
1040     assert(proved(F4 = [th(F6,A), [A,B] => [B]])),
1041     id(A,F5),
1042     id(B,F6).

1043 /* epsilon(A,B) : A & (A => B) -----> B */
1044 epsilon(A,B,Z) :-
1045     gensym(f,F1),
1046     assert(proved(Z = [aia(F1,A&(A -> B)), [A&(A -> B)] => [B]])),
1047     gensym(f,F2),
1048     gensym(f,F3),
1049     assert(proved(F1 = [hia(F2,F3,A -> B), [A,A -> B] => [B]])),
1050     id(A,F2),
1051     id(B,F3).

1052 /* ita(A,B) : B -----> A => (A & B)   */
1053 ita(A,B,Z) :-
1054     gensym(f,F1),
1055     assert(proved(Z = [his(F1,A -> (A&B)), [B] => [A -> (A&B)])),
1056     gensym(f,F2),
1057     gensym(f,F3),
1058     assert(proved(F1 = [ais(F2,F3,A & B), [A,B] => [A & B]])),
1059     gensym(f,F4),
1060     assert(proved(F2 = [th(F4,B), [A,B] => [A]])),
1061     gensym(f,F5),
1062     assert(proved(F3 = [th(F5,A), [A,B] => [B]])),
1063     id(A,F4),
1064     id(B,F5).

1065 /* hook(U,V) = (U => V) : (B => C) -----> (A => D)   where
1066    U : A -----> B ,   V : C -----> D   */
1067 hook(U,V,Z) :-
1068     alpha_right(epsilon_left(U,V), Z).

1069 part(X,Z) :-
1070     gensym(f,Z),
1071     ((X =.. [X1], ar(X,Z))
1072     ; (X =.. X1,
1073       append(X1,[Z],L1),
1074       L =.. L1,
1075       L)).

```

```

1077 normal(Z) :-
1078     cut_free(Z),
1079     (((thry(cartcl) ; thry(bicartcl)),
1080      hook_terminai
1081      ; true),
1082      term_intl,
1083      expth,
1084      put_in_llst(Z,L),
1085      abolish(proved,1),
1086      ass_llst(L),
1087      normal1,
1088      normal2,
1089      (((thry(cartcl) ; thry(bicartcl)),
1090       assert(second),
1091       normal1,
1092       normal2,
1093       retract(second) )
1094       ; true ).

1095 cut_free(Z) :-
1096     ( hook_cut
1097     ; ((inserted(X) ; cop),
1098       write('Proof is as follows :'), nl,
1099       write('===== '), nl, printall(Z),
1100       nl, nl, nl, assert(main(Z))),
1101       repeat,
1102         one_step,
1103         not(proved(_ = [cut(_,_), _ => _])),
1104     |,
1105     hook_cut
1106     ; (write('Cut_free proof is :'), nl,
1107       write('===== '), nl, nl,
1108       printall(Z), nl, nl)).

1109 one_step :-
1110     proved(Z = [cut(Z1,Z2,C), G => T]),
1111     proved(Z1 = [K1,_]), K1 =.. [H1,_], H1 \== cut,
1112     proved(Z2 = [K2,_]), K2 =.. [H2,_], H2 \== cut,
1113     cut_elm(proved(Z = [cut(Z1,Z2,C), G => T])),
1114     expcon,
1115     expinc,
1116     expth,
1117     purge,
1118     |.
1119 one_step. /* must succeed if no cuts in proof */

1120 purge :- /* purges unwanted proof steps */
1121     main(Y),
1122     assertz(proved(mark)),
1123     reassert_all(Y),
1124     repeat,
1125         retract(proved(Z)),
1126         Z = mark, /* retracts all upto mark */
1127     |.

1128 reassert_all(X) :-
1129     proved(X = Y),
1130     ( nullary(X)
1131     ; unary(X,L), reassert_all(L)
1132     ; binary(X,L,M), reassert_all(L), reassert_all(M)),
1133     retract(proved(X = Y)),
1134     assertz(proved(X = Y)).

1135 /* (A1,A1) - (C.1) */
1136 cut_elm(proved(Z = [cut(Z1,Z2,B), G => T])) :-
1137     proved(Z1 = [L, G => [B]]),
1138     proved(Z2 = [M, [B] => T]),
1139     (L =.. [X] ; L =.. [id,B]),
1140     (M =.. [Y] ; M =.. [id,B]),
1141     |,
1142     ((L =.. [id,B], N = M)
1143     ; (M =.. [id,B], N = L)
1144     ; composition(M,L,N)),
1145     assert(proved(Z = [N, G => T])),
1146     retract(proved(Z = [cut(Z1,Z2,B), G => T])).

```

```

1147 /* (A1,R2) - (C.2.1) */
1148 cut_elm(proved(Z = [cut(Z1,Z2,B), G => T])) :-
1149     proved(Z1 = [K1,[A]=>[B]]),
1150     (K1 =.. [X] ; K1 =.. [id,A]),
1151     proved(Z2 = [th(Z3,B),G1=>T]),
1152     proved(Z3 = [K,G3=>T]),
1153     append(Gamma, [A|Delta], G),
1154     append(Gamma, [B|Delta], G1),
1155     append(Gamma, Delta, G3),
1156     !,
1157     gensym(f,F1),
1158     assert(proved(F1 = [K,G3 => T])),
1159     assert(proved(Z = [th(F1,A),G => T])),
1160     retract(proved(Z = [cut(Z1,Z2,B), G => T])).

1161 /* (A3,R2) - (C.2.3) */
1162 cut_elm(proved(Z = [cut(Z1,Z2,tr), G => T])) :-
1163     proved(Z1 = [true,[]=>[tr]]),
1164     proved(Z2 = [th(Z3,tr),G1=>T]),
1165     proved(Z3 = [K,G=>T]),
1166     !,
1167     assert(proved(Z = [K,G=>T])),
1168     retract(proved(Z = [cut(Z1,Z2,tr),G=>T])).

1169 /* (R10,R2) - (C.3) */
1170 cut_elm(proved(Z = [cut(Z1,Z2,A & B), G => T])) :-
1171     proved(Z1 = [ois(Z3,Z4,A & B), G1 => T1]),
1172     proved(Z2 = [th(Z5,A & B), G2 => T2]),
1173     proved(Z3 = [K1,G1=>T3]),
1174     proved(Z4 = [K2,G1=>T4]),
1175     proved(Z5 = [K3,G3=>T2]),
1176     append(Phi, [A & B|Psi], T1),
1177     append(Phi, [A|Psi], T3),
1178     append(Phi, [B|Psi], T4),
1179     append3(Phi, T2, Psi, T),
1180     append(Delta, [A & B|Lambda], G2),
1181     append(Delta, Lambda, G3),
1182     append3(Delta, G1, Lambda, G),
1183     !,
1184     gensym(f,F1),
1185     assert(proved(F1 = [K3,G3=>T2])),
1186     append(Delta, [B|Lambda], G4),
1187     gensym(f,F2),
1188     assert(proved(F2 = [th(F1,B),G4 => T2])),
1189     append(Delta, [A|[B|Lambda]], G5),
1190     gensym(f,F3),
1191     assert(proved(F3 = [th(F2,A),G5 => T2])),
1192     gensym(f,F4),
1193     assert(proved(F4 = [K2,G1=>T4])),
1194     append3(Delta, [A|G1], Lambda, G6),
1195     gensym(f,F5),
1196     assert(proved(F5 = [cut(F4,F3,B),G6 => T])),
1197     gensym(f,F6),
1198     assert(proved(F6 = [K1,G1=>T3])),
1199     append4(Delta, G1, G1, Lambda, G7),
1200     append(Phi, T, T6),
1201     append(T6, Psi, T5),
1202     gensym(f,F7),
1203     assert(proved(F7 = [cut(F6,F5,A),G7 => T5])),
1204     gensym(f,F8),
1205     assert(proved(F8 = [con(F7,Psi),G7 => T6])),
1206     gensym(f,F9),
1207     assert(proved(F9 = [con(F8,Phi),G7 => T])),
1208     assert(proved(Z = [con(F9,G1),G => T])),
1209     retract(proved(Z = [cut(Z1,Z2,A & B),G => T])).

1210 /* (R13,R2) - (C.4) */
1211 cut_elm(proved(Z = [cut(Z1,Z2,A # B), G => T])) :-
1212     proved(Z1 = [ois(Z3,A # B),G1=>T1]),
1213     proved(Z2 = [th(Z4,A # B),G2=>T2]),
1214     proved(Z3 = [K1,G1=>T3]),
1215     proved(Z4 = [K2,G3=>T2]),
1216     append(Phi, [A # B|Psi], T1),
1217     append(Phi, [A|[B|Psi]], T3),
1218     append3(Phi, T2, Psi, T),
1219     append(Delta, [A # B|Lambda], G2),
1220     append(Delta, Lambda, G3),
1221     append3(Delta, G1, Lambda, G),
1222     !,

```

```

1223      gensym(f,F1),
1224      assert(proved(F1 = [K2,G3=>T2])),
1225      append(Delta, [B|Lambda], G4),
1226      gensym(f,F2),
1227      assert(proved( F2 = [th(F1,B),G4 => T2])),
1228      gensym(f,F3),
1229      assert(proved(F3 = [K1,G1=>T3])),
1230      append3(Phi, [A|T2], Psi, T4),
1231      gensym(f,F4),
1232      assert(proved( F4 = [cut(F3,F2,B),G => T4])),
1233      gensym(f,F5),
1234      assert(proved(F5 = [K2, G3 => T2])),
1235      append(Delta, [A|Lambda], G5),
1236      gensym(f,F6),
1237      assert(proved( F6 = [th(F5,A),G5 => T2])),
1238      append(Delta, G, G7),
1239      append(G7, Lambda, G6),
1240      append4(Phi, T2, T2, Psi, T5),
1241      gensym(f,F7),
1242      assert(proved( F7 = [cut(F4,F6,A),G6 => T5])),
1243      gensym(f,F8),
1244      assert(proved( F8 = [con(F7,T2),G6 => T])),
1245      gensym(f,F9),
1246      assert(proved( F9 = [con(F8,Lambda),G7 => T])),
1247      assert(proved( Z = [con(F9,Delta),G => T])),
1248      retract(proved(Z = [cut(Z1,Z2,A # B),G => T])).

1249 /* (R1b,R2) - (C.5) */
1250 cut_eim(proved(Z = [cut(Z1,Z2,A -> B), G => T])) :-
1251   proved(Z1 = [his(Z3,A -> B),G1=>[A -> B]]),
1252   proved(Z2 = [th(Z4,A -> B),G2=>T]),
1253   proved(Z3 = [K1,G3=>[B]]),
1254   proved(Z4 = [K2,G4=>T]),
1255   append(Gamma, [A|Delta], G3),
1256   append(Gamma, Delta, G1),
1257   append(Lambda, [A -> B|Zeta], G2),
1258   append(Lambda, Zeta, G4),
1259   append3(Lambda, G1, Zeta, G),
1260   !,
1261   gensym(f,F1),
1262   assert(proved(F1 = [K2,G4 => T])),
1263   append3(Lambda, Gamma, Zeta, G5),
1264   gensym(f,F2),
1265   assert(proved( F2 = [th(F1,Gamma),G5 => T])),
1266   assert(proved( Z = [th(F2,Delta),G => T])),
1267   retract(proved(Z = [cut(Z1,Z2,A -> B),G => T])).

1268 /* (R5,A1) - (C.7.1) */
1269 cut_eim(proved(Z = [cut(Z1,Z2,A),G => T])) :-
1270   proved(Z1 = [th(Z3,A),G=>T1]),
1271   proved(Z2 = [K,[A]==>[B]]),
1272   (K =.. [X] ; K =.. [id,A]),
1273   proved(Z3 = [K1,G=>T2]),
1274   append(Phi, [B|Psi], T),
1275   append(Phi, [A|Psi], T1),
1276   append(Phi, Psi, T2),
1277   !,
1278   gensym(f,F1),
1279   assert(proved(F1 = [K1,G => T2])),
1280   assert(proved(Z=[th(F1,B),G=>T])),
1281   retract(proved(Z=[cut(Z1,Z2,A),G=>T])).

1282 /* (R5,A4) - (C.7.2) */
1283 cut_eim(proved(Z = [cut(Z1,Z2,bo), G => T])) :-
1284   proved(Z1 = [th(Z3,bo),G=>T1]),
1285   proved(Z2 = [bottom,[bo]==>[]]),
1286   proved(Z3 = [K1,G=>T]),
1287   !,
1288   assert(proved(Z = [K1, G => T])),
1289   retract(proved(Z = [cut(Z1,Z2,bo),G=>T])).

1290 /* (R5,R11) - (C.8) */
1291 cut_eim(proved(Z = [cut(Z1,Z2,A & B), G => T])) :-
1292   proved(Z1 = [th(Z3,A & B),G1 => T1]),
1293   proved(Z2 = [ola(Z4,A & B),G2 => T2]),
1294   proved(Z3 = [K1,G1=>T3]),
1295   proved(Z4 = [K2,G3=>T2]),
1296   append(Phi, [A & B|Psi], T1),
1297   append(Phi, Psi, T3),

```



```

1298     append3(Phi, T2, Psi, T),
1299     append(Delta, [A & B|Lambda], G2),
1300     append(Delta, [A|[B|Lambda]], G3),
1301     append3(Delta, G1, Lambda, G),
1302
1303     gensym(f, F1),
1304     assert(proved(F1 = [K1, G1 => T3])),
1305     append(Phi, [A|Psi], T4),
1306     gensym(f, F2),
1307     assert(proved( F2 = [th(F1,A),G1 => T4])),
1308     gensym(f, F3),
1309     assert(proved(F3 = [K1, G1 => T3])),
1310     append(Phi, [B|Psi], T5),
1311     gensym(f, F4),
1312     assert(proved( F4 = [th(F3,B),G1 => T5])),
1313     gensym(f, F5),
1314     assert(proved(F5 = [K2, G3 => T2])),
1315     append3(Delta, [A|G1], Lambda, G4),
1316     gensym(f, F6),
1317     assert(proved( F6 = [cut(F4,F5,B),G4 => T])),
1318     append4(Delta, G1, G1, Lambda, G5),
1319     append(Phi, T, T7),
1320     append(T7, Psi, T6),
1321     gensym(f, F7),
1322     assert(proved( F7 = [cut(F2,F6,A),G5 => T6])),
1323     gensym(f, F8),
1324     assert(proved( F8 = [con(F7,Psi),G5 => T7])),
1325     gensym(f, F9),
1326     assert(proved( F9 = [con(F8,Phi),G5 => T])),
1327     assert(proved( Z = [con(F9,G1),G => T])),
1328     retract(proved(Z = [cut(Z1,Z2,A & B),G => T])).

```

```

1329 /* (R5,R12) - (C.9) */
1330 cut_alm(proved(Z = [cut(Z1,Z2,A # B), G => T])) :-
1331     proved(Z1 = [th(Z3,A # B),G1=>T1]),
1332     proved(Z2 = [ola(Z4,Z5,A # B),G2=>T2]),
1333     proved(Z3 = [K1,G1=>T3]),
1334     proved(Z4 = [K2,G3=>T2]),
1335     proved(Z5 = [K3,G4=>T2]),
1336     append(Phi, [A # B|Psi], T1),
1337     append(Phi, Psi, T3),
1338     append3(Phi, T2, Psi, T),
1339     append(Delta, [A # B|Lambda], G2),
1340     append(Delta, [A|Lambda], G3),
1341     append(Delta, [B|Lambda], G4),
1342     append3(Delta, G1, Lambda, G),
1343
1344     gensym(f, F1),
1345     assert(proved(F1 = [K1,G1 => T3])),
1346     append(Phi, [B|Psi], T4),
1347     gensym(f, F2),
1348     assert(proved( F2 = [th(F1,B),G1 => T4])),
1349     append(Phi, [A|[B|Psi]], T5),
1350     gensym(f, F3),
1351     assert(proved( F3 = [th(F2,A),G1 => T5])),
1352     gensym(f, F4),
1353     assert(proved(F4 = [K2, G3 => T2])),
1354     append3(Phi, T2, [B|Psi], T6),
1355     gensym(f, F5),
1356     assert(proved( F5 = [cut(F3,F4,A),G => T6])),
1357     gensym(f, F6),
1358     assert(proved(F6 = [K3, G4 => T2])),
1359     append(Delta, G, G6),
1360     append(G6, Lambda, G5),
1361     append4(Phi, T2, T2, Psi, T7),
1362     gensym(f, F7),
1363     assert(proved( F7 = [cut(F5,F6,B),G5 => T7])),
1364     gensym(f, F8),
1365     assert(proved( F8 = [con(F7,T2),G5 => T])),
1366     gensym(f, F9),
1367     assert(proved( F9 = [con(F8,Lambda),G6 => T])),
1368     assert(proved( Z = [con(F9,Delta),G => T])),
1369     retract(proved(Z = [cut(Z1,Z2,A # B),G => T])).

```

```

1370 /* (R5,R14) - (C.10) */
1371 cut_alm(proved(Z = [cut(Z1,Z2,A -> B), G => T])) :-
1372     proved(Z1 = [th(Z3,A -> B),G1=>T1]),
1373     proved(Z2 = [hia(Z4,Z5,A -> B),G2=>T2]),
1374     proved(Z3 = [K1,G1=>T3]),
1375     proved(Z4 = [K2,G3=>[A]]),
1376     proved(Z5 = [K3,G4=>T2]),
1377     append(Phi, [A -> B|Psi], T1),

```

```

1378     append(Phi, Psi, T3),
1379     append3(Phi, T2, Psi, T),
1380     append(S2, [A -> B|Zeta], G2),
1381     append(Lambda, [B|Zeta], G4),
1382     append(Lambda, G3, S2),
1383     append3(S2, G1, Zeta, G),
1384
1385     I,
1386     gensym(f, F1),
1387     assert(proved(F1 = [K1, G1 => T3])),
1388     gensym(f, F2),
1389     assert(proved(F2 = [th(F1, A), G5 => T3])),
1390     append(G3, G1, G6),
1391     gensym(f, F3),
1392     assert(proved(F3 = [K2, G3 => [A]])),
1393     gensym(f, F4),
1394     assert(proved(F4 = [cut(F3, F2, A), G6 => T3])),
1395     append(Phi, [B|Psi], T4),
1396     gensym(f, F5),
1397     assert(proved(F5 = [th(F4, B), G6 => T4])),
1398     gensym(f, F6),
1399     assert(proved(F6 = [K3, G4 => T2])),
1400     assert(proved(Z = [cut(F5, F6, B), G => T])),
1401     retract(proved(Z = [cut(Z1, Z2, A -> B), G => T])).

1402 /* (R5,R2) - (C.12) */
1403 cut_elm(proved(Z = [cut(Z1, Z2, A), G => T])) :-
1404     proved(Z1 = [th(Z3, A), G1 => T1]),
1405     proved(Z2 = [th(Z4, A), G2 => T2]),
1406     proved(Z3 = [K1, G1 => T3]),
1407     proved(Z4 = [G3 => T2]),
1408     append(Phi, [A|Psi], T1),
1409     append(Phi, Psi, T3),
1410     append3(Phi, T2, Psi, T),
1411     append(Delta, [A|Lambda], G2),
1412     append(Delta, Lambda, G3),
1413     append(G1, Lambda, G4),
1414     append(Delta, G4, G),
1415
1416     I,
1417     gensym(f, F1),
1418     assert(proved(F1 = [K1, G1 => T3])),
1419     gensym(f, F2),
1420     assert(proved(F2 = [th(F1, T2), G1 => T])),
1421     gensym(f, F3),
1422     assert(proved(F3 = [th(F2, Lambda), G4 => T])),
1423     assert(proved(Z = [th(F3, Delta), G => T])),
1424     retract(proved(Z = [cut(Z1, Z2, A), G => T])).

1424 /* (R10,R11) - (C.14) */
1425 cut_elm(proved(Z = [cut(Z1, Z2, A & B), G => T])) :-
1426     proved(Z1 = [ai3(Z3, Z4, A & B), G1 => T1]),
1427     proved(Z2 = [ai4(Z5, A & B), G2 => T2]),
1428     proved(Z3 = [K1, G1 => T3]),
1429     proved(Z4 = [K2, G1 => T4]),
1430     proved(Z5 = [K3, G3 => T2]),
1431     append(Phi, [A & B|Psi], T1),
1432     append(Phi, [A|Psi], T3),
1433     append(Phi, [B|Psi], T4),
1434     append3(Phi, T2, Psi, T),
1435     append(Delta, [A & B|Lambda], G2),
1436     append(Delta, [A|[B|Lambda]], G3),
1437     append3(Delta, G1, Lambda, G),
1438
1439     I,
1440     gensym(f, F1),
1441     assert(proved(F1 = [K2, G1 => T4])),
1442     gensym(f, F2),
1443     assert(proved(F2 = [K3, G3 => T2])),
1444     append3(Delta, [A|G1], Lambda, G4),
1445     gensym(f, F3),
1446     assert(proved(F3 = [cut(F1, F2, B), G4 => T])),
1447     gensym(f, F4),
1448     assert(proved(F4 = [K1, G1 => T3])),
1449     append4(Delta, G1, G1, Lambda, G5),
1450     append(Phi, T, T6),
1451     append(T6, Psi, T5),
1452     gensym(f, F5),
1453     assert(proved(F5 = [cut(F4, F3, A), G5 => T5])),
1454     gensym(f, F6),
1455     assert(proved(F6 = [con(F5, Psi), G5 => T6])),
1456     gensym(f, F7),
1457     assert(proved(F7 = [con(F6, Phi), G5 => T])),
1458     assert(proved(Z = [con(F7, G1), G => T])),
1459     retract(proved(Z = [cut(Z1, Z2, A & B), G => T])).

```

```

1459 /* (R13,R12) - (C.15) */
1460 cut_elm(proved(Z = [cut(Z1,Z2,A # B), G => T])) :-
1461     proved(Z1 = [oia(Z3,A # B), G1=>T1]),
1462     proved(Z2 = [oia(Z4,Z5,A # B), G2=>T2]),
1463     proved(Z3 = [K1, G1=>T3]),
1464     proved(Z4 = [K2, G3=>T2]),
1465     proved(Z5 = [K3, G4=>T2]),
1466     append(Phi, [A # B|Psi], T1),
1467     append(Phi, [A|B|Psi], T3),
1468     append3(Phi, T2, Psi, T),
1469     append(Delta, [A # B|Lambda], G2),
1470     append(Delta, [A|Lambda], G3),
1471     append(Delta, [B|Lambda], G4),
1472     append3(Delta, G1, Lambda, G),
1473     !,
1474     gensym(f, F1),
1475     assert(proved(F1 = [K1, G1 => T3])),
1476     gensym(f, F2),
1477     assert(proved(F2 = [K2, G3 => T2])),
1478     append3(Phi, T2, [B|Psi], T4),
1479     gensym(f, F3),
1480     assert(proved(F3 = [cut(F1,F2,A), G => T4])),
1481     gensym(f, F4),
1482     assert(proved(F4 = [K3, G4 => T2])),
1483     append(Delta, G, G6),
1484     append(G6, Lambda, G5),
1485     append4(Phi, T2, T2, Psi, T5),
1486     gensym(f, F5),
1487     assert(proved(F5 = [cut(F3,F4,B), G5 => T5])),
1488     gensym(f, F6),
1489     assert(proved(F6 = [con(F5,T2), G5 => T])),
1490     gensym(f, F7),
1491     assert(proved(F7 = [con(F6,Lambda), G6 => T])),
1492     assert(proved(Z = [con(F7,Delta), G => T])),
1493     retract(proved(Z = [cut(Z1,Z2,A # B), G => T])).

```

```

1494 /* (R15,R14) - (C.16) */
1495 cut_elm(proved(Z = [cut(Z1,Z2,A -> B), G => T])) :-
1496     proved(Z1 = [hia(Z3,A -> B), G1=>[A->B]]),
1497     proved(Z2 = [hia(Z4,Z5,A -> B), G2=>T]),
1498     proved(Z3 = [K1, G3=>[B]]),
1499     proved(Z4 = [K2, G4=>[A]]),
1500     proved(Z5 = [K3, G5=>T]),
1501     append(Gamma, [A|Delta], G3),
1502     append(Gamma, Delta, G1),
1503     append(S1, [A -> B|Sigma], G2),
1504     append(Zeta, [B|Sigma], G5),
1505     append(Zeta, G4, S1),
1506     append3(S1, G1, Sigma, G),
1507     !,
1508     gensym(f, F1),
1509     assert(proved(F1 = [K2, G4 => [A]])),
1510     gensym(f, F2),
1511     assert(proved(F2 = [K1, G3 => [B]])),
1512     append3(Gamma, G4, Delta, G6),
1513     gensym(f, F3),
1514     assert(proved(F3 = [cut(F1,F2,A), G6 => [B]])),
1515     gensym(f, F4),
1516     assert(proved(F4 = [K3, G5 => T])),
1517     append3(Zeta, G6, Sigma, G7),
1518     gensym(f, F5),
1519     assert(proved(F5 = [cut(F3,F4,B), G7 => T])),
1520     assert(proved(Z = [inc(F5,G4,Gamma), G => T])),
1521     retract(proved(Z = [cut(Z1,Z2,A -> B), G => T])).

```

```

1522 /* (_,R2) - (C.18.1,2) */
1523 cut_elm(proved(Z = [cut(Z1,Z2,C), G => T])) :-
1524     proved(Z1 = [K1, G1=>T1]),
1525     proved(Z2 = [th(Z3,A), G2=>T2]),
1526     proved(Z3 = [K2, G3=>T2]),
1527     append(Phi, [C|Psi], T1),
1528     append3(Phi, T2, Psi, T),
1529     ((append(S1, [A|Zeta], G2),
1530     append(S1, Zeta, G3),
1531     append(Delta, [C|Lambda], S1),
1532     append4(Delta, G1, Lambda, [A|Zeta], G),
1533     append4(Delta, G1, Lambda, Zeta, G4))
1534     ;(append(Delta, [A|E1], G2),
1535     append(Delta, E1, G3),

```

```

1536     append(Lambda, [C|Zeta], E1),
1537     append4(Delta, [A|Lambda], G1, Zeta, G),
1538     append4(Delta, Lambda, G1, Zeta, G4))),
1539     !,
1540     gensym(f, F1),
1541     assert(proved(F1 = [K1, G1 => T1])),
1542     gensym(f, F2),
1543     assert(proved(F2 = [K2, G3 => T2])),
1544     gensym(f, F3),
1545     assert(proved(F3 = [cut(F1, F2, C), G4 => T])),
1546     assert(proved(Z = [ih(F3, A), G => T])),
1547     retract(proved(Z = [cut(Z1, Z2, C), G => T])).

1548 /* (_,R3) - (C.19.1,2) */
1549 cut_eim(proved(Z = [cut(Z1, Z2, C), G => T])) :-
1550     proved(Z1 = [K1, G1 => T1]),
1551     proved(Z2 = [con(Z3, A), G2 => T2]),
1552     proved(Z3 = [K2, G3 => T2]),
1553     append(Phi, [C|Psi], T1),
1554     append3(Phi, T2, Psi, T),
1555     ((append(S1, [A|[A|Zeta]], G3),
1556     append(S1, [A|Zeta], G2),
1557     append(Delta, [C|Lambda], S1),
1558     append4(Delta, G1, Lambda, [A|Zeta], G),
1559     append4(Delta, G1, Lambda, [A|[A|Zeta]], G4))
1560     ; (append(Delta, [A|[A|E1]], G3),
1561     append(Delta, [A|E1], G2),
1562     append(Lambda, [C|Zeta], E1),
1563     append4(Delta, [A|Lambda], G1, Zeta, G),
1564     append4(Delta, [A|[A|Lambda]], G1, Zeta, G4))),
1565     !,
1566     gensym(f, F1),
1567     assert(proved(F1 = [K1, G1 => T1])),
1568     gensym(f, F2),
1569     assert(proved(F2 = [K2, G3 => T2])),
1570     gensym(f, F3),
1571     assert(proved(F3 = [cut(F1, F2, C), G4 => T])),
1572     assert(proved(Z = [con(F3, A), G => T])),
1573     retract(proved(Z = [cut(Z1, Z2, C), G => T])).

1574 /* (_,R3) - (C.19.3) */
1575 cut_eim(proved(Z = [cut(Z1, Z2, C), G => T])) :-
1576     proved(Z1 = [K1, G1 => T1]),
1577     proved(Z2 = [con(Z3, C), G2 => T2]),
1578     proved(Z3 = [K2, G3 => T2]),
1579     append(Phi, [C|Psi], T1),
1580     append3(Phi, T2, Psi, T),
1581     append(Delta, [C|[C|Lambda]], G3),
1582     append(Delta, [C|Lambda], G2),
1583     append3(Delta, G1, Lambda, G),
1584     !,
1585     gensym(f, F1),
1586     assert(proved(F1 = [K1, G1 => T1])),
1587     gensym(f, F2),
1588     assert(proved(F2 = [K2, G3 => T2])),
1589     append3(Delta, [C|G1], Lambda, G4),
1590     gensym(f, F3),
1591     assert(proved(F3 = [cut(F1, F2, C), G4 => T])),
1592     gensym(f, F4),
1593     assert(proved(F4 = [K1, G1 => T1])),
1594     append4(Delta, G1, G1, Lambda, G5),
1595     append(Phi, T, T4),
1596     append(T4, Psi, T3),
1597     gensym(f, F5),
1598     assert(proved(F5 = [cut(F4, F3, C), G5 => T3])),
1599     gensym(f, F6),
1600     assert(proved(F6 = [con(F5, Psi), G5 => T4])),
1601     gensym(f, F7),
1602     assert(proved(F7 = [con(F6, Phi), G5 => T])),
1603     assert(proved(Z = [con(F7, G1), G => T])),
1604     retract(proved(Z = [cut(Z1, Z2, C), G => T])).

1605 /* (_,R4) - (C.20.1,2) */
1606 cut_eim(proved(Z = [cut(Z1, Z2, C), G => T])) :-
1607     proved(Z1 = [K1, G1 => T1]),
1608     proved(Z2 = [inc(Z3, B, A), G2 => T2]),
1609     proved(Z3 = [K2, G3 => T2]),
1610     append(Phi, [C|Psi], T1),
1611     append3(Phi, T2, Psi, T),
1612     ((append(S1, [B|[A|Zeta]], G2),

```

```

1613     append(S1, [A|[B|Zeta]], G3),
1614     append(Delta, [C|Lambda], S1),
1615     append4(Delta, G1, Lambda, [B|[A|Zeta]], G),
1616     append4(Delta, G1, Lambda, [A|[B|Zeta]], G4)),
1617 : (append(Delta, [B|[A|E1]], G2),
1618   append(Delta, [A|[B|E1]], G3),
1619   append(Lambda, [C|Zeta], E1),
1620   append4(Delta, [B|[A|Lambda]], G1, Zeta, G),
1621   append4(Delta, [A|[B|Lambda]], G1, Zeta, G4))),
1622 1,
1623     gensym(f, F1),
1624     assert(proved(F1 = [K1, G1 => T1])),
1625     gensym(f, F2),
1626     assert(proved(F2 = [K2, G3 => T2])),
1627     gensym(f, F3),
1628     assert(proved(F3 = [cut(F1, F2, C), G4 => T])),
1629     assert(proved(Z = [inc(F3, B, A), G => T])),
1630     retract(proved(Z = [cut(Z1, Z2, C), G => T])).

1631 /* (_,R4) - (C.20.3) */
1632 cut_eim(proved(Z = [cut(Z1, Z2, C), G => T])) :-
1633     proved(Z1 = [K1, G1 => T1]),
1634     proved(Z2 = [inc(Z3, C, B), G2 => T2]),
1635     proved(Z3 = [K2, G3 => T2]),
1636     append(Phi, [C|Psi], T1),
1637     append3(Phi, T2, Psi, T),
1638     append(Delta, [C|[B|Lambda]], G2),
1639     append(Delta, [B|[C|Lambda]], G3),
1640     append3(Delta, G1, [B|Lambda], G),
1641 1,
1642     gensym(f, F1),
1643     assert(proved(F1 = [K1, G1 => T1])),
1644     gensym(f, F2),
1645     assert(proved(F2 = [K2, G3 => T2])),
1646     append3(Delta, [B|G1], Lambda, G4),
1647     gensym(f, F3),
1648     assert(proved(F3 = [cut(F1, F2, C), G4 => T])),
1649     assert(proved(Z = [inc(F3, G1, B), G => T])),
1650     retract(proved(Z = [cut(Z1, Z2, C), G => T])).

1651 /* (_,R4) - (C.20.4) */
1652 cut_eim(proved(Z = [cut(Z1, Z2, C), G => T])) :-
1653     proved(Z1 = [K1, G1 => T1]),
1654     proved(Z2 = [inc(Z3, B, C), G2 => T2]),
1655     proved(Z3 = [K2, G3 => T2]),
1656     append(Phi, [C|Psi], T1),
1657     append3(Phi, T2, Psi, T),
1658     append(Delta, [B|[C|Lambda]], G2),
1659     append(Delta, [C|[B|Lambda]], G3),
1660     append3(Delta, [B|G1], Lambda, G),
1661 1,
1662     gensym(f, F1),
1663     assert(proved(F1 = [K1, G1 => T1])),
1664     gensym(f, F2),
1665     assert(proved(F2 = [K2, G3 => T2])),
1666     append3(Delta, G1, [B|Lambda], G4),
1667     gensym(f, F3),
1668     assert(proved(F3 = [cut(F1, F2, C), G4 => T])),
1669     assert(proved(Z = [inc(F3, B, G1), G => T])),
1670     retract(proved(Z = [cut(Z1, Z2, C), G => T])).

1671 /* (_,R5) - (C.21) */
1672 cut_eim(proved(Z = [cut(Z1, Z2, C), G => T])) :-
1673     proved(Z1 = [K1, G1 => T1]),
1674     proved(Z2 = [th(Z3, A), G2 => T2]),
1675     proved(Z3 = [K2, G2 => T3]),
1676     append(Delta, [C|Lambda], G2),
1677     append3(Delta, G1, Lambda, G),
1678     append(Theta, [A|Omega], T2),
1679     append(Theta, Omega, T3),
1680     append(Phi, [C|Psi], T1),
1681     append3(Phi, T2, Psi, T),
1682 1,
1683     gensym(f, F1),
1684     assert(proved(F1 = [K1, G1 => T1])),
1685     gensym(f, F2),
1686     assert(proved(F2 = [K2, G2 => T3])),
1687     append3(Phi, T3, Psi, T4),
1688     gensym(f, F3),
1689     assert(proved(F3 = [cut(F1, F2, C), G => T4])),
1690     assert(proved(Z = [th(F3, A), G => T])),
1691     retract(proved(Z = [cut(Z1, Z2, C), G => T])).

```

```

1692 /* (_,R6) - (C.22) */
1693 cut_eim(proved(Z = [cut(Z1,Z2,C), G => T])) :-
1694     proved(Z1 = [K1,G1=>T1]),
1695     proved(Z2 = [con(Z3,A),G2=>T2]),
1696     proved(Z3 = [K2,G2=>T3]),
1697     append(Delta, [C|Lambda], G2),
1698     append3(Delta, G1, Lambda, G),
1699     append(Theta, [A|[A|Omega]], T3),
1700     append(Theta, [A|Omega], T2),
1701     append(Phi, [C|Psi], T1),
1702     append3(Phi, T2, Psi, T),
1703     !,
1704     gensym(f,F1),
1705     assert(proved(F1 = [K1, G1 => T1])),
1706     gensym(f,F2),
1707     assert(proved(F2 = [K2, G2 => T3])),
1708     append3(Phi, T3, Psi, T4),
1709     gensym(f,F3),
1710     assert(proved(F3 = [cut(F1,F2,C), G => T4])),
1711     assert(proved(Z = [con(F3,A), G => T])),
1712     retract(proved(Z = [cut(Z1,Z2,C), G => T])).

1713 /* (_,R7) - (C.23) */
1714 cut_eim(proved(Z = [cut(Z1,Z2,C), G => T])) :-
1715     proved(Z1 = [K1,G1=>T1]),
1716     proved(Z2 = [inc(Z3,B,A),G2=>T2]),
1717     proved(Z3 = [K2,G2=>T3]),
1718     append(Delta, [C|Lambda], G2),
1719     append3(Delta, G1, Lambda, G),
1720     append(Theta, [B|[A|Omega]], T2),
1721     append(Theta, [A|[B|Omega]], T3),
1722     append(Phi, [C|Psi], T1),
1723     append3(Phi, T2, Psi, T),
1724     !,
1725     gensym(f,F1),
1726     assert(proved(F1 = [K1, G1 => T1])),
1727     gensym(f,F2),
1728     assert(proved(F2 = [K2, G2 => T3])),
1729     append3(Phi, T3, Psi, T4),
1730     gensym(f,F3),
1731     assert(proved(F3 = [cut(F1,F2,C), G => T4])),
1732     assert(proved(Z = [inc(F3,B,A), G => T])),
1733     retract(proved(Z = [cut(Z1,Z2,C), G => T])).

1734 /* (_,R10) - (C.26) */
1735 cut_eim(proved(Z = [cut(Z1,Z2,C), G => T])) :-
1736     proved(Z1 = [K1,G1=>T1]),
1737     proved(Z2 = [ois(Z3,Z4,A & B),G2 => T2]),
1738     proved(Z3 = [K2,G2=>T3]),
1739     proved(Z4 = [K3,G2=>T4]),
1740     append(Delta, [C|Lambda], G2),
1741     append3(Delta, G1, Lambda, G),
1742     append(Theta, [A & B|Omega], T2),
1743     append(Theta, [A|Omega], T3),
1744     append(Theta, [B|Omega], T4),
1745     append(Phi, [C|Psi], T1),
1746     append3(Phi, T2, Psi, T),
1747     !,
1748     gensym(f,F1),
1749     assert(proved(F1 = [K1,G1=>T1])),
1750     gensym(f,F2),
1751     assert(proved(F2 = [K2, G2 => T3])),
1752     append3(Phi, T3, Psi, T5),
1753     gensym(f,F3),
1754     assert(proved(F3 = [cut(F1,F2,C), G => T5])),
1755     gensym(f,F4),
1756     assert(proved(F4 = [K1,G1=>T1])),
1757     gensym(f,F5),
1758     assert(proved(F5 = [K3, G2 => T4])),
1759     append3(Phi, T4, Psi, T6),
1760     gensym(f,F6),
1761     assert(proved(F6 = [cut(F4,F5,C), G => T6])),
1762     assert(proved(Z = [ois(F3,F6,A & B), G => T])),
1763     retract(proved(Z = [cut(Z1,Z2,C), G => T])).

```

```

1764 /* (_,R11) - (C.27.1,2) */
1765 cut_elm(proved(Z = [cut(Z1,Z2,C), G => T])) :-
1766     proved(Z1 = [K1,G1=>T1]),
1767     proved(Z2 = [oia(Z3,A & B),G2 => T2]),
1768     proved(Z3 = [K2,G3=>T2]),
1769     append(Phi, [C|Psi], T1),
1770     append3(Phi, T2, Psi, T),
1771     ((append(S1, [A & B|Zeta], G2),
1772      append(S1, [A|[B|Zeta]], G3),
1773      append(Delta, [C|Lambda], S1),
1774      append4(Delta, G1, Lambda, [A & B|Zeta], G),
1775      append4(Delta, G1, Lambda, [A|[B|Zeta]], G4)),
1776     : (append(Delta, [A & B|E1], G2),
1777       append(Delta, [A|[B|E1]], G3),
1778       append(Lambda, [C|Zeta], E1),
1779       append4(Delta, [A & B|Lambda], G1, Zeta, G),
1780       append4(Delta, [A|[B|Lambda]], G1, Zeta, G4))),
1781     !,
1782     gensym(f,F1),
1783     assert(proved(F1 = [K1, G1 => T1])),
1784     gensym(f,F2),
1785     assert(proved(F2 = [K2, G3 => T2])),
1786     gensym(f,F3),
1787     assert(proved(F3 = [cut(F1,F2,C), G4 => T])),
1788     assert(proved(Z = [oia(F3,A & B), G => T])),
1789     retract(proved(Z = [cut(Z1,Z2,C), G => T])).

```

```

1790 /* (_,R12) - (C.28.1,2) */
1791 cut_elm(proved(Z = [cut(Z1,Z2,C), G => T])) :-
1792     proved(Z1 = [K1,G1=>T1]),
1793     proved(Z2 = [oia(Z3,Z4,A # B),G2=>T2]),
1794     proved(Z3 = [K2,G3=>T2]),
1795     proved(Z4 = [K3,G4=>T2]),
1796     append(Phi, [C|Psi], T1),
1797     append3(Phi, T2, Psi, T),
1798     ((append(S1, [A # B|Zeta], G2),
1799      append(S1, [A|Zeta], G3),
1800      append(S1, [B|Zeta], G4),
1801      append(Delta, [C|Lambda], S1),
1802      append4(Delta, G1, Lambda, [A # B|Zeta], G),
1803      append4(Delta, G1, Lambda, [A|Zeta], G5),
1804      append4(Delta, G1, Lambda, [B|Zeta], G6)),
1805     : (append(Delta, [A # B|E1], G2),
1806       append(Delta, [A|E1], G3),
1807       append(Delta, [B|E1], G4),
1808       append(Lambda, [C|Zeta], E1),
1809       append4(Delta, [A # B|Lambda], G1, Zeta, G),
1810       append4(Delta, [A|Lambda], G1, Zeta, G5),
1811       append4(Delta, [B|Lambda], G1, Zeta, G6))),
1812     !,
1813     gensym(f,F1),
1814     assert(proved(F1 = [K1, G1 => T1])),
1815     gensym(f,F2),
1816     assert(proved(F2 = [K2, G3 => T2])),
1817     gensym(f,F3),
1818     assert(proved(F3 = [cut(F1,F2,C), G5 => T])),
1819     gensym(f,F4),
1820     assert(proved(F4 = [K1, G1 => T1])),
1821     gensym(f,F5),
1822     assert(proved(F5 = [K3, G4 => T2])),
1823     gensym(f,F6),
1824     assert(proved(F6 = [cut(F4,F5,C), G6 => T])),
1825     assert(proved(Z = [oia(F3,F6,A # B), G => T])),
1826     retract(proved(Z = [cut(Z1,Z2,C), G => T])).

```

```

1827 /* (_,R13) - (C.29) */
1828 cut_elm(proved(Z = [cut(Z1,Z2,C), G => T])) :-
1829     proved(Z1 = [K1,G1=>T1]),
1830     proved(Z2 = [ois(Z3,A # B),G2=>T2]),
1831     proved(Z3 = [K2,G2=>T3]),
1832     append(Delta, [C|Lambda], G2),
1833     append3(Delta, G1, Lambda, G),
1834     append(Theta, [A # B|Omega], T2),
1835     append(Theta, [A|[B|Omega]], T3),
1836     append(Phi, [C|Psi], T1),
1837     append3(Phi, T2, Psi, T),
1838     !,
1839     gensym(f,F1),
1840     assert(proved(F1 = [K1, G1 => T1])),

```

```

1841 gensym(f,F2),
1842 assert(proved(F2 = [K2, G2 => T3])),
1843 append3(Phi, T3, Psi, T4),
1844 gensym(f,F3),
1845 assert(proved(F3 = [cut(F1,F2,C), G => T4])),
1846 assert(proved(Z = [ois(F3,A # B), G => T])),
1847 retract(proved(Z = [cut(Z1,Z2,C), G => T])).

1848 /* (_,R14) - (C.30.1) */
1849 cut_eim(proved(Z = [cut(Z1,Z2,C), G => T])) :-
1850     proved(Z1 = [K1,G1=>[C]]),
1851     proved(Z2 = [hia(Z3,Z4,A -> B),G2=>T]),
1852     proved(Z3 = [K2,G3=>[A]]),
1853     proved(Z4 = [K3,G4=>T]),
1854     append(Zeta, [B|Pi], G4),
1855     append3(Zeta, G3, [A -> B|Pi], G2),
1856     append(Delta, [C|Lambda], G3),
1857     append5(Zeta, Delta, G1, Lambda, [A -> B|Pi], G),
1858     !,
1859     gensym(f,F1),
1860     assert(proved(F1 = [K1, G1 => [C]])),
1861     gensym(f,F2),
1862     assert(proved(F2 = [K2, G3 => [A]])),
1863     append3(Delta, G1, Lambda, G5),
1864     gensym(f,F3),
1865     assert(proved(F3 = [cut(F1,F2,C), G5 => [A]])),
1866     gensym(f,F4),
1867     assert(proved(F4 = [K3, G4 => T])),
1868     assert(proved(Z = [hia(F3,F4,A -> B), G => T])),
1869     retract(proved(Z = [cut(Z1,Z2,C), G => T])).

1870 /* (_,R14) - (C.30.2,3) */
1871 cut_eim(proved(Z = [cut(Z1,Z2,C), G => T])) :-
1872     proved(Z1 = [K1,G1=>[C]]),
1873     proved(Z2 = [hia(Z3,Z4,A -> B),G2=>T]),
1874     proved(Z3 = [K2,G3=>[A]]),
1875     proved(Z4 = [K3,G4=>T]),
1876     ((append(S1, [A -> B|Pi], G2),
1877     append(S2, [B|Pi], G4),
1878     append(S2, G3, S1),
1879     append(Lambda, [C|Zeta], S2),
1880     append5(Lambda, G1, Zeta, G3, [A -> B|Pi], G),
1881     append4(Lambda, G1, Zeta, [B|Pi], G5))
1882     ;(append(S3, [A -> B|E1], G2),
1883     append(Lambda, [B|E1], G4),
1884     append(Lambda, G3, S3),
1885     append(Zeta, [C|Pi], E1),
1886     append5(Lambda, G3, [A -> B|Zeta], G1, Pi, G),
1887     append4(Lambda, [B|Zeta], G1, Pi, G5))),
1888     !,
1889     gensym(f,F1),
1890     assert(proved(F1 = [K1, G1 => [C]])),
1891     gensym(f,F2),
1892     assert(proved(F2 = [K3, G4 => T])),
1893     gensym(f,F3),
1894     assert(proved(F3 = [cut(F1,F2,C), G5 => T])),
1895     gensym(f,F4),
1896     assert(proved(F4 = [K2, G3 => [A]])),
1897     assert(proved(Z = [hia(F4,F3,A -> B), G => T])),
1898     retract(proved(Z = [cut(Z1,Z2,C), G => T])).

1899 /* (_,R15) - (C.31.1,2) */
1900 cut_eim(proved(Z = [cut(Z1,Z2,C), G => [A -> B]])) :-
1901     proved(Z1 = [K1,G1=>[C]]),
1902     proved(Z2 = [his(Z3,A -> B),G2=>[A->B]]),
1903     proved(Z3 = [K2,G3=>[B]]),
1904     ((append(S1, [A|Zeta], G3),
1905     append(S1, Zeta, G2),
1906     append(Delta, [C|Lambda], S1),
1907     append4(Delta, G1, Lambda, Zeta, G),
1908     append4(Delta, G1, Lambda, [A|Zeta], G4))
1909     ;(append(Delta, [A|E1], G3),
1910     append(Delta, E1, G2),
1911     append(Lambda, [C|Zeta], E1),
1912     append4(Delta, Lambda, G1, Zeta, G),
1913     append4(Delta, [A|Lambda], G1, Zeta, G4))),
1914     !,

```



```

1915 gensym(f,F1),
1916 assert(proved(F1 = [K1, G1 => [C]])),
1917 gensym(f,F2),
1918 assert(proved(F2 = [K2, G3 => [B]])),
1919 gensym(f,F3),
1920 assert(proved(F3 = [cut(F1,F2,C), G4 => [B]])),
1921 assert(proved(Z = [his(F3,A -> B), G => [A -> B]])),
1922 retract(proved(Z = [cut(Z1,Z2,C), G => [A -> B]])).

```

```

1923 /* (R2,_) - (C.34) */
1924 cut_elm(proved(Z = [cut(Z1,Z2,C), G => T])) :-
1925 proved(Z1 = [th(Z3,A),G1=>T1]),
1926 proved(Z2 = [K1,G2=>T2]),
1927 proved(Z3 = [K2,G3=>T1]),
1928 append(Phi, [C|Psi], T1),
1929 append3(Phi, T2, Psi, T),
1930 append(Gamma, [A|Delta], G1),
1931 append(Gamma, Delta, G3),
1932 append(Lambda, [C|Zeta], G2),
1933 append3(Lambda, G1, Zeta, G),
1934
1935 gensym(f,F1),
1936 assert(proved(F1 = [K2, G3 => T])),
1937 gensym(f,F2),
1938 assert(proved(F2 = [K1, G2 => T])),
1939 append3(Lambda, G3, Zeta, G4),
1940 gensym(f,F3),
1941 assert(proved(F3 = [cut(F1,F2,C), G4 => T])),
1942 assert(proved(Z = [th(F3,A), G => T])),
1943 retract(proved(Z = [cut(Z1,Z2,C), G => T])).

```

```

1944 /* (R3,_) - (C.35) */
1945 cut_elm(proved(Z = [cut(Z1,Z2,C), G => T])) :-
1946 proved(Z1 = [con(Z3,A),G1=>T1]),
1947 proved(Z2 = [K1,G2=>T2]),
1948 proved(Z3 = [K2,G3=>T1]),
1949 append(Phi, [C|Psi], T1),
1950 append3(Phi, T2, Psi, T),
1951 append(Gamma, [A|[A|Delta]], G3),
1952 append(Gamma, [A|Delta], G1),
1953 append(Lambda, [C|Zeta], G2),
1954 append3(Lambda, G1, Zeta, G),
1955
1956 gensym(f,F1),
1957 assert(proved(F1 = [K2, G3 => T])),
1958 gensym(f,F2),
1959 assert(proved(F2 = [K1, G2 => T])),
1960 append3(Lambda, G3, Zeta, G4),
1961 gensym(f,F3),
1962 assert(proved(F3 = [cut(F1,F2,C), G4 => T])),
1963 assert(proved(Z = [con(F3,A), G => T])),
1964 retract(proved(Z = [cut(Z1,Z2,C), G => T])).

```

```

1965 /* (R4,_) - (C.36) */
1966 cut_elm(proved(Z = [cut(Z1,Z2,C), G => T])) :-
1967 proved(Z1 = [inc(Z3,B,A),G1=>T1]),
1968 proved(Z2 = [K1,G2=>T2]),
1969 proved(Z3 = [K2,G3=>T1]),
1970 append(Phi, [C|Psi], T1),
1971 append3(Phi, T2, Psi, T),
1972 append(Gamma, [B|[A|Delta]], G1),
1973 append(Gamma, [A|[B|Delta]], G3),
1974 append(Lambda, [C|Zeta], G2),
1975 append3(Lambda, G1, Zeta, G),
1976
1977 gensym(f,F1),
1978 assert(proved(F1 = [K2, G3 => T])),
1979 gensym(f,F2),
1980 assert(proved(F2 = [K1, G2 => T])),
1981 append3(Lambda, G3, Zeta, G4),
1982 gensym(f,F3),
1983 assert(proved(F3 = [cut(F1,F2,C), G4 => T])),
1984 assert(proved(Z = [inc(F3,B,A), G => T])),
1985 retract(proved(Z = [cut(Z1,Z2,C), G => T])).

```

```

1986 /* (R5,_) - (C.37.1,2) */
1987 cut_elm(proved(Z = [cut(Z1,Z2,C), G => T])) :-
1988 proved(Z1 = [th(Z3,A),G1=>T1]),
1989 proved(Z2 = [K1,G2=>T2]),

```

```

1990      proved(Z3 = [K2,G1=>T3]),
1991      append(Delta, [C|Lambda], G2),
1992      append3(Delta, G1, Lambda, G),
1993      ((append(S1, [A|Theta], T1),
1994       append(S1, Theta, T3),
1995       append(Phi, [C|Psi], S1),
1996       append4(Phi, T2, Psi, [A|Theta], T),
1997       append4(Phi, T2, Psi, Theta, T4))
1998      : (append(Phi, [A|E1], T1),
1999       append(Phi, E1, T3),
2000       append(Psi, [C|Theta], E1),
2001       append4(Phi, [A|Psi], T2, Theta, T),
2002       append4(Phi, Psi, T2, Theta, T4))),
2003      !,
2004      gensym(f, F1),
2005      assert(proved(F1 = [K2, G1 => T3])),
2006      gensym(f, F2),
2007      assert(proved(F2 = [K1, G2 => T2])),
2008      gensym(f, F3),
2009      assert(proved(F3 = [cut(F1, F2, C), G => T4])),
2010      assert(proved(Z = [th(F3, A), G => T])),
2011      retract(proved(Z = [cut(Z1, Z2, C), G => T])).

```

```

2012 /* (R6,_) - (C.38.1.2) */
2013 cut_olm(proved(Z = [cut(Z1, Z2, C), G => T])) :-
2014     proved(Z1 = [con(Z3, A), G1=>T1]),
2015     proved(Z2 = [K1, G2=>T2]),
2016     proved(Z3 = [K2, G1=>T3]),
2017     append(Delta, [C|Lambda], G2),
2018     append3(Delta, G1, Lambda, G),
2019     ((append(S1, [A|[A|Theta]], T3),
2020      append(S1, [A|Theta], T1),
2021      append(Phi, [C|Psi], S1),
2022      append4(Phi, T2, Psi, [A|Theta], T),
2023      append4(Phi, T2, Psi, [A|[A|Theta]], T4))
2024     : (append(Phi, [A|[A|E1]], T3),
2025      append(Phi, [A|E1], T1),
2026      append(Psi, [C|Theta], E1),
2027      append4(Phi, [A|Psi], T2, Theta, T),
2028      append4(Phi, [A|[A|Psi]], T2, Theta, T4))),
2029     !,
2030     gensym(f, F1),
2031     assert(proved(F1 = [K2, G1 => T3])),
2032     gensym(f, F2),
2033     assert(proved(F2 = [K1, G2 => T2])),
2034     gensym(f, F3),
2035     assert(proved(F3 = [cut(F1, F2, C), G => T4])),
2036     assert(proved(Z = [con(F3, A), G => T])),
2037     retract(proved(Z = [cut(Z1, Z2, C), G => T])).

```

```

2038 /* (R6,_) - (C.38.3) */
2039 cut_olm(proved(Z = [cut(Z1, Z2, C), G => T])) :-
2040     proved(Z1 = [con(Z3, C), G1=>T1]),
2041     proved(Z2 = [K1, G2=>T2]),
2042     proved(Z3 = [K2, G1=>T3]),
2043     append(Delta, [C|Lambda], G2),
2044     append3(Delta, G1, Lambda, G),
2045     append(Phi, [C|[C|Psi]], T3),
2046     append(Phi, [C|Psi], T1),
2047     append3(Phi, T2, Psi, T),
2048     !,
2049     gensym(f, F1),
2050     assert(proved(F1 = [K2, G1 => T3])),
2051     gensym(f, F2),
2052     assert(proved(F2 = [K1, G2 => T2])),
2053     append3(Phi, [C|T2], Psi, T4),
2054     gensym(f, F3),
2055     assert(proved(F3 = [cut(F1, F2, C), G => T4])),
2056     gensym(f, F4),
2057     assert(proved(F4 = [K1, G2 => T2])),
2058     append3(Delta, G, Lambda, G3),
2059     append4(Phi, T2, T2, Psi, T5),
2060     gensym(f, F5),
2061     assert(proved(F5 = [cut(F3, F4, C), G3 => T5])),
2062     gensym(f, F6),
2063     assert(proved(F6 = [con(F5, T2), G3 => T])),
2064     append(Delta, G, G4),
2065     gensym(f, F7),
2066     assert(proved(F7 = [con(F6, Lambda), G4 => T])),
2067     assert(proved(Z = [con(F7, Delta), G => T])),
2068     retract(proved(Z = [cut(Z1, Z2, C), G => T])).

```

```

2069 /* (R7,_) - (C.39.1,2) */
2070 cut_elm(proved(Z = [cut(Z1,Z2,C), G => T])) :-
2071   proved(Z1 = [inc(Z3,B,A),G1=>T1]),
2072   proved(Z2 = [K1,G2=>T2]),
2073   proved(Z3 = [K2,G1=>T3]),
2074   append(Delta, [C|Lambda], G2),
2075   append3(Delta, G1, Lambda, G),
2076   ((append(S1, [B|[A|Theta]], T1),
2077     append(S1, [A|[B|Theta]], T3),
2078     append(Phi, [C|Psi], S1),
2079     append4(Phi, T2, Psi, [B|[A|Theta]], T),
2080     append4(Phi, T2, Psi, [A|[B|Theta]], T4))
2081   ;(append(Phi, [B|[A|E1]], T1),
2082     append(Phi, [A|[B|E1]], T3),
2083     append(Psi, [C|Theta], E1),
2084     append4(Phi, [B|[A|Psi]], T2, Theta, T),
2085     append4(Phi, [A|[B|Psi]], T2, Theta, T4))),
2086   !,
2087   gensym(f,F1),
2088   assert(proved(F1 = [K2, G1 => T3])),
2089   gensym(f,F2),
2090   assert(proved(F2 = [K1, G2 => T2])),
2091   gensym(f,F3),
2092   assert(proved(F3 = [cut(F1,F2,C), G => T4])),
2093   assert(proved(Z = [inc(F3,B,A), G => T])),
2094   retract(proved(Z = [cut(Z1,Z2,C), G => T])).

```

```

2095 /* (R7,_) - (C.39.3) */
2096 cut_elm(proved(Z = [cut(Z1,Z2,C), G => T])) :-
2097   proved(Z1 = [inc(Z3,C,D),G1=>T1]),
2098   proved(Z2 = [K1,G2=>T2]),
2099   proved(Z3 = [K2,G1=>T3]),
2100   append(Delta, [C|Lambda], G2),
2101   append3(Delta, G1, Lambda, G),
2102   append(Phi, [C|[D|Psi]], T1),
2103   append(Phi, [D|[C|Psi]], T3),
2104   append3(Phi, T2, [D|Psi], T),
2105   !,
2106   gensym(f,F1),
2107   assert(proved(F1 = [K2, G1 => T3])),
2108   gensym(f,F2),
2109   assert(proved(F2 = [K1, G2 => T2])),
2110   append3(Phi, [D|T2], Psi, T4),
2111   gensym(f,F3),
2112   assert(proved(F3 = [cut(F1,F2,C), G => T4])),
2113   assert(proved(Z = [inc(F3,T2,D), G => T])),
2114   retract(proved(Z = [cut(Z1,Z2,C), G => T])).

```

```

2115 /* (R7,_) - (C.39.4) */
2116 cut_elm(proved(Z = [cut(Z1,Z2,C), G => T])) :-
2117   proved(Z1 = [inc(Z3,D,C),G1=>T1]),
2118   proved(Z2 = [K1,G2=>T2]),
2119   proved(Z3 = [K2,G1=>T3]),
2120   append(Delta, [C|Lambda], G2),
2121   append3(Delta, G1, Lambda, G),
2122   append(Phi, [D|[C|Psi]], T1),
2123   append(Phi, [C|[D|Psi]], T3),
2124   append3(Phi, [D|T2], Psi, T),
2125   !,
2126   gensym(f,F1),
2127   assert(proved(F1 = [K2, G1 => T3])),
2128   gensym(f,F2),
2129   assert(proved(F2 = [K1, G2 => T2])),
2130   append3(Phi, T2, [D|Psi], T4),
2131   gensym(f,F3),
2132   assert(proved(F3 = [cut(F1,F2,C), G => T4])),
2133   assert(proved(Z = [inc(F3,D,T2), G => T])),
2134   retract(proved(Z = [cut(Z1,Z2,C), G => T])).

```

```

2135 /* (R10,_) - (C.41.1,2) */
2136 cut_elm(proved(Z = [cut(Z1,Z2,C), G => T])) :-
2137   proved(Z1 = [ais(Z3,Z4,A & B),G1 => T1]),
2138   proved(Z2 = [K1,G2=>T2]),
2139   proved(Z3 = [K2,G1=>T3]),
2140   proved(Z4 = [K3,G1=>T4]),
2141   append(Delta, [C|Lambda], G2),
2142   append3(Delta, G1, Lambda, G),
2143   ((append(S1, [A & B|Theta], T1),

```

```

2144     append(S1, [A|Theta], T3),
2145     append(S1, [B|Theta], T4),
2146     append(Phi, [C|Psi], S1),
2147     append4(Phi, T2, Psi, [A & B|Theta], T),
2148     append4(Phi, T2, Psi, [A|Theta], T5),
2149     append4(Phi, T2, Psi, [B|Theta], T6)),
2150 : (append(Phi, [A & B|E1], T1),
2151     append(Phi, [A|E1], T3),
2152     append(Phi, [B|E1], T4),
2153     append(Psi, [C|Theta], E1),
2154     append4(Phi, [A & B|Psi], T2, Theta, T),
2155     append4(Phi, [A|Psi], T2, Theta, T5),
2156     append4(Phi, [B|Psi], T2, Theta, T6))),
2157 1,
2158     gensym(f, F1),
2159     assert(proved(F1 = [K2, G1 => T3])),
2160     gensym(f, F2),
2161     assert(proved(F2 = [K1, G2 => T2])),
2162     gensym(f, F3),
2163     assert(proved(F3 = [cut(F1, F2, C), G => T5])),
2164     gensym(f, F4),
2165     assert(proved(F4 = [K3, G1 => T4])),
2166     gensym(f, F5),
2167     assert(proved(F5 = [K1, G2 => T2])),
2168     gensym(f, F6),
2169     assert(proved(F6 = [cut(F4, F5, C), G => T6])),
2170     assert(proved(Z = [ois(F3, F6, A & B), G => T])),
2171     retract(proved(Z = [cut(Z1, Z2, C), G => T])).

```

```

2172 /* (R11,_) - (C.42) */
2173 cut_elm(proved(Z = [cut(Z1, Z2, C), G => T])) :-
2174     proved(Z1 = [aia(Z3, A & B), G1 => T1]),
2175     proved(Z2 = [K1, G2 => T2]),
2176     proved(Z3 = [K2, G3 => T1]),
2177     append(Phi, [C|Psi], T1),
2178     append3(Phi, T2, Psi, T),
2179     append(Gamma, [A & B|Delta], G1),
2180     append(Gamma, [A|B|Delta], G3),
2181     append(Lambda, [C|Zeta], G2),
2182     append3(Lambda, G1, Zeta, G),
2183 1,
2184     gensym(f, F1),
2185     assert(proved(F1 = [K2, G3 => T1])),
2186     gensym(f, F2),
2187     assert(proved(F2 = [K1, G2 => T2])),
2188     append3(Lambda, G3, Zeta, G4),
2189     gensym(f, F3),
2190     assert(proved(F3 = [cut(F1, F2, C), G4 => T])),
2191     assert(proved(Z = [aia(F3, A & B), G => T])),
2192     retract(proved(Z = [cut(Z1, Z2, C), G => T])).

```

```

2193 /* (R12,_) - (C.43) */
2194 cut_elm(proved(Z = [cut(Z1, Z2, C), G => T])) :-
2195     proved(Z1 = [oia(Z3, Z4, A # B), G1 => T1]),
2196     proved(Z2 = [K1, G2 => T2]),
2197     proved(Z3 = [K2, G3 => T1]),
2198     proved(Z4 = [K3, G4 => T1]),
2199     append(Phi, [C|Psi], T1),
2200     append3(Phi, T2, Psi, T),
2201     append(Gamma, [A # B|Delta], G1),
2202     append(Gamma, [A|Delta], G3),
2203     append(Gamma, [B|Delta], G4),
2204     append(Lambda, [C|Zeta], G2),
2205     append3(Lambda, G1, Zeta, G),
2206 1,
2207     gensym(f, F1),
2208     assert(proved(F1 = [K2, G3 => T1])),
2209     gensym(f, F2),
2210     assert(proved(F2 = [K1, G2 => T2])),
2211     append3(Lambda, G3, Zeta, G5),
2212     gensym(f, F3),
2213     assert(proved(F3 = [cut(F1, F2, C), G5 => T])),
2214     gensym(f, F4),
2215     assert(proved(F4 = [K3, G4 => T1])),
2216     gensym(f, F5),
2217     assert(proved(F5 = [K1, G2 => T2])),
2218     append3(Lambda, G4, Zeta, G6),
2219     gensym(f, F6),
2220     assert(proved(F6 = [cut(F4, F5, C), G6 => T])),
2221     assert(proved(Z = [oia(F3, F6, A # B), G => T])),
2222     retract(proved(Z = [cut(Z1, Z2, C), G => T])).

```

```

2223 /* (R13,_) - (C.44.1,2) */
2224 cut_elm(proved(Z = [cut(Z1,Z2,C), G => T])) :-
2225   proved(Z1 = [ois(Z3,A # B),G1=>T1]),
2226   proved(Z2 = [K1,G2=>T2]),
2227   proved(Z3 = [K2,G1=>T3]),
2228   append(Delta, [C|Lambda], G2),
2229   append3(Delta, G1, Lambda, G),
2230   ((append(S1, [A # B|Theta], T1);
2231     append(S1, [A|B|Theta], T3);
2232     append(Phi, [C|Psi], S1);
2233     append4(Phi, T2, Psi, [A # B|Theta], T);
2234     append4(Phi, T2, Psi, [A|B|Theta], T4))
2235   ;(append(Phi, [A # B|E1], T1);
2236     append(Phi, [A|B|E1], T3);
2237     append(Psi, [C|Theta], E1);
2238     append4(Phi, [A # B|Psi], T2, Theta, T);
2239     append4(Phi, [A|B|Psi], T2, Theta, T4))),
2240   !,
2241   gensym(f,F1),
2242   assert(proved(F1 = [K2, G1 => T3])),
2243   gensym(f,F2),
2244   assert(proved(F2 = [K1, G2 => T2])),
2245   gensym(f,F3),
2246   assert(proved(F3 = [cut(F1,F2,C), G => T4])),
2247   assert(proved(Z = [ois(F3,A # B), G => T])),
2248   retract(proved(Z = [cut(Z1,Z2,C), G => T])).

```

```

2249 /* (R14,_) - (C.45) */
2250 cut_elm(proved(Z = [cut(Z1,Z2,C), G => T])) :-
2251   proved(Z1 = [hia(Z3,Z4,A -> B),G1=>T1]),
2252   proved(Z2 = [K1,G2=>T2]),
2253   proved(Z3 = [K2,G3=>[A]]),
2254   proved(Z4 = [K3,G4=>T1]),
2255   append(Phi, [C|Psi], T1),
2256   append3(Phi, T2, Psi, T),
2257   append(S1, [A -> B|Lambda], G1),
2258   append(Delta, [B|Lambda], G4),
2259   append(Delta, G3, S1),
2260   append(Zeta, [C|Pi], G2),
2261   append3(Zeta, G1, Pi, G),
2262   !,
2263   gensym(f,F1),
2264   assert(proved(F1 = [K3, G4 => T1])),
2265   gensym(f,F2),
2266   assert(proved(F2 = [K1, G2 => T2])),
2267   append3(Zeta, G4, Pi, G5),
2268   gensym(f,F3),
2269   assert(proved(F3 = [cut(F1,F2,C), G5 => T])),
2270   gensym(f,F4),
2271   assert(proved(F4 = [K2, G3 => [A]])),
2272   assert(proved(Z = [hia(F4,F3,A -> B), G => T])),
2273   retract(proved(Z = [cut(Z1,Z2,C), G => T])).

```

```

2274 /* this part is to remove all steps of (R14)(hook_introduction
2275 in the antecedent with the property that the active formulas
2276 in their right premisses are isomorphic to T (the terminal
2277 object) */

2278 hook_terminal :-
2279     (thry(cartcl) ; thry(bicartcl)),
2280     proved(Z = [hia(Z1,Z2,A -> B), G => T]),
2281     isom_tr(B),
2282     hook_termin(Z = [hia(Z1,Z2,A -> B), G => T]),
2283     fail.

2284 hook_terminal.

2285 hook_termin(Z = [hia(Z1,Z2,A -> B), G => T]) :-
2286     proved(Z1 = [K1, G1 => [A]]),
2287     proved(Z2 = [K, G2 => T]),
2288     append(Delta, [B|Lambda], G2),
2289     append3(Delta, G1, [A -> B|Lambda], G),
2290     not(thinning(Z2,B,Delta,Lambda)),
2291     gensym(f,Z3),
2292     expterm(Z3, B),
2293     gensym(f,Z4),
2294     append(Delta, Lambda, G3),
2295     assert(proved(Z4 = [cut(Z3,Z2,B), G3 => T])),
2296     append(Delta, [A -> B|Lambda], G4),
2297     gensym(f,Z5),
2298     assert(proved(Z5 = [th(Z4,A -> B), G4 => T])),
2299     assert(proved(Z = [th(Z5,G1), G => T])),
2300     retract(proved(Z1 = [K1, G1 => [A]])),
2301     retract(proved(Z = [hia(Z1,Z2,A -> B), G => T])),
2302     main(X), assert(hook_cut),
2303     cut_free(X),
2304     retract(hook_cut),
2305     !.

2306 hook_termin(Z).

2307 thinning(Z2,B,Delta,Lambda) :-
2308     proved(Z2 = [th(Z3,B), G2 => T]),
2309     proved(Z3 = [_, G3 => T]),
2310     append(Delta, Lambda, G3).

2311 /* this part is to deal with the terminal and initial arrows */
2312 term_inti :-
2313     proved(X),
2314     terminal(X),
2315     initial(X),
2316     fail.

2317 term_inti.

2318 /* deal with terminal arrows */
2319 terminal(X = [L, G => T]) :-
2320     G \== [],
2321     ((thry(cart) ; thry(cartcl)),
2322     proved(X = [L, G => [Y]]),
2323     isom_tr(Y),
2324     gensym(f,X1),
2325     expterm(X1, Y),
2326     asserta(proved(X = [th(X1,G), G => T])))
2327 ; ((thry(bicart) ; thry(dbicart) ; thry(bicartcl)),
2328     append(Delta, [Y|Phi], T),
2329     isom_tr(Y),
2330     gensym(f,X1),
2331     expterm(X1, Y),
2332     gensym(f,X2),
2333     append([Y], Phi, T1),
2334     asserta(proved(X2 = [th(X1,Phi), [] => T1])),
2335     gensym(f,X3),
2336     asserta(proved(X3 = [th(X2,Delta), [] => T])),
2337     asserta(proved(X = [th(X3,G), G => T]))),
2338     retract(proved(X = [L, G => T])),
2339     !.

```

```

2340 terminal(X).

2342 /* to check if an object X is isomorphic to a terminal object */
2343 isom_tr(X) :-
2344     X = tr,
2345     !.

2346 isom_tr(X) :-
2347     X =.. [&,A,B],
2348     !,
2349     isom_tr(A), isom_tr(B).

2350 isom_tr(X) :-
2351     X =.. [->,B,A],
2352     !,
2353     ((thry(cartcl) ; thry(bicartcl))),
2354     isom_tr(A), assert(term_hook(A))),
2355     ;(thry(bicartcl),
2356     isom_bo(B), assert(bo_hook(B)))).

2357 isom_tr(X) :-
2358     X =.. [# ,A,B],
2359     !,
2360     ((isom_tr(A), isom_bo(B), assert(term_or(A))))),
2361     ;(isom_tr(B), isom_bo(A), assert(term_or(B)))).

2362 /* to produce a unique derivation of --> Y, where Y is
2363 isomorphic to a terminal object */
2364 expterm(X1, Y) :-
2365     ((Y = tr,
2366     asserta(proved(X1 = [true, [] => [tr]])))
2367     ;(Y =.. [&,A,B],
2368     gensym(f,X2),
2369     gensym(f,X3),
2370     expterm(X2, A),
2371     expterm(X3, B),
2372     asserta(proved(X1 = [als(X2,X3,A&B), [] => [A & B]])))
2373     ;(Y =.. [->,B,A],
2374     term_hook(A), retract(term_hook(A)),
2375     gensym(f,X2),
2376     expterm(X2, A),
2377     gensym(f, X3),
2378     asserta(proved(X3 = [th(X2,B), [B] => [A]])),
2379     asserta(proved(X1 = [his(X3,B -> A), [] => [B -> A]])))
2380     ;(Y =.. [->,B,A],
2381     bo_hook(B), retract(bo_hook(B)),
2382     gensym(f,X2),
2383     expinti(X2, B),
2384     gensym(f, X3),
2385     asserta(proved(X3 = [th(X2,A), [B] => [A]])),
2386     asserta(proved(X1 = [his(X3,B -> A), [] => [B -> A]])))
2387     ;(Y =.. [# ,A,B],
2388     gensym(f,X2),
2389     gensym(f,X3),
2390     ((term_or(A), retract(term_or(A)),
2391     expterm(X2, A),
2392     asserta(proved(X3 = [th(X2,B), [] => [A,B]])))
2393     ;(term_or(B), retract(term_or(B)),
2394     expterm(X2, B),
2395     asserta(proved(X3 = [th(X2,A), [] => [A,B]]))))),
2396     asserta(proved(X1 = [ois(X3,A # B), [] => [A # B]])))).

2398 /* deal with initial arrows */
2399 initial(X = [L, G => T]) :-
2400     (thry(bicart) ; thry(dbicart) ; thry(bicartcl)),
2401     T \== [],
2402     append(Psi, [E|Psi], T),
2403     not(isom_tr(E)),
2404     append(Delta, [Y|Lambda], G),
2405     isom_bo(Y),
2406     gensym(f,X1),
2407     expinti(X1, Y),
2408     gensym(f,X2),

```

```

2409         append([Y], Lambda, G1),
2410         asserta(proved(X2 = [th(X1,Lambda), G1 => []])),
2411         gensym(f,X3),
2412         asserta(proved(X3 = [th(X2,Delta), G => []])),
2413         asserta(proved(X = [th(X3,T), G => T])),
2414         retract(proved(X = [L, G => T])),
2415     ].

2416 initial(X).

2417 /* to check if an object X is isomorphic to an initial object */
2418 isom_bo(X) :-
2419     X = bo,
2420     !.

2421 isom_bo(X) :-
2422     X =.. [# ,A,B],
2423     !,
2424     isom_bo(A), isom_bo(B).

2425 isom_bo(X) :-
2426     X =.. [& ,A,B],
2427     !,
2428     ((thry(bicart),
2429      ((isom_tr(A), isom_bo(B), assert(inti_and(B)))
2430       ;(isom_tr(B), isom_bo(A), assert(inti_and(A))))))
2431      ;((thry(dbicart) ; thry(bicartcl)),
2432       ((isom_bo(A), assert(inti_and(A)))
2433        ;(isom_bo(B), assert(inti_and(B))))))).

2434 /* to produce a unique derivation of Y ---> , where Y is
2435    isomorphic to an initial object */

2436 expinti(X1,Y) :-
2437     ((Y = bo,
2438      asserta(proved(X1 = [bottom, [bo] => []])))
2439     ;(Y =.. [# ,A,B],
2440      gensym(f,X2),
2441      gensym(f,X3),
2442      expinti(X2, A),
2443      expinti(X3, B),
2444      asserta(proved(X1 = [oia(X2,X3,A#B), [A#B] => []])))
2445     ;(Y =.. [& ,A,B],
2446      gensym(f,X2),
2447      gensym(f,X3),
2448      ((inti_and(A), retract(inti_and(A)),
2449       expinti(X2, A),
2450       asserta(proved(X3 = [th(X2,B), [A,B] => []])))
2451      ;(inti_and(B), retract(inti_and(B)),
2452       expinti(X2, B),
2453       asserta(proved(X3 = [th(X2,A), [A,B] => []])))),
2454     asserta(proved(X1 = [oia(X3,A&B), [A&B] => []]))).

```



2455

/\* Contraction &amp; Interchange Elimination \*/

2456 /\* This is a program to eliminate contraction and interchange  
 2457 in the case of Fc(X), Fbc(X) and Fdbc(X) \*/

2458 con\_inc\_free(X) :-

```
2459     cop,
2460     write('Proof is as follows :'),nl,
2461     write('===== '),nl,nl,
2462     printall(X),nl,nl,
2463     assert(main(X)),
2464     normal1.
```

2465 normal1 :-

```
2466     main(Z),
2467     repeat,
2468         do1_step,
2469     (thry(cartcl)
2470     ; thry(bicartcl)
2471     ; (write('Contraction and interchange free proof is :'),nl,
2472       write('===== '),nl,nl,
2473       printall(Z), nl, nl)).
```

2474 do1\_step :-

```
2475     main(X),
2476     put_in_list(X,L),
2477     assertz(proved(mark)),
2478     proved(Z),
2479     norm1(proved(Z)),
2480     Z = mark,
2481     put_in_list(X,L1),
2482     abolish(proved,1),
2483     ass_list(L1), 1,
2484     L == L1.
```

2485 /\* (R2,R2) ~ (E.1) \*/

```
2486 norm1(proved(Z = [th(Z1,B), G => T])) :-
2487     proved(Z1 = [th(Z2,A), G1 => T]),
2488     proved(Z2 = [_, G2 => T]),
2489     append(Gamma, [A|E1], G1),
2490     append(Gamma, E1, G2),
2491     append(Gamma, [A|E], G),
2492     append(Delta, [B|Lambda], E),
2493     append(Delta, Lambda, E1),
2494     !,
2495     assertz(proved(Z = [th(Z1,A), G => T])),
2496     append(Gamma, E, G3),
2497     assertz(proved(Z1 = [th(Z2,B), G3 => T])),
2498     retract(proved(Z = [th(Z1,B), G => T])),
2499     retract(proved(Z1 = [th(Z2,A), G1 => T])),
2500     !.
```

2501 /\* (R2,R3) ~ (E.2.1) \*/

```
2502 norm1(proved(Z = [con(Z1,A), G => T])) :-
2503     proved(Z1 = [th(Z2,A), G1 => T]),
2504     proved(Z2 = [F, G => T]),
2505     append(Gamma, [A|[A|Delta]], G1),
2506     append(Gamma, [A|Delta], G),
2507     !,
2508     assertz(proved(Z = [F, G => T])),
2509     retract(proved(Z = [con(Z1,A), G => T])),
2510     retract(proved(Z1 = [th(Z2,A), G1 => T])),
2511     retract(proved(Z2 = [F, G => T])),
2512     !.
```

2513 /\* (R2,R3) ~ (E.2.2,3) \*/

```
2514 norm1(proved(Z = [con(Z1,A), G => T])) :-
2515     proved(Z1 = [th(Z2,B), G1 => T]),
2516     proved(Z2 = [_, G2 => T]),
2517     ((append(Gamma, [A|[A|E]], G1),
2518     append(Gamma, [A|E], G),
2519     append(Gamma, [A|[A|E1]], G2),
2520     append(Delta, [B|Lambda], E),
2521     append(Delta, Lambda, E1),
```

```

2522     append(Gamma, [A|E1], G3))
2523   : (append(S, [A|[A|Lambda]], G1),
2524     append(S, [A|Lambda], G),
2525     append(S1, [A|[A|Lambda]], G2),
2526     append(Gamma, [B|Delta], S),
2527     append(Gamma, Delta, S1),
2528     append(S1, [A|Lambda], G3))),
2529   |
2530   assertz(proved(Z = [th(Z1,B), G => T])),
2531   assertz(proved(Z1 = [con(Z2,A), G3 => T])),
2532   retract(proved(Z = [con(Z1,A), G => T])),
2533   retract(proved(Z1 = [th(Z2,B), G1 => T])),
2534   |.

2535 /* (R2,R4) - (E.3.1) */
2536 norm1(proved(Z = [inc(Z1,B,A), G => T])) :-
2537   proved(Z1 = [th(Z2,B), G1 => T]),
2538   proved(Z2 = [-, G2 => T]),
2539   append(Gamma, [A|[B|Delta]], G1),
2540   append(Gamma, [B|[A|Delta]], G),
2541   append(Gamma, [A|Delta], G2),
2542   |
2543   assertz(proved(Z = [th(Z2,B), G => T])),
2544   retract(proved(Z1 = [th(Z2,B), G1 => T])),
2545   retract(proved(Z = [inc(Z1,B,A), G => T])),
2546   |.

2547 /* (R2,R4) - (E.3.2) */
2548 norm1(proved(Z = [inc(Z1,A,B), G => T])) :-
2549   proved(Z1 = [th(Z2,B), G1 => T]),
2550   proved(Z2 = [-, G2 => T]),
2551   append(Gamma, [A|[B|Delta]], G),
2552   append(Gamma, [B|[A|Delta]], G1),
2553   append(Gamma, [A|Delta], G2),
2554   |
2555   assertz(proved(Z = [th(Z2,B), G => T])),
2556   retract(proved(Z1 = [th(Z2,B), G1 => T])),
2557   retract(proved(Z = [inc(Z1,A,B), G => T])),
2558   |.

2559 /* (R2,R4) - (E.3.3,4) */
2560 norm1(proved(Z = [inc(Z1,B,A), G => T])) :-
2561   proved(Z1 = [th(Z2,C), G1 => T]),
2562   proved(Z2 = [-, G2 => T]),
2563   ((append(Gamma, [B|[A|E]], G),
2564     append(Gamma, [A|[B|E]], G1),
2565     append(Gamma, [A|[B|E1]], G2),
2566     append(Delta, [C|Lambda], E),
2567     append(Delta, Lambda, E1),
2568     append(Gamma, [B|[A|E1]], G3))
2569   : (append(S, [B|[A|Lambda]], G),
2570     append(S, [A|[B|Lambda]], G1),
2571     append(S1, [A|[B|Lambda]], G2),
2572     append(Gamma, [C|Delta], S),
2573     append(Gamma, Delta, S1),
2574     append(S1, [B|[A|Lambda]], G3))),
2575   |
2576   assertz(proved(Z = [th(Z1,C), G => T])),
2577   assertz(proved(Z1 = [inc(Z2,B,A), G3 => T])),
2578   retract(proved(Z1 = [th(Z2,C), G1 => T])),
2579   retract(proved(Z = [inc(Z1,B,A), G => T])),
2580   |.

2581 /* (R2,R6) - (E.4) */
2582 norm1(proved(Z = [con(Z1,A), G => T])) :-
2583   proved(Z1 = [th(Z2,B), G => T]),
2584   proved(Z2 = [-, G1 => T]),
2585   append(Gamma, [B|Delta], G),
2586   append(Gamma, Delta, G1),
2587   append(Phi, [A|[A|Psi]], T1),
2588   append(Phi, [A|Psi], T),
2589   |
2590   assertz(proved(Z = [th(Z1,B), G => T])),
2591   assertz(proved(Z1 = [con(Z2,A), G1 => T])),
2592   retract(proved(Z = [con(Z1,A), G => T])),
2593   retract(proved(Z1 = [th(Z2,B), G => T])),
2594   |.

```

```

2595 /* (R2,R7) - (E.5) */
2596 norm1(proved(Z = [inc(Z1,B,A), G => T])) :-
2597   proved(Z1 = [th(Z2,C), G => T]),
2598   proved(Z2 = [-, G1 => T]),
2599   append(Gamma, [C|Delta], G),
2600   append(Gamma, Delta, G1),
2601   append(Phi, [B|[A|Psi]], T),
2602   append(Phi, [A|[B|Psi]], T1),
2603   !,
2604   assertz(proved(Z = [th(Z1,C), G => T])),
2605   assertz(proved(Z1 = [inc(Z2,B,A), G1 => T])),
2606   retract(proved(Z = [inc(Z1,B,A), G => T])),
2607   retract(proved(Z1 = [th(Z2,C), G => T])),
2608   !.

```

```

2609 /* (R3,R4) - (E.6.1,2) */
2610 norm1(proved(Z = [inc(Z1,C,B), G => T])) :-
2611   proved(Z1 = [con(Z2,A), G1 => T]),
2612   proved(Z2 = [-, G2 => T]),
2613   ((append(S1, [C|[B|Lambda]], G),
2614     append(S1, [B|[C|Lambda]], G1),
2615     append(S, [B|[C|Lambda]], G2),
2616     append(Gamma, [A|[A|Delta]], S),
2617     append(Gamma, [A|Delta], S1),
2618     append(S, [C|[B|Lambda]], G3))
2619   ;(append(Gamma, [C|[B|E1]], G),
2620     append(Gamma, [B|[C|E1]], G1),
2621     append(Gamma, [B|C|E]], G2),
2622     append(Delta, [A|[A|Lambda]], E),
2623     append(Delta, [A|Lambda], E1),
2624     append(Gamma, [C|[B|E]], G3))),
2625   !,
2626   assertz(proved(Z = [con(Z1,A), G => T])),
2627   assertz(proved(Z1 = [inc(Z2,C,B), G3 => T])),
2628   retract(proved(Z = [inc(Z1,C,B), G => T])),
2629   retract(proved(Z1 = [con(Z2,A), G1 => T])),
2630   !.

```

```

2631 /* (R3,R4) - (E.6.3) */
2632 norm1(proved(Z = [inc(Z1,B,A), G => T])) :-
2633   proved(Z1 = [con(Z2,A), G1 => T]),
2634   proved(Z2 = [-, G2 => T]),
2635   append(Gamma, [B|[A|Lambda]], G),
2636   append(Gamma, [A|[B|Lambda]], G1),
2637   append(Gamma, [A|[A|[B|Lambda]]], G2),
2638   !,
2639   assertz(proved(Z = [con(Z1,A), G => T])),
2640   append(Gamma, [B|[A|[A|Lambda]]], G3),
2641   gensym(f,F),
2642   assertz(proved(Z1 = [inc(F,B,A), G3 => T])),
2643   append(Gamma, [A|[B|[A|Lambda]]], G4),
2644   assertz(proved(F = [inc(Z2,B,A), G4 => T])),
2645   retract(proved(Z = [inc(Z1,B,A), G => T])),
2646   retract(proved(Z1 = [con(Z2,A), G1 => T])),
2647   !.

```

```

2648 /* (R3,R4) - (E.6.4) */
2649 norm1(proved(Z = [inc(Z1,A,B), G => T])) :-
2650   proved(Z1 = [con(Z2,A), G1 => T]),
2651   proved(Z2 = [-, G2 => T]),
2652   append(Gamma, [A|[B|Lambda]], G),
2653   append(Gamma, [B|[A|Lambda]], G1),
2654   append(Gamma, [B|[A|[A|Lambda]]], G2),
2655   !,
2656   assertz(proved(Z = [con(Z1,A), G => T])),
2657   append(Gamma, [A|[A|[B|Lambda]]], G3),
2658   gensym(f,F),
2659   assertz(proved(Z1 = [inc(F,A,B), G3 => T])),
2660   append(Gamma, [A|[B|[A|Lambda]]], G4),
2661   assertz(proved(F = [inc(Z2,A,B), G4 => T])),
2662   retract(proved(Z = [inc(Z1,A,B), G => T])),
2663   retract(proved(Z1 = [con(Z2,A), G1 => T])),
2664   !.

```

```

2665 /* (R3,R5) - (E.7) */
2666 norm1(proved(Z = [con(Z1,A), G => T])) :-
2667   proved(Z1 = [th(Z2,B), G1 => T]),
2668   proved(Z2 = [-, G1 => T]),
2669   append(Gamma, [A|[A|Delta]], G1),

```

```

2670      append(Gamma, [A|Delta], G),
2671      append(Phi, [B|Psi], T),
2672      append(Phi, Psi, T1),
2673      |
2674      assertz(proved(Z = [th(Z1,B), G => T])),
2675      assertz(proved(Z1 = [con(Z2,A), G => T1])),
2676      retract(proved(Z = [con(Z1,A), G => T])),
2677      retract(proved(Z1 = [th(Z2,B), G1 => T1])),
2678      |

```

```

2679 /* (R3,R7) - (E.8) */
2680 norm1(proved(Z = [inc(Z1,B,A), G => T])) :-
2681     proved(Z1 = [con(Z2,C), G => T1]),
2682     proved(Z2 = [_, G1 => T1]),
2683     append(Gamma, [C|Delta], G),
2684     append(Gamma, [C|[C|Delta]], G1),
2685     append(Phi, [B|[A|Psi]], T),
2686     append(Phi, [A|[B|Psi]], T1),
2687     |
2688     assertz(proved(Z = [con(Z1,C), G => T])),
2689     assertz(proved(Z1 = [inc(Z2,B,A), G1 => T1])),
2690     retract(proved(Z = [inc(Z1,B,A), G => T])),
2691     retract(proved(Z1 = [con(Z2,C), G => T1])),
2692     |

```

```

2693 /* (R3,R10) - (E.9) */
2694 norm1(proved(Z = [con(Z1,C), G => T])) :-
2695     proved(Z1 = [ais(Z2,Z3,A & B), G1 => T]),
2696     proved(Z2 = [_, G1 => T1]),
2697     proved(Z3 = [_, G1 => T2]),
2698     |
2699     gensym(f,F1),
2700     assertz(proved(Z = [ais(Z1,F1,A & B), G => T])),
2701     assertz(proved(Z1 = [con(Z2,C), G => T1])),
2702     assertz(proved(F1 = [con(Z3,C), G => T2])),
2703     retract(proved(Z = [con(Z1,C), G => T])),
2704     retract(proved(Z1 = [ais(Z2,Z3,A & B), G1 => T])),
2705     |

```

```

2706 /* (R3,R11) - (E.10.1,2) */
2707 norm1(proved(Z = [con(Z1,C), G => T])) :-
2708     proved(Z1 = [aia(Z2,A & B), G1 => T]),
2709     proved(Z2 = [_, G2 => T]),
2710     ((append(S1, [A & B|Lambda], G1),
2711      append(S1, [A|[B|Lambda]], G2),
2712      append(Gamma, [C|[C|Delta]], S1),
2713      append3(Gamma, [C|Delta], [A & B|Lambda], G),
2714      append3(Gamma, [C|Delta], [A|[B|Lambda]], G3))
2715     ;(append(Gamma, [A & B|E1], G1),
2716      append(Gamma, [A|[B|E1]], G2),
2717      append(Delta, [C|[C|Lambda]], E1),
2718      append3(Gamma, [A & B|Delta], [C|Lambda], G),
2719      append3(Gamma, [A|[B|Delta]], [C|Lambda], G3))),
2720     |
2721     assertz(proved(Z = [aia(Z1,A & B), G => T])),
2722     assertz(proved(Z1 = [con(Z2,C), G3 => T])),
2723     retract(proved(Z = [con(Z1,C), G => T])),
2724     retract(proved(Z1 = [aia(Z2,A & B), G1 => T])),
2725     |

```

```

2726 /* (R3,R11) - (E.10.3,4) */
2727 norm1(proved(Z = [con(Z1,A & A), G => T])) :-
2728     proved(Z1 = [aia(Z2,A & A), G1 => T]),
2729     proved(Z2 = [aia(Z3,A & A), G2 => T]),
2730     proved(Z3 = [_, G3 => T]),
2731     append(Gamma, [A & A|Delta], G),
2732     append(Gamma, [A & A|[A & A|Delta]], G1),
2733     (append(Gamma, [A|[A|[A & A|Delta]]], G2),
2734     ;append(Gamma, [A & A|[A|[A|Delta]]], G2)),
2735     append(Gamma, [A|[A|[A|[A|Delta]]]], G3),
2736     |
2737     assertz(proved(Z = [aia(Z1,A & A), G => T])),
2738     append(Gamma, [A|[A|Delta]], G4),
2739     assertz(proved(Z1 = [con(Z2,A), G4 => T])),
2740     append(Gamma, [A|[A|[A|Delta]]], G5),
2741     assertz(proved(Z2 = [con(Z3,A), G5 => T])),
2742     retract(proved(Z = [con(Z1,A & A), G => T])),
2743     retract(proved(Z1 = [aia(Z2,A & A), G1 => T])),
2744     retract(proved(Z2 = [aia(Z3,A & A), G2 => T])),
2745     |

```

```

2746 /* (R3,R11) - (E.10.5,6) */
2747 norm1(proved(Z = [con(Z1,A & B), G => T])) :-
2748   proved(Z1 = [oia(Z2,A & B), G1 => T]),
2749   proved(Z2 = [oia(Z3,A & B), G2 => T]),
2750   proved(Z3 = [-, G3 => T]),
2751   append(Gamma, [A & B|Delta], G),
2752   append(Gamma, [A & B|[A & B|Delta]], G1),
2753   (append(Gamma, [A|[B|[A & B|Delta]]], G2),
2754   ;append(Gamma, [A & B|[A|[B|Delta]]], G2)),
2755   append(Gamma, [A|[B|[A|[B|Delta]]]], G3),
2756   !,
2757   assertz(proved(Z = [oia(Z1,A & B), G => T])),
2758   append(Gamma, [A|[B|Delta]], G4),
2759   assertz(proved(Z1 = [con(Z2,A), G4 => T])),
2760   append(Gamma, [A|[A|[B|Delta]]], G5),
2761   gensym(f,F),
2762   assertz(proved(Z2 = [con(F,B), G5 => T])),
2763   append(Gamma, [A|[A|[B|[B|Delta]]]], G6),
2764   assertz(proved(F = [inc(Z3,A,B), G6 => T])),
2765   retract(proved(Z = [con(Z1,A & B), G => T])),
2766   retract(proved(Z1 = [oia(Z2,A & B), G1 => T])),
2767   retract(proved(Z2 = [oia(Z3,A & B), G2 => T])),
2768   !.

```

```

2769 /* (R3,R12) - (E.11.1,2) */
2770 norm1(proved(Z = [con(Z1,C), G => T])) :-
2771   proved(Z1 = [oia(Z2,Z3,A # B), G1 => T]),
2772   proved(Z2 = [-, G2 => T]),
2773   proved(Z3 = [-, G3 => T]),
2774   ((append(S, [C|[C|Lambda]], G1),
2775   append(S, [C|Lambda], G),
2776   append(S1, [C|[C|Lambda]], G2),
2777   append(S2, [C|[C|Lambda]], G3),
2778   append(Gamma, [A # B|Delta], S),
2779   append(Gamma, [A|Delta], S1),
2780   append(Gamma, [B|Delta], S2),
2781   append(S1, [C|Lambda], G4),
2782   append(S2, [C|Lambda], G5))
2783   ;(append(Gamma, [C|[C|E]], G1),
2784   append(Gamma, [C|E], G),
2785   append(Gamma, [C|[C|E1]], G2),
2786   append(Gamma, [C|[C|E2]], G3),
2787   append(Delta, [A # B|Lambda], E),
2788   append(Delta, [A|Lambda], E1),
2789   append(Delta, [B|Lambda], E2),
2790   append(Gamma, [C|E1], G4),
2791   append(Gamma, [C|E2], G5))),
2792   !,
2793   gensym(f,F),
2794   assertz(proved(Z = [oia(Z1,F,A # B), G => T])),
2795   assertz(proved(Z1 = [con(Z2,C), G4 => T])),
2796   assertz(proved(F = [con(Z3,C), G5 => T])),
2797   retract(proved(Z = [con(Z1,C), G => T])),
2798   retract(proved(Z1 = [oia(Z2,Z3,A # B), G1 => T])),
2799   !.

```

```

2800 /* (R3,R12) - (E.11.3,4) */
2801 norm1(proved(Z = [con(Z1,A # B), G => T])) :-
2802   proved(Z1 = [oia(Z2,Z3,A # B), G1 => T]),
2803   proved(Z2 = [oia(Z4,Z5,A # B), G2 => T]),
2804   proved(Z4 = [th(Z6,A), G3 => T]),
2805   proved(Z6 = [-, G4 => T]),
2806   proved(Z5 = [th(Z7,A), G5 => T]),
2807   proved(Z7 = [-, G6 => T]),
2808   proved(Z3 = [oia(Z8,Z9,A # B), G7 => T]),
2809   proved(Z8 = [th(Z10,B), G8 => T]),
2810   proved(Z10 = [-, G4 => T]),
2811   proved(Z9 = [th(Z11,B), G9 => T]),
2812   proved(Z11 = [-, G6 => T]),
2813   append(Gamma, [A # B|[A # B|Delta]], G1),
2814   append(Gamma, [A # B|Delta], G),
2815   append(Gamma, [A|[A|Delta]], G3),
2816   append(Gamma, [B|[B|Delta]], G9),
2817   append(Gamma, [A|Delta], G4),
2818   append(Gamma, [B|Delta], G6),
2819   ((append(Gamma, [A # B|[A|Delta]], G2),
2820   append(Gamma, [A # B|[B|Delta]], G7),
2821   append(Gamma, [B|[A|Delta]], G5),
2822   append(Gamma, [A|[B|Delta]], G8))

```

```

2823 ;(append(Gamma, [A|[A#B|Delta]], G2),
2824 append(Gamma, [B|[A#B|Delta]], G7),
2825 append(Gamma, [A|[B|Delta]], G5),
2826 append(Gamma, [B|[A|Delta]], G8)),
2827 |,
2828 assertz(proved(Z = [ola(Z6,Z11,A # B), G => T])),
2829 retract(proved(Z = [con(Z1,A # B), G => T])),
2830 |.

```

```

2831 /* (R3,R13) - (E.12) */
2832 norm1(proved(Z = [con(Z1,C), G => T])) :-
2833 proved(Z1 = [ois(Z2,A # B), G1 => T]),
2834 proved(Z2 = [_, G1 => T1]),
2835 |,
2836 assertz(proved(Z = [ois(Z1,A # B), G => T])),
2837 assertz(proved(Z1 = [con(Z2,C), G => T1])),
2838 retract(proved(Z = [con(Z1,C), G => T])),
2839 retract(proved(Z1 = [ois(Z2,A # B), G1 => T])),
2840 |.

```

```

2841 /* (R3,R14) - (E.13.1) */
2842 norm1(proved(Z = [con(Z1,C), G => T])) :-
2843 proved(Z1 = [hia(Z2,Z3,A -> B), G1 => T]),
2844 proved(Z2 = [_, G2 => [A]]),
2845 proved(Z3 = [_, G3 => T]),
2846 append(Lambda, [B|Sigma], G3),
2847 append3(Lambda, G2, [A->B|Sigma], G1),
2848 append(Gamma, [C|[C|Delta]], G2),
2849 append4(Lambda, Gamma, [C|Delta], [A->B|Sigma], G),
2850 |,
2851 assertz(proved(Z = [hia(Z1,Z3,A -> B), G => T])),
2852 append(Gamma, [C|Delta], G4),
2853 assertz(proved(Z1 = [con(Z2,C), G4 => [A]])),
2854 retract(proved(Z1 = [hia(Z2,Z3,A -> B), G1 => T])),
2855 retract(proved(Z = [con(Z1,C), G => T])),
2856 |.

```

```

2857 /* (R3,R14) - (E.13.2,3) */
2858 norm1(proved(Z = [con(Z1,C), G => T])) :-
2859 proved(Z1 = [hia(Z2,Z3,A -> B), G1 => T]),
2860 proved(Z2 = [_, Gamma => [A]]),
2861 proved(Z3 = [_, G2 => T]),
2862 ((append(S, [B|Sigma], G2),
2863 append3(S, Gamma, [A->B|Sigma], G1),
2864 append(Delta, [C|[C|Lambda]], S),
2865 append3(S1, Gamma, [A -> B|Sigma], G),
2866 append(Delta, [C|Lambda], S1),
2867 append(S1, [B|Sigma], G3))
2868 ;(append(Delta, [B|E], G2),
2869 append3(Delta, Gamma, [A->B|E], G1),
2870 append(Lambda, [C|[C|Sigma]], E),
2871 append3(Delta, Gamma, [A->B|E1], G),
2872 append(Lambda, [C|Sigma], E1),
2873 append(Delta, [B|E1], G3))),
2874 |,
2875 assertz(proved(Z = [hia(Z2,Z1,A->B), G => T])),
2876 assertz(proved(Z1 = [con(Z3,C), G3 => T])),
2877 retract(proved(Z1 = [hia(Z2,Z3,A->B), G1 => T])),
2878 retract(proved(Z = [con(Z1,C), G => T])),
2879 |.

```

```

2880 /* (R3,R15) - (E.14.1,2) */
2881 norm1(proved(Z = [con(Z1,C), G => [A -> B]])) :-
2882 proved(Z1 = [his(Z2,A -> B), G1 => [A -> B]]),
2883 proved(Z2 = [_, G2 => [B]]),
2884 ((append(Gamma, [C|[C|E1]], G1),
2885 append(Gamma, [C|E1], G),
2886 append(Gamma, [C|[C|E]], G2),
2887 append(Delta, [A|Lambda], E),
2888 append(Delta, Lambda, E1),
2889 append(Gamma, [C|E], G3))
2890 ;(append(S1, [C|[C|Lambda]], G1),
2891 append(S1, [C|Lambda], G),
2892 append(S, [C|[C|Lambda]], G2),
2893 append(Gamma, [A|Delta], S),
2894 append(Gamma, Delta, S1),
2895 append(S, [C|Lambda], G3))),
2896 |.

```

```

2897      assertz(proved(Z = [his(Z1,A->B), G => [A->B]])),
2898      assertz(proved(Z1 = [con(Z2,C), G3 => [B]])),
2899      retract(proved(Z1 = [his(Z2,A->B), G1 => [A->B]])),
2900      retract(proved(Z = [con(Z1,C), G => [A->B]])),
2901      !.

```

```

2902 /* (R3,R15) - (E.14.3) */
2903 norm1(proved(Z = [con(Z1,C), G => [A->B]])) :-
2904     proved(Z1 = [his(Z2,A->B), G1 => [A->B]]),
2905     proved(Z2 = [_, G2 => [B]]),
2906     append(Gamma, [C|[A|[C|Delta]]], G2),
2907     append(Gamma, [C|[C|Delta]], G1),
2908     append(Gamma, [C|Delta], G),
2909     !.
2910     assertz(proved(Z = [his(Z1,A->B), G => [A->B]])),
2911     append(Gamma, [C|[A|Delta]], G3),
2912     assertz(proved(Z1 = [con(Z3,C), G3 => [B]])),
2913     append(Gamma, [C|[C|[A|Delta]]], G4),
2914     assertz(proved(Z3 = [inc(Z2,C,A), G4 => [B]])),
2915     retract(proved(Z1 = [his(Z2,A->B), G1 => [A->B]])),
2916     retract(proved(Z = [con(Z1,C), G => [A->B]])),
2917     !.

```

```

2918 /* (R4,R5) - (E.15) */
2919 norm1(proved(Z = [inc(Z1,B,A), G => T])) :-
2920     proved(Z1 = [th(Z2,C), G1 => T]),
2921     proved(Z2 = [_, G1 => T1]),
2922     append(Gamma, [B|[A|Delta]], G),
2923     append(Gamma, [A|[B|Delta]], G1),
2924     append(Phi, [C|Psi], T),
2925     append(Phi, Psi, T1),
2926     !.
2927     assertz(proved(Z = [th(Z1,C), G => T])),
2928     assertz(proved(Z1 = [inc(Z2,B,A), G => T1])),
2929     retract(proved(Z = [inc(Z1,B,A), G => T])),
2930     retract(proved(Z1 = [th(Z2,C), G1 => T])),
2931     !.

```

```

2932 /* (R4,R6) - (E.16) */
2933 norm1(proved(Z = [inc(Z1,B,A), G => T])) :-
2934     proved(Z1 = [con(Z2,C), G1 => T]),
2935     proved(Z2 = [_, G1 => T1]),
2936     append(Gamma, [B|[A|Delta]], G),
2937     append(Gamma, [A|[B|Delta]], G1),
2938     append(Phi, [C|[C|Psi]], T1),
2939     append(Phi, [C|Psi], T),
2940     !.
2941     assertz(proved(Z = [con(Z1,C), G => T])),
2942     assertz(proved(Z1 = [inc(Z2,B,A), G => T1])),
2943     retract(proved(Z = [inc(Z1,B,A), G => T])),
2944     retract(proved(Z1 = [con(Z2,C), G1 => T])),
2945     !.

```

```

2946 /* (R4,R10) - (E.17) */
2947 norm1(proved(Z = [inc(Z1,D,C), G => T])) :-
2948     proved(Z1 = [ais(Z2,Z3,A & B), G1 => T]),
2949     proved(Z2 = [_, G1 => T1]),
2950     proved(Z3 = [_, G1 => T2]),
2951     !.
2952     gensym(f, F1),
2953     assertz(proved(Z = [ais(Z1,F1,A & B), G => T])),
2954     assertz(proved(Z1 = [inc(Z2,D,C), G => T1])),
2955     assertz(proved(F1 = [inc(Z3,D,C), G => T2])),
2956     retract(proved(Z = [inc(Z1,D,C), G => T])),
2957     retract(proved(Z1 = [ais(Z2,Z3,A & B), G1 => T])),
2958     !.

```

```

2959 /* (R4,R11) - (E.18.1,2) */
2960 norm1(proved(Z = [inc(Z1,B,A), G => T])) :-
2961     proved(Z1 = [ais(Z2,C & D), G1 => T]),
2962     proved(Z2 = [_, G2 => T]),
2963     ((append(S, [C & D|Lambda], G1),
2964     append(S, [C|[D|Lambda]], G2),
2965     append(Gamma, [A|[B|Delta]], S),
2966     append3(Gamma, [B|[A|Delta]], [C & D|Lambda], G),
2967     append3(Gamma, [B|[A|Delta]], [C|[D|Lambda]], G3))

```

```

2968      : (append(Gamma, [C & D|E], G1),
2969      append(Gamma, [C|[D|E]], G2),
2970      append(Delta, [A|[B|Lambda]], E),
2971      append3(Gamma, [C & D|Delta], [B|[A|Lambda]], G),
2972      append3(Gamma, [C|[D|Delta]], [B|[A|Lambda]], G3))),
2973      ],
2974      assertz(proved(Z = [cia(Z1,C & D), G => T])),
2975      assertz(proved(Z1 = [inc(Z2,B,A), G3 => T])),
2976      retract(proved(Z = [inc(Z1,B,A), G => T])),
2977      retract(proved(Z1 = [cia(Z2,C & D), G1 => T])),
2978      ].

```

```

2979      /* (R4,R11) - (E.18.3) */
2980      norm1(proved(Z = [inc(Z1,C,A & B), G => T])) :-
2981      proved(Z1 = [cia(Z2,A & B), G1 => T]),
2982      proved(Z2 = [_, G2 => T]),
2983      append(Gamma, [C|[A & B|Delta]], G),
2984      append(Gamma, [A & B|[C|Delta]], G1),
2985      append(Gamma, [A|[B|[C|Delta]]], G2),
2986      ],
2987      assertz(proved(Z = [cia(Z1,A & B), G => T])),
2988      gensym(f,Z3),
2989      append(Gamma, [C|[A|[B|Delta]]], G4),
2990      assertz(proved(Z1 = [inc(Z3,C,A), G4 => T])),
2991      append(Gamma, [A|[C|[B|Delta]]], G3),
2992      assertz(proved(Z3 = [inc(Z2,C,B), G3 => T])),
2993      retract(proved(Z = [inc(Z1,C,A & B), G => T])),
2994      retract(proved(Z1 = [cia(Z2,A & B), G1 => T])),
2995      ].

```

```

2996      /* (R4,R11) - (E.18.4) */
2997      norm1(proved(Z = [inc(Z1,B & C,A), G => T])) :-
2998      proved(Z1 = [cia(Z2,B & C), G1 => T]),
2999      proved(Z2 = [_, G2 => T]),
3000      append(Gamma, [B & C|[A|Delta]], G),
3001      append(Gamma, [A|[B & C|Delta]], G1),
3002      append(Gamma, [A|[B|[C|Delta]]], G2),
3003      ],
3004      assertz(proved(Z = [cia(Z1,B & C), G => T])),
3005      gensym(f1,Z3),
3006      append(Gamma, [B|[C|[A|Delta]]], G3),
3007      assertz(proved(Z1 = [inc(Z3,C,A), G3 => T])),
3008      append(Gamma, [B|[A|[C|Delta]]], G4),
3009      assertz(proved(Z3 = [inc(Z2,B,A), G4 => T])),
3010      retract(proved(Z = [inc(Z1,B & C,A), G => T])),
3011      retract(proved(Z1 = [cia(Z2,B & C), G1 => T])),
3012      ].

```

```

3013      /* (R4,R12) - (E.19.1,2) */
3014      norm1(proved(Z = [inc(Z1,D,C), G => T])) :-
3015      proved(Z1 = [cia(Z2,Z3,A # B), G1 => T]),
3016      proved(Z2 = [_, G2 => T]),
3017      proved(Z3 = [_, G3 => T]),
3018      ((append(Gamma, [D|[C|E]], G),
3019      append(Gamma, [C|[D|E]], G1),
3020      append(Gamma, [C|[D|E1]], G2),
3021      append(Gamma, [C|[D|E2]], G3),
3022      append(Delta, [A # B|Lambda], E),
3023      append(Delta, [A|Lambda], E1),
3024      append(Delta, [B|Lambda], E2),
3025      append(Gamma, [D|[C|E1]], G4),
3026      append(Gamma, [D|[C|E2]], G5))
3027      : (append(S, [D|[C|Lambda]], G),
3028      append(S, [C|[D|Lambda]], G1),
3029      append(S1, [C|[D|Lambda]], G2),
3030      append(S2, [C|[D|Lambda]], G3),
3031      append(Gamma, [A # B|Delta], S),
3032      append(Gamma, [A|Delta], S1),
3033      append(Gamma, [B|Delta], S2),
3034      append(S1, [D|[C|Lambda]], G4),
3035      append(S2, [D|[C|Lambda]], G5))),
3036      ],
3037      gensym(f,F),
3038      assertz(proved(Z = [cia(Z1,F,A # B), G => T])),
3039      assertz(proved(Z1 = [inc(Z2,D,C), G4 => T])),
3040      assertz(proved(F = [inc(Z3,D,C), G5 => T])),
3041      retract(proved(Z = [inc(Z1,D,C), G => T])),
3042      retract(proved(Z1 = [cia(Z2,Z3,A # B), G1 => T])),
3043      ].

```



```

3044 /* (R4,R12) - (E.19.3) */
3045 norm1(proved(Z = [inc(Z1,C,A # B), G => T])) :-
3046   proved(Z1 = [ola(Z2,Z3,A # B), G1 => T]),
3047   proved(Z2 = [-, G2 => T]),
3048   proved(Z3 = [-, G3 => T]),
3049   append(Gamma, [C|[A # B|Delta]], G),
3050   append(Gamma, [A # B|[C|Delta]], G1),
3051   append(Gamma, [A|[C|Delta]], G2),
3052   append(Gamma, [B|[C|Delta]], G3),
3053   !,
3054   gensym(f,F),
3055   assertz(proved(Z = [ola(Z1,F,A # B), G => T])),
3056   append(Gamma, [C|[A|Delta]], G4),
3057   assertz(proved(Z1 = [inc(Z2,C,A), G4 => T])),
3058   append(Gamma, [C|[B|Delta]], G5),
3059   assertz(proved(F = [inc(Z3,C,B), G5 => T])),
3060   retract(proved(Z = [inc(Z1,C,A # B), G => T])),
3061   retract(proved(Z1 = [ola(Z2,Z3,A # B), G1 => T])),
3062   !.

```

```

3063 /* (R4,R12) - (E.19.4) */
3064 norm1(proved(Z = [inc(Z1,A # B,C), G => T])) :-
3065   proved(Z1 = [ola(Z2,Z3,A # B), G1 => T]),
3066   proved(Z2 = [-, G2 => T]),
3067   proved(Z3 = [-, G3 => T]),
3068   append(Gamma, [A # B|[C|Delta]], G),
3069   append(Gamma, [C|[A # B|Delta]], G1),
3070   append(Gamma, [C|[A|Delta]], G2),
3071   append(Gamma, [C|[B|Delta]], G3),
3072   !,
3073   gensym(f,F),
3074   assertz(proved(Z = [ola(Z1,F,A # B), G => T])),
3075   append(Gamma, [A|[C|Delta]], G4),
3076   assertz(proved(Z1 = [inc(Z2,A,C), G4 => T])),
3077   append(Gamma, [B|[C|Delta]], G5),
3078   assertz(proved(F = [inc(Z3,B,C), G5 => T])),
3079   retract(proved(Z = [inc(Z1,A # B,C), G => T])),
3080   retract(proved(Z1 = [ola(Z2,Z3,A # B), G1 => T])),
3081   !.

```

```

3082 /* (R4,R13) - (E.20) */
3083 norm1(proved(Z = [inc(Z1,D,C), G => T])) :-
3084   proved(Z1 = [ois(Z2,A # B), G1 => T]),
3085   proved(Z2 = [-, G1 => T]),
3086   append(Gamma, [D|[C|Delta]], G),
3087   append(Gamma, [C|[D|Delta]], G1),
3088   append(Phi, [A # B|Psi], T),
3089   append(Phi, [A|[B|Psi]], T1),
3090   !,
3091   assertz(proved(Z = [ois(Z1,A # B), G => T])),
3092   assertz(proved(Z1 = [inc(Z2,D,C), G => T])),
3093   retract(proved(Z = [inc(Z1,D,C), G => T])),
3094   retract(proved(Z1 = [ois(Z2,A # B), G1 => T])),
3095   !.

```

```

3096 /* (R4,R14) - (E.21.1) */
3097 norm1(proved(Z = [inc(Z1,D,C), G => T])) :-
3098   proved(Z1 = [hia(Z2,Z3,A -> B), G1 => T]),
3099   proved(Z2 = [-, G2 => [A]]),
3100   proved(Z3 = [-, G3 => T]),
3101   append(Lambda, [B|Sigma], G3),
3102   append3(Lambda,G2,[A->B|Sigma],G1),
3103   append(Gamma, [C|[D|Delta]], G2),
3104   append4(Lambda,Gamma,[D|[C|Delta]], [A->B|Sigma],G),
3105   !,
3106   assertz(proved(Z = [hia(Z1,Z3,A -> B), G => T])),
3107   append(Gamma, [D|[C|Delta]], G4),
3108   assertz(proved(Z1 = [inc(Z2,D,C), G4 => [A]])),
3109   retract(proved(Z1 = [hia(Z2,Z3,A -> B), G1 => T])),
3110   retract(proved(Z = [inc(Z1,D,C), G => T])),
3111   !.

```

```

3112 /* (R4,R14) - (E.21.2,3) */
3113 norm1(proved(Z = [inc(Z1,D,C), G => T])) :-
3114   proved(Z1 = [hia(Z2,Z3,A -> B), G1 => T]),
3115   proved(Z2 = [-, Gamma => [A]]),
3116   proved(Z3 = [-, G2 => T]),
3117   ((append(S, [B|Sigma], G2),

```

```

3118     append3(S, Gamma, [A->B|Sigma], G1),
3119     append(Delta, [C|[D|Lambda]], S),
3120     append3(S1, Gamma, [A -> B|Sigma], G),
3121     append(Delta, [D|[C|Lambda]], S1),
3122     append(S1, [B|Sigma], G3))
3123   ;(append(Delta, [B|E], G2),
3124     append3(Delta, Gamma, [A->B|E], G1),
3125     append(Lambda, [C|[D|Sigma]], E),
3126     append3(Delta, Gamma, [A->B|E1], G),
3127     append(Lambda, [D|[C|Sigma]], E1),
3128     append(Delta, [B|E1], G3)))
3129   !,
3130   assertz(proved(Z = [hia(Z2, Z1, A->B), G => T])),
3131   assertz(proved(Z1 = [inc(Z3, D, C), G3 => T])),
3132   retract(proved(Z1 = [hia(Z2, Z3, A->B), G1 => T])),
3133   retract(proved(Z = [inc(Z1, D, C), G => T])),
3134   !.

```

```

3135 /* (R4,R15) - (E.22.1,2) */
3136 norm1(proved(Z = [inc(Z1, D, C), G => [A -> B]])) :-
3137   proved(Z1 = [his(Z2, A -> B), G1 => [A -> B]]),
3138   proved(Z2 = [_, G2 => [B]]),
3139   ((append(Gamma, [C|[D|E1]], G1),
3140     append(Gamma, [D|[C|E1]], G),
3141     append(Gamma, [C|[D|E]], G2),
3142     append(Delta, [A|Lambda], E),
3143     append(Delta, [Lambda], E1),
3144     append(Gamma, [D|[C|E]], G3))
3145   ;(append(S1, [C|[D|Lambda]], G1),
3146     append(S1, [D|[C|Lambda]], G),
3147     append(S, [C|[D|Lambda]], G2),
3148     append(Gamma, [A|Delta], S),
3149     append(Gamma, [Delta], S1),
3150     append(S, [D|[C|Lambda]], G3)))
3151   !,
3152   assertz(proved(Z = [his(Z1, A->B), G => [A->B]])),
3153   assertz(proved(Z1 = [inc(Z2, D, C), G3 => [B]])),
3154   retract(proved(Z1 = [his(Z2, A->B), G1 => [A->B]])),
3155   retract(proved(Z = [inc(Z1, D, C), G => [A -> B]])),
3156   !.

```

```

3157 /* (R5,R5) - (E.23) */
3158 norm1(proved(Z = [th(Z1, B), G => T])) :-
3159   proved(Z1 = [th(Z2, A), G => T1]),
3160   proved(Z2 = [_, G => T2]),
3161   append(S, [B|Theta], T),
3162   append(S, [Theta], T1),
3163   append(S1, [Theta], T2),
3164   append(Phi, [A|Psi], S),
3165   append(Phi, [Psi], S1),
3166   !,
3167   assertz(proved(Z = [th(Z1, A), G => T])),
3168   append3(Phi, [Psi], [B|Theta], T3),
3169   assertz(proved(Z1 = [th(Z2, B), G => T3])),
3170   retract(proved(Z = [th(Z1, B), G => T])),
3171   retract(proved(Z1 = [th(Z2, A), G => T1])),
3172   !.

```

```

3173 /* (R5,R6) - (E.24.1) */
3174 norm1(proved(Z = [con(Z1, A), G => T])) :-
3175   proved(Z1 = [th(Z2, A), G => T1]),
3176   proved(Z2 = [K, G => T]),
3177   append(Phi, [A|[A|Psi]], T1),
3178   append(Phi, [A|Psi], T),
3179   !,
3180   assertz(proved(Z = [K, G => T])),
3181   retract(proved(Z = [con(Z1, A), G => T])),
3182   retract(proved(Z1 = [th(Z2, A), G => T1])),
3183   !.

```

```

3184 /* (R5,R6) - (E.24.2,3) */
3185 norm1(proved(Z = [con(Z1, A), G => T])) :-
3186   proved(Z1 = [th(Z2, B), G => T1]),
3187   proved(Z2 = [_, G => T2]),
3188   ((append(Phi, [A|E], T),
3189     append(Phi, [A|[A|E]], T1),
3190     append(Phi, [A|[A|E1]], T2),
3191     append(Psi, [B|Theta], E),
3192     append(Psi, [Theta], E1),

```

```

3193      append(Phi, [A|E1], T3))
3194      ;(append(S, [A|Theta], T),
3195      append(S, [A|[A|Theta]], T1),
3196      append(S1, [A|[A|Theta]], T2),
3197      append(Phi, [B|Psi], S),
3198      append(Phi, Psi, S1),
3199      append(S1, [A|Theta], T3))),
3200      |,
3201      assertz(proved(Z = [th(Z1,B), G => T])),
3202      assertz(proved(Z1 = [con(Z2,A), G => T3])),
3203      retract(proved(Z = [con(Z1,A), G => T1])),
3204      retract(proved(Z1 = [th(Z2,B), G => T1])),
3205      |.

3206      /* (R5,R7) - (E.25.1,2) */
3207      norm1(proved(Z = [inc(Z1,B,A), G => T])) :-
3208      proved(Z1 = [th(Z2,C), G => T1]),
3209      proved(Z2 = [_, G => T2]),
3210      ((append(S, [B|[A|Theta]], T),
3211      append(S, [A|[B|Theta]], T1),
3212      append(S1, [A|[B|Theta]], T2),
3213      append(Phi, [C|Psi], S),
3214      append(Phi, Psi, S1),
3215      append(S1, [B|[A|Theta]], T3))
3216      ;(append(Phi, [B|[A|E]], T),
3217      append(Phi, [A|[B|E]], T1),
3218      append(Phi, [A|[B|E1]], T2),
3219      append(Psi, [C|Theta], E),
3220      append(Psi, Theta, E1),
3221      append(Phi, [B|[A|E1]], T3))),
3222      |,
3223      assertz(proved(Z = [th(Z1,C), G => T])),
3224      assertz(proved(Z1 = [inc(Z2,B,A), G => T3])),
3225      retract(proved(Z = [inc(Z1,B,A), G => T])),
3226      retract(proved(Z1 = [th(Z2,C), G => T1])),
3227      |.

3228      /* (R5,R7) - (E.25.3) */
3229      norm1(proved(Z = [inc(Z1,B,A), G => T])) :-
3230      proved(Z1 = [th(Z2,B), G => T1]),
3231      proved(Z2 = [_, G => T2]),
3232      append(Phi, [B|[A|Psi]], T),
3233      append(Phi, [A|[B|Psi]], T1),
3234      append(Phi, [A|Psi], T2),
3235      |,
3236      assertz(proved(Z = [th(Z2,B), G => T])),
3237      retract(proved(Z = [inc(Z1,B,A), G => T])),
3238      retract(proved(Z1 = [th(Z2,B), G => T1])),
3239      |.

3240      /* (R5,R7) - (E.25.4) */
3241      norm1(proved(Z = [inc(Z1,A,B), G => T])) :-
3242      proved(Z1 = [th(Z2,B), G => T1]),
3243      proved(Z2 = [_, G => T2]),
3244      append(Phi, [A|[B|Psi]], T),
3245      append(Phi, [B|[A|Psi]], T1),
3246      append(Phi, [A|Psi], T2),
3247      |,
3248      assertz(proved(Z = [th(Z2,B), G => T])),
3249      retract(proved(Z = [inc(Z1,A,B), G => T])),
3250      retract(proved(Z1 = [th(Z2,B), G => T1])),
3251      |.

3252      /* (R6,R7) - (E.26.1,2) */
3253      norm1(proved(Z = [inc(Z1,B,A), G => T])) :-
3254      proved(Z1 = [con(Z2,C), G => T1]),
3255      proved(Z2 = [_, G => T2]),
3256      ((append(S, [B|[A|Theta]], T),
3257      append(S, [A|[B|Theta]], T1),
3258      append(S1, [A|[B|Theta]], T2),
3259      append(Phi, [C|[C|Psi]], S1),
3260      append(Phi, [C|Psi], S),
3261      append(S1, [B|[A|Theta]], T3))
3262      ;(append(Phi, [B|[A|E]], T),
3263      append(Phi, [A|[B|E]], T1),
3264      append(Phi, [A|[B|E1]], T2),
3265      append(Psi, [C|[C|Theta]], E1),
3266      append(Psi, [C|Theta], E),
3267      append(Phi, [B|[A|E1]], T3))),
3268      |,

```

```

3269      assertz(proved(Z = [con(Z1,C), G => T])).
3270      assertz(proved(Z1 = [inc(Z2,B,A), G => T3])).
3271      retract(proved(Z = [inc(Z1,B,A), G => T])).
3272      retract(proved(Z1 = [con(Z2,C), G => T1])).
3273      !.

```

```

3274 /* (R6,R7) - (E.26.3) */
3275 norm1(proved(Z = [inc(Z1,B,A), G => T])) :-
3276     proved(Z1 = [con(Z2,A), G => T1]),
3277     proved(Z2 = [_, G => T2]),
3278     append(Phi, [B|[A|Psi]], T),
3279     append(Phi, [A|[B|Psi]], T1),
3280     append(Phi, [A|[A|[B|Psi]]], T2),
3281     !,
3282     assertz(proved(Z = [con(Z1,A), G => T])),
3283     append(Phi, [B|[A|[A|Psi]]], T3),
3284     gensym(f,F),
3285     assertz(proved(Z1 = [inc(F,B,A), G => T3])),
3286     append(Phi, [A|[B|[A|Psi]]], T4),
3287     assertz(proved(F = [inc(Z2,B,A), G => T4])),
3288     retract(proved(Z = [inc(Z1,B,A), G => T])),
3289     retract(proved(Z1 = [con(Z2,A), G => T1])).
3290     !.

```

```

3291 /* (R6,R7) - (E.26.4) */
3292 norm1(proved(Z = [inc(Z1,A,B), G => T])) :-
3293     proved(Z1 = [con(Z2,A), G => T1]),
3294     proved(Z2 = [_, G => T2]),
3295     append(Phi, [A|[B|Psi]], T),
3296     append(Phi, [B|[A|Psi]], T1),
3297     append(Phi, [B|[A|[A|Psi]]], T2),
3298     !,
3299     assertz(proved(Z = [con(Z1,A), G => T])),
3300     append(Phi, [A|[A|[B|Psi]]], T3),
3301     gensym(f,F),
3302     assertz(proved(Z1 = [inc(F,A,B), G => T3])),
3303     append(Phi, [A|[B|[A|Psi]]], T4),
3304     assertz(proved(F = [inc(Z2,A,B), G => T4])),
3305     retract(proved(Z = [inc(Z1,A,B), G => T])),
3306     retract(proved(Z1 = [con(Z2,A), G => T1])).
3307     !.

```

```

3308 /* (R6,R10) - (E.27.1,2) */
3309 norm1(proved(Z = [con(Z1,C), G => T])) :-
3310     proved(Z1 = [dis(Z2,Z3,A & B), G => T1]),
3311     proved(Z2 = [_, G => T2]),
3312     proved(Z3 = [_, G => T3]),
3313     ((append(S, [C|[C|Theta]], T1),
3314     append(S, [C|Theta], T),
3315     append(S1, [C|[C|Theta]], T2),
3316     append(S2, [C|[C|Theta]], T3),
3317     append(Phi, [A & B|Psi], S),
3318     append(Phi, [A|Psi], S1),
3319     append(Phi, [B|Psi], S2),
3320     append(S1, [C|Theta], T4),
3321     append(S2, [C|Theta], T5))
3322     ;(append(Phi, [C|[C|E]], T1),
3323     append(Phi, [C|E], T),
3324     append(Phi, [C|[C|E1]], T2),
3325     append(Phi, [C|[C|E2]], T3),
3326     append(Psi, [A & B|Theta], E),
3327     append(Psi, [A|Theta], E1),
3328     append(Psi, [B|Theta], E2),
3329     append(Phi, [C|E1], T4),
3330     append(Phi, [C|E2], T5))),
3331     !,
3332     gensym(f,F),
3333     assertz(proved(Z = [dis(Z1,F,A & B), G => T])),
3334     assertz(proved(Z1 = [con(Z2,C), G => T4])),
3335     assertz(proved(F = [con(Z3,C), G => T5])),
3336     retract(proved(Z = [con(Z1,C), G => T])),
3337     retract(proved(Z1 = [dis(Z2,Z3,A & B), G => T1])).
3338     !.

```

```

3339 /* (R6,R10) - (E.27.3,4) */
3340 norm1(proved(Z = [con(Z1,A&B), G => T])) :-
3341     proved(Z1 = [dis(Z2,Z3,A & B), G => T1]),
3342     proved(Z2 = [dis(Z4,Z5,A & B), G => T2]),
3343     proved(Z4 = [th(Z6,A), G => T3]),
3344     proved(Z6 = [_, G => T4]),

```

```

3345 proved(Z5 = [th(Z7,A), G => T5]),
3346 proved(Z7 = [K1, G => T6]),
3347 proved(Z3 = [ais(Z8,Z9,A & B), G => T7]),
3348 proved(Z8 = [th(Z10,B), G => T8]),
3349 proved(Z10 = [K2, G => T4]),
3350 proved(Z9 = [th(Z11,B), G => T9]),
3351 proved(Z11 = [_, G => T6]),
3352 append(Phi, [A&B|[A&B|Psi]], T1),
3353 append(Phi, [A&B|Psi], T),
3354 append(Phi, [A|Psi], T4),
3355 append(Phi, [B|Psi], T6),
3356 append(Phi, [A|[A|Psi]], T3),
3357 append(Phi, [B|[B|Psi]], T9),
3358 ((append(Phi, [A|[A&B|Psi]], T2),
3359 append(Phi, [B|[A&B|Psi]], T7),
3360 append(Phi, [A|[B|Psi]], T5),
3361 append(Phi, [B|[A|Psi]], T8))),
3362 ;(append(Phi, [A&B|[A|Psi]], T2),
3363 append(Phi, [A&B|[B|Psi]], T7),
3364 append(Phi, [B|[A|Psi]], T5),
3365 append(Phi, [A|[B|Psi]], T8))),
3366 !,
3367 assertz(proved(Z = [ais(Z6,Z11,A & B), G => T])),
3368 retract(proved(Z = [con(Z1,A & B), G => T])),
3369 !.

```

```

3370 /* (R6,R10) - (E.27.5,6) */
3371 norm1(proved(Z = [con(Z1,A&B), G => T])) :-
3372 proved(Z1 = [ais(Z2,Z3,A & B), G => T1]),
3373 proved(Z2 = [ais(Z4,Z5,A & B), G => T2]),
3374 proved(Z4 = [th(Z6,A), G => T3]),
3375 proved(Z6 = [K1, G => T4]),
3376 proved(Z5 = [th(Z7,B), G => T5]),
3377 proved(Z7 = [K1, G => T4]),
3378 proved(Z3 = [ais(Z8,Z9,A & B), G => T6]),
3379 proved(Z8 = [th(Z10,A), G => T7]),
3380 proved(Z10 = [K2, G => T8]),
3381 proved(Z9 = [th(Z11,B), G => T9]),
3382 proved(Z11 = [K2, G => T8]),
3383 append(Phi, [A&B|[A&B|Psi]], T1),
3384 append(Phi, [A&B|Psi], T),
3385 append(Phi, [A|Psi], T4),
3386 append(Phi, [B|Psi], T8),
3387 append(Phi, [A|[A|Psi]], T3),
3388 append(Phi, [B|[B|Psi]], T9),
3389 ((append(Phi, [A|[A&B|Psi]], T2),
3390 append(Phi, [B|[A&B|Psi]], T6),
3391 append(Phi, [A|[B|Psi]], T5),
3392 append(Phi, [B|[A|Psi]], T7))),
3393 ;(append(Phi, [A&B|[A|Psi]], T2),
3394 append(Phi, [A&B|[B|Psi]], T6),
3395 append(Phi, [B|[A|Psi]], T5),
3396 append(Phi, [A|[B|Psi]], T7))),
3397 !,
3398 assertz(proved(Z = [ais(Z6,Z11,A & B), G => T])),
3399 retract(proved(Z = [con(Z1,A & B), G => T])),
3400 !.

```

```

3401 /* (R6,R10) - (E.27.7,8) */
3402 norm1(proved(Z = [con(Z1,A&B), G => T])) :-
3403 proved(Z1 = [ais(Z2,Z3,A & B), G => T1]),
3404 proved(Z2 = [th(Z4,A), G => T2]),
3405 proved(Z4 = [ais(Z5,Z6,A & B), G => T]),
3406 proved(Z5 = [_, G => T3]),
3407 proved(Z6 = [_, G => T4]),
3408 proved(Z3 = [th(Z7,B), G => T5]),
3409 proved(Z7 = [ais(Z8,Z9,A & B), G => T]),
3410 proved(Z8 = [_, G => T3]),
3411 proved(Z9 = [_, G => T4]),
3412 append(Phi, [A&B|[A&B|Psi]], T1),
3413 append(Phi, [A&B|Psi], T),
3414 append(Phi, [A|Psi], T3),
3415 append(Phi, [B|Psi], T4),
3416 ((append(Phi, [A|[A&B|Psi]], T2),
3417 append(Phi, [B|[A&B|Psi]], T5))),
3418 ;(append(Phi, [A&B|[A|Psi]], T2),
3419 append(Phi, [A&B|[B|Psi]], T5))),
3420 !,
3421 assertz(proved(Z = [ais(Z5,Z6,A & B), G => T])),
3422 retract(proved(Z = [con(Z1,A & B), G => T])),
3423 !.

```

```

3424 /* (R6,R10) - (E.27.9,10) */
3425 norm1(proved(Z = [con(Z1,A&B), G => T])) :-
3426   proved(Z1 = [ais(Z2,Z3,A & B), G => T1]),
3427   proved(Z2 = [th(Z4,A), G => T2]),
3428   proved(Z4 = [ais(Z5,Z6,A & B), G => T]),
3429   proved(Z5 = [_, G => T3]),
3430   proved(Z6 = [_, G => T4]),
3431   proved(Z3 = [ais(Z7,Z8,A & B), G => T5]),
3432   proved(Z7 = [th(Z9,A), G => T6]),
3433   proved(Z9 = [_, G => T4]),
3434   proved(Z8 = [th(Z10,B), G => T7]),
3435   proved(Z10 = [_, G => T4]),
3436   append(Phi, [A&B|[A&B|Psi]], T1),
3437   append(Phi, [A&B|Psi], T),
3438   append(Phi, [A|Psi], T3),
3439   append(Phi, [B|Psi], T4),
3440   append(Phi, [B|[B|Psi]], T7),
3441   ((append(Phi, [A|[A&B|Psi]], T2);
3442     append(Phi, [B|[A&B|Psi]], T5);
3443     append(Phi, [B|[A|Psi]], T6))
3444   );(append(Phi, [A&B|[A|Psi]], T2);
3445     append(Phi, [A&B|[B|Psi]], T5);
3446     append(Phi, [A|[B|Psi]], T6))),
3447   !,
3448   assertz(proved(Z = [ais(Z5,Z6,A & B), G => T])),
3449   retract(proved(Z = [con(Z1,A & B), G => T])),
3450   !.

```

```

3451 /* (R6,R10) - (E.27.11,12) */
3452 norm1(proved(Z = [con(Z1,A&B), G => T])) :-
3453   proved(Z1 = [ais(Z2,Z3,A & B), G => T1]),
3454   proved(Z2 = [ais(Z4,Z5,A & B), G => T2]),
3455   proved(Z4 = [th(Z6,A), G => T3]),
3456   proved(Z6 = [_, G => T4]),
3457   proved(Z5 = [th(Z7,B), G => T5]),
3458   proved(Z7 = [_, G => T4]),
3459   proved(Z3 = [th(Z8,B), G => T6]),
3460   proved(Z8 = [ais(Z9,Z10,A & B), G => T]),
3461   proved(Z9 = [_, G => T4]),
3462   proved(Z10 = [_, G => T7]),
3463   append(Phi, [A&B|[A&B|Psi]], T1),
3464   append(Phi, [A&B|Psi], T),
3465   append(Phi, [A|Psi], T4),
3466   append(Phi, [B|Psi], T7),
3467   append(Phi, [A|[A|Psi]], T3),
3468   ((append(Phi, [A|[A&B|Psi]], T2);
3469     append(Phi, [B|[A&B|Psi]], T6);
3470     append(Phi, [A|[B|Psi]], T5))
3471   );(append(Phi, [A&B|[A|Psi]], T2);
3472     append(Phi, [A&B|[B|Psi]], T6);
3473     append(Phi, [B|[A|Psi]], T5))),
3474   !,
3475   assertz(proved(Z = [ais(Z9,Z10,A & B), G => T])),
3476   retract(proved(Z = [con(Z1,A & B), G => T])),
3477   !.

```

```

3478 /* (R6,R11) - (E.28) */
3479 norm1(proved(Z = [con(Z1,C), G => T])) :-
3480   proved(Z1 = [aia(Z2,A & B), G => T1]),
3481   proved(Z2 = [_, G1 => T1]),
3482   append(Gamma, [A & B|Delta], G),
3483   append(Gamma, [A|[B|Delta]], G1),
3484   append(Phi, [C|[C|Psi]], T),
3485   append(Phi, [C|Psi], T),
3486   !,
3487   assertz(proved(Z = [aia(Z1,A & B), G => T])),
3488   assertz(proved(Z1 = [con(Z2,C), G1 => T])),
3489   retract(proved(Z = [con(Z1,C), G => T])),
3490   retract(proved(Z1 = [aia(Z2,A & B), G => T1])),
3491   !.

```

```

3492 /* (R6,R12) - (E.29) */
3493 norm1(proved(Z = [con(Z1,C), G => T])) :-
3494   proved(Z1 = [oia(Z2,Z3,A # B), G => T1]),
3495   proved(Z2 = [_, G1 => T1]),
3496   proved(Z3 = [_, G2 => T1]),
3497   append(Gamma, [A # B|Delta], G),
3498   append(Gamma, [A|Delta], G1),

```

```

3499     append(Gamma, [B|Delta], G2),
3500     append(Phi, [C|[C|Psi]], T1),
3501     append(Phi, [C|Psi], T),
3502   |,
3503     gensym(f,F),
3504     assertz(proved(Z = [ola(Z1,F,A # B), G => T])),
3505     assertz(proved(Z1 = [con(Z2,C), G1 => T])),
3506     assertz(proved(F = [con(Z3,C), G2 => T])),
3507     retract(proved(Z = [con(Z1,C), G => T])),
3508     retract(proved(Z1 = [ola(Z2,Z3,A # B), G => T])),
3509   |.

```

```

3510 /* (R5,R13) - (E.30.1,2) */
3511 norm1(proved(Z = [con(Z1,C), G => T])) :-
3512     proved(Z1 = [ois(Z2,A # B), G => T1]),
3513     proved(Z2 = [_, G => T2]),
3514     ((append(S, [C|[C|Theta]], T1),
3515      append(S, [C|Theta], T),
3516      append(S1, [C|[C|Theta]], T2),
3517      append(Phi, [A # B|Psi], S),
3518      append(Phi, [A|[B|Psi]], S1),
3519      append(S1, [C|Theta], T3))
3520     ;(append(Phi, [C|[C|E]], T1),
3521        append(Phi, [C|E], T),
3522        append(Phi, [C|[C|E1]], T2),
3523        append(Psi, [A # B|Theta], E),
3524        append(Psi, [A|[B|Theta]], E1),
3525        append(Phi, [C|E1], T3))),
3526   |,
3527     assertz(proved(Z = [ois(Z1,A # B), G => T])),
3528     assertz(proved(Z1 = [con(Z2,C), G => T3])),
3529     retract(proved(Z = [con(Z1,C), G => T])),
3530     retract(proved(Z1 = [ois(Z2,A # B), G => T])),
3531   |.

```

```

3532 /* (R6,R13) - (E.30.3,4) */
3533 norm1(proved(Z = [con(Z1,A # B), G => T])) :-
3534     proved(Z1 = [ois(Z2,A # B), G => T1]),
3535     proved(Z2 = [ois(Z3,A # B), G => T2]),
3536     proved(Z3 = [_, G => T3]),
3537     append(Phi, [A # B|Psi], T),
3538     append(Phi, [A # B|[A # B|Psi]], T1),
3539     (append(Phi, [A|[B|[A # B|Psi]]], T2),
3540      ;append(Phi, [A # B|[A|[B|Psi]]], T2)),
3541     append(Phi, [A|[B|[A|[B|Psi]]]], T3),
3542   |,
3543     assertz(proved(Z = [ois(Z1,A # B), G => T])),
3544     append(Phi, [A|[B|Psi]], T4),
3545     assertz(proved(Z1 = [con(Z2,A), G => T4])),
3546     append(Phi, [A|[A|[B|Psi]]], T5),
3547     gensym(f,F),
3548     assertz(proved(Z2 = [con(F,B), G => T5])),
3549     append(Phi, [A|[A|[B|[B|Psi]]]], T6),
3550     assertz(proved(F = [inc(Z3,A,B), G => T6])),
3551     retract(proved(Z = [con(Z1,A # B), G => T])),
3552     retract(proved(Z1 = [ois(Z2,A # B), G => T1])),
3553     retract(proved(Z2 = [ois(Z3,A # B), G => T2])),
3554   |.

```

```

3555 /* (R6,R14) - (E.31) */
3556 norm1(proved(Z = [con(Z1,C), G => T])) :-
3557     proved(Z1 = [hia(Z2,Z3,A -> B), G => T1]),
3558     proved(Z2 = [_, G1 => [A]]),
3559     proved(Z3 = [_, G2 => T1]),
3560     append(G1, [A => B], M),
3561     append(Delta, [B|Lambda], G2),
3562     append3(Delta, M, Lambda, G),
3563     append(Phi, [C|[C|Psi]], T1),
3564     append(Phi, [C|Psi], T),
3565   |,
3566     assertz(proved(Z = [hia(Z2,Z1,A -> B), G => T])),
3567     assertz(proved(Z1 = [con(Z3,C), G2 => T])),
3568     retract(proved(Z = [con(Z1,C), G => T])),
3569     retract(proved(Z1 = [hia(Z2,Z3,A -> B), G => T])),
3570   |.

```

```

3571 /* (R7,R10) - (E.32.1,2) */
3572 norm1(proved(Z = [inc(Z1,D,C), G => T])) :-
3573   proved(Z1 = [als(Z2,Z3,A & B), G => T1]),
3574   proved(Z2 = [-, G => T2]),
3575   proved(Z3 = [-, G => T3]),
3576   ((append(Phi, [D|[C|E]], T),
3577    append(Phi, [C|[D|E]], T1),
3578    append(Phi, [C|[D|E1]], T2),
3579    append(Phi, [C|[D|E2]], T3),
3580    append(Psi, [A & B|Theta], E),
3581    append(Psi, [A|Theta], E1),
3582    append(Psi, [B|Theta], E2),
3583    append(Phi, [D|[C|E1]], T4),
3584    append(Phi, [D|[C|E2]], T5))
3585   ;(append(S, [D|[C|Theta]], T),
3586    append(S, [C|[D|Theta]], T1),
3587    append(S1, [C|[D|Theta]], T2),
3588    append(S2, [C|[D|Theta]], T3),
3589    append(Phi, [A & B|Psi], S),
3590    append(Phi, [A|Psi], S1),
3591    append(Phi, [B|Psi], S2),
3592    append(S1, [D|[C|Theta]], T4),
3593    append(S2, [D|[C|Theta]], T5))),
3594   !,
3595   gensym(f,F),
3596   assertz(proved(Z = [als(Z1,F,A & B), G => T])),
3597   assertz(proved(Z1 = [inc(Z2,D,C), G => T4])),
3598   assertz(proved(F = [inc(Z3,D,C), G => T5])),
3599   retract(proved(Z = [inc(Z1,D,C), G => T])),
3600   retract(proved(Z1 = [als(Z2,Z3,A & B), G => T1])),
3601   !.

```

```

3602 /* (R4,R12) - (E.32.3) */
3603 norm1(proved(Z = [inc(Z1,C,A & B), G => T])) :-
3604   proved(Z1 = [als(Z2,Z3,A & B), G => T1]),
3605   proved(Z2 = [-, G => T2]),
3606   proved(Z3 = [-, G => T3]),
3607   append(Phi, [C|[A & B|Psi]], T),
3608   append(Phi, [A & B|[C|Psi]], T1),
3609   append(Phi, [A|[C|Psi]], T2),
3610   append(Phi, [B|[C|Psi]], T3),
3611   !,
3612   gensym(f,F),
3613   assertz(proved(Z = [als(Z1,F,A & B), G => T])),
3614   append(Phi, [C|[A|Psi]], T4),
3615   assertz(proved(Z1 = [inc(Z2,C,A), G => T4])),
3616   append(Phi, [C|[B|Psi]], T5),
3617   assertz(proved(F = [inc(Z3,C,B), G => T5])),
3618   retract(proved(Z = [inc(Z1,C,A & B), G => T])),
3619   retract(proved(Z1 = [als(Z2,Z3,A & B), G => T1])),
3620   !.

```

```

3621 /* (R4,R12) - (E.32.4) */
3622 norm1(proved(Z = [inc(Z1,A & B,C), G => T])) :-
3623   proved(Z1 = [als(Z2,Z3,A & B), G => T1]),
3624   proved(Z2 = [-, G => T2]),
3625   proved(Z3 = [-, G => T3]),
3626   append(Phi, [A & B|[C|Psi]], T),
3627   append(Phi, [C|[A & B|Psi]], T1),
3628   append(Phi, [C|[A|Psi]], T2),
3629   append(Phi, [C|[B|Psi]], T3),
3630   !,
3631   gensym(f,F),
3632   assertz(proved(Z = [als(Z1,F,A & B), G => T])),
3633   append(Phi, [A|[C|Psi]], T4),
3634   assertz(proved(Z1 = [inc(Z2,A,C), G => T4])),
3635   append(Phi, [B|[C|Psi]], T5),
3636   assertz(proved(F = [inc(Z3,B,C), G => T5])),
3637   retract(proved(Z = [inc(Z1,A & B,C), G => T])),
3638   retract(proved(Z1 = [als(Z2,Z3,A & B), G => T1])),
3639   !.

```

```

3640 /* (R7,R11) - (E.33) */
3641 norm1(proved(Z = [inc(Z1,D,C), G => T])) :-
3642   proved(Z1 = [ala(Z2,A & B), G => T1]),
3643   proved(Z2 = [-, G1 => T1]),
3644   append(Gamma, [A & B|Delta], G),
3645   append(Gamma, [A|[B|Delta]], G1),
3646   append(Phi, [D|[C|Psi]], T),
3647   append(Phi, [C|[D|Psi]], T1),
3648   !,

```



```

3649      assertz(proved(Z = [oia(Z1,A & B), G => T])),
3650      assertz(proved(Z1 = [inc(Z2,D,C), G1 => T])),
3651      retract(proved(Z = [inc(Z1,D,C), G => T])),
3652      retract(proved(Z1 = [oia(Z2,A & B), G => T])),
3653  .

```

```

3654 /* (R7,R12) - (E.34) */
3655 norm1(proved(Z = [inc(Z1,D,C), G => T])) :-
3656   proved(Z1 = [oia(Z2,Z3,A#B), G => T1]),
3657   proved(Z2 = [-, G1 => T1]),
3658   proved(Z3 = [-, G2 => T1]),
3659   append(Gamma, [A#B|Delta], G),
3660   append(Gamma, [A|Delta], G1),
3661   append(Gamma, [B|Delta], G2),
3662   append(Phi, [D|[C|Psi]], T),
3663   append(Phi, [C|[D|Psi]], T1),
3664 .
3665   gensym(f,F),
3666   assertz(proved(Z = [oia(Z1,F,A#B), G => T])),
3667   assertz(proved(Z1 = [inc(Z2,D,C), G1 => T])),
3668   assertz(proved(F = [inc(Z3,D,C), G2 => T])),
3669   retract(proved(Z = [inc(Z1,D,C), G => T])),
3670   retract(proved(Z1 = [oia(Z2,Z3,A#B), G => T])),
3671 .

```

```

3672 /* (R7,R13) - (E.35.1,2) */
3673 norm1(proved(Z = [inc(Z1,D,C), G => T])) :-
3674   proved(Z1 = [ois(Z2,A#B), G => T1]),
3675   proved(Z2 = [-, G => T2]),
3676   ((append(S, [D|[C|Theta]], T),
3677    append(S, [C|[D|Theta]], T1),
3678    append(S1, [C|[D|Theta]], T2),
3679    append(Phi, [A#B|Psi], S),
3680    append(Phi, [A|[B|Psi]], S1),
3681    append(S1, [D|[C|Theta]], T3))
3682  ; (append(Phi, [D|[C|E]], T),
3683    append(Phi, [C|[D|E]], T1),
3684    append(Phi, [C|[D|E1]], T2),
3685    append(Psi, [A#B|Theta], E),
3686    append(Psi, [A|[B|Theta]], E1),
3687    append(Phi, [D|[C|E1]], T3))),
3688 .
3689   assertz(proved(Z = [ois(Z1,A # B), G => T])),
3690   assertz(proved(Z1 = [inc(Z2,D,C), G => T3])),
3691   retract(proved(Z = [inc(Z1,D,C), G => T])),
3692   retract(proved(Z1 = [ois(Z2,A # B), G => T])),
3693 .

```

```

3694 /* (R7,R13) - (E.35.3) */
3695 norm1(proved(Z = [inc(Z1,C,A # B), G => T])) :-
3696   proved(Z1 = [ois(Z2,A # B), G => T1]),
3697   proved(Z2 = [-, G => T2]),
3698   append(Phi, [C|[A#B|Theta]], T),
3699   append(Phi, [A#B|[C|Theta]], T1),
3700   append(Phi, [A|[B|[C|Theta]]], T2),
3701 .
3702   assertz(proved(Z = [ois(Z1,A # B), G => T])),
3703   append(Phi, [C|[A|[B|Theta]]], T3),
3704   gensym(f,F),
3705   assertz(proved(Z1 = [inc(F,C,A), G => T3])),
3706   append(Phi, [A|[C|[B|Theta]]], T4),
3707   assertz(proved(F = [inc(Z2,C,B), G => T4])),
3708   retract(proved(Z = [inc(Z1,C,A # B), G => T])),
3709   retract(proved(Z1 = [ois(Z2,A # B), G => T])),
3710 .

```

```

3711 /* (R7,R13) - (E.35.4) */
3712 norm1(proved(Z = [inc(Z1,A # B,C), G => T])) :-
3713   proved(Z1 = [ois(Z2,A # B), G => T1]),
3714   proved(Z2 = [-, G => T2]),
3715   append(Phi, [A#B|[C|Theta]], T),
3716   append(Phi, [C|[A#B|Theta]], T1),
3717   append(Phi, [C|[A|[B|Theta]]], T2),
3718 .
3719   assertz(proved(Z = [ois(Z1,A # B), G => T])),
3720   append(Phi, [A|[B|[C|Theta]]], T3),
3721   gensym(f,F),
3722   assertz(proved(Z1 = [inc(F,B,C), G => T3])),
3723   append(Phi, [A|[C|[B|Theta]]], T4),

```

```

3724      assertz(proved(F = [inc(Z2,A,C), G => T4])),
3725      retract(proved(Z = [inc(Z1,A # B,C), G => T])),
3726      retract(proved(Z1 = [ois(Z2,A # B), G => T1])),
3727

```

```

3728 /* (R7,R14) - (E.36) */
3729 norm1(proved(Z = [inc(Z1,D,C), G => T])) :-
3730     proved(Z1 = [hia(Z2,Z3,A -> B), G => T1]),
3731     proved(Z2 = [-, G1 => [A]]),
3732     proved(Z3 = [-, G2 => T1]),
3733     append(G1, [A -> B], M),
3734     append(Delta, [B|Lambda], G2),
3735     append3(Delta, M, Lambda, G),
3736     append(Phi, [D|[C|Psi]], T),
3737     append(Phi, [C|[D|Psi]], T1).
3738
3739     ,
3740     assertz(proved(Z = [hia(Z2,Z1,A -> B), G => T])),
3741     assertz(proved(Z1 = [inc(Z3,D,C), G2 => T])),
3742     retract(proved(Z = [inc(Z1,D,C), G => T])),
3743     retract(proved(Z1 = [hia(Z2,Z3,A -> B), G => T1])),
3744

```

```

3744 /* (R2,R3,R11) - (E.37.1,2) */
3745 norm1(proved(Z = [con(Z1,A & B), G => T])) :-
3746     proved(Z1 = [aia(Z2,A & B), G1 => T]),
3747     proved(Z2 = [th(Z3,A), G2 => T]),
3748     proved(Z3 = [th(Z4,B), G3 => T]),
3749     proved(Z4 = [K, G => T]),
3750     append(Gamma, [A & B|[A & B|Delta]], G1),
3751     ((append(Gamma, [A & B|[A|[B|Delta]]], G2),
3752     append(Gamma, [A & B|[B|Delta]], G3))
3753     : (append(Gamma, [A|[B|[A & B|Delta]]], G2),
3754     append(Gamma, [B|[A & B|Delta]], G3))),
3755     append(Gamma, [A & B|Delta], G),
3756
3757     ,
3758     assertz(proved(Z = [K, G => T])),
3759     retract(proved(Z4 = [K, G => T])),
3760     retract(proved(Z3 = [th(Z4,B), G3 => T])),
3761     retract(proved(Z2 = [th(Z3,A), G2 => T])),
3762     retract(proved(Z1 = [aia(Z2,A & B), G1 => T])),
3763     retract(proved(Z = [con(Z1,A & B), G => T])),
3764

```

```

3764 /* (R2,R3,R11) - (E.37.3) */
3765 norm1(proved(Z = [con(Z1,A & B), G => T])) :-
3766     proved(Z1 = [aia(Z2,A & B), G1 => T]),
3767     proved(Z2 = [th(Z3,A), G2 => T]),
3768     proved(Z3 = [aia(Z4,A & B), G3 => T]),
3769     proved(Z4 = [-, G4 => T]),
3770     append(Gamma, [A & B|Delta], G),
3771     append(Gamma, [A & B|[A & B|Delta]], G1),
3772     append(Gamma, [A & B|[A|[B|Delta]]], G2),
3773     append(Gamma, [A & B|[B|Delta]], G3),
3774     append(Gamma, [A|[B|[B|Delta]]], G4).
3775
3776     ,
3777     assertz(proved(Z = [aia(Z1,A & B), G => T]),
3778     append(Gamma, [A|[B|Delta]], G5),
3779     assertz(proved(Z1 = [con(Z4,B), G5 => T])),
3780     retract(proved(Z = [con(Z1,A & B), G => T])),
3781     retract(proved(Z1 = [aia(Z2,A & B), G1 => T])),
3782     retract(proved(Z2 = [th(Z3,A), G2 => T])),
3783     retract(proved(Z3 = [aia(Z4,A & B), G3 => T])),
3784

```

```

3784 /* (R2,R3,R11) - (E.37.4) */
3785 norm1(proved(Z = [con(Z1,A & B), G => T])) :-
3786     proved(Z1 = [aia(Z2,A & B), G1 => T]),
3787     proved(Z2 = [th(Z3,A), G2 => T]),
3788     proved(Z3 = [aia(Z4,A & B), G3 => T]),
3789     proved(Z4 = [-, G4 => T]),
3790     append(Gamma, [A & B|Delta], G),
3791     append(Gamma, [A & B|[A & B|Delta]], G1),
3792     append(Gamma, [A|[B|[A & B|Delta]]], G2),
3793     append(Gamma, [B|[A & B|Delta]], G3),
3794     append(Gamma, [B|[A|[B|Delta]]], G4).
3795
3796     ,
3797     assertz(proved(Z = [aia(Z1,A & B), G => T]),
3798     append(Gamma, [A|[B|Delta]], G5),
3799     assertz(proved(Z1 = [con(Z2,B), G5 => T])),

```

```

3799      append(Gamma, [A|[B|[B|Delta]]], G6),
3800      assertz(proved(Z2 = [inc(Z4,A,B), G6 => T])),
3801      retract(proved(Z = [con(Z1,A & B), G => T])),
3802      retract(proved(Z1 = [aia(Z2,A & B), G1 => T])),
3803      retract(proved(Z2 = [th(Z3,A), G2 => T])),
3804      retract(proved(Z3 = [aia(Z4,A & B), G3 => T])),
3805

```

```

3806 /* (R2,R3,R11) - (E.37.5) */
3807 norm1(proved(Z = [con(Z1,A & B), G => T])) :-
3808     proved(Z1 = [aia(Z2,A & B), G1 => T]),
3809     proved(Z2 = [th(Z3,B), G2 => T]),
3810     proved(Z3 = [aia(Z4,A & B), G3 => T]),
3811     proved(Z4 = [_, G4 => T]),
3812     append(Gamma, [A & B|Delta], G),
3813     cappend(Gamma, [A & B|[A & B|Delta]], G1),
3814     append(Gamma, [A & B|[A|[B|Delta]]], G2),
3815     append(Gamma, [A & B|[A|Delta]], G3),
3816     append(Gamma, [A|[B|[A|Delta]]], G4),
3817
3818     assertz(proved(Z = [aia(Z1,A & B), G => T])),
3819     append(Gamma, [A|[B|Delta]], G5),
3820     assertz(proved(Z1 = [con(Z2,B), G5 => T])),
3821     append(Gamma, [A|[A|[B|Delta]]], G6),
3822     assertz(proved(Z2 = [inc(Z4,A,B), G6 => T])),
3823     retract(proved(Z = [con(Z1,A & B), G => T])),
3824     retract(proved(Z1 = [aia(Z2,A & B), G1 => T])),
3825     retract(proved(Z2 = [th(Z3,B), G2 => T])),
3826     retract(proved(Z3 = [aia(Z4,A & B), G3 => T])),
3827

```

```

3828 /* (R2,R3,R11) - (E.37.6) */
3829 norm1(proved(Z = [con(Z1,A & B), G => T])) :-
3830     proved(Z1 = [aia(Z2,A & B), G1 => T]),
3831     proved(Z2 = [th(Z3,B), G2 => T]),
3832     proved(Z3 = [aia(Z4,A & B), G3 => T]),
3833     proved(Z4 = [_, G4 => T]),
3834     append(Gamma, [A & B|Delta], G),
3835     append(Gamma, [A & B|[A & B|Delta]], G1),
3836     append(Gamma, [A|[B|[A & B|Delta]]], G2),
3837     append(Gamma, [A|[A & B|Delta]], G3),
3838     append(Gamma, [A|[A|[B|Delta]]], G4),
3839
3840     assertz(proved(Z = [aia(Z1,A & B), G => T])),
3841     append(Gamma, [A|[B|Delta]], G5),
3842     assertz(proved(Z1 = [con(Z4,A), G5 => T])),
3843     retract(proved(Z = [con(Z1,A & B), G => T])),
3844     retract(proved(Z1 = [aia(Z2,A & B), G1 => T])),
3845     retract(proved(Z2 = [th(Z3,B), G2 => T])),
3846     retract(proved(Z3 = [aia(Z4,A & B), G3 => T])),
3847

```

```

3848 /* (R5,R6,R13) - (E.38.1,2) */
3849 norm1(proved(Z = [con(Z1,A # B), G => T])) :-
3850     proved(Z1 = [ois(Z2,A # B), G => T1]),
3851     proved(Z2 = [th(Z3,A), G => T2]),
3852     proved(Z3 = [th(Z4,B), G => T3]),
3853     proved(Z4 = [K, G => T]),
3854     append(Phi, [A # B|[A # B|Psi]], T1),
3855     ((append(Phi, [A # B|[A|[B|Psi]]], T2),
3856     append(Phi, [A # B|[B|Psi]], T3))
3857     : (append(Phi, [A|[B|[A # B|Psi]]], T2),
3858     append(Phi, [B|[A # B|Psi]], T3))),
3859     append(Phi, [A # B|Psi], T),
3860
3861     assertz(proved(Z = [K, G => T])),
3862     retract(proved(Z4 = [K, G => T])),
3863     retract(proved(Z3 = [th(Z4,B), G => T3])),
3864     retract(proved(Z2 = [th(Z3,A), G => T2])),
3865     retract(proved(Z1 = [ois(Z2,A # B), G => T1])),
3866     retract(proved(Z = [con(Z1,A # B), G => T])),
3867

```

```

3868 /* (R2,R3,R11) - (E.38.3) */
3869 norm1(proved(Z = [con(Z1,A # B), G => T])) :-
3870     proved(Z1 = [ois(Z2,A # B), G => T1]),
3871     proved(Z2 = [th(Z3,A), G => T2]),
3872     proved(Z3 = [ois(Z4,A # B), G => T3]),
3873     proved(Z4 = [_, G => T4]),
3874     append(Phi, [A # B|Psi], T),

```

```

3875     append(Phi, [A # B | [A # B | Psi]], T1);
3876     append(Phi, [A # B | [A | [B | Psi]]], T2);
3877     append(Phi, [A # B | [B | Psi]], T3);
3878     append(Phi, [A | [B | [B | Psi]]], T4);
3879
3880     assertz(proved(Z = [ois(Z1, A # B), G => T])).
3881     append(Phi, [A | [B | Psi]], T5);
3882     assertz(proved(Z1 = [con(Z4, B), G => T5])).
3883     retract(proved(Z = [con(Z1, A # B), G => T1])).
3884     retract(proved(Z1 = [ois(Z2, A # B), G => T1])).
3885     retract(proved(Z2 = [th(Z3, A), G => T2])).
3886     retract(proved(Z3 = [ois(Z4, A # B), G => T3])).
3887

```

```

3888 /* (R2,R3,R11) - (E.38.4) */
3889 norm1(proved(Z = [con(Z1, A # B), G => T])) :-
3890     proved(Z1 = [ois(Z2, A # B), G => T1]),
3891     proved(Z2 = [th(Z3, A), G => T2]),
3892     proved(Z3 = [ois(Z4, A # B), G => T3]),
3893     proved(Z4 = [-, G => T4]),
3894     append(Phi, [A # B | Psi], T),
3895     append(Phi, [A # B | [A # B | Psi]], T1),
3896     append(Phi, [A | [B | [A # B | Psi]]], T2),
3897     append(Phi, [B | [A # B | Psi]], T3),
3898     append(Phi, [B | [A | [B | Psi]]], T4),
3899
3900     assertz(proved(Z = [ois(Z1, A # B), G => T])).
3901     append(Phi, [A | [B | Psi]], T5);
3902     assertz(proved(Z1 = [con(Z2, B), G => T5])).
3903     append(Phi, [A | [B | [B | Psi]]], T6);
3904     assertz(proved(Z2 = [inc(Z4, A, B), G => T6])).
3905     retract(proved(Z = [con(Z1, A # B), G => T1])).
3906     retract(proved(Z1 = [ois(Z2, A # B), G => T1])).
3907     retract(proved(Z2 = [th(Z3, A), G => T2])).
3908     retract(proved(Z3 = [ois(Z4, A # B), G => T3])).
3909

```

```

3910 /* (R2,R3,R11) - (E.38.5) */
3911 norm1(proved(Z = [con(Z1, A # B), G => T])) :-
3912     proved(Z1 = [ois(Z2, A # B), G => T1]),
3913     proved(Z2 = [th(Z3, B), G => T2]),
3914     proved(Z3 = [ois(Z4, A # B), G => T3]),
3915     proved(Z4 = [-, G => T4]),
3916     append(Phi, [A # B | Psi], T),
3917     append(Phi, [A # B | [A # B | Psi]], T1),
3918     append(Phi, [A # B | [A | [B | Psi]]], T2),
3919     append(Phi, [A # B | [A | Psi]], T3),
3920     append(Phi, [A | [B | [A | Psi]]], T4),
3921
3922     assertz(proved(Z = [ois(Z1, A # B), G => T])).
3923     append(Phi, [A | [B | Psi]], T5);
3924     assertz(proved(Z1 = [con(Z2, B), G => T5])).
3925     append(Phi, [A | [A | [B | Psi]]], T6);
3926     assertz(proved(Z2 = [inc(Z4, A, B), G => T6])).
3927     retract(proved(Z = [con(Z1, A # B), G => T1])).
3928     retract(proved(Z1 = [ois(Z2, A # B), G => T1])).
3929     retract(proved(Z2 = [th(Z3, B), G => T2])).
3930     retract(proved(Z3 = [ois(Z4, A # B), G => T3])).
3931

```

```

3932 /* (R2,R3,R11) - (E.38.6) */
3933 norm1(proved(Z = [con(Z1, A # B), G => T])) :-
3934     proved(Z1 = [ois(Z2, A # B), G => T1]),
3935     proved(Z2 = [th(Z3, B), G => T2]),
3936     proved(Z3 = [ois(Z4, A # B), G => T3]),
3937     proved(Z4 = [-, G => T4]),
3938     append(Phi, [A # B | Psi], T),
3939     append(Phi, [A # B | [A # B | Psi]], T1),
3940     append(Phi, [A | [B | [A # B | Psi]]], T2),
3941     append(Phi, [A | [A # B | Psi]], T3),
3942     append(Phi, [A | [A | [B | Psi]]], T4),
3943
3944     assertz(proved(Z = [ois(Z1, A # B), G => T])).
3945     append(Phi, [A | [B | Psi]], T5);
3946     assertz(proved(Z1 = [con(Z4, A), G => T5])).
3947     retract(proved(Z = [con(Z1, A # B), G => T1])).
3948     retract(proved(Z1 = [ois(Z2, A # B), G => T1])).
3949     retract(proved(Z2 = [th(Z3, B), G => T2])).
3950     retract(proved(Z3 = [ois(Z4, A # B), G => T3])).
3951

```

```

3952 /* (R3,R12,R14) -- (E.39) */
3953 norm1(proved(Z = [con(Z1,A # B), G => T])) :-
3954   proved(Z1 = [hia(Z2,Z3,C -> D), G1 => T]),
3955   proved(Z2 = [oia(Z4,Z5,A#B), [A#B|Gamma] => [C]]),
3956   proved(Z4 = [A|Gamma] => [C]),
3957   proved(Z5 = [B|Gamma] => [C]),
3958   proved(Z3 = [oia(Z6,Z7,A # B), G2 => T]),
3959   proved(Z6 = [G3 => T]),
3960   proved(Z7 = [G4 => T]),
3961   append(Delta,[A#B|[A#B|E]],G1),
3962   append(Delta,[A#B|E],G),
3963   append(Gamma,[C -> D|Lambda],E),
3964   append(Delta,[A # B|[D|Lambda]],G2),
3965   append(Delta,[A|[D|Lambda]],G3),
3966   append(Delta,[B|[D|Lambda]],G4),
3967
3968   assertz(proved(Z = [oia(Z1,Z2,A # B), G => T])),
3969   append(Delta,[A|E],G5),
3970   assertz(proved(Z1 = [con(Z3,A), G5 => T])),
3971   append(Delta,[A|[A|E]],G6),
3972   assertz(proved(Z3 = [hia(Z4,Z6,C->D), G6 => T])),
3973   append(Delta,[B|E],G7),
3974   assertz(proved(Z2 = [con(Z8,B), G7 => T])),
3975   append(Delta,[B|[B|E]],G8),
3976   assertz(proved(Z8 = [hia(Z5,Z7,C->D), G8 => T])),
3977   retract(proved(Z1 = [hia(Z2,Z3,C->D), G1 => T])),
3978   retract(proved(Z2=[oia(Z4,Z5,A#B), [A#B|Gamma] => [C]])),
3979   retract(proved(Z3 = [oia(Z6,Z7,A # B), G2 => T])),
3980   retract(proved(Z = [con(Z1,A # B), G => T])),
3981
3982

```

```

3983 /* (R4,R12,R14) -- (E.40.1) */
3984 norm1(proved(Z = [inc(Z1,A # B,E), G => T])) :-
3985   proved(Z1 = [hia(Z2,Z3,C -> D), G1 => T]),
3986   proved(Z2 = [oia(Z4,Z5,A#B), [A#B|Gamma] => [C]]),
3987   proved(Z4 = [A|Gamma] => [C]),
3988   proved(Z5 = [B|Gamma] => [C]),
3989   proved(Z3 = [K, G2 => T]),
3990   append(Delta,[E|[A#B|R]],G1),
3991   append(Delta,[A#B|[E|R]],G),
3992   append(Gamma,[C -> D|Lambda],R),
3993   append(Delta,[E|[D|Lambda]],G2),
3994
3995   assertz(proved(Z = [oia(Z1,Z2,A # B), G => T])),
3996   append(Delta,[A|[E|R]],G3),
3997   assertz(proved(Z1 = [inc(Z6,A,E), G3 => T])),
3998   append(Delta,[E|[A|R]],G4),
3999   assertz(proved(Z6 = [hia(Z4,Z3,C->D), G4 => T])),
4000   append(Delta,[B|[E|R]],G5),
4001   assertz(proved(Z2 = [inc(Z7,B,E), G5 => T])),
4002   append(Delta,[E|[B|R]],G6),
4003   assertz(proved(Z7 = [hia(Z5,Z8,C->D), G6 => T])),
4004   retract(proved(Z1 = [K, G2 => T])),
4005   retract(proved(Z1 = [hia(Z2,Z3,C -> D), G1 => T])),
4006   retract(proved(Z2=[oia(Z4,Z5,A#B), [A#B|Gamma] => [C]])),
4007   retract(proved(Z = [inc(Z1,A # B,E), G => T])),
4008
4009

```

```

4010 /* (R4,R12,R14) -- (E.40.2) */
4011 norm1(proved(Z = [inc(Z1,E,A # B), G => T])) :-
4012   proved(Z1 = [hia(Z2,Z3,C -> D), G1 => T]),
4013   proved(Z2 = [K, [E|Gamma] => [C]]),
4014   proved(Z3 = [oia(Z4,Z5,A # B), G2 => T]),
4015   proved(Z4 = [G3 => T]),
4016   proved(Z5 = [G4 => T]),
4017   append(Delta,[E|[A#B|R]],G),
4018   append(Delta,[A#B|[E|R]],G1),
4019   append(Gamma,[C -> D|Lambda],R),
4020   append(Delta,[A # B|[D|Lambda]],G2),
4021   append(Delta,[A|[D|Lambda]],G3),
4022   append(Delta,[B|[D|Lambda]],G4),
4023
4024   assertz(proved(Z = [oia(Z1,Z3,A # B), G => T])),
4025   append(Delta,[E|[A|R]],G5),
4026   assertz(proved(Z1 = [inc(Z6,E,A), G5 => T])),
4027   append(Delta,[A|[E|R]],G6),
4028   assertz(proved(Z6 = [hia(Z2,Z4,C -> D), G6 => T])),
4029   append(Delta,[E|[B|R]],G7),
4030   assertz(proved(Z3 = [inc(Z7,E,B), G7 => T])),

```

```

4031      append(Delta, [B|[E|R]], G8),
4032      assertz(proved(Z = [inc(Z8, Z5, C -> D), G8 => T])),
4033      assertz(proved(Z8 = [K, [E|Gamma] => [C]])),
4034      retract(proved(Z1 = [hia(Z2, Z3, C -> D), G1 => T])),
4035      retract(proved(Z3 = [oia(Z4, Z5, A # B), G2 => T])),
4036      retract(proved(Z = [inc(Z1, E, A # B), G => T])).
4037
4038

```

```

4039 /* (R4, R12, R14) - (E.40.3) */
4040 norm1(proved(Z = [inc(Z1, A # B, C -> D), G => T])) :-
4041     proved(Z1 = [hia(Z2, Z3, C -> D), G1 => T]),
4042     proved(Z2 = [K, Gamma => [C]]),
4043     proved(Z3 = [oia(Z4, Z5, A # B), G2 => T]),
4044     proved(Z4 = [-, G3 => T]),
4045     proved(Z5 = [-, G4 => T]),
4046     append(S, [C->D|[A#B|Lambda]], G1),
4047     append(S, [A#B|[C->D|Lambda]], G),
4048     append(Delta, Gamma, S),
4049     append(Delta, [D|[A # B|Lambda]], G2),
4050     append(Delta, [D|[A|Lambda]], G3),
4051     append(Delta, [D|[B|Lambda]], G4),
4052
4053     assertz(proved(Z = [oia(Z1, Z3, A # B), G => T])),
4054     append(S, [A|[C->D|Lambda]], G5),
4055     assertz(proved(Z1 = [inc(Z6, A, C->D), G5 => T])),
4056     append(S, [C->D|[A|Lambda]], G6),
4057     assertz(proved(Z6 = [hia(Z2, Z4, C -> D), G6 => T])),
4058     append(S, [B|[C->D|Lambda]], G7),
4059     assertz(proved(Z3 = [inc(Z7, B, C->D), G7 => T])),
4060     append(S, [C->D|[B|Lambda]], G8),
4061     assertz(proved(Z7 = [hia(Z8, Z5, C -> D), G8 => T])),
4062     assertz(proved(Z8 = [K, Gamma => [C]])),
4063     retract(proved(Z1 = [hia(Z2, Z3, C -> D), G1 => T])),
4064     retract(proved(Z3 = [oia(Z4, Z5, A # B), G2 => T])),
4065     retract(proved(Z = [inc(Z1, A # B, C -> D), G => T])).
4066
4067

```

```

4068 /* (R4, R12, R14) - (E.40.4) */
4069 norm1(proved(Z = [inc(Z1, C -> D, A # B), G => T])) :-
4070     proved(Z1 = [hia(Z2, Z3, C -> D), G1 => T]),
4071     proved(Z2 = [oia(Z4, Z5, A#B), [Gamma|A#B] => [C]]),
4072     proved(Z4 = [-, [Gamma|A] => [C]]),
4073     proved(Z5 = [-, [Gamma|B] => [C]]),
4074     proved(Z3 = [K, G2 => T]),
4075     append(S, [A#B|[C->D|Lambda]], G1),
4076     append(S, [C->D|[A#B|Lambda]], G),
4077     append(Delta, Gamma, S),
4078     append(Delta, [D|Lambda], G2),
4079
4080     assertz(proved(Z = [oia(Z1, Z2, A # B), G => T])),
4081     append(S, [C->D|[A|Lambda]], G3),
4082     assertz(proved(Z1 = [inc(Z6, C -> D, A), G3 => T])),
4083     append(S, [A|[C->D|Lambda]], G4),
4084     assertz(proved(Z6 = [hia(Z4, Z3, C -> D), G4 => T])),
4085     append(S, [C->D|[B|Lambda]], G5),
4086     assertz(proved(Z2 = [inc(Z7, C -> D, B), G5 => T])),
4087     append(S, [B|[C->D|Lambda]], G6),
4088     assertz(proved(Z7 = [hia(Z5, Z8, C -> D), G6 => T])),
4089     assertz(proved(Z8 = [K, G2 => T])),
4090     retract(proved(Z1 = [hia(Z2, Z3, C -> D), G1 => T])),
4091     retract(proved(Z2 = [oia(Z4, Z5, A#B), [Gamma|A#B] => [C]])),
4092     retract(proved(Z = [inc(Z1, C->D, A#B), G => T])),
4093
4094

```

```

4095 /* (R2) - (E.41.1) */
4096 norm1(proved(Z = [th(Z1, A & B), G => T])) :-
4097     proved(Z1 = [-, G1 => T]),
4098     append(Gamma, [A & B|Delta], G),
4099     append(Gamma, Delta, G1),
4100
4101     gensym(f, F1),
4102     assertz(proved(Z = [aia(F1, A & B), G => T])),
4103     append(Gamma, [A|[B|Delta]], G2),
4104     gensym(f, F2),
4105     assertz(proved(F1 = [th(F2, A), G2 => T])),
4106     append(Gamma, [B|Delta], G3),
4107     assertz(proved(F2 = [th(Z1, B), G3 => T])),
4108     retract(proved(Z = [th(Z1, A & B), G => T])),
4109

```

```

4110 /* (R2) - (E.41.2) */
4111 norm1(proved(Z = [th(Z1,A # B), G => T])) :-
4112     proved(Z1 = [K, G1 => T]),
4113     append(Gamma, [A # B|Delta], G),
4114     append(Gamma, Delta, G1),
4115     ((thry(bicart), Gamma = [], Delta = []))
4116     ; thry(dbicart)
4117     ; thry(bicartcl)),
4118     !,
4119     gensym(f,F1),
4120     gensym(f,F2),
4121     assertz(proved(Z = [ola(F1,F2,A # B), G => T])),
4122     append(Gamma, [A|Delta], G2),
4123     append(Gamma, [B|Delta], G3),
4124     assertz(proved(F1 = [th(Z1,A), G2 => T])),
4125     assertz(proved(F2 = [th(Z1,B), G3 => T])),
4126     branch(F2),
4127     retract(proved(Z = [th(Z1,A # B), G => T])),
4128     !.

```

```

4129 /* (R5) - (E.42.1) */
4130 norm1(proved(Z = [th(Z1,A & B), G => T])) :-
4131     proved(Z1 = [K, G => T1]),
4132     append(Phi, [A & B|Psi], T),
4133     append(Phi, Psi, T1),
4134     ((thry(bicart), Phi = [], Psi = []))
4135     ; thry(dbicart)
4136     ; thry(bicartcl)),
4137     !,
4138     gensym(f,F1),
4139     gensym(f,F2),
4140     assertz(proved(Z = [als(F1,F2,A & B), G => T])),
4141     append(Phi, [A|Psi], T2),
4142     assertz(proved(F1 = [th(Z1,A), G => T2])),
4143     append(Phi, [B|Psi], T3),
4144     assertz(proved(F2 = [th(Z1,B), G => T3])),
4145     branch(F2),
4146     retract(proved(Z = [th(Z1,A & B), G => T])),
4147     !.

```

```

4148 /* (R5) - (E.42.2) */
4149 norm1(proved(Z = [th(Z1,A # B), G => T])) :-
4150     proved(Z1 = [_, G => T1]),
4151     append(Phi, [A # B|Psi], T),
4152     append(Phi, Psi, T1),
4153     !,
4154     gensym(f,F1),
4155     assertz(proved(Z = [ola(F1,A # B), G => T])),
4156     gensym(f,F2),
4157     append(Phi, [A|[B|Psi]], T2),
4158     assertz(proved(F1 = [th(F2,A), G => T2])),
4159     append(Phi, [B|Psi], T3),
4160     assertz(proved(F2 = [th(Z1,B), G => T3])),
4161     retract(proved(Z = [th(Z1,A # B), G => T])),
4162     !.

```

```

4163 norm1(proved(Z)) :-
4164     retract(proved(Z)),
4165     assertz(proved(Z)), !.

```

4186

/\* Normalization Program \*/

```

4167 normal2 :-
4168     main(Z),
4169     repeat,
4170     do2_step,
4171     ((thry(cartcl) ; thry(bicartcl)), not(second) )
4172     ; (write('Normal proof is :'), nl,
4173       write('====='), nl, nl,
4174       printall(Z), nl, nl, nl) ).

```

```

4175 do2_step :-
4176     main(X),
4177     put_in_list(X,L),
4178     assertz(proved(mark)),
4179     proved(Z),
4180     norm2(proved(Z)),
4181     Z = mark,
4182     expinc,
4183     expcon,
4184     expth,
4185     put_in_list(X,L1),
4186     abolish(proved,_),
4187     ass_list(L1), !,
4188     L == L1.

```

```

4189 ass_list([]).
4190 ass_list([X|Y]) :-
4191     assert(proved(X)),
4192     ass_list(Y).

```

```

4193 /* (R2,R3) - (D.2.1,2) */
4194 norm2(proved(Z = [con(Z1,A), G => T])) :-
4195     proved(Z1 = [th(Z2,B), G1 => T]),
4196     proved(Z2 = [_, G2 => T]),
4197     ((append(Gamma, [A|[A|E]], G1),
4198      append(Gamma, [A|E], G),
4199      append(Gamma, [A|[A|E1]], G2),
4200      append(Delta, [B|Lambda], E),
4201      append(Delta, Lambda, E1),
4202      append(Gamma, [A|E1], G3))
4203     ; (append(S, [A|[A|Lambda]], G1),
4204        append(S, [A|Lambda], G),
4205        append(S1, [A|[A|Lambda]], G2),
4206        append(Gamma, [B|Delta], S),
4207        append(Gamma, Delta, S1),
4208        append(S1, [A|Lambda], G3))),
4209     !,
4210     assertz(proved(Z = [th(Z1,B), G => T])),
4211     assertz(proved(Z1 = [con(Z2,A), G3 => T])),
4212     retract(proved(Z = [con(Z1,A), G => T])),
4213     retract(proved(Z1 = [th(Z2,B), G1 => T])),
4214     !.

```

```

4215 /* (R2,R3) - (D.2.3,4) */
4216 norm2(proved(Z = [con(Z1,A), G => T])) :-
4217     proved(Z1 = [th(Z2,A), G1 => T]),
4218     proved(Z2 = [F, G => T]),
4219     append(Gamma, [A|[A|Delta]], G1),
4220     append(Gamma, [A|Delta], G),
4221     !,
4222     assertz(proved(Z = [F, G => T])),
4223     retract(proved(Z = [con(Z1,A), G => T])),
4224     retract(proved(Z1 = [th(Z2,A), G1 => T])),
4225     retract(proved(Z2 = [F, G => T])),
4226     !.

```

```

4227 /* (R2, R4) - (D.3.1,2) */
4228 norm2(proved(Z = [th(Z1,C), G => T])) :-
4229     proved(Z1 = [inc(Z2,B,A), G1 => T]),
4230     proved(Z2 = [_, G2 => T]),
4231     ((append(Gamma, [B|[A|E]], G),
4232      append(Gamma, [B|[A|E1]], G1),
4233      append(Gamma, [A|[B|E1]], G2),
4234      append(Delta, [C|Lambda], E),
4235      append(Delta, Lambda, E1),
4236      append(Gamma, [A|[B|E]], G3))
4237     ; (append(S, [B|[A|Lambda]], G),

```



```

4238     append(S1, [B|[A|Lambda]], G1),
4239     append(S1, [A|[B|Lambda]], G2),
4240     append(Gamma, [C|Delta], S),
4241     append(Gamma, Delta, S1),
4242     append(S, [A|[B|Lambda]], G3))),
4243 1,
4244     assertz(proved(Z = [inc(Z1,B,A), G => T])),
4245     assertz(proved(Z1 = [th(Z2,C), G3 => T])),
4246     retract(proved(Z1 = [inc(Z2,B,A), G1 => T])),
4247     retract(proved(Z = [th(Z1,C), G => T])),
4248 1.

4249 /* (R2,R4) - (D.3.3) */
4250 norm2(proved(Z = [inc(Z1,B,A), G => T])) :-
4251     proved(Z1 = [th(Z2,B), G1 => T]),
4252     proved(Z2 = [F, G2 => T]),
4253     append(Gamma, [A|[B|Delta]], G1),
4254     append(Gamma, [B|[A|Delta]], G),
4255     append(Gamma, [A|Delta], G2),
4256 1,
4257     assertz(proved(Z = [th(Z2,B), G => T])),
4258     retract(proved(Z1 = [th(Z2,B), G1 => T])),
4259     retract(proved(Z = [inc(Z1,B,A), G => T])),
4260 1.

4261 /* (R2,R4) - (D.3.4) */
4262 norm2(proved(Z = [inc(Z1,A,B), G => T])) :-
4263     proved(Z1 = [th(Z2,B), G1 => T]),
4264     proved(Z2 = [F, G2 => T]),
4265     append(Gamma, [A|[B|Delta]], G),
4266     append(Gamma, [B|[A|Delta]], G1),
4267     append(Gamma, [A|Delta], G2),
4268 1,
4269     assertz(proved(Z = [th(Z2,B), G => T])),
4270     retract(proved(Z1 = [th(Z2,B), G1 => T])),
4271     retract(proved(Z = [inc(Z1,A,B), G => T])),
4272 1.

4273 /* (R2,R5) - (D.4) */
4274 norm2(proved(Z = [th(Z1,B), G => T])) :-
4275     proved(Z1 = [th(Z2,A), G => T1]),
4276     proved(Z2 = [_, G1 => T1]),
4277     append(Gamma, [A|Delta], G),
4278     append(Gamma, Delta, G1),
4279     append(Phi, [B|Psi], T),
4280     append(Phi, Psi, T1),
4281 1,
4282     assertz(proved(Z = [th(Z1,A), G => T])),
4283     assertz(proved(Z1 = [th(Z2,B), G1 => T1])),
4284     retract(proved(Z1 = [th(Z2,A), G => T1])),
4285     retract(proved(Z = [th(Z1,B), G => T])),
4286 1.

4287 /* (R2,R10) - (D.7) */
4288 norm2(proved(Z = [ais(Z1,Z2,A & B), G => T])) :-
4289     proved(Z1 = [th(Z3,C), G => T1]),
4290     proved(Z2 = [th(Z4,C), G => T2]),
4291     proved(Z3 = [_, G1 => T1]),
4292     proved(Z4 = [_, G1 => T2]),
4293     append(Gamma, [C|Delta], G),
4294     append(Gamma, Delta, G1),
4295     append(Phi, [A & B|Psi], T),
4296     append(Phi, [A|Psi], T1),
4297     append(Phi, [B|Psi], T2),
4298 1,
4299     assertz(proved(Z = [th(Z1,C), G => T])),
4300     assertz(proved(Z1 = [ais(Z3,Z4,A & B), G1 => T])),
4301     retract(proved(Z1 = [th(Z3,C), G => T1])),
4302     retract(proved(Z2 = [th(Z4,C), G => T2])),
4303     retract(proved(Z = [ais(Z1,Z2,A & B), G => T])),
4304 1.

4305 /* (R2,R11) - (D.8.1,2) */
4306 norm2(proved(Z = [aia(Z1,A & B), G => T])) :-
4307     proved(Z1 = [th(Z2,C), G1 => T]),
4308     proved(Z2 = [_, G2 => T]),
4309     ((append(S, [A & B|Lambda], G),
4310     append(S, [A|[B|Lambda]], G1),

```

```

4311     append(S1, [A|[B|Lambda]], G2),
4312     append(Gamma, [C|Delta], S),
4313     append(Gamma, Delta, S1),
4314     append(S1, [A & B|Lambda], G3))
4315 : (append(Gamma, [A & B|E], G),
4316     append(Gamma, [A|[B|E]], G1),
4317     append(Gamma, [A|[B|E1]], G2),
4318     append(Delta, [C|Lambda], E),
4319     append(Delta, Lambda, E1),
4320     append(Gamma, [A & B|E1], G3))),
4321 1.
4322     assertz(proved(Z = [th(Z1,C), G => T])),
4323     assertz(proved(Z1 = [oia(Z2,A & B), G3 => T])),
4324     retract(proved(Z1 = [th(Z2,C), G1 => T])),
4325     retract(proved(Z = [oia(Z1,A & B), G => T])).
4326 1.

4327 /* (R2,R12) - (D.9.1,2) */
4328 norm2(proved(Z = [oia(Z1,Z2,A # B), G => T])) :-
4329     proved(Z1 = [th(Z3,C), G1 => T]),
4330     proved(Z2 = [th(Z4,C), G2 => T]),
4331     proved(Z3 = [-, G3 => T]),
4332     proved(Z4 = [-, G4 => T]),
4333     ((append(S, [A # B|Lambda], G),
4334        append(S, [A|Lambda], G1),
4335        append(S, [B|Lambda], G2),
4336        append(S1, [A|Lambda], G3),
4337        append(S1, [B|Lambda], G4),
4338        append(Gamma, [C|Delta], S),
4339        append(Gamma, Delta, S1),
4340        append(S1, [A # B|Lambda], G5))
4341 : (append(Gamma, [A # B|E], G),
4342     append(Gamma, [A|E], G1),
4343     append(Gamma, [B|E], G2),
4344     append(Gamma, [A|E1], G3),
4345     append(Gamma, [B|E1], G4),
4346     append(Delta, [C|Lambda], E),
4347     append(Delta, Lambda, E1),
4348     append(Gamma, [A # B|E1], G5))),
4349 1.
4350     assertz(proved(Z = [th(Z1,C), G => T])),
4351     assertz(proved(Z1 = [oia(Z3,Z4,A # B), G5 => T])),
4352     retract(proved(Z1 = [th(Z3,C), G1 => T])),
4353     retract(proved(Z2 = [th(Z4,C), G2 => T])),
4354     retract(proved(Z = [oia(Z1,Z2,A # B), G => T])).
4355 1.

4356 /* (R2,R13) - (D.10) */
4357 norm2(proved(Z = [ois(Z1,A # B), G => T])) :-
4358     proved(Z1 = [th(Z2,C), G => T]),
4359     proved(Z2 = [-, G1 => T]),
4360 1.
4361     assertz(proved(Z1 = [ois(Z2,A # B), G1 => T])),
4362     assertz(proved(Z = [th(Z1,C), G => T])),
4363     retract(proved(Z1 = [th(Z2,C), G => T])),
4364     retract(proved(Z = [ois(Z1,A # B), G => T])).
4365 1.

4366 /* (R2,R14) - (D.11.1) */
4367 norm2(proved(Z = [hia(Z1,Z2,A -> B), G => T])) :-
4368     proved(Z1 = [th(Z3,C), G1 => [A]]),
4369     proved(Z2 = [-, G2 => T]),
4370     proved(Z3 = [-, G3 => [A]]),
4371     append(Gamma, [C|Delta], G1),
4372     append(Gamma, Delta, G3),
4373     append(Lambda, [B|Zeta], G2),
4374     append3(Lambda, G1, [A -> B|Zeta], G),
4375 1.
4376     assertz(proved(Z = [th(Z1,C), G => T])),
4377     append3(Lambda, G3, [A -> B|Zeta], G4),
4378     assertz(proved(Z1 = [hia(Z3,Z2,A -> B), G4 => T])),
4379     retract(proved(Z1 = [th(Z3,C), G1 => [A]])),
4380     retract(proved(Z = [hia(Z1,Z2,A -> B), G => T])).
4381 1.

4382 /* (R2,R14) - (D.11.2,3) */
4383 norm2(proved(Z = [hia(Z1,Z2,A -> B), G => T])) :-
4384     proved(Z1 = [-, G1 => [A]]),
4385     proved(Z2 = [th(Z3,C), G2 => T]),

```

```

4386     proved(Z3 = [_, G3 => T])).
4387     ((append(Delta, [C|E], G2),
4388      append(Delta, E, G3),
4389      append(Lambda, [B|Zeta], E),
4390      append4(Delta, [C|Lambda], G1, [A -> B|Zeta], G),
4391      append4(Delta, Lambda, G1, [A -> B|Zeta], G4))
4392     ;(append(S, [C|Zeta], G2),
4393      append(S, Zeta, G3),
4394      append(Delta, [B|Lambda], S),
4395      append4(Delta, G1, [A -> B|Lambda], [C|Zeta], G),
4396      append4(Delta, G1, [A -> B|Lambda], Zeta, G4))).
4397     |
4398     assertz(proved(Z = [th(Z2,C), G => T])).
4399     assertz(proved(Z2 = [hia(Z1,Z3,A -> B), G4 => T])).
4400     retract(proved(Z2 = [th(Z3,C), G2 => T])).
4401     retract(proved(Z = [hia(Z1,Z2,A -> B), G => T])).
4402     |

```

```

4403 /* (R2,R14) - (D.11.4) */
4404 norm2(proved(Z = [hia(Z1,Z2,A -> B), G => T])) :-
4405     proved(Z1 = [K, Gamma => [C]]),
4406     proved(Z2 = [th(Z3,B), G1 => T]),
4407     proved(Z3 = [_, G2 => T]),
4408     append(Delta, [B|Lambda], G1),
4409     append(Delta, Lambda, G2),
4410     append3(Delta, Gamma, [A -> B|Lambda], G),
4411     |
4412     assertz(proved(Z = [th(Z1,Gamma), G => T])),
4413     append(Delta, [A -> B|Lambda], G3),
4414     assertz(proved(Z1 = [th(Z3,A -> B), G3 => T])),
4415     retract(proved(Z1 = [K, Gamma => [C]])),
4416     retract(proved(Z2 = [th(Z3,B), G1 => T])),
4417     retract(proved(Z = [hia(Z1,Z2,A -> B), G => T])).
4418     |

```

```

4419 /* (R2,R15) - (D.12.1.2) */
4420 norm2(proved(Z = [hia(Z1, A -> B), G => [A -> B]])) :-
4421     proved(Z1 = [th(Z2,C), G1 => [B]]),
4422     proved(Z2 = [_, G2 => [B]]),
4423     ((append(S, [C|Lambda], G),
4424      append(S1, [C|Lambda], G1),
4425      append(S1, Lambda, G2),
4426      append(Gamma, [A|Delta], S1),
4427      append(Gamma, Delta, S),
4428      append(S, Lambda, G3))
4429     ;(append(Gamma, [C|E], G),
4430      append(Gamma, [C|E1], G1),
4431      append(Gamma, E1, G2),
4432      append(Delta, [A|Lambda], E1),
4433      append(Delta, Lambda, E),
4434      append(Gamma, E, G3))),
4435     |
4436     assertz(proved(Z = [th(Z1,C), G => [A -> B]])),
4437     assertz(proved(Z1 = [hia(Z2,A->B), G3 => [A->B]])),
4438     retract(proved(Z1 = [th(Z2,C), G1 => [B]])),
4439     retract(proved(Z = [hia(Z1,A->B), G => [A->B]])),
4440     |

```

```

4441 /* (R3,R3) - (D.15.1) */
4442 norm2(proved(Z = [con(Z1,B), G => T])) :-
4443     proved(Z1 = [con(Z2,A), G1 => T]),
4444     proved(Z2 = [_, G2 => T]),
4445     append(E, [B|Lambda], G),
4446     append(E, [B|[B|Lambda]], G1),
4447     append(E1, [B|[B|Lambda]], G2),
4448     append(Gamma, [A|Delta], E),
4449     append(Gamma, [A|[A|Delta]], E1),
4450     |
4451     assertz(proved(Z = [con(Z1,A), G => T])),
4452     append(E1, [B|Lambda], G3),
4453     assertz(proved(Z1 = [con(Z2,B), G3 => T])),
4454     retract(proved(Z1 = [con(Z2,A), G1 => T])),
4455     retract(proved(Z = [con(Z1,B), G => T])),
4456     |

```

```

4457 /* (R3,R4) - (D.16.1.2) */
4458 norm2(proved(Z = [con(Z1,A), G => T])) :-
4459     proved(Z1 = [inc(Z2,C,B), G1 => T]),
4460     proved(Z2 = [_, G2 => T]),

```

```

4461 ((append(S, [C|[B|Lambda]], G1),
4462 append(S, [B|[C|Lambda]], G2),
4463 append(Gamma, [A|[A|Delta]], S),
4464 append3(Gamma, [A|Delta], [C|[B|Lambda]], G),
4465 append3(Gamma, [A|Delta], [B|[C|Lambda]], G3))
4466 : (append(Gamma, [C|[B|E]], G1),
4467 append(Gamma, [B|[C|E]], G2),
4468 append(Delta, [A|[A|Lambda]], E),
4469 append3(Gamma, [C|[B|Delta]], [A|Lambda], G),
4470 append3(Gamma, [B|[C|Delta]], [A|Lambda], G3))),
4471 !,
4472 assertz(proved(Z = [inc(Z1,C,B), G => T])),
4473 assertz(proved(Z1 = [con(Z2,A), G3 => T])),
4474 retract(proved(Z1 = [inc(Z2,C,B), G1 => T])),
4475 retract(proved(Z = [con(Z1,A), G => T])),
4476 !.

```

```

4477 /* (R3,R4) - (D.16.3) */
4478 norm2(proved(Z = [con(Z1,A), G => T])) :-
4479 proved(Z1 = [inc(Z2,A,A), G1 => T]),
4480 proved(Z2 = [F, G1 => T]),
4481 append(Gamma, [A|Delta], G),
4482 append(Gamma, [A|[A|Delta]], G1),
4483 !,
4484 assertz(proved(Z = [con(Z2,A), G => T])),
4485 retract(proved(Z1 = [inc(Z2,A,A), G1 => T])),
4486 retract(proved(Z = [con(Z1,A), G => T])),
4487 !.

```

```

4488 /* (R3,R5) - (D.17) */
4489 norm2(proved(Z = [th(Z1,B), G => T])) :-
4490 proved(Z1 = [con(Z2,A), G => T]),
4491 proved(Z2 = [_, G1 => T]),
4492 append(Gamma, [A|[A|Delta]], G1),
4493 append(Gamma, [A|Delta], G),
4494 append(Phi, [B|Psi], T),
4495 append(Phi, Psi, T1),
4496 !,
4497 assertz(proved(Z = [con(Z1,A), G => T])),
4498 assertz(proved(Z1 = [th(Z2,B), G1 => T])),
4499 retract(proved(Z1 = [con(Z2,A), G => T])),
4500 retract(proved(Z = [th(Z1,B), G => T])),
4501 !.

```

```

4502 /* (R3,R10) - (D.18) */
4503 norm2(proved(Z = [ais(Z1,Z2,A & B), G => T])) :-
4504 proved(Z1 = [con(Z3,C), G => T]),
4505 proved(Z2 = [con(Z4,C), G => T]),
4506 proved(Z3 = [_, G1 => T]),
4507 proved(Z4 = [_, G1 => T]),
4508 !,
4509 assertz(proved(Z1 = [ais(Z3,Z4,A & B), G1 => T])),
4510 assertz(proved(Z = [con(Z1,C), G => T])),
4511 retract(proved(Z1 = [con(Z3,C), G => T])),
4512 retract(proved(Z2 = [con(Z4,C), G => T])),
4513 retract(proved(Z = [ais(Z1,Z2,A & B), G => T])),
4514 !.

```

```

4515 /* (R3,R11) - (D.19.3) */
4516 norm2(proved(Z = [con(Z1,A & A), G => T])) :-
4517 proved(Z1 = [aia(Z2,A & A), G1 => T]),
4518 proved(Z2 = [aia(Z3,A & A), G2 => T]),
4519 proved(Z3 = [_, G3 => T]),
4520 append(Gamma, [A & A|Delta], G),
4521 append(Gamma, [A & A|[A & A|Delta]], G1),
4522 append(Gamma, [A|[A|[A & A|Delta]]], G2),
4523 append(Gamma, [A|[A|[A|[A|Delta]]]]], G3),
4524 !,
4525 assertz(proved(Z = [aia(Z1,A & A), G => T])),
4526 append(Gamma, [A|[A|Delta]], G4),
4527 assertz(proved(Z1 = [con(Z2,A), G4 => T])),
4528 append(Gamma, [A|[A|[A|Delta]]], G5),
4529 assertz(proved(Z2 = [con(Z3,A), G5 => T])),
4530 retract(proved(Z = [con(Z1,A & A), G => T])),
4531 retract(proved(Z1 = [aia(Z2,A & A), G1 => T])),
4532 retract(proved(Z2 = [aia(Z3,A & A), G2 => T])),
4533 !.

```

```

4534 /* (R3,R11) - (D.19.1,2) */
4535 norm2(proved(Z = [con(Z1,A), G => T])) :-
4536   proved(Z1 = [aia(Z2,B & C), G1 => T]),
4537   proved(Z2 = [-, G2 => T]),
4538   ((append(S1, [B & C|Lambda], G1),
4539    append(S1, [B|[C|Lambda]], G2),
4540    append(Gamma, [A|[A|Delta]], S1),
4541    append3(Gamma, [A|Delta], [B & C|Lambda], G),
4542    append3(Gamma, [A|Delta], [A|[B|Lambda]], G3))
4543  ;(append(Gamma, [B & C|E1], G1),
4544   append(Gamma, [B|[C|E1]], G2),
4545   append(Delta, [A|[A|Lambda]], E1),
4546   append3(Gamma, [B & C|Delta], [A|Lambda], G),
4547   append3(Gamma, [B|[C|Delta]], [A|Lambda], G3))),
4548  1,
4549  assertz(proved(Z = [aia(Z1,B & C), G => T])),
4550  assertz(proved(Z1 = [con(Z2,A), G3 => T])),
4551  retract(proved(Z = [con(Z1,A), G => T])),
4552  retract(proved(Z1 = [aia(Z2,B & C), G1 => T])),
4553  1.

```

```

4554 /* (R3,R12) - (D.20.1,2) */
4555 norm2(proved(Z = [oia(Z1,Z2,A # B), G => T])) :-
4556   proved(Z1 = [con(Z3,C), G1 => T]),
4557   proved(Z2 = [con(Z4,C), G2 => T]),
4558   proved(Z3 = [-, G3 => T]),
4559   proved(Z4 = [-, G4 => T]),
4560   ((append(S, [A # B|Lambda], G),
4561    append(S, [A|Lambda], G1),
4562    append(S, [B|Lambda], G2),
4563    append(S1, [A|Lambda], G3),
4564    append(S1, [B|Lambda], G4),
4565    append(Gamma, [A|Delta], S),
4566    append(Gamma, [A|[A|Delta]], S1),
4567    append(S1, [A # B|Lambda], G5))
4568  ;(append(Gamma, [A # B|E], G),
4569   append(Gamma, [A|E], G1),
4570   append(Gamma, [B|E], G2),
4571   append(Gamma, [A|E1], G3),
4572   append(Gamma, [B|E1], G4),
4573   append(Delta, [C|Lambda], E),
4574   append(Delta, [C|[C|Lambda]], E1),
4575   append(Gamma, [A # B|E1], G5))),
4576  1,
4577  assertz(proved(Z = [con(Z1,C), G => T])),
4578  assertz(proved(Z1 = [oia(Z3,Z4,A # B), G5 => T])),
4579  retract(proved(Z = [oia(Z1,Z2,A # B), G => T])),
4580  retract(proved(Z1 = [con(Z3,C), G1 => T])),
4581  retract(proved(Z2 = [con(Z4,C), G2 => T])),
4582  1.

```

```

4583 /* (R3,R13) - (D.21) */
4584 norm2(proved(Z = [ois(Z1,A # B), G => T])) :-
4585   proved(Z1 = [con(Z2,C), G => T]),
4586   proved(Z2 = [-, G1 => T]),
4587  1,
4588  assertz(proved(Z = [con(Z1,C), G => T])),
4589  assertz(proved(Z1 = [ois(Z2,A # B), G1 => T])),
4590  retract(proved(Z = [ois(Z1,A # B), G => T])),
4591  retract(proved(Z1 = [con(Z2,C), G => T])),
4592  1.

```

```

4593 /* (R3,R4) - (D.22.1) */
4594 norm2(proved(Z = [hia(Z1,Z2,A -> B), G => T])) :-
4595   proved(Z1 = [con(Z3,C), G1 => [A]]),
4596   proved(Z2 = [-, G2 => T]),
4597   proved(Z3 = [-, G3 => [A]]),
4598   append(Gamma, [C|[C|Delta]], G3),
4599   append(Gamma, [C|Delta], G1),
4600   append(Lambda, [B|Zeta], G2),
4601   append3(Lambda, G1, [A -> B|Zeta], G),
4602  1,
4603  assertz(proved(Z = [con(Z1,C), G => T])),
4604  append3(Lambda, G3, [A -> B|Zeta], G4),
4605  assertz(proved(Z1 = [hia(Z3,Z2,A -> B), G4 => T])),
4606  retract(proved(Z = [hia(Z1,Z2,A -> B), G => T])),
4607  retract(proved(Z1 = [con(Z3,C), G1 => [A]])),
4608  1.

```

```

4609 /* (R3,R14) - (D.22.2,3) */
4610 norm2(proved(Z = [hia(Z1,Z2,A -> B), G => T])) :-
4611   proved(Z1 = [-, G1 => [A]]),
4612   proved(Z2 = [con(Z3,C), G2 => T]),
4613   proved(Z3 = [-, G3 => T]),
4614   ((append(Delta, [C|[C|E]], G3),
4615    append(Delta, [C|E], G2),
4616    append(Lambda, [B|Zeta], E),
4617    append4(Delta, [C|Lambda], G1, [A -> B|Zeta], G),
4618    append4(Delta, [C|[C|Lambda]], G1, [A -> B|Zeta], G4))
4619   ;(append(S, [C|[C|Zeta]], G3),
4620    append(S, [C|Zeta], G2),
4621    append(Delta, [B|Lambda], S),
4622    append4(Delta, G1, [A -> B|Lambda], [C|Zeta], G),
4623    append4(Delta, G1, [A->B|Lambda], [C|[C|Zeta]], G4))),
4624   !,
4625   assertz(proved(Z = [con(Z2,C), G => T])),
4626   assertz(proved(Z2 = [hia(Z1,Z3,A -> B), G4 => T])),
4627   retract(proved(Z = [hia(Z1,Z2,A -> B), G => T])),
4628   retract(proved(Z2 = [con(Z3,C), G2 => T])),
4629   !.

```

```

4630 /* (R3,R15) - (D.23.1,2) */
4631 norm2(proved(Z = [his(Z1,A -> B), G => [A -> B]])) :-
4632   proved(Z1 = [con(Z2,C), G1 => [B]]),
4633   proved(Z2 = [-, G2 => [B]]),
4634   ((append(Gamma, [C|[C|E1]], G2),
4635    append(Gamma, [C|E1], G1),
4636    append(Delta, [A|Lambda], E1),
4637    append3(Gamma, [C|Delta], Lambda, G),
4638    append3(Gamma, [C|[C|Delta]], Lambda, G3))
4639   ;(append(S1, [C|[C|Lambda]], G2),
4640    append(S1, [C|Lambda], G1),
4641    append(Gamma, [A|Delta], S1),
4642    append3(Gamma, Delta, [C|Lambda], G),
4643    append3(Gamma, Delta, [C|[C|Lambda]], G3))),
4644   !,
4645   assertz(proved(Z = [con(Z1,C), G => [A -> B]])),
4646   assertz(proved(Z1 = [his(Z2,A->B), G3 => [A->B]])),
4647   retract(proved(Z = [his(Z1,A->B), G => [A->B]])),
4648   retract(proved(Z1 = [con(Z3,C), G1 => [B]])),
4649   !.

```

```

4650 /* (R4,R5) - (D.25) */
4651 norm2(proved(Z = [th(Z1,C), G => T])) :-
4652   proved(Z1 = [inc(Z2,B,A), G => T1]),
4653   proved(Z2 = [-, G1 => T1]),
4654   append(Phi, [C|Psi], T),
4655   append(Phi, Psi, T1),
4656   append(Gamma, [B|[A|Delta]], G),
4657   append(Gamma, [A|[B|Delta]], G1),
4658   !,
4659   assertz(proved(Z = [inc(Z1,B,A), G => T])),
4660   assertz(proved(Z1 = [th(Z2,C), G1 => T])),
4661   retract(proved(Z = [th(Z1,C), G => T])),
4662   retract(proved(Z1 = [inc(Z2,B,A), G => T1])),
4663   !.

```

```

4664 /* (R4,R10) - (D.28) */
4665 norm2(proved(Z = [ais(Z1,Z2,A & B), G => T])) :-
4666   proved(Z1 = [inc(Z3,D,C), G => T1]),
4667   proved(Z2 = [inc(Z4,D,C), G => T2]),
4668   proved(Z3 = [-, G1 => T1]),
4669   proved(Z4 = [-, G1 => T2]),
4670   !,
4671   assertz(proved(Z = [inc(Z1,D,C), G => T])),
4672   assertz(proved(Z1 = [ais(Z3,Z4,A & B), G1 => T])),
4673   retract(proved(Z = [ais(Z1,Z2,A & B), G => T])),
4674   retract(proved(Z1 = [inc(Z3,D,C), G => T1])),
4675   retract(proved(Z2 = [inc(Z4,D,C), G => T2])),
4676   !.

```

```

4677 /* (R4,R11) - (D.29.1,2) */
4678 norm2(proved(Z = [inc(Z1,B,A), G => T])) :-
4679   proved(Z1 = [ala(Z2,C & D), G1 => T]),
4680   proved(Z2 = [-, G2 => T]),
4681   ((append(S, [C & D|Lambda], G1),
4682    append(S, [C|[D|Lambda]], G2),
4683    append(Gamma, [A|[B|Delta]], S),

```

```

4684      append3(Gamma, [B|[A|Delta]], [C & D|Lambda], G),
4685      append3(Gamma, [B|[A|Delta]], [C|[D|Lambda]], G3))
4686  :(append(Gamma, [C & D|E], G1),
4687     append(Gamma, [C|[D|E]], G2),
4688     append(Delta, [A|[B|Lambda]], E),
4689     append3(Gamma, [C & D|Delta], [B|[A|Lambda]], G),
4690     append3(Gamma, [C|[D|Delta]], [B|[A|Lambda]], G3)))
4691  |
4692     assertz(proved(Z = [aia(Z1,C & D), G => T])):
4693     assertz(proved(Z1 = [inc(Z2,B,A), G3 => T])):
4694     retract(proved(Z = [inc(Z1,B,A), G => T])):
4695     retract(proved(Z1 = [aia(Z2,C & D), G1 => T])).
4696  |

```

```

4697  /* (R4,R11) - (D.29.3) */
4698  norm2(proved(Z = [inc(Z1,C,A & B), G => T])) :-
4699     proved(Z1 = [aia(Z2,A & B), G1 => T]),
4700     proved(Z2 = [-, G2 => T]),
4701     append(Gamma, [C|[A & B|Delta]], G),
4702     append(Gamma, [A & B|[C|Delta]], G1),
4703     append(Gamma, [A|[B|[C|Delta]]], G2).
4704  |
4705     assertz(proved(Z = [aia(Z1,A & B), G => T])),
4706     gensym(f,Z3),
4707     append(Gamma, [C|[A|[B|Delta]]], G4),
4708     assertz(proved(Z1 = [inc(Z3,C,A), G4 => T])),
4709     append(Gamma, [A|[C|[B|Delta]]], G3),
4710     assertz(proved(Z3 = [inc(Z2,C,B), G3 => T])),
4711     retract(proved(Z = [inc(Z1,C,A & B), G => T])),
4712     retract(proved(Z1 = [aia(Z2,A & B), G1 => T])).
4713  |

```

```

4714  /* (R4,R11) - (D.29.4) */
4715  norm2(proved(Z = [inc(Z1,B & C,A), G => T])) :-
4716     proved(Z1 = [aia(Z2,B & C), G1 => T]),
4717     proved(Z2 = [-, G2 => T]),
4718     append(Gamma, [B & C|[A|Delta]], G),
4719     append(Gamma, [A|[B & C|Delta]], G1),
4720     append(Gamma, [A|[B|[C|Delta]]], G2).
4721  |
4722     assertz(proved(Z = [aia(Z1,B & C), G => T])),
4723     gensym(f1,Z3),
4724     append(Gamma, [B|[C|[A|Delta]]], G3),
4725     assertz(proved(Z1 = [inc(Z3,C,A), G3 => T])),
4726     append(Gamma, [B|[A|[C|Delta]]], G4),
4727     assertz(proved(Z3 = [inc(Z2,B,A), G4 => T])),
4728     retract(proved(Z = [inc(Z1,B & C,A), G => T])),
4729     retract(proved(Z1 = [aia(Z2,B & C), G1 => T])).
4730  |

```

```

4731  /* (R4,R12) - (D.30.1,2) */
4732  norm2(proved(Z = [oia(Z1,Z2,A # B), G => T])) :-
4733     proved(Z1 = [inc(Z3,D,C), G1 => T]),
4734     proved(Z2 = [inc(Z4,D,C), G2 => T]),
4735     proved(Z3 = [-, G3 => T]),
4736     proved(Z4 = [-, G4 => T]),
4737     ((append(S, [A # B|Lambda], G),
4738     append(S, [A|Lambda], G1),
4739     append(S, [B|Lambda], G2),
4740     append(S1, [A|Lambda], G3),
4741     append(S1, [B|Lambda], G4),
4742     append(Gamma, [D|[C|Delta]], S),
4743     append(Gamma, [C|[D|Delta]], S1),
4744     append(S1, [A # B|Lambda], G5))
4745     :(append(Gamma, [A # B|E], G),
4746     append(Gamma, [A|E], G1),
4747     append(Gamma, [B|E], G2),
4748     append(Gamma, [A|E1], G3),
4749     append(Gamma, [B|E1], G4),
4750     append(Delta, [D|[C|Lambda]], E),
4751     append(Delta, [C|[D|Lambda]], E1),
4752     append(Gamma, [A # B|E1], G5))).
4753  |
4754     assertz(proved(Z = [inc(Z1,D,C), G => T])),
4755     assertz(proved(Z1 = [oia(Z3,Z4,A # B), G5 => T])),
4756     retract(proved(Z = [oia(Z1,Z2,A # B), G => T])),
4757     retract(proved(Z1 = [inc(Z3,D,C), G1 => T])),
4758     retract(proved(Z2 = [inc(Z4,D,C), G2 => T])).
4759  |

```

```

4760 /* (R4,R13) - (D.31) */
4761 norm2(proved(Z = [ois(Z1,A # B), G => T])) :-
4762   proved(Z1 = [inc(Z2,D,C), G1 => T1]),
4763   proved(Z2 = [-, G1 => T1]),
4764   append(Gamma, [D|[C|Delta]], G),
4765   append(Gamma, [C|[D|Delta]], G1),
4766   append(Phi, [A # B|Psi], T),
4767   append(Phi, [A|[B|Psi]], T1),
4768   !,
4769   assertz(proved(Z = [inc(Z1,D,C), G => T])),
4770   assertz(proved(Z1 = [ois(Z2,A # B), G1 => T])),
4771   retract(proved(Z = [ois(Z1,A # B), G => T])),
4772   retract(proved(Z1 = [inc(Z2,D,C), G => T])),
4773   !.

4774 /* (R4,R14) - (D.32.1) */
4775 norm2(proved(Z = [hia(Z1,Z2,A -> B), G => T])) :-
4776   proved(Z1 = [inc(Z3,D,C), G1 => [A]]),
4777   proved(Z2 = [-, G2 => T]),
4778   proved(Z3 = [-, G3 => [A]]),
4779   append(S, [A -> B|Zeta], G),
4780   append(Lambda, [B|Zeta], G2),
4781   append(Lambda, G1, S),
4782   append(Gamma, [D|[C|Delta]], G1),
4783   append(Gamma, [C|[D|Delta]], G3),
4784   !,
4785   assertz(proved(Z = [inc(Z1,D,C), G => T])),
4786   append(Lambda, G3, S1),
4787   append(S1, [A -> B|Zeta], G4),
4788   assertz(proved(Z1 = [hia(Z3,Z2,A -> B), G4 => T])),
4789   retract(proved(Z = [hia(Z1,Z2,A -> B), G => T])),
4790   retract(proved(Z1 = [inc(Z3,D,C), G1 => [A]])),
4791   !.

4792 /* (R4,R14) - (D.32.2,3) */
4793 norm2(proved(Z = [hia(Z1,Z2,A -> B), G => T])) :-
4794   proved(Z1 = [-, G1 => [A]]),
4795   proved(Z2 = [inc(Z3,D,C), G2 => T]),
4796   proved(Z3 = [-, G3 => T]),
4797   ((append(S, [A -> B|Zeta], G),
4798    append(S1, [B|Zeta], G2),
4799    append(S2, [B|Zeta], G3),
4800    append(S1, G1, S),
4801    append(Delta, [D|[C|Lambda]], S1),
4802    append(Delta, [C|[D|Lambda]], S2),
4803    append(S2, G1, S3),
4804    append(S3, [A -> B|Zeta], G4))
4805   : (append(S, [A -> B|E], G),
4806      append(Delta, [B|E], G2),
4807      append(Delta, [B|E1], G3),
4808      append(Delta, G1, S),
4809      append(Lambda, [D|[C|Zeta]], E),
4810      append(Lambda, [C|[D|Zeta]], E1),
4811      append(S, [A -> B|E1], G4))),
4812   !,
4813   assertz(proved(Z = [inc(Z2,D,C), G => T])),
4814   assertz(proved(Z2 = [hia(Z1,Z3,A -> B), G4 => T])),
4815   retract(proved(Z = [hia(Z1,Z2,A -> B), G => T])),
4816   retract(proved(Z2 = [inc(Z3,D,C), G2 => T])),
4817   !.

4818 /* (R4,R14) - (D.32.4) */
4819 norm2(proved(Z = [hia(Z1,Z2,A -> B), G => T])) :-
4820   proved(Z1 = [-, G1 => [A]]),
4821   proved(Z2 = [inc(Z3,B,C), G2 => T]),
4822   proved(Z3 = [-, G3 => T]),
4823   append(Delta, [C|[B|Lambda]], G3),
4824   append(Delta, [B|[C|Lambda]], G2),
4825   append(Delta, G1, S),
4826   append(S, [A -> B|[C|Lambda]], G),
4827   !,
4828   assertz(proved(Z = [inc(Z2,A -> B,C), G => T])),
4829   append(S, [C|[A -> B|Lambda]], G4),
4830   gensym(f,F1),
4831   assertz(proved(Z2 = [inc(F1,G1,C), G4 => T])),
4832   append(Delta, [C|G1], S1),
4833   append(S1, [A -> B|Lambda], G5),
4834   assertz(proved(F1 = [hia(Z1,Z3,A -> B), G5 => T])),
4835   retract(proved(Z = [hia(Z1,Z2,A -> B), G => T])),
4836   retract(proved(Z2 = [inc(Z3,B,C), G2 => T])),
4837   !.

```



```

4838 /* (R4,R14) - (D.32.5) */
4839 norm2(proved(Z = [hia(Z1,Z2,A -> B), G => T])) :-
4840   proved(Z1 = [-, G1 => [A]]),
4841   proved(Z2 = [inc(Z3,C,B), G2 => T]),
4842   proved(Z3 = [-, G3 => T]),
4843   append(Delta, [B|[C|Lambda]], G3),
4844   append(Delta, [C|[B|Lambda]], G2),
4845   append(Delta, [C|G1], S),
4846   append(S, [A -> B|Lambda], G),
4847   !,
4848   assertz(proved(Z = [inc(Z2,C,G1), G => T])),
4849   append(Delta, G1, S1),
4850   append(S1, [C|[A -> B|Lambda]], G4),
4851   gensym(f,F1),
4852   assertz(proved(Z2 = [inc(F1,C,A -> B), G4 => T])),
4853   append(S1, [A -> B|[C|Lambda]], G5),
4854   assertz(proved(F1 = [hia(Z1,Z3,A -> B), G5 => T])),
4855   retract(proved(Z = [hia(Z1,Z2,A -> B), G => T])),
4856   retract(proved(Z2 = [inc(Z3,C,B), G2 => T])),
4857   !.

```

```

4858 /* (R4,R15) - (D.33.1,2) */
4859 norm2(proved(Z = [hia(Z1,A -> B), G => [A -> B]])) :-
4860   proved(Z1 = [inc(Z2,D,C), G1 => [B]]),
4861   proved(Z2 = [-, G2 => [B]]),
4862   ((append(S, [D|[C|Lambda]], G),
4863     append(S1, [D|[C|Lambda]], G1),
4864     append(S1, [C|[D|Lambda]], G2),
4865     append(Gamma, [A|Delta], S1),
4866     append(Gamma, Delta, S),
4867     append(S, [C|[D|Lambda]], G3))
4868   ;(append(Gamma, [D|[C|E]], G),
4869     append(Gamma, [D|[C|E1]], G1),
4870     append(Gamma, [C|[D|E1]], G2),
4871     append(Delta, [A|Lambda], E1),
4872     append(Delta, Lambda, E),
4873     append(Gamma, [C|[D|E]], G3))),
4874   !,
4875   assertz(proved(Z = [inc(Z1,D,C), G => [A -> B]])),
4876   assertz(proved(Z1 = [hia(Z2,A->B), G3 => [A->B]])),
4877   retract(proved(Z = [hia(Z1,A->B), G => [A->B]])),
4878   retract(proved(Z1 = [inc(Z2,D,C), G1 => [B]])),
4879   !.

```

```

4880 /* (R4,R15) - (D.33.3,4) */
4881 norm2(proved(Z = [hia(Z1,A -> B), G => [A -> B]])) :-
4882   ((proved(Z1 = [inc(Z2,A,C), G1 => [B]]),
4883     proved(Z2 = [-, G2 => [B]]),
4884     append(Gamma, [C|[A|Delta]], G2),
4885     append(Gamma, [A|[C|Delta]], G1)
4886   ;(proved(Z1 = [inc(Z2,C,A), G1 => [B]]),
4887     proved(Z2 = [-, G2 => [B]]),
4888     append(Gamma, [A|[C|Delta]], G2),
4889     append(Gamma, [C|[A|Delta]], G1))),
4890   append(Gamma, [C|Delta], G),
4891   !,
4892   assertz(proved(Z = [hia(Z2,A -> B), G => [A->B]])),
4893   retract(proved(Z = [hia(Z1,A -> B), G => [A->B]])),
4894   retract(proved(Z1 = [inc(Z2,A,C), G1 => [B]])),
4895   !.

```

```

4896 /* (R5,R5) - (D.34) */
4897 norm2(proved(Z = [th(Z1,B), G => T])) :-
4898   proved(Z1 = [th(Z2,A), G => T1]),
4899   proved(Z2 = [-, G => T2]),
4900   append(Phi, [A|E], T1),
4901   append(Phi, E, T2),
4902   append(Psi, Theta, E),
4903   append3(Phi, [A|Psi], [B|Theta], T),
4904   !,
4905   assertz(proved(Z = [th(Z1,A), G => T])),
4906   append3(Phi, Psi, [B|Theta], T3),
4907   assertz(proved(Z1 = [th(Z2,B), G => T3])),
4908   retract(proved(Z = [th(Z1,B), G => T])),
4909   retract(proved(Z1 = [th(Z2,A), G => T1])),
4910   !.

```

```

4911 /* (R5,R10) - (D.35.1,2) */
4912 norm2(proved(Z = [th(Z1,C), G => T])) :-
4913   proved(Z1 = [ais(Z2,Z3,A & B), G => T1]),
4914   proved(Z2 = [-, G => T2]),
4915   proved(Z3 = [-, G => T3]),
4916   ((append(S1, [A & B|Theta], T1),
4917    append(S1, [A|Theta], T2),
4918    append(S1, [B|Theta], T3),
4919    append(S, [A & B|Theta], T),
4920    append(Phi, [C|Psi], S),
4921    append(Phi, Psi, S1),
4922    append(S, [A|Theta], T4),
4923    append(S, [B|Theta], T5))
4924   ;(append(Phi, [A & B|E1], T1),
4925    append(Phi, [A|E1], T2),
4926    append(Phi, [B|E1], T3),
4927    append(Phi, [A & B|E], T),
4928    append(Psi, [C|Theta], E),
4929    append(Psi, Theta, E1),
4930    append(Phi, [A|E], T4),
4931    append(Phi, [B|E], T5))),
4932   !,
4933   gonsym(f,F1),
4934   assertz(proved(Z = [ais(Z1,F1,A & B), G => T])),
4935   assertz(proved(Z1 = [th(Z2,C), G => T4])),
4936   assertz(proved(F1 = [th(Z3,C), G => T5])),
4937   retract(proved(Z = [th(Z1,C), G => T])),
4938   retract(proved(Z1 = [ais(Z2,Z3,A & B), G => T1])),
4939   !.

```

```

4940 /* (R5,R10) - (D.35.3,4) */
4941 norm2(proved(Z = [ais(Z1,Z2,B & C), G => T])) :-
4942   proved(Z1 = [th(Z3,B), G => T1]),
4943   proved(Z2 = [th(Z4,A), G => T2]),
4944   proved(Z3 = [K1, G => T3]),
4945   proved(Z4 = [K2, G => T4]),
4946   ((append(Phi, [A], T3), append(Phi, [C], T4),
4947    append(Phi, [A,B], T1), append(Phi, [A,C], T2),
4948    append(Phi, [A,B & C], T))
4949   ;(append([A], Psi, T3), append([C], Psi, T4),
4950    append([B,A], Psi, T1), append([C,A], Psi, T2),
4951    append([B & C,A], Psi, T))),
4952   !,
4953   assertz(proved(Z = [ais(Z1,Z2,B & C), G => T])),
4954   assertz(proved(Z2 = [th(Z4,C), G => T2])),
4955   assertz(proved(Z4 = [K1, G => T3])),
4956   retract(proved(Z = [ais(Z1,Z2,B & C), G => T])),
4957   retract(proved(Z2 = [th(Z4,A), G => T2])),
4958   retract(proved(Z4 = [K2, G => T4])),
4959   !.

```

```

4960 /* (R5,R10) - (D.35.5) */
4961 norm2(proved(Z = [ais(Z1,Z2,B & C), G => T])) :-
4962   proved(Z1 = [th(Z3,B), G => T1]),
4963   proved(Z2 = [th(Z4,C), G => T2]),
4964   proved(Z3 = [th(Z5,D), G => T3]),
4965   proved(Z4 = [th(Z6,A), G => T3]),
4966   proved(Z5 = [K1, G => T4]),
4967   proved(Z6 = [K2, G => T5]),
4968   append(Phi, [A|[B & C|[D|Psi]]], T),
4969   append(Phi, [A|[B|[D|Psi]]], T1),
4970   append(Phi, [A|[C|[D|Psi]]], T2),
4971   append(Phi, [A|[D|Psi]], T3),
4972   append(Phi, [A|Psi], T4),
4973   append(Phi, [D|Psi], T5),
4974   !,
4975   assertz(proved(Z = [ais(Z1,Z2,B & C), G => T])),
4976   assertz(proved(Z4 = [th(Z6,D), G => T3])),
4977   assertz(proved(Z6 = [K1, G => T4])),
4978   retract(proved(Z = [ais(Z1,Z2,B & C), G => T])),
4979   retract(proved(Z4 = [th(Z6,A), G => T3])),
4980   retract(proved(Z6 = [K2, G => T5])),
4981   !.

```

```

4982 /* (R5,R10) - (D.35.6) */
4983 norm2(proved(Z = [ais(Z1,Z2,B & C), G => T])) :-
4984   proved(Z1 = [th(Z3,B), G => T1]),
4985   proved(Z2 = [th(Z4,A), G => T2]),
4986   proved(Z3 = [th(Z5,D), G => T3]),
4987   proved(Z4 = [th(Z6,D), G => T4]),

```

```

4988 proved(Z5 = [K1, G => T5]),
4989 proved(Z6 = [K2, G => T6]),
4990 append(Phi, [A [B & C [D Psi]]], T),
4991 append(Phi, [A [B [D Psi]]], T1),
4992 append(Phi, [A [C [D Psi]]], T2),
4993 append(Phi, [A [D Psi]]], T3),
4994 append(Phi, [C [D Psi]]], T4),
4995 append(Phi, [A Psi], T5),
4996 append(Phi, [C Psi], T6),
4997
4998 assertz(proved(Z = [ais(Z1,Z2,B & C), G => T])),
4999 assertz(proved(Z2 = [th(Z4,C), G => T2])),
5000 assertz(proved(Z4 = [th(Z6,D), G => T3])),
5001 assertz(proved(Z6 = [K1, G => T5])),
5002 retract(proved(Z = [ais(Z1,Z2,B & C), G => T])),
5003 retract(proved(Z2 = [th(Z4,A), G => T2])),
5004 retract(proved(Z4 = [th(Z6,D), G => T4])),
5005 retract(proved(Z6 = [K2, G => T6])),
5006

```

```

5007 /* (R5,R10) - (D.35.7) */
5008 norm2(proved(Z = [ais(Z1,Z2,B & C), G => T])) :-
5009 proved(Z1 = [th(Z3,B), G => T1]),
5010 proved(Z2 = [th(Z4,A), G => T2]),
5011 proved(Z3 = [th(Z5,D), G => T3]),
5012 proved(Z4 = [th(Z6,C), G => T4]),
5013 proved(Z5 = [K1, G => T5]),
5014 proved(Z6 = [K2, G => T6]),
5015 append(Phi, [A [B & C [D Psi]]], T),
5016 append(Phi, [A [B [D Psi]]], T1),
5017 append(Phi, [A [C [D Psi]]], T2),
5018 append(Phi, [A [D Psi]]], T3),
5019 append(Phi, [C [D Psi]]], T4),
5020 append(Phi, [A Psi], T5),
5021 append(Phi, [D Psi], T6),
5022
5023 assertz(proved(Z = [ais(Z1,Z2,B & C), G => T])),
5024 assertz(proved(Z1 = [th(Z3,A), G => T1])),
5025 append(Phi, [B [D Psi]]], T7),
5026 assertz(proved(Z3 = [th(Z5,B), G => T7])),
5027 assertz(proved(Z5 = [K2, G => T6])),
5028 retract(proved(Z = [ais(Z1,Z2,B & C), G => T])),
5029 retract(proved(Z1 = [th(Z3,B), G => T1])),
5030 retract(proved(Z3 = [th(Z5,D), G => T3])),
5031 retract(proved(Z5 = [K1, G => T5])),
5032

```

```

5033 /* (R5,R10) - (D.35.8) */
5034 norm2(proved(Z = [ais(Z1,Z2,B & C), G => T])) :-
5035 proved(Z1 = [th(Z3,A), G => T1]),
5036 proved(Z2 = [th(Z4,C), G => T2]),
5037 proved(Z3 = [th(Z5,D), G => T3]),
5038 proved(Z4 = [th(Z6,D), G => T4]),
5039 proved(Z5 = [K1, G => T5]),
5040 proved(Z6 = [K2, G => T6]),
5041 append(Phi, [A [B & C [D Psi]]], T),
5042 append(Phi, [A [B [D Psi]]], T1),
5043 append(Phi, [A [C [D Psi]]], T2),
5044 append(Phi, [B [D Psi]]], T3),
5045 append(Phi, [A [D Psi]]], T4),
5046 append(Phi, [B Psi], T5),
5047 append(Phi, [A Psi], T6),
5048
5049 assertz(proved(Z = [ais(Z1,Z2,B & C), G => T])),
5050 assertz(proved(Z1 = [th(Z3,B), G => T1])),
5051 assertz(proved(Z3 = [th(Z5,D), G => T4])),
5052 assertz(proved(Z5 = [K2, G => T6])),
5053 retract(proved(Z = [ais(Z1,Z2,B & C), G => T])),
5054 retract(proved(Z1 = [th(Z3,A), G => T1])),
5055 retract(proved(Z3 = [th(Z5,D), G => T3])),
5056 retract(proved(Z5 = [K1, G => T5])),
5057

```

```

5058 /* (R5,R10) - (D.35.9) */
5059 norm2(proved(Z = [ais(Z1,Z2,B & C), G => T])) :-
5060 proved(Z1 = [th(Z3,A), G => T1]),
5061 proved(Z2 = [th(Z4,A), G => T2]),
5062 proved(Z3 = [th(Z5,D), G => T3]),
5063 proved(Z4 = [th(Z6,C), G => T4]),
5064 proved(Z5 = [K1, G => T5]),
5065 proved(Z6 = [K2, G => T6]),

```

```

5066      append(Phi, [A|[B & C|[D|Psi]]], T),
5067      append(Phi, [A|[B|[D|Psi]]], T1),
5068      append(Phi, [A|[C|[D|Psi]]], T2),
5069      append(Phi, [B|[D|Psi]]], T3),
5070      append(Phi, [C|[D|Psi]]], T4),
5071      append(Phi, [B|Psi], T5),
5072      append(Phi, [D|Psi], T6).
5073  1.
5074      assertz(proved(Z = [ais(Z1,Z2,B & C), G => T])),
5075      assertz(proved(Z3 = [th(Z5,B), G => T3])),
5076      assertz(proved(Z5 = [K2, G => T6])),
5077      retract(proved(Z = [ais(Z1,Z2,B & C), G => T])),
5078      retract(proved(Z3 = [th(Z5,D), G => T3])),
5079      retract(proved(Z5 = [K1, G => T5])),
5080  1.

```

```

5081 /* (R5,R10) -- (D.35.10) */
5082 norm2(proved(Z = [ais(Z1,Z2,B & C), G => T])) :-
5083     proved(Z1 = [th(Z3,A), G => T1]),
5084     proved(Z2 = [th(Z4,C), G => T2]),
5085     proved(Z3 = [th(Z5,B), G => T3]),
5086     proved(Z4 = [th(Z6,D), G => T4]),
5087     proved(Z5 = [K1, G => T5]),
5088     proved(Z6 = [K2, G => T6]),
5089     append(Phi, [A|[B & C|[D|Psi]]], T),
5090     append(Phi, [A|[B|[D|Psi]]], T1),
5091     append(Phi, [A|[C|[D|Psi]]], T2),
5092     append(Phi, [B|[D|Psi]]], T3),
5093     append(Phi, [A|[D|Psi]]], T4),
5094     append(Phi, [D|Psi], T5),
5095     append(Phi, [A|Psi], T6).
5096  1.
5097      assertz(proved(Z = [ais(Z1,Z2,B & C), G => T])),
5098      assertz(proved(Z2 = [th(Z4,A), G => T2])),
5099      append(Phi, [C|[D|Psi]]], T7),
5100      assertz(proved(Z4 = [th(Z6,C), G => T7])),
5101      assertz(proved(Z6 = [K1, G => T5])),
5102      retract(proved(Z = [ais(Z1,Z2,B & C), G => T])),
5103      retract(proved(Z2 = [th(Z4,C), G => T2])),
5104      retract(proved(Z4 = [th(Z6,D), G => T4])),
5105      retract(proved(Z6 = [K2, G => T6])),
5106  1.

```

```

5107 /* (R5,R10) -- (D.35.11) */
5108 norm2(proved(Z = [ais(Z1,Z2,B & C), G => T])) :-
5109     proved(Z1 = [th(Z3,A), G => T1]),
5110     proved(Z2 = [th(Z4,A), G => T2]),
5111     proved(Z3 = [th(Z5,B), G => T3]),
5112     proved(Z4 = [th(Z6,D), G => T4]),
5113     proved(Z5 = [K1, G => T5]),
5114     proved(Z6 = [K2, G => T6]),
5115     append(Phi, [A|[B & C|[D|Psi]]], T),
5116     append(Phi, [A|[B|[D|Psi]]], T1),
5117     append(Phi, [A|[C|[D|Psi]]], T2),
5118     append(Phi, [B|[D|Psi]]], T3),
5119     append(Phi, [C|[D|Psi]]], T4),
5120     append(Phi, [D|Psi], T5),
5121     append(Phi, [C|Psi], T6).
5122  1.
5123      assertz(proved(Z = [ais(Z1,Z2,B & C), G => T])),
5124      assertz(proved(Z4 = [th(Z6,C), G => T4])),
5125      assertz(proved(Z6 = [K1, G => T5])),
5126      retract(proved(Z = [ais(Z1,Z2,B & C), G => T])),
5127      retract(proved(Z4 = [th(Z6,D), G => T4])),
5128      retract(proved(Z6 = [K2, G => T6])),
5129  1.

```

```

5130 /* (R5,R11) -- (D.36) */
5131 norm2(proved(Z = [th(Z1,C), G => T])) :-
5132     proved(Z1 = [ais(Z2,A & B), G => T1]),
5133     proved(Z2 = [-, G1 => T1]),
5134     append(Gamma, [A & B|Delta], G),
5135     append(Gamma, [A|[B|Delta]], G1),
5136     append(Phi, [C|Psi], T),
5137     append(Phi, Psi, T1),
5138  1.
5139      assertz(proved(Z = [ais(Z1,A & B), G => T])),
5140      assertz(proved(Z1 = [th(Z2,C), G1 => T1])),
5141      retract(proved(Z = [th(Z1,C), G => T])),
5142      retract(proved(Z1 = [ais(Z2,A & B), G => T1])),
5143  1.

```

```

5144 /* (R5,R12) - (D.37) */
5145 norm2(proved(Z = [th(Z1,C), G => T])) :-
5146   proved(Z1 = [ola(Z2,Z3,A # B), G => T1]),
5147   proved(Z2 = [-, G1 => T1]),
5148   proved(Z3 = [-, G2 => T1]),
5149   append(Gamma, [A # B|Delta], G),
5150   append(Gamma, [A|Delta], G1),
5151   append(Gamma, [B|Delta], G2),
5152   append(Phi, [C|Psi], T),
5153   append(Phi, Psi, T1),
5154   !,
5155   gensym(f, F1),
5156   assertz(proved(Z = [ola(Z1,F1,A # B), G => T])),
5157   assertz(proved(Z1 = [th(Z2,C), G1 => T])),
5158   assertz(proved(F1 = [th(Z3,C), G2 => T])),
5159   retract(proved(Z = [th(Z1,C), G => T])),
5160   retract(proved(Z1 = [ola(Z2,Z3,A # B), G => T1])),
5161   !.

```

```

5162 /* (R5,R13) - (D.38.1.2) */
5163 norm2(proved(Z = [th(Z1,C), G => T])) :-
5164   proved(Z1 = [ois(Z2,A # B), G => T1]),
5165   proved(Z2 = [-, G => T2]),
5166   ((append(S1, [A # B|Theta], T1),
5167    append(S1, [A|B|Theta], T2),
5168    append(S, [A # B|Theta], T),
5169    append(Phi, [C|Psi], S),
5170    append(Phi, Psi, S1),
5171    append(S, [A|B|Theta], T3))
5172   : (append(Phi, [A # B|E1], T1),
5173      append(Phi, [A|B|E1], T2),
5174      append(Phi, [A # B|E], T),
5175      append(Psi, [C|Theta], E),
5176      append(Psi, Theta, E1),
5177      append(Phi, [A|B|E], T3))),
5178   !,
5179   assertz(proved(Z = [ois(Z1,A # B), G => T])),
5180   assertz(proved(Z1 = [th(Z2,C), G => T3])),
5181   retract(proved(Z = [th(Z1,C), G => T])),
5182   retract(proved(Z1 = [ois(Z2,A # B), G => T1])),
5183   !.

```

```

5184 /* (R5,R14) - (D.39) */
5185 norm2(proved(Z = [th(Z1,C), G => T])) :-
5186   proved(Z1 = [hia(Z2,Z3,A -> B), G => T1]),
5187   proved(Z2 = [-, G1 => [A]]),
5188   proved(Z3 = [-, G2 => T1]),
5189   append(S, [A -> B|Lambda], G),
5190   append(Delta, [B|Lambda], G2),
5191   append(Phi, [C|Psi], T),
5192   append(Phi, Psi, T1),
5193   !,
5194   assertz(proved(Z = [hia(Z2,Z1,A -> B), G => T])),
5195   assertz(proved(Z1 = [th(Z3,C), G2 => T])),
5196   retract(proved(Z = [th(Z1,C), G => T])),
5197   retract(proved(Z1 = [hia(Z2,Z3,A -> B), G => T1])),
5198   !.

```

```

5199 /* (R10,R10) - (D.48) */
5200 norm2(proved(Z = [ais(Z1,Z2,A & A), G => T])) :-
5201   proved(Z1 = [ais(Z3,Z4,A & A), G => T1]),
5202   proved(Z2 = [ais(Z5,Z6,A & A), G => T1]),
5203   proved(Z3 = [-, G => T2]),
5204   proved(Z4 = [-, G => T2]),
5205   proved(Z5 = [-, G => T2]),
5206   proved(Z6 = [-, G => T2]),
5207   append(Phi, [A & A|E1], T1),
5208   append(Phi, [A|E1], T2),
5209   append(Phi, [A & A|E], T),
5210   append(Psi, [A & A|Theta], E),
5211   append(Psi, [A|Theta], E1),
5212   !,
5213   assertz(proved(Z = [ais(Z1,Z2,A & A), G => T])),
5214   append(Phi, [A|Psi], S1),
5215   append(S1, [A & A|Theta], T3),
5216   assertz(proved(Z1 = [ais(Z3,Z4,A & A), G => T3])),
5217   assertz(proved(Z2 = [ais(Z5,Z6,A & A), G => T3])),
5218   retract(proved(Z = [ais(Z1,Z2,A & A), G => T])),
5219   retract(proved(Z1 = [ais(Z3,Z4,A & A), G => T1])),
5220   retract(proved(Z2 = [ais(Z5,Z6,A & A), G => T1])),
5221   !.

```

```

5222 /* (R10,R11) - (D.49) */
5223 norm2(proved(Z = [als(Z1,Z2,C & D), G => T])) :-
5224   proved(Z1 = [als(Z3,Z4,A & B), G => T1]),
5225   proved(Z2 = [als(Z4,A & B), G => T2]),
5226   proved(Z3 = [-, G1 => T1]),
5227   proved(Z4 = [-, G1 => T2]),
5228   append(Gamma, [A & B|Delta], G),
5229   append(Gamma, [A|B|Delta], G1),
5230   append(Phi, [C & D|Psi], T),
5231   append(Phi, [C|Psi], T1),
5232   append(Phi, [D|Psi], T2),
5233   !,
5234   assertz(proved(Z = [als(Z1,A & B), G => T])),
5235   assertz(proved(Z1 = [als(Z3,Z4,C & D), G1 => T1])),
5236   retract(proved(Z = [als(Z1,Z2,C & D), G => T])),
5237   retract(proved(Z1 = [als(Z3,A & B), G => T1])),
5238   retract(proved(Z2 = [als(Z4,A & B), G => T2])),
5239   !.

```

```

5240 /* (R10,R12) - (D.50) */
5241 norm2(proved(Z = [als(Z1,Z2,C & D), G => T])) :-
5242   proved(Z1 = [oia(Z3,Z4,A # B), G => T1]),
5243   proved(Z2 = [oia(Z5,Z6,A # B), G => T2]),
5244   proved(Z3 = [-, G1 => T1]),
5245   proved(Z4 = [-, G2 => T1]),
5246   proved(Z5 = [-, G1 => T2]),
5247   proved(Z6 = [-, G2 => T2]),
5248   append(Gamma, [A # B|Delta], G),
5249   append(Gamma, [A|Delta], G1),
5250   append(Gamma, [B|Delta], G2),
5251   append(Phi, [C & D|Psi], T),
5252   append(Phi, [C|Psi], T1),
5253   append(Phi, [D|Psi], T2),
5254   !,
5255   assertz(proved(Z = [oia(Z1,Z2,A # B), G => T])),
5256   assertz(proved(Z1 = [als(Z3,Z5,C & D), G1 => T1])),
5257   assertz(proved(Z2 = [als(Z4,Z6,C & D), G2 => T2])),
5258   retract(proved(Z = [als(Z1,Z2,C & D), G => T])),
5259   retract(proved(Z1 = [oia(Z3,Z4,A # B), G => T1])),
5260   retract(proved(Z2 = [oia(Z5,Z6,A # B), G => T2])),
5261   !.

```

```

5262 /* (R10,R13) - (D.51.1,2) */
5263 norm2(proved(Z = [ois(Z1,C # D), G => T])) :-
5264   proved(Z1 = [als(Z2,Z3,A & B), G => T1]),
5265   proved(Z2 = [-, G => T2]),
5266   proved(Z3 = [-, G => T3]),
5267   ((append(Phi, [C # D|E], T),
5268     append(Phi, [C|D|E], T1),
5269     append(Phi, [C|D|E1], T2),
5270     append(Phi, [C|D|E2], T3),
5271     append(Phi, [C # D|E1], T4),
5272     append(Phi, [C # D|E2], T5))
5273   ;(append(S, [C # D|Theta], T),
5274     append(S, [C|D|Theta], T1),
5275     append(S1, [C|D|Theta], T2),
5276     append(S2, [C|D|Theta], T3),
5277     append(S1, [C # D|Theta], T4),
5278     append(S2, [C # D|Theta], T5))),
5279   !,
5280   gensym(f,F1),
5281   assertz(proved(Z = [als(Z1,F1,A & B), G => T])),
5282   assertz(proved(Z1 = [ois(Z2,C # D), G => T4])),
5283   assertz(proved(F1 = [ois(Z3,C # D), G => T5])),
5284   retract(proved(Z = [ois(Z1,C # D), G => T])),
5285   retract(proved(Z1 = [als(Z2,Z3,A & B), G => T1])),
5286   !.

```

```

5287 /* (R10,R14) - (D.52) */
5288 norm2(proved(Z = [als(Z1,Z2,C & D), G => T])) :-
5289   proved(Z1 = [hla(Z3,Z4,A -> B), G => T1]),
5290   proved(Z2 = [hla(Z5,Z6,A -> B), G => T2]),
5291   proved(Z3 = [-, G1 => [A]]),
5292   proved(Z4 = [-, G2 => T1]),
5293   proved(Z5 = [K, G1 => [A]]),
5294   proved(Z6 = [-, G2 => T2]),
5295   append(S, [A -> B|Lambda], G),
5296   append(Delta, [B|Lambda], G2),

```

```

5297     append(Phi, [C & D|Psi], T),
5298     append(Phi, [C|Psi], T1),
5299     append(Phi, [D|Psi], T2)),
5300   I,
5301   assertz(proved(Z = [hia(Z3,Z1,A -> B), G => T])),
5302   assertz(proved(Z1 = [ais(Z4,Z6,C & D), G2 => T])),
5303   retract(proved(Z = [ais(Z1,Z2,C & D), G => T])),
5304   retract(proved(Z1 = [hia(Z3,Z4,A -> B), G => T1])),
5305   retract(proved(Z2 = [hia(Z5,Z6,A -> B), G => T2])),
5306   retract(proved(Z5 = [K, G1 => [A]])),
5307   I.

```

```

5308 /* (R11,R11) - (D.53) */
5309 norm2(proved(Z = [aia(Z1,C & D), G => T])) :-
5310   proved(Z1 = [aia(Z2,A & B), G1 => T]),
5311   proved(Z2 = [-, G2 => T]),
5312   append(Gamma, [A & B|E1], G1),
5313   append(Gamma, [A|[B|E1]], G2),
5314   append(Gamma, [A & B|E], G),
5315   append(Delta, [C & D|Lambda], E),
5316   append(Delta, [C|[D|Lambda]], E1),
5317   I,
5318   assertz(proved(Z = [aia(Z1,A & B), G => T])),
5319   append(Gamma, [A|[B|Delta]], S),
5320   append(S, [C & D|Lambda], G3),
5321   assertz(proved(Z1 = [aia(Z2,C & D), G3 => T])),
5322   retract(proved(Z = [aia(Z1,C & D), G => T])),
5323   retract(proved(Z1 = [aia(Z2,A & B), G1 => T])),
5324   I.

```

```

5325 /* (R11,R12) - (D.54.1,2) */
5326 norm2(proved(Z = [oia(Z1,Z2,A # B), G => T])) :-
5327   proved(Z1 = [aia(Z3,C & D), G1 => T]),
5328   proved(Z2 = [aia(Z4,C & D), G2 => T]),
5329   proved(Z3 = [-, G3 => T]),
5330   proved(Z4 = [-, G4 => T]),
5331   ((append(Gamma, [C & D|E1], G1),
5332     append(Gamma, [C|[D|E1]], G3),
5333     append(Gamma, [C & D|E2], G2),
5334     append(Gamma, [C|[D|E2]], G4),
5335     append(Gamma, [C & D|E], G),
5336     append(Gamma, [C|[D|E]], G5)),
5337   (append(S1, [C & D|Lambda], G1),
5338     append(S1, [C|[D|Lambda]], G3),
5339     append(S2, [C & D|Lambda], G2),
5340     append(S2, [C|[D|Lambda]], G4),
5341     append(S, [C & D|Lambda], G),
5342     append(S, [C|[D|Lambda]], G5))),
5343   I,
5344   assertz(proved(Z = [aia(Z1,C & D), G => T])),
5345   assertz(proved(Z1 = [oia(Z3,Z4,A # B), G5 => T])),
5346   retract(proved(Z = [oia(Z1,Z2,A # B), G => T])),
5347   retract(proved(Z1 = [aia(Z3,C & D), G1 => T])),
5348   retract(proved(Z2 = [aia(Z4,C & D), G2 => T])),
5349   I.

```

```

5350 /* (R11,R13) - (D.55) */
5351 norm2(proved(Z = [ois(Z1,C # D), G => T])) :-
5352   proved(Z1 = [aia(Z2,A & B), G => T1]),
5353   proved(Z2 = [-, G1 => T1]),
5354   append(Gamma, [A & B|Delta], G),
5355   append(Gamma, [A|[B|Delta]], G1),
5356   append(Phi, [C # D|Psi], T),
5357   append(Phi, [C|[D|Psi]], T1),
5358   I,
5359   assertz(proved(Z = [aia(Z1,A & B), G => T])),
5360   assertz(proved(Z1 = [ois(Z2,C # D), G1 => T])),
5361   retract(proved(Z = [ois(Z1,C # D), G => T])),
5362   retract(proved(Z1 = [aia(Z2,A & B), G => T])),
5363   I.

```

```

5364 /* (R11,R14) - (D.56.1) */
5365 norm2(proved(Z = [hia(Z1,Z2,A -> B), G => T])) :-
5366   proved(Z1 = [aia(Z3,C & D), G1 => [A]]),
5367   proved(Z2 = [-, G2 => T]),
5368   proved(Z3 = [-, G3 => [A]]),
5369   append(Gamma, [C & D|Delta], G1),
5370   append(Gamma, [C|[D|Delta]], G3),
5371   append(S, [A -> B|Zeta], G),

```

```

5372      append(Lambda, [B|Zeta], G2),
5373      |,
5374      assertz(proved(Z = [aia(Z1,C & D), G => T])),
5375      append(Lambda, G3, S1),
5376      append(S1, [A -> B|Zeta], G4),
5377      assertz(proved(Z1 = [hia(Z3,Z2,A -> B),G4 => T])),
5378      retract(proved(Z = [hia(Z1,Z2,A -> B), G => T])),
5379      retract(proved(Z1 = [aia(Z3,C & D), G1 => [A]])),
5380      |.

```

```

5381 /* (R11,R14) - (D.56.2,3) */
5382 norm2(proved(Z = [hia(Z1,Z2,A -> B), G => T])) :-
5383     proved(Z1 = [_, G1 => [A]]),
5384     proved(Z2 = [aia(Z3,C & D), G2 => T]),
5385     proved(Z3 = [_, G3 => T]),
5386     ((append(Delta, [C & D|E1], G2),
5387      append(Delta, [C|[D|E1]], G3),
5388      append(Delta, [C & D|E], G),
5389      append(Delta, [C|[D|E]], G4))
5390     ;(append(S1, [C & D|Zeta], G2),
5391      append(S1, [C|[D|Zeta]], G3),
5392      append(S, [C & D|Zeta], G),
5393      append(S, [C|[D|Zeta]], G4))),
5394     |,
5395     assertz(proved(Z = [aia(Z2,C & D), G => T])),
5396     assertz(proved(Z2 = [hia(Z1,Z3,A -> B),G4 => T])),
5397     retract(proved(Z = [hia(Z1,Z2,A -> B), G => T])),
5398     retract(proved(Z2 = [aia(Z3,C & D), G2 => T])),
5399     |.

```

```

5400 /* (R11,R15) - (D.57.1,2) */
5401 norm2(proved(Z = [his(Z1,A -> B), G => [A -> B]])) :-
5402     proved(Z1 = [aia(Z2,C & D), G1 => [B]]),
5403     proved(Z2 = [_, G2 => [B]]),
5404     ((append(S1, [C & D|Lambda], G1),
5405      append(S1, [C|[D|Lambda]], G2),
5406      append(S, [C & D|Lambda], G),
5407      append(S, [C|[D|Lambda]], G3))
5408     ;(append(Gamma, [C & D|E1], G1),
5409      append(Gamma, [C|[D|E1]], G2),
5410      append(Gamma, [C & D|E], G),
5411      append(Gamma, [C|[D|E]], G3))),
5412     |,
5413     assertz(proved(Z = [aia(Z1,C & D),G => [A -> B]])),
5414     assertz(proved(Z1 = [his(Z2,A->B),G3 => [A->B]])),
5415     retract(proved(Z = [his(Z1,A->B),G => [A->B]])),
5416     retract(proved(Z1 = [aia(Z2,C & D), G1 => [B]])),
5417     |.

```

```

5418 /* (R11,R15) - (D.57.3) */
5419 norm2(proved(Z = [his(Z1,(A&B) -> L),G => [(A&B) -> L]])) :-
5420     proved(Z1 = [aia(Z2,A & B), G1 => [L]]),
5421     proved(Z2 = [aia(Z3,C & D), G2 => [L]]),
5422     proved(Z3 = [_, G3 => [L]]),
5423     append(S1, [C & D|Lambda], G2),
5424     append(S1, [C|[D|Lambda]], G3),
5425     append(S2, [C & D|Lambda], G1),
5426     append(S, [C & D|Lambda], G),
5427     append(Gamma, [A & B|Delta], S2),
5428     append(Gamma, Delta, S),
5429     |,
5430     assertz(proved(Z = [aia(Z1,C&D),G => [(A&B)->L]])),
5431     append(S, [C|[D|Lambda]], G4),
5432     assertz(proved(Z1 = [his(Z2,(A&B)->L),G4=>[(A&B)->L]])),
5433     append(S2, [C|[D|Lambda]], G5),
5434     assertz(proved(Z2 = [aia(Z3,A & B), G5 => [L]])),
5435     retract(proved(Z = [his(Z1,(A&B)->L),G=>[(A&B)->L]])),
5436     retract(proved(Z1 = [aia(Z2,A & B), G1 => [L]])),
5437     retract(proved(Z2 = [aia(Z3,C & D), G2 => [L]])),
5438     |.

```

```

5439 /* (R12,R12) - (D.58) */
5440 norm2(proved(Z = [oia(Z1,Z2,A # A), G => T])) :-
5441     proved(Z1 = [oia(Z3,Z4,A # A), G1 => T]),
5442     proved(Z2 = [oia(Z5,Z6,A # A), G1 => T]),
5443     proved(Z3 = [_, G2 => T]),
5444     proved(Z4 = [_, G2 => T]),
5445     proved(Z5 = [_, G2 => T]),
5446     proved(Z6 = [_, G2 => T]),

```



```

5447      append(Gamma, [A # A|E1], G1),
5448      append(Gamma, [A|E1], G2),
5449      append(Gamma, [A # A|E], G),
5450      append(Delta, [A # A|Lambda], E),
5451      append(Delta, [A|Lambda], E1),
5452      |
5453      assertz(proved(Z = [oia(Z1,Z2,A # A), G => T])),
5454      append(Gamma, [A|E], G3),
5455      assertz(proved(Z1 = [oia(Z3,Z4,A # A), G3 => T])),
5456      assertz(proved(Z2 = [oia(Z5,Z6,A # A), G3 => T])),
5457      retract(proved(Z = [oia(Z1,Z2,A # A), G => T])),
5458      retract(proved(Z1 = [oia(Z3,Z4,A # A), G1 => T])),
5459      retract(proved(Z2 = [oia(Z5,Z6,A # A), G1 => T])),
5460      |.

```

```

5461 /* (R12,R13) - (D.59) */
5462 norm2(proved(Z = [ois(Z1,C # D), G => T])) :-
5463      proved(Z1 = [oia(Z2,Z3,A # B), G => T1]),
5464      proved(Z2 = [-, G1 => T1]),
5465      proved(Z3 = [-, G2 => T1]),
5466      append(Gamma, [A # B|Delta], G),
5467      append(Gamma, [A|Delta], G1),
5468      append(Gamma, [B|Delta], G2),
5469      append(Phi, [C # D|Psi], T),
5470      append(Phi, [C|D|Psi], T1),
5471      |
5472      gensym(f,F1),
5473      assertz(proved(Z = [oia(Z1,F1,A # B), G => T])),
5474      assertz(proved(Z1 = [ois(Z2,C # D), G1 => T])),
5475      assertz(proved(F1 = [ois(Z3,C # D), G2 => T])),
5476      retract(proved(Z = [ois(Z1,C # D), G => T])),
5477      retract(proved(Z1 = [oia(Z2,Z3,A # B), G => T1])),
5478      |.

```

```

5479 /* (R12,R14) - (D.60.1,2) */
5480 norm2(proved(Z = [oia(Z1,Z2,C # D), G => T])) :-
5481      proved(Z1 = [hia(Z3,Z4,A -> B), G1 => T1]),
5482      proved(Z2 = [hia(Z5,Z6,A -> B), G2 => T1]),
5483      proved(Z3 = [-, G3 => [A]]),
5484      proved(Z4 = [-, G4 => T]),
5485      proved(Z5 = [K, G3 => [A]]),
5486      proved(Z6 = [-, G5 => T]),
5487      ((append(S, [C # D|Zeta], G),
5488      append(S, [C|Zeta], G1),
5489      append(S, [D|Zeta], G2),
5490      append(S1, [C|Zeta], G4),
5491      append(S1, [D|Zeta], G5),
5492      append(S1, [C # D|Zeta], G6))
5493      ;(append(Delta, [C # D|E], G),
5494      append(Delta, [C|E], G1),
5495      append(Delta, [D|E], G2),
5496      append(Delta, [C|E], G4),
5497      append(Delta, [D|E], G5),
5498      append(Delta, [C # D|E], G6))),
5499      |
5500      assertz(proved(Z = [hia(Z3,Z1,A -> B), G => T])),
5501      assertz(proved(Z1 = [oia(Z4,Z6,C # D), G6 => T])),
5502      retract(proved(Z = [oia(Z1,Z2,C # D), G => T])),
5503      retract(proved(Z1 = [hia(Z3,Z4,A -> B), G1 => T])),
5504      retract(proved(Z2 = [hia(Z5,Z6,A -> B), G2 => T])),
5505      retract(proved(Z5 = [K, G3 => [A]])),
5506      |.

```

```

5507 /* (R12,R15) - (D.61.1,2) */
5508 norm2(proved(Z = [his(Z1,C -> D), G => [C -> D]])) :-
5509      proved(Z1 = [oia(Z2,Z3,A # B), G1 => [D]]),
5510      proved(Z2 = [-, G2 => [D]]),
5511      proved(Z3 = [-, G3 => [D]]),
5512      ((append(S1, [A # B|Lambda], G1),
5513      append(S1, [A|Lambda], G2),
5514      append(S1, [B|Lambda], G3),
5515      append(S, [A # B|Lambda], G),
5516      append(S, [A|Lambda], G4),
5517      append(S, [B|Lambda], G5))
5518      ;(append(Gamma, [A # B|E1], G1),
5519      append(Gamma, [A|E1], G2),
5520      append(Gamma, [B|E1], G3),
5521      append(Gamma, [A # B|E], G),
5522      append(Gamma, [A|E], G4),
5523      append(Gamma, [B|E], G5))),

```

```

5524      1,
5525      gensym(f, F1),
5526      assertz(proved(Z = [oia(Z1, F1, A#B), G => [C->D]]))),
5527      assertz(proved(Z1 = [his(Z2, C->D), G4 => [C->D]]))),
5528      assertz(proved(F1 = [his(Z3, C->D), G5 => [C->D]]))),
5529      retract(proved(Z = [his(Z1, C->D), G => [C->D]]))),
5530      retract(proved(Z1 = [oia(Z2, Z3, A#B), G1 => [D]]))),
5531      1,

5532 /* (R13,R13) - (D.62) */
5533 norm2(proved(Z = [ois(Z1, C # D), G => T])) :-
5534     proved(Z1 = [ois(Z2, A # B), G => T1]),
5535     proved(Z2 = [_, G => T2]),
5536     append(Phi, [A # B|E1], T1),
5537     append(Phi, [A|B|E1], T2),
5538     append(Phi, [A # B|E], T),
5539     append(Psi, [C # D|Theta], E),
5540     append(Psi, [C|D|Theta], E1),
5541     1,
5542     assertz(proved(Z = [ois(Z1, A # B), G => T])),
5543     append(Phi, [A|B|E], T3),
5544     assertz(proved(Z1 = [ois(Z2, C # D), G => T3])),
5545     retract(proved(Z = [ois(Z1, C # D), G => T])),
5546     retract(proved(Z1 = [ois(Z2, A # B), G => T1])),
5547     1,

5548 /* (R13,R14) - (D.63) */
5549 norm2(proved(Z = [ois(Z1, C # D), G => T])) :-
5550     proved(Z1 = [hia(Z2, Z3, A -> B), G => T1]),
5551     proved(Z2 = [_, G1 => [A]]),
5552     proved(Z3 = [_, G2 => T1]),
5553     append(S, [A -> B|Lambda], G),
5554     append(Si, [B|Lambda], G2),
5555     append(Phi, [C # D|Psi], T),
5556     append(Phi, [C|D|Psi], T1),
5557     1,
5558     assertz(proved(Z = [hia(Z2, Z1, A -> B), G => T])),
5559     assertz(proved(Z1 = [ois(Z3, C # D), G2 => T])),
5560     retract(proved(Z = [ois(Z1, C # D), G => T])),
5561     retract(proved(Z1 = [hia(Z2, Z3, A -> B), G => T1])),
5562     1,

5563 /* (R14,R14) - (D.64) */
5564 norm2(proved(Z = [hia(Z1, Z2, C -> D), G => T])) :-
5565     proved(Z1 = [_, G1 => [C]]),
5566     proved(Z2 = [hia(Z3, Z4, A -> B), G2 => T]),
5567     proved(Z3 = [_, G3 => [A]]),
5568     proved(Z4 = [_, G4 => T]),
5569     append(S, [C -> D|Ba], G),
5570     append(S1, [D|Ba], G2),
5571     append(S2, [D|Ba], G4),
5572     append(S3, [A -> B|Zeta], S1),
5573     append(Lambda, [B|Zeta], S2),
5574     append(S1, G1, S),
5575     1,
5576     assertz(proved(Z = [hia(Z3, Z2, A -> B), G => T])),
5577     append(S2, G1, E),
5578     append(E, [C -> D|Ba], G5),
5579     assertz(proved(Z2 = [hia(Z1, Z4, C -> D), G5 => T])),
5580     retract(proved(Z = [hia(Z1, Z2, C -> D), G => T])),
5581     retract(proved(Z2 = [hia(Z3, Z4, A -> B), G2 => T])),
5582     1,

5583 /* (R14,R15) - (D.65.1,2) */
5584 norm2(proved(Z = [his(Z1, C -> D), G => [C -> D]])) :-
5585     proved(Z1 = [hia(Z2, Z3, A -> B), G1 => [D]]),
5586     proved(Z2 = [_, G2 => [A]]),
5587     proved(Z3 = [_, G3 => [D]]),
5588     ((append(S1, [A -> B|Zeta], G1),
5589     append(S2, [B|Zeta], G3),
5590     append(S, [A -> B|Zeta], G),
5591     append(Delta_Lambda, G2, S),
5592     append(Delta_Lambda, [B|Zeta], G4))
5593     ; (append(S, [A -> B|E1], G1),
5594     append(Delta, [B|E1], G3),
5595     append(S, [A -> B|E], G),
5596     append(Delta, [B|E], G4))),
5597     1,
5598     assertz(proved(Z = [hia(Z2, Z1, A->B), G => [C->D]])),

```

```

5599      assertz(proved(Z1 = [his(Z3,C->D),G4 => [C->D]])),
5600      retract(proved(Z = [his(Z1,C -> D), G => [C->D]])),
5601      retract(proved(Z1 = [hia(Z2,Z3,A->B),G1 => [D]])),
5602

```

```

5603 /* (R14,R15) - (D.65.3) */
5604 norm2(proved(Z = [his(Z1,(A->B)->F), G => [(A->B)->F]])) :-
5605     proved(Z1 = [hia(Z2,Z3,A -> B), G1 => [F]]),
5606     proved(Z2 = [-, G2 => [A]]),
5607     proved(Z3 = [hia(Z4,Z5,C -> D), G3 => [F]]),
5608     proved(Z4 = [-, G4 => [C]]),
5609     proved(Z5 = [-, G5 => [F]]),
5610     append(S1, [C -> D|Bai], G3),
5611     append(S2, [D|Bai], G5),
5612     append(S3, [C -> D|Bai], G1),
5613     append(S, [C -> D|Bai], G),
5614     append(Lambda_Delta, [A -> B|E1], S3),
5615     append(Lambda, [B|E1], S1),
5616     append(Lambda_Delta_Zeta, G4, S),
5617
5618     assertz(proved(Z=[hia(Z4,Z1,C->D),G=>[(A->B)->F]])),
5619     append(Lambda_Delta_Zeta, [D|Bai], G6),
5620     assertz(proved(Z1=[his(Z3,(A->B)->F),G6=>[(A->B)->F]])),
5621     append(K, G4, S3),
5622     append(K, [D|Bai], G7),
5623     assertz(proved(Z3 = [hia(Z2,Z5,A -> B), G7 => [F]])),
5624     retract(proved(Z=[his(Z1,(A->B)->F),G=>[(A->B)->F]])),
5625     retract(proved(Z1 = [hia(Z2,Z3,A->B),G1 => [F]])),
5626     retract(proved(Z3 = [hia(Z4,Z5,C->D),G3 => [F]])),
5627

```

```

5628 /* (R14,R15) - (D.65.4,5) */
5629 norm2(proved(Z = [his(Z1,(C&D) -> F), G => [(C&D) -> F]])):-
5630     proved(Z1 = [aia(Z2,C & D), G1 => [F]]),
5631     proved(Z2 = [hia(Z3,Z4,A -> B), G2 => [F]]),
5632     proved(Z3 = [-, G3 => [A]]),
5633     proved(Z4 = [-, G4 => [F]]),
5634     ((append(Delta, [C & D|E], G1),
5635     append(Delta, [C|[D|E]], G2),
5636     append(Delta, [C|[D|E1]], G4),
5637     append(Delta, E, G),
5638     append(Lambda, [B|Zeta], E1),
5639     append(S1, [A -> B|Zeta], E),
5640     append(Delta, E1, G5),
5641     append(Delta, [C & D|E1], G6))
5642     :(append(S, [C & D|Zeta], G1),
5643     append(S, [C|[D|Zeta]], G2),
5644     append(S1, [C|[D|Zeta]], G4),
5645     append(S, Zeta, G),
5646     append(Delta, [B|Lambda], S1),
5647     append(S2, [A -> B|Lambda], S),
5648     append(S1, Zeta, G5),
5649     append(S1, [C & D|Zeta], G6)),
5650
5651     assertz(proved(Z=[hia(Z3,Z1,A->B),G=>[(C&D)->F]])),
5652     assertz(proved(Z1=[his(Z2,(C&D)->F),G5=>[(C&D)->F]])),
5653     assertz(proved(Z2 = [aia(Z4,C & D), G6 => [F]])),
5654     retract(proved(Z=[his(Z1,(C&D)->F),G=>[(C&D)->F]])),
5655     retract(proved(Z1 = [aia(Z2,C & D), G1 => [F]])),
5656     retract(proved(Z2 = [hia(Z3,Z4,A->B),G2 => [F]])),
5657

```

```

5658 /* (R14,R15) - (D.65.6,7) */
5659 norm2(proved(Z = [his(Z1,C -> F), G => [C -> F]])) :-
5660     proved(Z1 = [th(Z2,C), G1 => [F]]),
5661     proved(Z2 = [hia(Z3,Z4,A -> B), G => [F]]),
5662     proved(Z3 = [-, G2 => [A]]),
5663     proved(Z4 = [-, G3 => [F]]),
5664     ((append(S, [C|Zeta], G1),
5665     append(S, Zeta, G),
5666     append(S1, Zeta, G3),
5667     append(Delta, [B|Lambda], S1),
5668     append(Delta_Gamma, [A -> B|Lambda], S),
5669     append(S1, [C|Zeta], G4))
5670     :(append(Delta, [C|E], G1),
5671     append(Delta, E, G),
5672     append(Delta, E1, G3),
5673     append(Lambda, [B|Zeta], E1),
5674     append(Lambda_Gamma, [A -> B|Zeta], E),

```

```

5675     append(Delta, [C|E1], G4))).
5676
5677     assertz(proved(Z = [hia(Z3,Z1,A->B), G => [C->F]])),
5678     assertz(proved(Z1 = [hia(Z2,C->F), G3 => [C->F]])),
5679     assertz(proved(Z2 = [th(Z4,C), G4 => [F]])),
5680     retract(proved(Z = [hia(Z1,C->F), G => [C->F]])),
5681     retract(proved(Z1 = [th(Z2,C), G1 => [F]])),
5682     retract(proved(Z2 = [hia(Z3,Z4,A->B), G => [F]])),
5683

```

```

5684 /* (R2,R3,R10) - (D.73.1) */
5685 norm2(proved(Z = [ais(Z1,Z2,A & B), G => T])) :-
5686     proved(Z1 = [-, G => T1]),
5687     proved(Z2 = [con(Z3,C), G => T2]),
5688     proved(Z3 = [-, G1 => T2]),
5689     append(Gamma, [C|[C|Delta]], G1),
5690     append(Gamma, [C|Delta], G),
5691     append(Phi, [A & B|Psi], T),
5692     append(Phi, [A|Psi], T1),
5693     append(Phi, [B|Psi], T2),
5694
5695     assertz(proved(Z = [con(Z2,C), G => T])),
5696     gensym(f,F1),
5697     assertz(proved(Z2 = [ais(F1,Z3,A & B), G1 => T])),
5698     assertz(proved(F1 = [th(Z1,C), G1 => T1])),
5699     retract(proved(Z = [ais(Z1,Z2,A & B), G => T])),
5700     retract(proved(Z2 = [con(Z3,C), G => T2])),
5701

```

```

5702 /* (R2,R3,R10) - (D.73.2) */
5703 norm2(proved(Z = [ais(Z1,Z2,A & B), G => T])) :-
5704     proved(Z1 = [con(Z3,C), G => T1]),
5705     proved(Z2 = [-, G => T2]),
5706     proved(Z3 = [-, G1 => T1]),
5707     append(Gamma, [C|[C|Delta]], G1),
5708     append(Gamma, [C|Delta], G),
5709     append(Phi, [A & B|Psi], T),
5710     append(Phi, [A|Psi], T1),
5711     append(Phi, [B|Psi], T2),
5712
5713     assertz(proved(Z = [con(Z1,C), G => T])),
5714     gensym(f,F1),
5715     assertz(proved(Z1 = [ais(Z3,F1,A & B), G1 => T])),
5716     assertz(proved(F1 = [th(Z2,C), G1 => T2])),
5717     retract(proved(Z = [ais(Z1,Z2,A & B), G => T])),
5718     retract(proved(Z1 = [con(Z3,C), G => T1])),
5719

```

```

5720 /* (R2,R3,R12) - (D.74.1,3) */
5721 norm2(proved(Z = [oia(Z1,Z2,A # B), G => T])) :-
5722     proved(Z1 = [-, G1 => T]),
5723     proved(Z2 = [con(Z3,C), G2 => T]),
5724     proved(Z3 = [-, G3 => T]),
5725     ((append(S, [A # B|Lambda], G),
5726     append(S, [A|Lambda], G1),
5727     append(S, [B|Lambda], G2),
5728     append(S1, [B|Lambda], G3),
5729     append(Gamma, [C|[C|Delta]], S1),
5730     append(Gamma, [C|Delta], S),
5731     append(S1, [A # B|Lambda], G4),
5732     append(S1, [A|Lambda], G5))
5733     ;(append(Gamma, [A # B|E], G),
5734     append(Gamma, [A|E], G1),
5735     append(Gamma, [B|E], G2),
5736     append(Gamma, [B|E1], G3),
5737     append(Delta, [C|[C|Lambda]], E1),
5738     append(Delta, [C|Lambda], E),
5739     append(Gamma, [A # B|E1], G4),
5740     append(Gamma, [A|E1], G5)),
5741
5742     assertz(proved(Z = [con(Z2,C), G => T])),
5743     gensym(f,F1),
5744     assertz(proved(Z2 = [oia(F1,Z3,A # B), G4 => T])),
5745     assertz(proved(F1 = [th(Z1,C), G5 => T])),
5746     retract(proved(Z = [oia(Z1,Z2,A # B), G => T])),
5747     retract(proved(Z2 = [con(Z3,C), G2 => T])),
5748

```

```

5749 /* (R2,R3,R12) - (D.74.2,4) */
5750 norm2(proved(Z = [ola(Z1,Z2,A # B), G => T])) :-
5751   proved(Z1 = [con(Z3,C), G1 => T]),
5752   proved(Z2 = [_, G2 => T]),
5753   proved(Z3 = [_, G3 => T]),
5754   ((append(S, [A # B|Lambda], G),
5755     append(S, [A|Lambda], G1),
5756     append(S, [B|Lambda], G2),
5757     append(S1, [A|Lambda], G3),
5758     append(Gamma, [C|C|Delta], S1),
5759     append(Gamma, [C|Delta], S),
5760     append(S1, [A # B|Lambda], G4),
5761     append(S1, [A|Lambda], G5))
5762   : (append(Gamma, [A # B|E], G),
5763     append(Gamma, [A|E], G1),
5764     append(Gamma, [B|E], G2),
5765     append(Gamma, [A|E1], G3),
5766     append(Delta, [C|C|Lambda], E1),
5767     append(Delta, [C|Lambda], E),
5768     append(Gamma, [A # B|E1], G4),
5769     append(Gamma, [A|E1], G5))),
5770   !,
5771   assertz(proved(Z = [con(Z1,C), G => T])),
5772   gensym(f,Fi),
5773   assertz(proved(Z1 = [ola(Z3,F1,A # B), G4 => T])),
5774   assertz(proved(Fi = [th(Z2,C), G5 => T])),
5775   retract(proved(Z = [ola(Z1,Z2,A # B), G => T])),
5776   retract(proved(Z1 = [con(Z3,C), G1 => T])),
5777   !.

```

```

5778 /* (R2,R4,R10) - (D.75.1) */
5779 norm2(proved(Z = [ais(Z1,Z2,A & B), G => T])) :-
5780   proved(Z1 = [th(Z3,D), G => T1]),
5781   proved(Z2 = [inc(Z4,D,C), G => T2]),
5782   proved(Z3 = [_, G1 => T1]),
5783   proved(Z4 = [_, G2 => T2]),
5784   append(Gamma, [D|C|Delta], G),
5785   append(Gamma, [C|Delta], G1),
5786   append(Gamma, [C|D|Delta], G2),
5787   append(Phi, [A & B|Psi], T),
5788   append(Phi, [A|Psi], T1),
5789   append(Phi, [B|Psi], T2),
5790   !,
5791   assertz(proved(Z = [inc(Z2,D,C), G => T])),
5792   assertz(proved(Z2 = [ais(Z1,Z4,A & B), G2 => T])),
5793   assertz(proved(Z1 = [th(Z3,D), G2 => T1])),
5794   retract(proved(Z = [ais(Z1,Z2,A & B), G => T])),
5795   retract(proved(Z1 = [th(Z3,D), G => T1])),
5796   retract(proved(Z2 = [inc(Z4,D,C), G => T2])),
5797   !.

```

```

5798 /* (R2,R4,R10) - (D.75.2) */
5799 norm2(proved(Z = [ais(Z1,Z2,A & B), G => T])) :-
5800   proved(Z1 = [th(Z3,C), G => T1]),
5801   proved(Z2 = [inc(Z4,D,C), G => T2]),
5802   proved(Z3 = [_, G1 => T1]),
5803   proved(Z4 = [_, G2 => T2]),
5804   append(Gamma, [D|C|Delta], G),
5805   append(Gamma, [D|Delta], G1),
5806   append(Gamma, [C|D|Delta], G2),
5807   append(Phi, [A & B|Psi], T),
5808   append(Phi, [A|Psi], T1),
5809   append(Phi, [B|Psi], T2),
5810   !,
5811   assertz(proved(Z = [inc(Z2,D,C), G => T])),
5812   assertz(proved(Z2 = [ais(Z1,Z4,A & B), G2 => T])),
5813   assertz(proved(Z1 = [th(Z3,C), G2 => T1])),
5814   retract(proved(Z = [ais(Z1,Z2,A & B), G => T])),
5815   retract(proved(Z1 = [th(Z3,C), G => T1])),
5816   retract(proved(Z2 = [inc(Z4,D,C), G => T2])),
5817   !.

```

```

5818 /* (R2,R4,R10) - (D.75.3) */
5819 norm2(proved(Z = [ais(Z1,Z2,B & A), G => T])) :-
5820   proved(Z1 = [inc(Z3,D,C), G => T1]),
5821   proved(Z2 = [th(Z4,D), G => T2]),
5822   proved(Z3 = [_, G1 => T1]),
5823   proved(Z4 = [_, G2 => T2]),
5824   append(Gamma, [D|C|Delta], G),
5825   append(Gamma, [C|D|Delta], G1),

```

```

5826      append(Gamma, [C|Delta], G2),
5827      append(Phi, [B & A|Psi], T),
5828      append(Phi, [B|Psi], T1),
5829      append(Phi, [A|Psi], T2);
5830  I,
5831      assertz(proved(Z = [inc(Z1,D,C), G => T])),
5832      assertz(proved(Z1 = [ois(Z3,Z2,B & A), G1 => T])),
5833      assertz(proved(Z2 = [th(Z4,D), G1 => T2])),
5834      retract(proved(Z = [ois(Z1,Z2,B & A), G => T])),
5835      retract(proved(Z1 = [inc(Z3,D,C), G => T1])),
5836      retract(proved(Z2 = [th(Z4,D), G => T2])),
5837  I.

```

```

5838 /* (R2,R4,R10) - (D.75.4) */
5839 norm2(proved(Z = [ois(Z1,Z2,B & A), G => T])) :-
5840     proved(Z1 = [inc(Z3,D,C), G => T1]),
5841     proved(Z2 = [th(Z4,C), G => T2]),
5842     proved(Z3 = [-, G1 => T1]),
5843     proved(Z4 = [-, G2 => T2]),
5844     append(Gamma, [D|[C|Delta]], G),
5845     append(Gamma, [C|[D|Delta]], G1),
5846     append(Gamma, [D|Delta], G2),
5847     append(Phi, [B & A|Psi], T),
5848     append(Phi, [B|Psi], T1),
5849     append(Phi, [A|Psi], T2);
5850 I,
5851     assertz(proved(Z = [inc(Z1,D,C), G => T])),
5852     assertz(proved(Z1 = [ois(Z3,Z2,B & A), G1 => T])),
5853     assertz(proved(Z2 = [th(Z4,C), G1 => T2])),
5854     retract(proved(Z = [ois(Z1,Z2,B & A), G => T])),
5855     retract(proved(Z1 = [inc(Z3,D,C), G => T1])),
5856     retract(proved(Z2 = [th(Z4,C), G => T2])),
5857 I.

```

```

5858 /* (R2,R4,R12) - (D.76.1,5) */
5859 norm2(proved(Z = [ois(Z1,Z2,A # B), G => T])) :-
5860     proved(Z1 = [th(Z3,D), G1 => T]),
5861     proved(Z2 = [inc(Z4,D,C), G2 => T]),
5862     proved(Z3 = [-, G3 => T]),
5863     proved(Z4 = [-, G4 => T]),
5864     ((append(Gamma, [D|[C|E1]], G1),
5865     append(Gamma, [C|[E1]], G3),
5866     append(Gamma, [D|[C|E2]], G2),
5867     append(Gamma, [C|[D|E2]], G4),
5868     append(Gamma, [D|[C|E]], G),
5869     append(Delta, [A # B|Lambda], E),
5870     append(Delta, [A|Lambda], E1),
5871     append(Delta, [B|Lambda], E2),
5872     append(Gamma, [C|[D|E]], G5),
5873     append(Gamma, [C|[D|E1]], G6))
5874 ; (append(S1, [D|[C|Lambda]], G1),
5875     append(S1, [C|Lambda], G3),
5876     append(S2, [D|[C|Lambda]], G2),
5877     append(S2, [C|[D|Lambda]], G4),
5878     append(S, [D|[C|Lambda]], G),
5879     append(Gamma, [A # B|Delta], S),
5880     append(Gamma, [A|Delta], S1),
5881     append(Gamma, [B|Delta], S2),
5882     append(S, [C|[D|Lambda]], G5),
5883     append(S1, [C|[D|Lambda]], G6));
5884 I,
5885     assertz(proved(Z = [inc(Z2,D,C), G => T])),
5886     assertz(proved(Z2 = [ois(Z1,Z4,A # B), G5 => T])),
5887     assertz(proved(Z1 = [th(Z3,D), G6 => T])),
5888     retract(proved(Z = [ois(Z1,Z2,A # B), G => T])),
5889     retract(proved(Z1 = [th(Z3,D), G1 => T])),
5890     retract(proved(Z2 = [inc(Z4,D,C), G2 => T])),
5891 I.

```

```

5892 /* (R2,R4,R12) - (D.76.2,6) */
5893 norm2(proved(Z = [ois(Z1,Z2,A # B), G => T])) :-
5894     proved(Z1 = [th(Z3,C), G1 => T]),
5895     proved(Z2 = [inc(Z4,D,C), G2 => T]),
5896     proved(Z3 = [-, G3 => T]),
5897     proved(Z4 = [-, G4 => T]),
5898     ((append(Gamma, [D|[C|E1]], G1),
5899     append(Gamma, [D|E1], G3),
5900     append(Gamma, [D|[C|E2]], G2),
5901     append(Gamma, [C|[D|E2]], G4),
5902     append(Gamma, [D|[C|E]], G),

```

```

5903     append(Delta, [A # B|Lambda], E),
5904     append(Delta, [A|Lambda], E1),
5905     append(Delta, [B|Lambda], E2),
5906     append(Gamma, [C|[D|E]], G5),
5907     append(Gamma, [C|[D|E1]], G6))
5908 ;(append(S1, [D|[C|Lambda]], G1),
5909     append(S1, [D|Lambda], G3),
5910     append(S2, [D|[C|Lambda]], G2),
5911     append(S2, [C|[D|Lambda]], G4),
5912     append(S, [D|[C|Lambda]], G),
5913     append(Gamma, [A # B|Delta], S),
5914     append(Gamma, [A|Delta], S1),
5915     append(Gamma, [B|Delta], S2),
5916     append(S, [C|[D|Lambda]], G5),
5917     append(S1, [C|[D|Lambda]], G6))),
5918 !,
5919 assertz(proved(Z = [inc(Z2,D,C), G => T])),
5920 assertz(proved(Z2 = [oia(Z1,Z4,A # B), G5 => T])),
5921 assertz(proved(Z1 = [th(Z3,C), G6 => T])),
5922 retract(proved(Z = [oia(Z1,Z2,A # B), G => T])),
5923 retract(proved(Z1 = [th(Z3,C), G1 => T])),
5924 retract(proved(Z2 = [inc(Z4,D,C), G2 => T])),
5925 !.

```

```

5926 /* (R2,R4,R12) - (D.76.3,7) */
5927 norm2(proved(Z = [oia(Z1,Z2,B # A), G => T])) :-
5928     proved(Z1 = [inc(Z3,D,C), G1 => T]),
5929     proved(Z2 = [th(Z4,D), G2 => T]),
5930     proved(Z3 = [-, G3 => T]),
5931     proved(Z4 = [-, G4 => T]),
5932     ((append(Gamma, [D|[C|E1]], G1),
5933      append(Gamma, [C|[D|E1]], G3),
5934      append(Gamma, [D|[C|E2]], G2),
5935      append(Gamma, [C|E2], G4),
5936      append(Gamma, [D|[C|E]], G),
5937      append(Delta, [B # A|Lambda], E),
5938      append(Delta, [B|Lambda], E1),
5939      append(Delta, [A|Lambda], E2),
5940      append(Gamma, [C|[D|E]], G5),
5941      append(Gamma, [C|[D|E2]], G6))
5942 ;(append(S1, [D|[C|Lambda]], G1),
5943     append(S1, [C|[D|Lambda]], G3),
5944     append(S2, [D|[C|Lambda]], G2),
5945     append(S2, [C|Lambda], G4),
5946     append(S, [D|[C|Lambda]], G),
5947     append(Gamma, [B # A|Delta], S),
5948     append(Gamma, [B|Delta], S1),
5949     append(Gamma, [A|Delta], S2),
5950     append(S, [C|[D|Lambda]], G5),
5951     append(S2, [C|[D|Lambda]], G6))),
5952 !,
5953 assertz(proved(Z = [inc(Z1,D,C), G => T])),
5954 assertz(proved(Z1 = [oia(Z3,Z2,B # A), G5 => T])),
5955 assertz(proved(Z2 = [th(Z4,D), G6 => T])),
5956 retract(proved(Z = [oia(Z1,Z2,B # A), G => T])),
5957 retract(proved(Z1 = [inc(Z3,D,C), G1 => T])),
5958 retract(proved(Z2 = [th(Z4,D), G2 => T])),
5959 !.

```

```

5960 /* (R2,R4,R12) - (D.76.4,8) */
5961 norm2(proved(Z = [oia(Z1,Z2,B # A), G => T])) :-
5962     proved(Z1 = [inc(Z3,D,C), G1 => T]),
5963     proved(Z2 = [th(Z4,C), G2 => T]),
5964     proved(Z3 = [-, G3 => T]),
5965     proved(Z4 = [-, G4 => T]),
5966     ((append(Gamma, [D|[C|E1]], G1),
5967      append(Gamma, [C|[D|E1]], G3),
5968      append(Gamma, [D|[C|E2]], G2),
5969      append(Gamma, [D|E2], G4),
5970      append(Gamma, [D|[C|E]], G),
5971      append(Delta, [B # A|Lambda], E),
5972      append(Delta, [B|Lambda], E1),
5973      append(Delta, [A|Lambda], E2),
5974      append(Gamma, [C|[D|E]], G5),
5975      append(Gamma, [C|[D|E2]], G6))
5976 ;(append(S1, [D|[C|Lambda]], G1),
5977     append(S1, [C|[D|Lambda]], G3),
5978     append(S2, [D|[C|Lambda]], G2),
5979     append(S2, [D|Lambda], G4),
5980     append(S, [D|[C|Lambda]], G),
5981     append(Gamma, [B # A|Delta], S),

```

```

5982     append(Gamma, [B|Delta], S1),
5983     append(Gamma, [A|Delta], S2),
5984     append(S, [C|[D|Lambda]], G5),
5985     append(S2, [C|[D|Lambda]], G6))),
5986
5987 1.
5988     assertz(proved(Z = [inc(Z1,D,C), G => T])),
5989     assertz(proved(Z1 = [ola(Z3,Z2,B # A), G5 => T])),
5990     assertz(proved(Z2 = [th(Z4,C), G6 => T])),
5991     retract(proved(Z = [ola(Z1,Z2,B # A), G => T])),
5992     retract(proved(Z1 = [inc(Z3,D,C), G1 => T])),
5993     retract(proved(Z2 = [th(Z4,C), G2 => T])).

```

```

5994 /* (R2,R4,R12) - (D.76.9,13) */
5995 norm2(proved(Z = [inc(Z1,L1,L2), G => T])) :-
5996 ((L1 = A, L2 = B # C) ; (L1 = B # C, L2 = A)),
5997 proved(Z1 = [ola(Z2,Z3,B # C), G1 => T]),
5998 proved(Z2 = [th(Z4,B), G2 => T]),
5999 proved(Z3 = [th(Z5,C), G3 => T]),
6000 proved(Z4 = [-, G4 => T]),
6001 proved(Z5 = [-, G4 => T]),
6002 ((append(Gamma, [A|[B # C|Delta]], G),
6003  append(Gamma, [B # C|[A|Delta]], G1),
6004  append(Gamma, [B|[A|Delta]], G2),
6005  append(Gamma, [C|[A|Delta]], G3),
6006  append(Gamma, [A|Delta], G4),
6007  append(Gamma, [A|[B|Delta]], G5),
6008  append(Gamma, [A|[C|Delta]], G6))
6009 ; (append(Gamma, [B # C|[A|Delta]], G),
6010  append(Gamma, [A|[B # C|Delta]], G1),
6011  append(Gamma, [A|[B|Delta]], G2),
6012  append(Gamma, [A|[C|Delta]], G3),
6013  append(Gamma, [A|Delta], G4),
6014  append(Gamma, [B|[A|Delta]], G5),
6015  append(Gamma, [C|[A|Delta]], G6))),
6016
6017 1.
6018     assertz(proved(Z = [ola(Z2,Z3,B # C), G => T])),
6019     assertz(proved(Z2 = [th(Z4,B), G5 => T])),
6020     assertz(proved(Z3 = [th(Z5,C), G6 => T])),
6021     retract(proved(Z = [inc(Z1,L1,L2), G => T])),
6022     retract(proved(Z1 = [ola(Z2,Z3,B # C), G1 => T])),
6023     retract(proved(Z2 = [th(Z4,B), G2 => T])),
6024     retract(proved(Z3 = [th(Z5,C), G3 => T])).

```

```

6025 /* (R2,R4,R12) - (D.76.10,14) */
6026 norm2(proved(Z = [inc(Z1,L1,L2), G => T])) :-
6027 ((L1 = A, L2 = B # C) ; (L1 = B # C, L2 = A)),
6028 proved(Z1 = [ola(Z2,Z3,B # C), G1 => T]),
6029 proved(Z2 = [th(Z4,A), G2 => T]),
6030 proved(Z3 = [th(Z5,C), G3 => T]),
6031 proved(Z4 = [-, G4 => T]),
6032 proved(Z5 = [-, G5 => T]),
6033 ((append(Gamma, [A|[B # C|Delta]], G),
6034  append(Gamma, [B # C|[A|Delta]], G1),
6035  append(Gamma, [B|[A|Delta]], G2),
6036  append(Gamma, [C|[A|Delta]], G3),
6037  append(Gamma, [B|Delta], G4),
6038  append(Gamma, [A|Delta], G5),
6039  append(Gamma, [A|[B|Delta]], G6),
6040  append(Gamma, [A|[C|Delta]], G7))
6041 ; (append(Gamma, [B # C|[A|Delta]], G),
6042  append(Gamma, [A|[B # C|Delta]], G1),
6043  append(Gamma, [A|[B|Delta]], G2),
6044  append(Gamma, [A|[C|Delta]], G3),
6045  append(Gamma, [B|Delta], G4),
6046  append(Gamma, [A|Delta], G5),
6047  append(Gamma, [B|[A|Delta]], G6),
6048  append(Gamma, [C|[A|Delta]], G7))),
6049
6050 1.
6051     assertz(proved(Z = [ola(Z2,Z3,B # C), G => T])),
6052     assertz(proved(Z2 = [th(Z4,A), G6 => T])),
6053     assertz(proved(Z3 = [th(Z5,C), G7 => T])),
6054     retract(proved(Z = [inc(Z1,L1,L2), G => T])),
6055     retract(proved(Z1 = [ola(Z2,Z3,B # C), G1 => T])),
6056     retract(proved(Z2 = [th(Z4,A), G2 => T])),
6057     retract(proved(Z3 = [th(Z5,C), G3 => T])).

```



```

6058 /* (R2,R4,R12) - (D.76.11,15) */
6059 norm2(proved(Z = [inc(Z1,L1,L2), G => T])) :-
6060 ((L1 = A, L2 = B # C) ; (L1 = B # C, L2 = A)),
6061 proved(Z1 = [oia(Z2,Z3,B # C), G1 => T]),
6062 proved(Z2 = [th(Z4,B), G2 => T]),
6063 proved(Z3 = [th(Z5,A), G3 => T]),
6064 proved(Z4 = [_, G4 => T]),
6065 proved(Z5 = [_, G5 => T]),
6066 ((append(Gamma, [A|B # C|Delta]], G),
6067 append(Gamma, B # C|[A|Delta]], G1),
6068 append(Gamma, B|[A|Delta]], G2),
6069 append(Gamma, C|[A|Delta]], G3),
6070 append(Gamma, A|Delta], G4),
6071 append(Gamma, C|Delta], G5),
6072 append(Gamma, A|[B|Delta]], G6),
6073 append(Gamma, A|[C|Delta]], G7))
6074 ;(append(Gamma, B # C|[A|Delta]], G),
6075 append(Gamma, A|[B # C|Delta]], G1),
6076 append(Gamma, A|[B|Delta]], G2),
6077 append(Gamma, A|[C|Delta]], G3),
6078 append(Gamma, A|Delta], G4),
6079 append(Gamma, C|Delta], G5),
6080 append(Gamma, B|[A|Delta]], G6),
6081 append(Gamma, C|[A|Delta]], G7))),
6082 !,
6083 assertz(proved(Z = [oia(Z2,Z3,B # C), G => T])),
6084 assertz(proved(Z2 = [th(Z4,B), G6 => T])),
6085 assertz(proved(Z3 = [th(Z5,A), G7 => T])),
6086 retract(proved(Z = [inc(Z1,L1,L2), G => T])),
6087 retract(proved(Z1 = [oia(Z2,Z3,B # C), G1 => T])),
6088 retract(proved(Z2 = [th(Z4,B), G2 => T])),
6089 retract(proved(Z3 = [th(Z5,A), G3 => T])),
6090 !.

```

```

6091 /* (R2,R4,R12) - (D.76.12,16) */
6092 norm2(proved(Z = [inc(Z1,L1,L2), G => T])) :-
6093 ((L1 = A, L2 = B # C) ; (L1 = B # C, L2 = A)),
6094 proved(Z1 = [oia(Z2,Z3,B # C), G1 => T]),
6095 proved(Z2 = [th(Z4,A), G2 => T]),
6096 proved(Z3 = [th(Z5,A), G3 => T]),
6097 proved(Z4 = [_, G4 => T]),
6098 proved(Z5 = [_, G5 => T]),
6099 ((append(Gamma, [A|B # C|Delta]], G),
6100 append(Gamma, B # C|[A|Delta]], G1),
6101 append(Gamma, B|[A|Delta]], G2),
6102 append(Gamma, C|[A|Delta]], G3),
6103 append(Gamma, B|Delta], G4),
6104 append(Gamma, C|Delta], G5),
6105 append(Gamma, A|[B|Delta]], G6),
6106 append(Gamma, A|[C|Delta]], G7))
6107 ;(append(Gamma, B # C|[A|Delta]], G),
6108 append(Gamma, A|[B # C|Delta]], G1),
6109 append(Gamma, A|[B|Delta]], G2),
6110 append(Gamma, A|[C|Delta]], G3),
6111 append(Gamma, B|Delta], G4),
6112 append(Gamma, C|Delta], G5),
6113 append(Gamma, B|[A|Delta]], G6),
6114 append(Gamma, C|[A|Delta]], G7))),
6115 !,
6116 assertz(proved(Z = [oia(Z2,Z3,B # C), G => T])),
6117 assertz(proved(Z2 = [th(Z4,A), G6 => T])),
6118 assertz(proved(Z3 = [th(Z5,A), G7 => T])),
6119 retract(proved(Z = [inc(Z1,L1,L2), G => T])),
6120 retract(proved(Z1 = [oia(Z2,Z3,B # C), G1 => T])),
6121 retract(proved(Z2 = [th(Z4,A), G2 => T])),
6122 retract(proved(Z3 = [th(Z5,A), G3 => T])),
6123 !.

```

```

6124 /* (R3,R3,R10) - (D.77.1) */
6125 norm2(proved(Z = [ais(Z1,Z2,A & B), G => T])) :-
6126 proved(Z1 = [con(Z3,C), G => T1]),
6127 proved(Z2 = [con(Z4,D), G => T2]),
6128 proved(Z3 = [_, G1 => T1]),
6129 proved(Z4 = [_, G2 => T2]),
6130 append(Phi, [A & B|Psi], T),
6131 append(Phi, [A|Psi], T1),
6132 append(Phi, [B|Psi], T2),
6133 append(Gamma, [C|[D|Delta]], G),
6134 append(Gamma, [C|[C|[D|Delta]]], G1),
6135 append(Gamma, [C|[D|[D|Delta]]], G2),
6136 !.

```

```

6137 gensym(f,F1),
6138 assertz(proved(Z = [con(F1,C), G => T])),
6139 gensym(f,F2),
6140 assertz(proved(F1 = [con(F2,D), G1 => T])),
6141 append(Gamma, [C|[C|D|[D|Delta]]], G3),
6142 assertz(proved(F2 = [ais(Z1,Z2,A & B), G3 => T])),
6143 assertz(proved(Z1 = [th(Z3,D), G3 => T1])),
6144 assertz(proved(Z2 = [th(Z4,C), G3 => T2])),
6145 retract(proved(Z = [ais(Z1,Z2,A & B), G => T])),
6146 retract(proved(Z1 = [con(Z3,C), G => T1])),
6147 retract(proved(Z2 = [con(Z4,D), G => T2])),
6148

```

```

6149 /* (R3,R3,R10) - (D.77.2) */
6150 norm2(proved(Z = [ais(Z1,Z2,A & B), G => T])) :-
6151 proved(Z1 = [con(Z3,D), G => T1]),
6152 proved(Z2 = [con(Z4,C), G => T2]),
6153 proved(Z3 = [-, G1 => T1]),
6154 proved(Z4 = [-, G2 => T2]),
6155 append(Phi, [A & B|Psi], T),
6156 append(Phi, [A|Psi], T1),
6157 append(Phi, [B|Psi], T2),
6158 append(Gamma, [C|[D|Delta]], G),
6159 append(Gamma, [C|[D|Delta]], G1),
6160 append(Gamma, [C|[C|[D|Delta]]], G2),
6161
6162 gensym(f,F1),
6163 assertz(proved(Z = [con(F1,C), G => T])),
6164 gensym(f,F2),
6165 assertz(proved(F1 = [con(F2,D), G2 => T])),
6166 append(Gamma, [C|[C|D|[D|Delta]]], G3),
6167 assertz(proved(F2 = [ais(Z1,Z2,A & B), G3 => T])),
6168 assertz(proved(Z1 = [th(Z3,C), G3 => T1])),
6169 assertz(proved(Z2 = [th(Z4,D), G3 => T2])),
6170 retract(proved(Z = [ais(Z1,Z2,A & B), G => T])),
6171 retract(proved(Z1 = [con(Z3,D), G => T1])),
6172 retract(proved(Z2 = [con(Z4,C), G => T2])),
6173

```

```

6174 /* (R3,R3,R12) - (D.78.1,3) */
6175 norm2(proved(Z = [oia(Z1,Z2,A # B), G => T])) :-
6176 proved(Z1 = [con(Z3,C), G1 => T1]),
6177 proved(Z2 = [con(Z4,D), G2 => T2]),
6178 proved(Z3 = [-, G3 => T1]),
6179 proved(Z4 = [-, G4 => T2]),
6180 ((append(Gamma, [C|[D|E1]], G1),
6181 append(Gamma, [C|[C|[D|E1]]], G3),
6182 append(Gamma, [C|[D|E2]], G2),
6183 append(Gamma, [C|[D|[D|E2]]], G4),
6184 append(Gamma, [C|[D|E]], G),
6185 append(Delta, [A # B|Lambda], E),
6186 append(Delta, [A|Lambda], E1),
6187 append(Delta, [B|Lambda], E2),
6188 append(Gamma, [C|[C|[D|E]], G5),
6189 append(Gamma, [C|[C|[D|[D|E]]], G6),
6190 append(Gamma, [C|[C|[D|[D|E1]]], G7),
6191 append(Gamma, [C|[C|[D|[D|E2]]], G8))),
6192 :(append(S1, [C|[D|Lambda]], G1),
6193 append(S1, [C|[C|[D|Lambda]]], G3),
6194 append(S2, [C|[D|Lambda]], G2),
6195 append(S2, [C|[D|[D|Lambda]]], G4),
6196 append(S, [C|[D|Lambda]], G),
6197 append(Gamma, [A # B|Delta], S),
6198 append(Gamma, [A|Delta], S1),
6199 append(Gamma, [B|Delta], S2),
6200 append(S, [C|[C|[D|Lambda]]], G5),
6201 append(S, [C|[C|[D|[D|Lambda]]], G6),
6202 append(S1, [C|[C|[D|[D|Lambda]]], G7),
6203 append(S2, [C|[C|[D|[D|Lambda]]], G8))),
6204
6205 gensym(f,F1),
6206 assertz(proved(Z = [con(F1,C), G => T])),
6207 gensym(f,F2),
6208 assertz(proved(F1 = [con(F2,D), G5 => T])),
6209 assertz(proved(F2 = [oia(Z1,Z2,A # B), G6 => T])),
6210 assertz(proved(Z1 = [th(Z3,D), G7 => T1])),
6211 assertz(proved(Z2 = [th(Z4,C), G8 => T2])),
6212 retract(proved(Z = [oia(Z1,Z2,A # B), G => T])),
6213 retract(proved(Z1 = [con(Z3,C), G1 => T1])),
6214 retract(proved(Z2 = [con(Z4,D), G2 => T2])),
6215

```

```

6216 /* (R3,R3,R12) - (D.78.2.4) */
6217 norm2(proved(Z = [oia(Z1,Z2,A # B), G => T])) :-
6218   proved(Z1 = [con(Z3,D), G1 => T]),
6219   proved(Z2 = [con(Z4,C), G2 => T]),
6220   proved(Z3 = [-, G3 => T]),
6221   proved(Z4 = [-, G4 => T]),
6222   ((append(Gamma, [C|D|E1], G1),
6223     append(Gamma, [C|D|[D|E1]], G3),
6224     append(Gamma, [C|D|E2], G2),
6225     append(Gamma, [C|C|[D|E2]], G4),
6226     append(Gamma, [C|D|E1], G),
6227     append(Delta, [A # B|Lambda], E),
6228     append(Delta, [A|Lambda], F1),
6229     append(Delta, [B|Lambda], E2),
6230     append(Gamma, [C|C|[D|E1]], G5),
6231     append(Gamma, [C|C|[D|D|E1]], G6),
6232     append(Gamma, [C|C|[D|D|E1]], G7),
6233     append(Gamma, [C|C|[D|D|E2]], G8)),
6234   ; (append(S1, [C|D|Lambda], G1),
6235     append(S1, [C|D|[D|Lambda]], G3),
6236     append(S2, [C|D|Lambda], G2),
6237     append(S2, [C|[D|Lambda]], G4),
6238     append(S, [C|[D|Lambda]], G),
6239     append(Gamma, [A # B|Delta], S),
6240     append(Gamma, [A|Delta], S1),
6241     append(Gamma, [B|Delta], S2),
6242     append(S, [C|[C|[D|Lambda]], G5),
6243     append(S, [C|[C|[D|[D|Lambda]], G6),
6244     append(S1, [C|[C|[D|[D|Lambda]], G7),
6245     append(S2, [C|[C|[D|[D|Lambda]], G8))),
6246   !,
6247   gensym(f,F1),
6248   assertz(proved(Z = [con(F1,C), G => T])),
6249   gensym(f,F2),
6250   assertz(proved(F1 = [con(F2,D), G5 => T])),
6251   assertz(proved(F2 = [oia(Z1,Z2,A # B), G6 => T])),
6252   assertz(proved(Z1 = [th(Z3,C), G7 => T])),
6253   assertz(proved(Z2 = [th(Z4,D), G8 => T])),
6254   retract(proved(Z = [oia(Z1,Z2,A # B), G => T])),
6255   retract(proved(Z1 = [con(Z3,D), G1 => T])),
6256   retract(proved(Z2 = [con(Z4,C), G2 => T])),
6257   !.

```

```

6258 /* (R4,R4,R10) - (D.79) */
6259 norm2(proved(Z = [ais(Z1,Z2,A & B), G => T])) :-
6260   proved(Z1 = [inc(Z3,D,C), G => T1]),
6261   proved(Z2 = [-, G => T2]),
6262   proved(Z3 = [-, G1 => T1]),
6263   append(Gamma, [D|[C|Delta], G),
6264   append(Gamma, [C|[D|Delta], G1),
6265   append(Phi, [A & B|Psi], T),
6266   append(Phi, [A|Psi], T1),
6267   append(Phi, [B|Psi], T2),
6268   !,
6269   assertz(proved(Z = [inc(Z1,D,C), G => T])),
6270   gensym(f,F1),
6271   assertz(proved(Z1 = [ais(Z3,F1,A & B), G1 => T])),
6272   assertz(proved(F1 = [inc(Z2,C,D), G1 => T2])),
6273   retract(proved(Z = [ais(Z1,Z2,A & B), G => T])),
6274   retract(proved(Z1 = [inc(Z3,D,C), G => T1])),
6275   !.

```

```

6276 /* (R2) - (D.81.3) */
6277 norm2(proved(Z = [th(Z1,A -> B), G => []])) :-
6278   proved(Z1 = [K, G1 => []]),
6279   append(Gamma, [A -> B|Delta], G),
6280   append(Gamma, Delta, G1),
6281   !,
6282   gensym(f,F1),
6283   assertz(proved(Z = [con(F1,Gamma), G => []])),
6284   gensym(f,F2),
6285   append(Gamma, G, G2),
6286   assertz(proved(F1 = [con(F2,Delta), G2 => []])),
6287   append(G2, Delta, G3),
6288   gensym(Step,F3),
6289   assertz(proved(F2 = [inc(F3,A->B,Delta), G3=>[]])),
6290   gensym(f,F4),

```

```

6291      gensym(f,F5),
6292      append(Gamma, G1, S),
6293      append(S, [A -> B|Delta], G4),
6294      assertz(proved(F3 = [his(F4,F5,A->B), G4 => []])),
6295      assertz(proved(F4 = [th(Z1,A), G1 => [A]])),
6296      gensym(f,F6),
6297      append(Gamma, [B|Delta], G5),
6298      assertz(proved(F5 = [th(F6,B), G5 => []])),
6299      assertz(proved(F6 = [K, G1 => []])),
6300      retract(proved(Z = [th(Z1,A -> B), G => []])).
6301      !.

```

```

6302 /* (R5) - (D.82.3) */
6303 norm2(proved(Z = [th(Z1,A -> B), G => [A -> B]])) :-
6304     proved(Z1 = [-, G => []]),
6305     !,
6306     gensym(f,F1),
6307     assertz(proved(Z = [his(F1,A->B), G => [A->B]])),
6308     gensym(f,F2),
6309     append([A], G, G1),
6310     assertz(proved(F1 = [th(F2,A), G1 => [B]])),
6311     assertz(proved(F2 = [th(Z1,B), G => [B]])),
6312     retract(proved(Z = [th(Z1,A -> B), G => [A->B]])),
6313     !.

```

```

6314 /* (R5) - (D.82.4) */
6315 norm2(proved(Z = [his(Z1,A -> B), G => [A -> B]])) :-
6316     proved(Z1 = [th(Z2,A), G1 => [B]]),
6317     proved(Z2 = [th(Z3,B), G => [B]]),
6318     proved(Z3 = [-, G => []]),
6319     append(Gamma, [A|Delta], G1),
6320     append(Gamma, Delta, G),
6321     !,
6322     assertz(proved(Z = [his(Z1,A->B), G => [A->B]])),
6323     append([A], G, G2),
6324     assertz(proved(Z1 = [th(Z2,A), G2 => [B]])),
6325     retract(proved(Z = [his(Z1,A->B), G => [A->B]])),
6326     retract(proved(Z1 = [th(Z2,A), G1 => [B]])),
6327     !.

```

```

6328 norm2(proved(Z)) :-
6329     retract(proved(Z)),
6330     assertz(proved(Z)), !.

```

6331

/\* expinc \*/

6332 /\* This part is to deal with proof-steps with interchange  
6333 \* formulae : either empty or sequences of formulae \*/

```

6334 expinc :-
6335   proved(Z = [inc(Z1,A,B),K]),
6336   (A == [ ]
6337   ;B == [ ]
6338   ;A == [B]
6339   ;B == [A]
6340   ;B = A
6341   ;(A == [D],B == [D])),
6342   !,
6343   ((proved(Z2 = [F,M]),F == H,member(Z,H),sub(Z,Z1,H,H1),
6344     F1 == M1,assorta(proved(Z2 = [F1,M]))),
6345     retract(proved(Z2 = [F,M])),
6346     retract(proved(Z = [inc(Z1,A,B),K])))
6347   ;(retract(proved(Z = [inc(Z1,A,B),K])),proved(Z1 = X)),
6348     assorta(proved(Z = X)),retract(proved(Z1 = X))),
6349   expinc.

```

```

6350 expinc :-
6351   (proved(Z = [inc(Z1,[A],[B]),K])
6352   ;(proved(Z = [inc(Z1,[A],B),K]),not(islist(B)))
6353   ;(proved(Z = [inc(Z1,A,[B]),K]),not(islist(A)))).
6354   !,
6355   assorta(proved(Z = [inc(Z1,A,B),K])),
6356   (retract(proved(Z = [inc(Z1,[A],[B]),K]))
6357   ;retract(proved(Z = [inc(Z1,[A],B),K]))
6358   ;retract(proved(Z = [inc(Z1,A,[B]),K]))),
6359   expinc.

```

```

6360 expinc :-
6361   proved(Z = [inc(Z1,L,A),K]),
6362   not(islist(A)),L = [_|_],
6363   !,
6364   assorta(proved(Z = [inc(Z1,L,[A]),K])),
6365   retract(proved(Z = [inc(Z1,L,A),K])),
6366   expinc.

```

```

6367 expinc :-
6368   proved(Z = [inc(Z1,A,L),K]),
6369   not(islist(A)),L = [_|_],
6370   !,
6371   assorta(proved(Z = [inc(Z1,[A],L),K])),
6372   retract(proved(Z = [inc(Z1,A,L),K])),
6373   expinc.

```

6374 expinc :- expinca, expincs.

```

6375 expinca :-
6376   proved(Z = [inc(Z1,L2,L1),G => T]),
6377   proved(Z1 = [_,G1 => T]),
6378   L1 = [A|Rest1],L2 = [B|Rest2],
6379   (Rest1 \== [ ] ; Rest2 \== [ ]),
6380   !,
6381   break_up(G1,L1,L2,G,S,F),
6382   !,
6383   move_overo(L1,L2,S,F,Z1,T),
6384   retract(proved(Z = [inc(Z1,L2,L1),G => T])),
6385   proved(Y = [K,G => T]),
6386   !,
6387   assorta(proved(Z = [K,G => T])),
6388   retract(proved(Y = [K,G => T])),
6389   expinca.

```

6390 expinca.

```

6391 move_overo(L1,[],S,F,Z1,T) :-
6392   !.
6393 move_overo(N,M,S,F,Z1,T) :-
6394   M = [A|M1],
6395   singlemoveo(N,[A|M1],S,F,Z1,T),
6396   append(S,[A],S1),
6397   append(N,M1,N1),
6398   append(S1,N1,S2),
6399   append(S2,F,G),
6400   proved(F1 = [K,G => T]),
6401   move_overo(N,M1,S1,F,F1,T).

```

```

6402 singlemovea([],M,S,F,Z1,T) :-
6403     !.
6404 singlemovea(N,[A|M1],S,F,Z1,T) :-
6405     append(N1,[B],N),
6406     append([B],M1,M2),
6407     append(N1,[A|M2],M),
6408     append(S,M,S1),
6409     append(S1,F,G),
6410     gensym(f,F1),
6411     asserta(proved(F1 = [inc(Z1,A,B), G => T])),
6412     singlemovea(N1,[A|M2],S,F,F1,T).

6413 expincs :-
6414     proved(Z = [inc(Z1,L2,L1), G => T]),
6415     proved(Z1 = [_, G => T1]),
6416     L1 = [A|Rest1], L2 = [B|Rest2],
6417     (Rest1 \== [] ; Rest2 \== []),
6418     !,
6419     break_up(T1,L1,L2,T,S,F),
6420     !,
6421     move_overs(L1,L2,S,F,Z1,G),
6422     retract(proved(Z = [inc(Z1,L2,L1), G => T])),
6423     proved(Y = [K, G => T]),
6424     !,
6425     asserta(proved(Z = [K, G => T])),
6426     retract(proved(Y = [K, G => T])),
6427     expincs.
6428 expincs.

6429 break_up(E1,L1,L2,E,S,F) :-
6430     append(S,L1,S1),
6431     append(S1,L2,F1),
6432     append(F1,F,E1),
6433     append(S,L2,S2),
6434     append(S2,L1,F2),
6435     append(F2,F,E).

6436 move_overs(L1,[],S,F,K,E) :-
6437     !.
6438 move_overs(N,M,S,F,K,E) :-
6439     M = [A|M1],
6440     singlemoves(N,[A|M1],S,F,K,E),
6441     append(S,[A],S1),
6442     append(N,M1,N1),
6443     append(S1,N1,S2),
6444     append(S2,F,T),
6445     proved(F1 = [H,E => T]),
6446     move_overs(N,M1,S1,F,F1,E).

6447 singlemoves([],M,S,F,K,E) :-
6448     !.
6449 singlemoves(N,[A|M1],S,F,K,E) :-
6450     append(N1,[B],N),
6451     append([B],M1,M2),
6452     append(N1,[A|M2],M),
6453     append(S,M,S1),
6454     append(S1,F,T),
6455     gensym(f,F1),
6456     asserta(proved(F1 = [inc(K,A,B), E => T])),
6457     singlemoves(N1,[A|M2],S,F,F1,E).

6458                                     /* expcon */

6459 /* This part deals with proof-steps which have contraction
6460    formula either empty or a sequence of formulae */

6461 expcon :-
6462     proved(Z = [con(Z1,[],K)]),
6463     !,
6464     ((proved(Z2 = [F,M]),F =.. H,member(Z,H),sub(Z,Z1,H,H1),
6465        F1 =.. H1, asserta(proved(Z2 = [F1,M]))),
6466     retract(proved(Z2 = [F,M])),
6467     retract(proved(Z = [con(Z1,[],K)])))
6468     ;(retract(proved(Z = [con(Z1,[],K)]), proved(Z1 = X),
6469     asserta(proved(Z = X)), retract(proved(Z1 = X)))).
6470 expcon.

```

```

6471 expcon :-
6472     proved(Z = [con(Z1,[A]), K]),
6473     !,
6474     asserta(proved(Z = [con(Z1,A), K])),
6475     retract(proved(Z = [con(Z1,[A]), K])),
6476     expcon.
6477 expcon :- expcona, expcons.

6478 expcona :-
6479     proved(Z = [con(Z1,L), G => T]),
6480     proved(Z1 = [-, G1 => T]),
6481     L = [A|Rest], [Rest] \== [],
6482     !,
6483     break_up_con(G1,L,G,S,F),
6484     !,
6485     move_over_alla(L,L,S,F,Z1,T),
6486     retract(proved(Z = [con(Z1,L), G => T])),
6487     proved(Y = [K, G => T]),
6488     !,
6489     asserta(proved(Z = [K, G => T])),
6490     retract(proved(Y = [K,G => T])),
6491     expcona.
6492 expcona.

6493 move_over_alla(L,[],S,F,Z1,T) :-
6494     !.
6495 move_over_alla(L,N,S,F,Z1,T) :-
6496     N = [A|M],
6497     append(L1,[A|M],L),
6498     singlemovea(L1,A,M,[A|M],S,F,Z1,T),
6499     proved(F1 = [con(_,A),_]),
6500     move_over_alla(L,M,S,F,F1,T).

6501 singlemovea(L1,A,[],[A|M],S,F,K,T) :-
6502     !,
6503     append(L1,[A|M],L),
6504     append(S,L,S1),
6505     append(S1,F,G),
6506     gensym(f,F1),
6507     asserta(proved(F1 = [con(K,A), G => T])).
6508 singlemovea(L1,A,L2,[A|M],S,F,Z1,T) :-
6509     append(L3,[B],L2),
6510     append([B],M,N),
6511     append(L1,[A],L4),
6512     append(L4,L3,L5),
6513     append(L5,[A|N],L6),
6514     append(S,L6,S1),
6515     append(S1,F,G),
6516     gensym(f,F1),
6517     asserta(proved(F1 = [inc(Z1,A,B), G => T])),
6518     singlemovea(L1,A,L3,[A|N],S,F,F1,T).

6519 expcons :-
6520     proved(Z = [con(Z1,L), G => T]),
6521     proved(Z1 = [-, G => T]),
6522     L = [A|Rest], [Rest] \== [],
6523     !,
6524     break_up_con(T1,L,T,S,F),
6525     !,
6526     move_over_alla(L,L,S,F,Z1,G),
6527     retract(proved(Z = [con(Z1,L), G => T])),
6528     proved(Y = [K, G => T]),
6529     !,
6530     asserta(proved(Z = [K, G => T])),
6531     retract(proved(Y = [K,G => T])),
6532     expcons.
6533 expcons.

6534 break_up_con(E1,L,E,S,F) :-
6535     /* E is S then L then F */
6536     /* E1 is S then L then L then F */
6537     /* E1,L,E are imported */
6538     /* S, F are exported */
6539     append(S,L,S1),
6540     append(S1,F,E),
6541     append(S1,L,S2),
6542     append(S2,F,E1).

```

```

6543 move_over_alla(L, [], S, F, Z1, E) :-
6544     !.
6545 move_over_alla(L, N, S, F, Z1, E) :-
6546     N = [A|M],
6547     append(L1, [A|M], L),
6548     singlemoves(L1, A, M, [A|M], S, F, Z1, E),
6549     proved(F1 = [con(_, A), _]),
6550     move_over_alla(L, M, S, F, F1, E).

6551 singlemoves(L1, A, [], [A|M], S, F, K, E) :-
6552     !,
6553     append(L1, [A|M], L),
6554     append(S, L, S1),
6555     append(S1, F, T),
6556     gensym(f, F1),
6557     asserta(proved(F1 = [con(K, A), E => T])).
6558 singlemoves(L1, A, L2, [A|M], S, F, K, E) :-
6559     append(L3, [B], L2),
6560     append([B], M, N),
6561     append(L1, [A], L4),
6562     append(L4, L3, L5),
6563     append(L5, [A|N], L6),
6564     append(S, L6, S1),
6565     append(S1, F, T),
6566     gensym(f, F1),
6567     asserta(proved(F1 = [inc(K, A, B), E => T])),
6568     singlemoves(L1, A, L3, [A|N], S, F, F1, E).

6589 /* expth */

6570 /* This part deals with proof-steps which have thinning
6571     formula either empty or a sequence of formulae */

6572 expth :-
6573     proved(Z = [th(Z1, []), K]),
6574     !,
6575     ((proved(Z2 = [F, M]), F =.. H, member(Z, H), sub(Z, Z1, H, H1),
6576     F1 =.. H1, asserta(proved(Z2 = [F1, M])),
6577     retract(proved(Z2 = [F, M])),
6578     retract(proved(Z = [th(Z1, []), K])))
6579     ;(retract(proved(Z = [th(Z1, []), K])), proved(Z1 = X),
6580     asserta(proved(Z = X)), retract(proved(Z1 = X))),
6581     expth.

6582 expth :-
6583     proved(Z = [th(Z1, [A]), K]),
6584     !,
6585     asserta(proved(Z = [th(Z1, A), K])),
6586     retract(proved(Z = [th(Z1, [A]), K])),
6587     expth.

6588 expth :- exptha, expths.

6589 exptha :-
6590     proved(Z = [th(Z1, L), G => T]),
6591     proved(Z1 = [_, G1 => T]),
6592     L = [A|Rest], Rest \== [],
6593     !,
6594     break_up_th(G, L, G1, S, F),
6595     !,
6596     add_alla(L, S, F, Z1, T),
6597     retract(proved(Z = [th(Z1, L), G => T])),
6598     proved(Y = [K, G => T]),
6599     !,
6600     asserta(proved(Z = [K, G => T])),
6601     retract(proved(Y = [K, G => T])),
6602     exptha.
6603 expths.

6604 add_alla([], S, F, Z1, T) :-
6605     !.
6606 add_alla(N, S, F, Z1, T) :-
6607     N = [A|M],
6608     append(S, [A], S1),
6609     append(S1, F, G),
6610     gensym(f, F1),
6611     asserta(proved(F1 = [th(Z1, A), G => T])),
6612     add_alla(M, S1, F, F1, T).

```



```

6613  expths :-
6614      proved(Z = [th(Z1,L), G => T]),
6615      proved(Z1 = [_, G => T1]),
6616      L = [A|Rest], Rest \== [],
6617      !,
6618      break_up_th(T,L,T1,S,F),
6619      !,
6620      add_allis(L,S,F,Z1,G),
6621      retract(proved(Z = [th(Z1,L), G => T])),
6622      proved(Y = [K, G => T]),
6623      !,
6624      asserta(proved(Z = [K, G => T])),
6625      retract(proved(Y = [K, G => T])),
6626      expths.
6627  expths.

6628  break_up_th(E,L,E1,S,F):-
6629      append(S,L,S1),
6630      append(S1,F,E),
6631      append(S,F,E1).

6632  add_allis([],S,F,K,E) :-
6633      !.
6634  add_allis(N,S,F,K,E) :-
6635      N = [A|M],
6636      append(S,[A],S1),
6637      append(S1,F,T),
6638      gensym(f,F1),
6639      asserta(proved(F1 = [th(K,A), E => T])),
6640      add_allis(M,S1,F,F1,E).

6641                                     /* printall */

6642  /* This part is used to put all the proof in a list */
6643  put_in_list(X,L) :- proved(X = _),
6644                      prinlist(X,L).
6645  prinlist(X,L) :-
6646      prinlist(X,[],L).

6647  prinlist(X, Sofar, K) :-
6648      proved(X = Y), append([X = Y],Sofar,L2),
6649      ((nullary(X), K = L2)
6650      ;(unary(X,L), prinlist(L,L2,K))
6651      ;(binary(X,L,M), prinlist(L,L2,L4), prinlist(M,L4,K))
6652      ).

6653  /* gives a print on the screen for the proof in the database */
6654  printall(X) :- proved(X = Y),
6655                (nullary(X)
6656                ;(unary(X,L), printall(L))
6657                ;(binary(X,L,M), printall(L), printall(M))),
6658                nl, write(X = Y).

6659  nullary(X) :-
6660      proved(X = [Z,_]),
6661      (Z =.. [A] ; Z =.. [id,C]).

6662  unary(X,L) :-
6663      proved(X = [Z,_]),
6664      Z =.. [A|[L[_]],
6665      (A = th
6666      ;A = inc
6667      ;A = con
6668      ;A = ois
6669      ;A = ala
6670      ;A = his).

```

```

6671 binary(X, L, M) :-
6672     proved(X = [Z,_]),
6673     Z =.. [A,L,M,B],
6674     (A = cut
6675     ;A = oia
6676     ;A = oia
6677     ;A = hia).

6678 member(H, [H|_]).
6679 member(H, [_|_]) :- member(H,L).

6680 sub(X,C,[_],[_]).
6681 sub(X,C,[X|L],[C|L1]) :- !,
6682     sub(X,C,L,L1).
6683 sub(X,C,[Y|M],[Y|M1]) :-
6684     sub(X,C,M,M1).

6685 gensym(Root,Atom) :- /* gensym means generate symbols,
6686     gensym(f,N). it will give you N=f1,
6687     and continue giving you f2,f3,...
6688     if you repeat this question many
6689     times */
6690     get_num(Root,Num),
6691     name(Root,Name1),
6692     integer_name(Num,Name2),
6693     append(Name1,Name2,Name),
6694     name(Atom,Name).

6695 get_num(Root,Num) :-
6696     retract(current_num(Root,Num1)), !,
6697     Num is Num1 + 1,
6698     asserta(current_num(Root,Num)).
6699 get_num(Root,1) :- asserta(current_num(Root,1)).

6700 integer_name(Int,List) :- integer_name(Int,[],List).
6701 integer_name(I,Sofar,[C|Sofar]) :-
6702     I<10, !, C is I+48.
6703 integer_name(I,Sofar,List) :-
6704     Tophalf is floor(I/10), /* floor(I/10) gives the integer
6705     part of the division */
6706     Bothalf is I mod 10,
6707     C is Bothalf+48,
6708     integer_name(Tophalf,[C|Sofar],List).

6709 append([],L,L).
6710 append([M|L1],L2,[M|L3]) :-
6711     append(L1,L2,L3).

6712 append3(X,Y,Z,U) :-
6713     append(X,Y,Z1),
6714     append(Z1,Z,U).

6715 append4(X,Y,Z,V,U) :-
6716     append3(X,Y,Z,Z1),
6717     append(Z1,V,U).

6718 append5(X,Y,Z,U,V,W) :-
6719     append4(X,Y,Z,U,U1),
6720     append(U1,V,W).

6721 islist([]).
6722 islist([_|_]).
6723 islist([A|B]) :- islist(B).

6724 cop :-
6725     proved(Z1 = [K1,H1]),
6726     K1 =.. [L1|Arg1],member(X,Arg1),
6727     proved(Z2 = [K2,H2]), Z2 \== Z1,
6728     K2 =.. [L2|Arg2],member(X,Arg2),
6729     proved(X = [F,G]).

```

```

6730     gensym(f,F1),
6731     assert(proved(F1 = [F,G])),
6732     sub(X,F1,Arg1,Arg11), K11 =.. [L1|Arg11],
6733     assert(proved(Z1 = [K11,H1])),
6734     retract(proved(Z1 = [K1,H1])),
6735     gensym(f,F2),
6736     assert(proved(F2 = [F,G])),
6737     sub(X,F2,Arg2,Arg22), K22 =.. [L2|Arg22],
6738     assert(proved(Z2 = [K22,H2])),
6739     retract(proved(Z2 = [K2,H2])),
6740     cop.
6741 cop.

```

```

6742 branch(Z) :-
6743     proved(Z = Y),
6744     ( bra_nul(Z)
6745     ;(bra_unar(Z,Z1), branch(Z1))
6746     ;(bra_binar(Z,Z1,Z2), branch(Z1), branch(Z2))).

```

```

6747 bra_nul(Z) :-
6748     proved(Z = [X,Y]),
6749     (X =.. [_] ; X =.. [Id,_]).

```

```

6750 bra_unar(Z,Z1) :-
6751     proved(Z = [X,Y]),
6752     gensym(f,Z1),
6753     ((X =.. [H, L, T],
6754     X1 =.. [H, Z1, T])
6755     ;(X =.. [inc, L, T1, T2],
6756     X1 =.. [inc, Z1, T1, T2])),
6757     proved(L = K),
6758     assertz(proved(Z = [X1,Y])),
6759     assertz(proved(Z1 = K)),
6760     retract(proved(Z = [X,Y])).

```

```

6761 bra_binar(Z,Z1,Z2) :-
6762     proved(Z = [X,Y]),
6763     X =.. [H, L1, L2, T],
6764     proved(L1 = K1),
6765     proved(L2 = K2),
6766     gensym(f,Z1),
6767     gensym(f,Z2),
6768     X1 =.. [H, Z1, Z2, T],
6769     assertz(proved(Z = [X1,Y])),
6770     assertz(proved(Z1 = K1)),
6771     assertz(proved(Z2 = K2)),
6772     retract(proved(Z = [X,Y])).

```