

A Decision Procedure Revisited: Notes on Direct Logic, Linear Logic and its Implementation

by

Gianluigi Bellin and Jussi Ketonen

A Decision Procedure revisited:

LFCS Report Series

ECS-LFCS-91-167

LFCS

July 1991

Department of Computer Science
University of Edinburgh
The King's Buildings
Edinburgh EH9 3JZ

Copyright © 1991, LFCS

**Copyright © 1991, Laboratory for Foundations of Computer Science,
University of Edinburgh. All rights reserved.**

**Reproduction of all or part of this work
is permitted for educational or research use
on condition that this copyright notice is
included in any copy.**

**A Decision Procedure Revisited:
Notes on Direct Logic, Linear Logic
and its Implementation**

by
Gianluigi Bellin and Jussi Ketonen

(to appear in Theoretical Computer Science)

Research Sponsored by
Advanced Research Projects Agency
and
National Science Foundation

Department of Computer Science
Stanford University
Stanford, CA 94305

This version of the PROCES was printed on July 15, 1991.

CONTENTS

1.	Introduction	2
1.1.	Notation	3
1.2.	Sequents	5
1.3.	Relations with the Classical System	7
1.4.	Basic Proof Theory	8
2.	Linear logic and Direct Predicate Calculus	10
2.1.	Proof Nets	11
2.2.	Multiple Conclusion Natural Deduction	13
2.3.	Proof search in Sequent Calculi	14
3.	A Decision Procedure for Direct Predicate Logic	16
3.1.	Paths	16
3.2.	Chains	17
3.3.	The Main Theorem	18
3.4.	An Example	18
4.	Some Properties of Graphs	20
4.1.	Definitions	20
4.2.	Basic Properties of Chains	21
5.	Proof of the Main Theorem	23
5.1.	From Proofs to Chains	23
5.2.	From Chains to Proofs	24
6.	Equivalence with Girard's Proof Nets	27
7.	Implementation of the Decision Procedure for DPC	29
7.1.	The Goal	29
7.2.	Finding All Matches	30
7.3.	Simple Depth First Search	30
7.4.	Breadth First Search: Pre-Computing Valid Path Combinations	31
7.5.	Static Irrelevance Elimination: the First Refinement	31
7.6.	Modified Breadth First Search: Partial Pre-Computation	32
7.7.	Searching for Paths	33
7.8.	Dealing with Cycles	33
8.	Bibliography	35

Abstract

This paper studies decidable fragments of predicate calculus. We will focus on the structure of Direct Predicate Calculus as defined in [KETONEN and WEYHRAUCH 1984] in the light of the recent work of Girard [GIRARD 1987, 1988 A and B] on Linear Logic. Several graph-theoretic results are used to prove correspondences between systems of Natural Deduction, Direct Predicate Logic, and Linear Logic. In addition, the implementation of a decision procedure for Direct Predicate Logic is sketched.

1. Introduction

Much of the recent work in Artificial Intelligence (AI) and its applications uses predicate logic as its foundation. Yet, full predicate logic is known to be undecidable. Given an arbitrary formula, one cannot predict whether the formula is provable, and if so, how fast one can verify it. We are faced with the problem of *combinatorial explosion*: programs exhibiting supposedly intelligent behavior can get swamped in many unexpected ways. This situation can be somewhat remedied with a better understanding of heuristic approaches and programming tricks in universal proof methods such as resolution. However, we feel strongly that this methodology — derived from the everyday practice of AI programming — must be coupled with a better formal understanding of the use of predicate logic. For example, it is intuitively clear that in most situations the full power of predicate logic is never used: our intention is to mechanize simple-minded reasoning. We are led to study decidable fragments of predicate logic. For example, propositional logic is quite sufficient for simple knowledge representation tasks. Even though all known bounds for procedures for checking propositional tautologies are exponential, they are good enough in practice. However, if we go further and add unary predicates (i.e., study decision procedures for full monadic predicate calculus), we are faced with double exponential bounds [LEWIS 1978]. Other fragments defined by restricting the number of quantifiers that can occur also be proven to be decidable with extremely high bounds [DREBEN and GOLDFARB 1979]. We do not view these types of approaches — syntactic restrictions — as intuitively satisfactory. Our intention is not to restrict the expressiveness of our base language, merely the methods of proof and rules of deduction.

The paper of Ketonen and Weyhrauch [KETONEN and WEYHRAUCH 1984] defines a fragment of predicate calculus — Direct Predicate Calculus (DPC) — by eliminating the use of the rule of contraction. Intuitively, this means that every formula can be ‘used’ at most once in a proof. Thus, ‘tricky’ proofs such as proof by cases are not covered. For example, the formula

$$\exists y. \forall x. A(y) \supset A(x),$$

while provable, is not provable within DPC. It was shown that DPC admits a relatively simple decision procedure.

The work of Girard [GIRARD 1987, 1988] can be viewed as a logical extension of this research; by defining *multiplicative* and *linear* versions of all propositional connectives and their corresponding proof rules, one can gain a more refined analysis of decidable proof procedures.

Our paper will explore the connections between these two approaches. In particular, we demonstrate correspondences between the basic data structures used in Natural Deduction (proof trees), Linear Logic (proof nets), and Direct Predicate Logic (chains). Finally, we will sketch an implementation approach for the decision procedure for DPC.

1.1. Notation

Our basic language is that of pure first-order predicate logic. We are given a language \mathcal{L}_0 consisting of an infinite list of variable symbols x, y, z, \dots , a list P_i ($i = 1, 2, \dots$) of predicate symbols of arity $n_i \geq 1$, and a list of function symbols f_i ($i = 1, 2, \dots$) of arity $m_i \geq 1$. The notion of a term and a formula in the language \mathcal{L}_0 is then defined in the usual manner.

Definition: Any variable is a term. If t_1, \dots, t_n is a list of terms and f is an n -ary function symbol, then $f(t_1, \dots, t_n)$ is a term. If t_1, \dots, t_n is a list of terms and P is an n -ary predicate symbol, then $P(t_1, \dots, t_n)$ is an atomic formula. If x is a variable and ϕ, ψ are formulas, then so are

$$\neg\phi, \quad \phi \supset \psi, \quad \phi \wedge \psi, \quad \phi \vee \psi, \quad \forall x.\phi, \quad \exists x.\phi.$$

One can now go on and formulate a system of predicate logic in the expected fashion. Our interest lies in analyzing the notion of provability in this context. For this reason, we need to often distinguish between a *term* and its *occurrence* in a proof. For example, the two occurrences of a formula “ A ” in “ $A \wedge (A \supset C) \supset D$ ” might arise in different ways in a proof. We wish to make this explicit by constructing another language \mathcal{L} which will allow us to denote any term of \mathcal{L}_0 in infinitely many different ways. \mathcal{L} is constructed by assigning to each symbol s of \mathcal{L}_0 a countable list of distinct symbols O_s^1, O_s^2, \dots of the same arity and type.

In general, we use the letters A, B, C, \dots to denote formulas of \mathcal{L} and $\Gamma, \Pi, \Sigma, \dots$ for finite sequences of formulas. Comma will be used as a concatenation operator for sequences. The symbol “ \emptyset ” denotes the empty sequence. For any formula A , we use the notation $A(t_1, \dots, t_n)$ for A with all the free occurrences of the variable x_i replaced by t_i for $i = 1, 2, \dots$.

Define a partial mapping π from the terms and formulas of \mathcal{L} onto the terms and formulas of \mathcal{L}_0 by setting

$$\pi(O_s^i) = s$$

for all symbols and extending this by a straightforward induction to all objects of \mathcal{L} . Clearly, the set of pre-images of any term or a formula of \mathcal{L}_0 under π is infinite. We view π as a *fibration* of the base language \mathcal{L}_0 .

Two terms t, u of \mathcal{L} are called *similar* ($t \approx u$) if $\pi(t) = \pi(u)$.

A term t of \mathcal{L} is *separated* if any two similar occurrences of subterms in t are distinct elements of \mathcal{L} .

For any t in \mathcal{L} there is a separated u similar to it. Using this representation one can uniquely identify a term (or a formula) through its position in the proof of some other term or its appearance in some other term. This fact will make our arguments somewhat easier to formulate.

Define a partial order \prec on the set of all terms and formulas of \mathcal{L} as follows:

$$t \prec u \equiv_{\text{def}} t \text{ occurs in } u.$$

Thus $A \prec B$ if and only if A is a subformula of B if and only if A occurs above B in the tree of formulas.

A *positive* or *negative occurrence* of a subformula B in a formula A is defined inductively as follows:

- A is positive;
- let B be $C \wedge D$, $C \vee D$, $\forall x.C(x)$ or $\exists x.C(x)$: if B is positive [negative], then C , D , $C(t)$ are positive [negative];
- let B be $C \supset D$ or $\neg C$: if B is positive [negative] then C is negative [positive] and D is positive [negative].

A quantifier is called *essentially universal* if it is universal in a positive occurrence or existential in a negative occurrence; otherwise, the quantifier is called *essentially existential*. This terminology is extended to the variables bound by such quantifiers.

A *conjunctive subformula* in a set of formulas is either (i) a positive occurrence of a subformula of the form $A \wedge B$ or (ii) a negative occurrence of $A \supset B$ or (iii) a negative occurrence of $A \vee B$. Similarly, a *disjunctive subformula* is either a positive occurrence of $A \vee B$ or of $A \supset B$ or a negative occurrence of $A \wedge B$.

1.2. Sequents

A sequent S is a pair of sequences Γ, Δ of formulas, usually written as $\Gamma \vdash \Delta$. Γ is called the antecedent and Δ the succedent. For our purpose it is convenient to use the notation

$$\vdash \neg\Gamma, \Delta,$$

or even $\vdash \Gamma$, assuming an implicit partition between positive and negative formulas in Γ .

We set $\neg\neg B \equiv_{\text{def}} B$.

All of the notions of occurrences for formulas can be extended to sequents by interpreting sequents of the form $\vdash \neg A_1, \dots, \neg A_n, B_1, \dots, B_m$ say as implications of the form: $A_1 \wedge \dots \wedge A_n \supset B_1 \vee \dots \vee B_m$.

We define Axioms and Rules of Inference as follows.

Axioms are sequents $\vdash \neg A, B$ such that $A \approx B$.

An **Inference** is a relation between sequent(s) — the premise(s) — and a sequent — the conclusion —, written as usual

$$\frac{S_1}{S} \qquad \frac{S_1 \quad S_2}{S}$$

according to the rules indicated below.

Structural Rules are Weakening and Exchange; Contraction is excluded:

Contraction

$$\frac{\vdash \Gamma, A, A}{\vdash \Gamma, A}$$

Logical Rules are $\neg, \vee, \wedge, \supset, \forall, \exists$ Left and Right.

In this context it becomes essential to decide whether the rules for disjunction and conjunction are given an *additive* interpretation, e.g.,

$$\frac{\Gamma \vdash \Delta, A \quad \Gamma \vdash \Delta, B}{\Gamma \vdash \Delta, A \wedge B} \qquad \frac{A, \Gamma \vdash \Delta}{A \wedge B, \Gamma \vdash \Delta} \qquad \frac{\neg B, \Gamma \vdash \Delta}{A \wedge B, \Gamma \vdash \Delta}$$

or a *multiplicative* interpretation:

$$\frac{\Gamma \vdash \Delta, A \quad \Pi \vdash \Lambda, B}{\Gamma, \Pi \vdash \Delta, \Lambda, A \wedge B} \qquad \frac{A, B, \Gamma \vdash \Delta}{A \wedge B, \Gamma \vdash \Delta}$$

Symmetrically for disjunction.

Left Rules	Right Rules
<p style="text-align: center;">Weakening</p> $\frac{\vdash \Gamma}{\vdash \Gamma, A}$	
<p style="text-align: center;">Exchange</p> $\frac{\vdash \Gamma, A, B, \Delta}{\vdash \Gamma, B, A, \Delta}$	
<p style="text-align: center;">Conjunction</p> $\frac{\vdash \neg A, \neg B, \Gamma}{\vdash \neg(A \wedge B), \Gamma} \quad \bigg \quad \frac{\vdash \Gamma, A \quad \vdash \Delta, B}{\vdash \Gamma, \Delta, A \wedge B}$	
<p style="text-align: center;">Disjunction</p> $\frac{\vdash \neg A, \Gamma \quad \vdash \neg B, \Delta}{\vdash \neg(A \vee B), \Gamma, \Delta} \quad \bigg \quad \frac{\vdash \Gamma, A, B}{\vdash \Gamma, A \vee B}$	
<p style="text-align: center;">Implication</p> $\frac{\vdash \Gamma, A \quad \vdash \neg B, \Delta}{\vdash \neg(A \supset B), \Gamma, \Delta} \quad \bigg \quad \frac{\vdash \Gamma, \neg A, B, \Delta}{\vdash \Gamma, A \supset B, \Delta}$	
<p style="text-align: center;">Universal Quantification</p> $\frac{\vdash \neg A(t), \Gamma}{\vdash \neg \forall x. A(x), \Gamma} \quad \bigg \quad \frac{\vdash \Gamma, A(a)}{\vdash \Gamma, \forall x. A(x)}$ <p style="text-align: right;">where a does not occur in Γ, Δ.</p>	
<p style="text-align: center;">Existential Quantification</p> $\frac{\vdash \neg A(a), \Gamma}{\vdash \neg \exists y. A(y), \Delta} \quad \bigg \quad \frac{\vdash \Gamma, A(t)}{\vdash \Gamma, \exists x. A(x)}$ <p style="text-align: left;">where a does not occur in Γ, Δ.</p>	

In addition, the **Cut** rule has the form

$$\frac{\vdash \Gamma, A \quad \vdash B, \Delta}{\vdash \Gamma, \Delta}$$

where $A \approx \neg B$.

In each rule, the indicated formulas in the premise(s) are the *active* formulas and the indicated formula in the conclusion the *principal* formula. All the formulas in the indicated sets are *side* formulas. The active formulas of Cut are the *Cut formulas*. The active formula(s) are the *immediate ancestors* of the principal formula and the relation “... is an ancestor of ...” is the transitive closure of “... is an immediate ancestor of ...”.

Derivations are inductively defined as usual. The definition of a *branch* β in a derivation is also routine.

Definition. A derivation \mathcal{D} is *separated* if for every pair of distinct branches β_1, β_2 and every pair of terms t_1, t_2 with $t_1 \in \beta_1, t_2 \in \beta_2, t_1 \approx t_2$ implies $t_1 \neq t_2$.

We work with separated proofs.

1.3. Relations with the Classical System

The relations between the languages \mathcal{L}_0 and \mathcal{L} and between Classical and Direct Sequent Calculus could be further explored by introducing in \mathcal{L} the notation $A^{(n)}$, with the meaning $A \circ A \circ \dots \circ A$ (n times), where “ \circ ” is “ \vee ” in a positive context and “ \wedge ” in a negative context. One can then introduce the following approximation of Contraction

$$\frac{\vdash \Gamma, A^{(n)}, B^{(m)}, \Delta}{\vdash \Gamma, A^{(n+m)}, \Delta}$$

where $A \approx B$.

Next, one may consider a Generalized Cut rule, or Mix

$$\frac{\vdash \Gamma, A^{(n)} \quad \vdash B^{(m)}, \Delta}{\vdash \Gamma, \Delta}$$

where $A \approx \neg B$. The trouble with Mix is that it cannot be reduced to Cut: one completely loses control over the number of iterations. Thus Mix must be forbidden from Direct Logic. On the other hand, the rule

$$\frac{\vdash \Gamma, A \quad \vdash B^{(m)}, \Delta}{\vdash \Gamma^{(m)}, \Delta}$$

with $A \approx \neg B$, is definable in terms of Cut, and yields an (exponential) bound on the number of needed duplication. The dynamical analysis of Contraction (i.e., with respect to Cut elimination) in classical logic is a main concern and motivation for a logic as ours. However, the issue will not be pursued further in this paper.

The following is clear:

Approximation Theorem. *Let S be any sequent in \mathcal{L}_0 . S is derivable in classical sequent calculus LK iff there is a sequent S^* in \mathcal{L} , derivable in Direct sequent calculus such that $\pi(S^*) = S$. ■*

1.4. Basic Proof Theory

Cut Elimination Theorem: *Every derivation \mathcal{D} in Direct Sequent Calculus can be transformed into a Cut free derivation \mathcal{D}' . ■*

The length of \mathcal{D}' grows exponentially, as usual.

Let \mathcal{D} be a cut free derivation of S in which all axioms involve atomic formulas. Without loss of generality, we may also assume that all principal formulas in applications of a Weakening are atomic.

Corollary (Strong Subformula Property): *There exists a bijection between the subformulas of S and the active formulas of inferences of \mathcal{D} . ■*

Corollary (Midsequent Theorem): *Let S be a sequent containing only prenex formulas, derivable in Direct Sequent Calculus. There exists a Cut free derivation \mathcal{D} of S and a sequent S_0 in \mathcal{D} with the following properties:*

- (i): S_0 results from S by erasing all quantifiers and replacing the variables by terms;
- (ii): every inference above S_0 is propositional and every inference below S_0 is quantificational. ■

We may assume that different variables are used in different occurrences of quantifiers.

Herbrand's Theorem has nice properties in Direct Logic. We recall the definition of *Herbrand function*. In a sequent S , let $Qy.A(y)$ be an essentially universal subformula which lies in the scope of the essentially existential quantifiers Qx_1, \dots, Qx_n : then $y[x_1, \dots, x_n]$ is the Herbrand term associated with Qy .

The *Herbrand form* $S_H(x_1, \dots, x_n)$ of a sequent S is the result of erasing all quantifiers of S and of replacing each essentially universal variable with the corresponding Herbrand function. (Here x_1, \dots, x_n are all the essentially existential variables in S).

Corollary (Herbrand Theorem):[†] *Let S be a sequent containing only prenex formulas. Then S is derivable in Direct Predicate Calculus iff there are terms t_1, \dots, t_n such that $S_H(t_1, \dots, t_n)$ is derivable in Direct Propositional Calculus. ■*

[†] The theorem is not true for non prenex formulas. Counterexample:

$$(\exists y.A(y)) \vee B \supset (\exists x.A(x) \vee B).$$

It is true, however, that given S , there exists $S^* \approx S$ such that some subformulas of S are fibrated in S^* and the theorem holds for S^* . We will not pursue this topic here.

2. Linear logic and Direct Predicate Calculus

Girard's Linear Logic provides a comprehensive framework in which the above issues can be discussed. The context is particularly stimulating, since connections are exhibited between results of Proof Theory, Category and Domain Theory, Linear Algebra and Banach Algebras [GIRARD 1987, 1988 A and B].

Developing ideas already present in the literature on Contraction free systems [ONO 1988] and in Relevance Logic (see [AVRON 1987]), Girard gives an instructive picture of the structure underlying Classical and Intuitionistic Logic.

The connectives of Linear Logic and their dual are organized in 5 levels:

- (1) the self-dual *linear negation* $(.)^\perp$;
- (2) multiplicative conjunction \otimes (*times*) and the dual disjunction \sqcup (*par*), with their identities, namely, 1 and \perp ;
- (3) additive disjunction \oplus (*plus*), and the dual conjunction $\&$ (*with*), with the identities 0 and \top , respectively;
- (4) the *exponentials*: $!$ (*of course*) and its dual $?$ (*why not*)
- (5) additive quantifiers \bigwedge (*every*) and the dual \bigvee (*some*).

Linear negation is defined for non atomic formulas, linear implication is defined as multiplicative.

At the moment (November 1988) we are aware of 5 different semantics for Linear Logic: Girard's *Phase Semantics*, *Coherent Semantics* [GIRARD 1987] and the interpretation in C^* Algebras [GIRARD 1988 B]; Lafont's interpretation in Linear Algebra [LAFONT 1988] and Sambin's in *Formal Spaces* [SAMBIN 1988]. †

Sequent Calculi for Linear Logic are obtained by eliminating both structural rules of Weakening and Contraction.

The *Multiplicative Fragment* (levels (1) and (2)) has logical rules for \otimes and \sqcup

$$\frac{\vdash \Gamma, A \quad \vdash \Delta, B}{\vdash \Gamma, \Delta, A \otimes B} \qquad \frac{\vdash A, B, \Gamma}{\vdash A \sqcup B, \Gamma}$$

the axiom $\vdash 1$ and the rule Weakening, but only with the constant \perp as principal formula.

The *Additives* (level 3) satisfy the rules

$$\frac{\vdash \Gamma, A \quad \vdash \Gamma, B}{\vdash \Gamma, A \& B} \qquad \frac{\vdash A, \Gamma}{\vdash A \oplus B, \Gamma} \qquad \frac{\vdash B, \Gamma}{\vdash A \oplus B, \Gamma}$$

† We will not say anything about semantics. The reader of [GIRARD 1987] may want do the following easy Exercise for Chapter 1: *Direct logic is sound and complete with respect to Phase Structures with $0 = \perp$* . In this Chapter we work within the limits of Proof Theory — with the prospect of applications for automatic formalization of proofs.

and the axiom $\vdash \top, \Gamma$ (given that \top and $\mathbf{0}$ are dual, this means “ $\mathbf{0}$ linearly implies everything”).

Finally, for the *Exponentials* (level 4) there are the rules

$$\frac{\vdash ?\Gamma, A}{\vdash ?\Gamma, !A} \qquad \frac{\vdash \Gamma, A}{\vdash \Gamma, ?A}$$

where $?\Gamma$ means that all formulas in Γ begin with $?$; in addition, Contraction and Weakening are allowed for formulas beginning with $?$. It is easy to see that the rule of Mix is not derivable in this system (see Section 1.3).

It is clear that Propositional Direct Logic is exactly the Multiplicative fragment with Weakening for arbitrary formulas.

2.1. Proof Nets

In addition to Sequent Calculus, the Proof Theory of Linear Logic consists of the new and suggestive notion of Proof Net. We will be mainly interested in this notion for the Multiplicative Fragment.

Given a set S of propositional formulas in the multiplicative language, a *Proof Net* comprises the following set of data:

(1) the set S^\prec of subformulas of S arranged in the obvious tree structure. A *link* is the relation between a formula (the *conclusion* of the link) and its immediate subformulas (the *premises* of the link);

(2) a set \mathcal{P} of *Axiom links*, i.e., connections between positive and negative occurrences of the same formula (the *conclusions* of the axiom link). Here we consider axiom links with atomic formulas only. Each occurrence of atomic formula in S^\prec is a conclusion of *at most one* axiom link.

A pair (S^\prec, \mathcal{P}) is a *Proof Structure* for (Multiplicative) Linear Logic if
— (*relevance condition*) all atoms of S^\prec are conclusions of *exactly one* axiom link.

(3) A Proof Structure is a Proof Net if it satisfies a graph theoretic condition.

Thus, a Proof Structure for the Multiplicative Fragment is built using the following links, with the condition that every formula occurrence is a consequence of one link and premise of at most one link.

Girard Multiplicative Links	
Axioms $\frac{}{A \quad A^\perp}$	
Times $\frac{A \quad B}{A \otimes B}$	
Par $\frac{A \quad B}{A \sqcup B}$	

The graph theoretic condition (3) is defined in terms of *trips* over (S^\prec, \mathcal{P}) . A *trip* visits the formulas of a proof structure in two directions, \uparrow and \downarrow , the movements being determined by the nature of the link and by arbitrary choices (*switches*) as follows:

Axioms

$$A^\wedge, A^\perp_v; \quad A^{\perp^\wedge}, A_v$$

Conclusions

$$C_v, C^\wedge$$

Times

Switch L	$A \otimes B^\wedge, B^\wedge;$	$B_v, A^\wedge;$	$A_v, A \otimes B_v;$
Switch R	$A \otimes B^\wedge, A^\wedge;$	$A_v, B^\wedge;$	$B_v, A \otimes B_v;$

Par

Switch L	$A \sqcup B^\wedge, A^\wedge;$	$B_v, B^\wedge;$	$A_v, A \sqcup B_v;$
Switch R	$A \sqcup B^\wedge, B^\wedge;$	$B_v, A^\wedge;$	$B_v, A \sqcup B_v;$

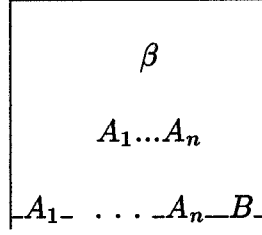
A proof structure β with n formula occurrences is a *proof net* if for every position of the switches the resulting trip does not return to the starting point in the same direction in less than $2n$ steps.

A main theorem of [GIRARD 1987] is

Theorem S is provable in *Multiplicative Linear Sequent Calculus* if and only if there is a \mathcal{P} such that (S^{\neg}, \mathcal{P}) is a *Proof Net*. ■

(The “if” direction is called *Sequentialization Theorem*. For a simplification of the proof, see also [GIRARD 1988 A], *Quantifiers in Linear Logic*, II.1., Remark 2, p. 21.)

Proof structures for the multiplicative fragment can be extended to a rule of Weakening by using a *box*: if β is a proof structure (with boxes) with conclusions A_1, \dots, A_n , then the following is a proof structure with boxes:



The list of formulas at the bottom of the box may be regarded as an extended Axiom for the structure outside it. If σ is any cyclic permutation of $n + 1$, then $C_i^{\wedge}, C_{\sigma(i)}^{\vee}$ (for $C_i = A_1, \dots, A_n, B$) determines the trip at the outside of the box — independently of the trips inside the box.

A proof structure with boxes is defined to be a proof net if for each box both the structure inside each box and the structure outside are proof nets.

The main theorem can be extended to Proof Nets with Weakening Boxes.

2.2. Multiple Conclusion Natural Deduction

Girard’s notion of *Proof Net* requires us to reconsider the following question:

What are the relevant features of Natural Deduction Systems vs. Sequent Calculi?

The issue is not just one of terminology. Both Natural Deduction and Sequent Calculi

(i) create links between occurrences of formulas — roughly corresponding, in informal reasoning, to the sequence of statements in an argument. In Natural Deduction these links are given by the *Rules of Inference*. In Sequent Calculus they are the relation between *active* and *principal* formulas in an inference.

(ii) establish relations between the occurrences of formulas in (i) and the context — roughly corresponding in informal reasoning to the *structure of an argument* (what are the presuppositions? what is the conclusion?) and *conditions for its correctness*. For instance, a sequent is precisely a notational device that keeps track of *side* formulas. In Natural Deduction such specifications are given, e.g., as *Rules of Deduction* (see [PRAWITZ 1965], Chapter I). In both formalisms there are restrictions on *eigenvariables*, i.e., the free variables that become bound in an (essentially) universal quantification. Typically, these restrictions are relations between the active premise of a quantification and its context.

A Sequent Calculus can be characterized as a formal system where (i) and (ii) are local, i.e., given simultaneously by the rules of inference — this is one of the reasons why Sequent Calculus is an efficient mathematical tool. On the contrary, Natural Deduction is a formal system in which (i) is local and (ii) is global.

According to the above classification, Proof Nets are clearly to be regarded as a Natural Deduction system. A Multiple Conclusion Natural Deduction (MCND) system is one in which formula occurrences are arranged as a directed acyclic graph, rather than as a tree. Proof Nets are a MCND system.

This view may encounter objections:

(a) Natural Deduction is a system for deducibility from assumptions as opposed to derivability from logical axioms, as in Sequent Calculus;

(b) essential feature of Natural Deduction is the presence of Introduction and Elimination rules and a certain logical priority of the Introduction rules.

To (a): The present arrangement of Multiplicative Proof Structures is very convenient and economical and makes assumptions and conclusion interchangeable, at the cost of giving up the functional character of the implication rule. However, another MCND system for the Multiplicative Fragment could be designed that has multiple-premise rules for \otimes -Introduction and \sqcup -Introduction, multiple-conclusion rules for \otimes -Elimination and \sqcup -Elimination, as well as Introduction and Elimination rules for Linear Implication of more traditional kind. Constraints to guarantee consistency of such system could easily be described by adapting the “No Short Trip” condition.

To (b): Proof Nets are a system with only Introduction rules, thanks to the duality of the connectives. However, something of the intuition behind the Introduction-Elimination classification is preserved, in a certain priority of the connectives themselves: for instance, the notion of \otimes seems to be more easy to understand than that of \sqcup and \oplus easier than $\&$.

2.3. Proof search in Sequent Calculi

The above considerations are relevant to the discussion of more mundane issues, like the computational content of derivations in Classical Logic. It is inefficiently represented by the standard systems of Sequent Calculus and Natural Deduction.

Prawitz’s treatment of Intuitionistic Logic in Natural Deduction seems definitive from the point of view of Proof Theory: the proofs of very strong results (e.g., Strong Normalization) are elegant and the weak spots (e.g., the permutability of \exists -elimination and \vee -elimination) may be regarded as relatively minor inconveniences. Also, the computational content of Intuitionistic Logic is well represented by Natural Deduction, given the Curry-Howard analogy with λ -calculus. However, Prawitz’s Natural Deduction system for Classical Logic is inadequate, since it uses essentially the negative translation into Intuitionistic Logic.

Sequent Calculi for Classical Logic effectively express semantical ideas and arguments (see Schutte's *Valuations* and Girard's work on the semantics of Cut-free proofs). Computationally, sequent calculus is less satisfactory. For example, the standard algorithm for searching for proofs — the *Wang algorithm* — (based on the invertibility of the rules of Sequent Calculus (cfr. [O. KETONEN 1944])), is relatively inefficient even for propositional calculus.

Here we write the tree from bottom up, breaking alternately the leftmost formula in the antecedent (i.e., at the left of \vdash) and the leftmost formula in the consequent (i.e., at the right of \vdash). In a system with Contraction, we also rewrite the formula under consideration in the upper sequent(s). Then we keep going upwards and break formulas on each branch until the process enters a loop on each branch — or, in the case of a system without Contraction, until there is no formula to break.

The very locality of the rules of sequent calculus forces us to neglect the global 'leit-motifs' of proofs. For example, the mechanical application of inverted 2-premise rules not only duplicates the work to be done in every succeeding step but may also be unnecessary. The 'global picture', i.e., the natural order of application of inference rules, cannot be deciphered through the application of this kind of formalism. The problems presented by the classical resolution approach are similar: again we are faced with the necessity of a global, conjunctive normal form — a transformation that erases the local connections used in natural proof generation.

Consider, e.g., the treatment of the right conjunction rule in a system without Contraction: an *additive* interpretation must be taken. Notice that this is incompatible with the notion of "formulas as trees". A *multiplicative* interpretation would be attractive because formulas in Γ and Δ would be broken just once. However to implement it *intelligently*, we need a global consideration of the role of the side formulas in the proof.

In terms of Girard's *trips* one can effectively express the main task of any reasonable procedure: to continue the search from a formula occurrence of the form $A \wedge B$ (multiplicative interpretation) one needs two separate 'explorations' of the context, one from A and the other from B , or:

Lemma: *Any trip in a proof net containing the link*

$$\frac{A \quad B}{A \otimes B}$$

has either the form

$$A^\wedge, \dots, A_\vee, B^\wedge, \dots, B_\vee, (A \otimes B)_\vee, \dots, (A \otimes B)^\wedge, A^\wedge$$

or

$$A^\wedge, \dots, A_\vee, (A \otimes B)_\vee, \dots, (A \otimes B)^\wedge, B^\wedge, \dots, B_\vee, A^\wedge,$$

where here are no visits to A , B and $A \otimes B$ other than the indicated ones. (see [GIRARD 1984], Lemma 2.9.1.).

3. A Decision Procedure for Direct Predicate Logic

Let S be a set of closed first order formulas of \mathcal{L} in prenex normal form. We shall outline below a procedure (and the appropriate abstract data structures) for deciding whether

$$\vdash S$$

in Direct Predicate Calculus.

We can consider the Herbrand form $S_H(x_1, \dots, x_n)$ of S (as in Section 1.4). Our problem reduces to finding a substitution $\sigma = (x_1/t_1, \dots, x_n/t_n)$ such that

$$S_H(t_1, \dots, t_n)$$

is provable in Direct Propositional Logic.

3.1. Paths

The first step of the procedure is the search for an *open path* through $S_H(x_1, \dots, x_n)$, i.e., a set $\mathcal{P}(x_1, \dots, x_n)$ of pairs of atoms (P, P') .

Definitions.(i) An *open path* \mathcal{P} is a set of pairs of atomic formulas such that:

- (a) if $(P, P'), (Q, Q')$ are two distinct members of \mathcal{P} then $P \neq Q$ and $P' \neq Q'$.
- (ii) We say that \mathcal{P} *satisfies* a formula A (in symbols $\mathcal{P} \mapsto A$) if there is a pair (P, P') in \mathcal{P} such that either $P \prec A$ or $P' \prec A$.
- (iii) Let S be a set of formulas. We say that \mathcal{P} is a *path for* S if
 - (b) \mathcal{P} satisfies some formula in S ;
 - (c) for all conjunctive subformulas $A \circ B$ in S , if $\mathcal{P} \mapsto A \circ B$ then $\mathcal{P} \mapsto A$ and $\mathcal{P} \mapsto B$ (*relevance condition for conjunctions*);
 - (d) for all $(P, P') \in \mathcal{P}$, P occurs positively in S and P' occurs negatively in S .
- (iv) a path \mathcal{P} for S is *minimal* if no proper subset \mathcal{P}' of \mathcal{P} is a path for S .

3.2. Chains

Consider the tree S_H^\prec of all subformulas of $S_H(x_1, \dots, x_n)$, as in Section 2.1 .

It is convenient to mark the conjunctive subformulas of S_H^\prec :

Multiplicative Links	
And	
$\frac{\neg A \quad \neg B}{\dots\dots\dots \neg(A \wedge B)}$	$\frac{A \quad B}{A \wedge B}$
Or	
$\frac{\neg A \quad \neg B}{\neg(A \vee B)}$	$\frac{A \quad B}{\dots\dots\dots A \vee B}$
Implies	
$\frac{A \quad \neg B}{\neg(A \supset B)}$	$\frac{\neg A \quad B}{\dots\dots\dots A \supset B}$

Consider the tree $S_H^\prec(x_1, \dots, x_n)$, together with an open path $\mathcal{P}(x_1, \dots, x_n)$ and a substitution $\sigma = (x_1/t_1, \dots, x_n/t_n)$ such that

(α) If $(P, P') \in \mathcal{P}$, then $P[x_1/t_1, \dots, x_n/t_n]$ and $P'[x_1/t_1, \dots, x_n/t_n]$ are similar.

Then the pair $(S^{\sigma, \prec}, \mathcal{P}^\sigma)$ with $S^{\sigma, \prec} = S_H^\prec(t_1, \dots, t_n)$ and $\mathcal{P}^\sigma = \mathcal{P}(t_1, \dots, t_n)$ (*closed path*) can be regarded as a proof structure in the sense of Section 2.1 , except that in Direct Logic the relevance condition is relaxed to a *relevance condition for conjunctive subformulas*.

Definitions. (i) For $A, B \in S^{\sigma, \prec}$ $A \neq B$, say that A and B are *connected* (write $A \parallel B$) if there is a pair $(P, P') \in \mathcal{P}^\sigma$ such that $P \prec A$ and $P' \prec B$ (or viceversa).

(ii) Let $X, Y \in S^{\sigma, \prec}$ be such that $X \not\prec Y$ and $Y \not\prec X$, with $A \prec X$ and $B \prec Y$. We write $X_A -_B Y$ if $A \parallel B$.

(iii) For $X \in S^{\sigma, \prec}$ we write $_A X_B$ if $A \circ B$ is a conjunctive subformula of X .

(iv) Let $\mathcal{C} = C_0, \dots, C_n$ be a set of subformulas in $(S^{\sigma, \prec}, \mathcal{P}^\sigma)$ with $n > 0$ and $C_i \neq C_j$ for all $0 \leq i \neq j \leq n$. \mathcal{C} is called a *cycle* if

$$_A C_{0B} -_{A_1} C_{1B_1} - \dots -_{A_n} C_{nB_n} -_A C_0.$$

The graph theoretic conditions that select the proof structures corresponding to a correct proof are the following:

- (β) If $\mathcal{P} \mapsto A \wedge B$, then not $A \parallel B$.
- (γ) There is no conjunctive cycle.

3.3. The Main Theorem

The main theorem (Theorem 4.4.) of [KETONEN and WEYHRAUCH 1984] is:

Theorem *For any sequence t_1, \dots, t_n of terms, the sequent $S_H[x_1/t_1, \dots, x_n/t_n]$ is provable in Direct Sequent Calculus if and only if we can find a minimal path \mathcal{P} for $S_H[x_1/t_1, \dots, x_n/t_n]$ satisfying:*

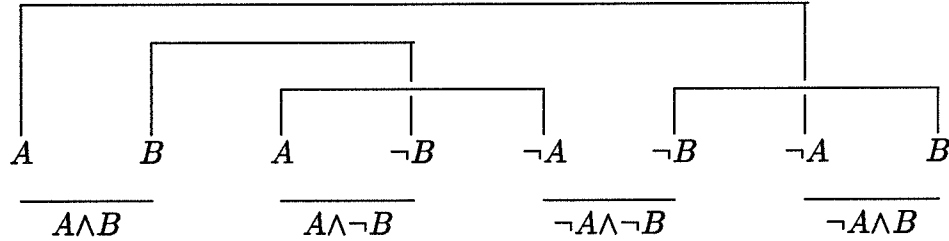
- (α) *If $(P, P') \in \mathcal{P}$, then $P[x_1/t_1, \dots, x_n/t_n]$ and $P'[x_1/t_1, \dots, x_n/t_n]$ are similar.*
- (β) *If $A \circ B$ is a conjunctive subformula and $\mathcal{P} \mapsto A \circ B$, then not $A \parallel B$.*
- (γ) *There is no conjunctive cycle.*

3.4. An Example

Consider the set of formulas

$$S = \{A \wedge B, A \wedge \neg B, \neg A \wedge \neg B, \neg A \wedge B\}.$$

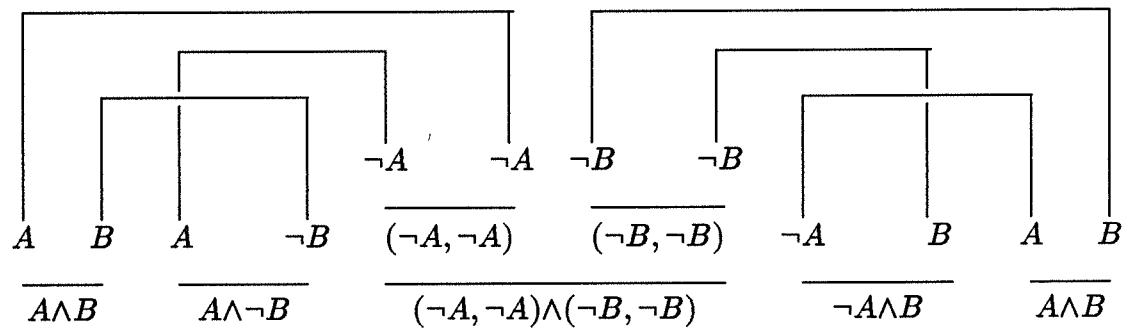
Construct the tree of subformulas S^\prec : every path \mathcal{P} for S^\prec creates a conjunctive cycle. For instance:



Therefore S is not provable in Direct Logic. To find a set S' , provable in Direct Logic, such that its projection $\pi(S')$ is S , notice that we must break the conjunctive cycle into two loops. The simplest S' will have the form

$$S' = \{(A \wedge B, A \wedge B), A \wedge \neg B, (\neg A, \neg A) \wedge (\neg B, \neg B), \neg A \wedge B\}.$$

In other words, we need to duplicate both a 'top level conjunction' and the subformulas of a conjunction.



4. Some Properties of Graphs

In this section we formulate abstractly some properties of graphs to be applied to formulas in trees of subformulas.

4.1. Definitions

Let $G = (V, E)$ be a graph with V finite and let F be a set of subsets of V . Also, let C be a set of distinguished elements of F , such that each $C \in C$ is partitioned by some elements A_1, \dots, A_n of F . (The set of singleton subsets of V will be interpreted as the set of all atomic formulas, F as the set of all formulas and C as the set of *conjunctive* subformulas, where each $C \in C$ is written in conjunctive normal form as $A_1 \wedge \dots \wedge A_n$. Finally, E as the set of connections on V determined by a unification of positive and negative atoms.

(i) For $A, B \in F$ with $A \cap B = \emptyset$, say that A and B are *connected* (write $A \parallel B$) if there is an edge e in E with vertices v_1 and v_2 such that $v_1 \in A$ and $v_2 \in B$.

(ii) Let $X, Y, A, B \in C$. We write $X_A -_B Y$ if $X \cap Y = \emptyset$, $A \subset X$, $B \subset Y$ and, moreover, $A \parallel B$. Think of A and B as “doors” of X and Y , respectively.

(iii) We write $_A X_B$ if A and B are different sets in the partition of X . A *chain* is a sequence X_1, \dots, X_n such that $Y -_{A_1} X_{1B_1} - \dots -_{A_n} X_{nB_n} - Z$.

(iv) A chain is *pure* if for all $i \neq j$ with $1 \leq i, j \leq n$, $X_i \neq X_j$.

Let $C = X_1 - \dots - X_n$ be a *nonempty* pure chain.

(v) $_A Y_B - C -_A Y$ is called a *cycle*.

(vi) $Y_B - C -_A Y$ is called a *loop* if $A = B$. Y is the *exit* of the loop.

(vii) A chain C is *terminal* if there is a formula in C from which C cannot be continued.

(viii) $X \gg^1 Y$ iff there is a loop

$$\mathcal{L} : Z_A - \dots - X - \dots -_A Z$$

and either $Y = Z$ or there is a pure chain

$$C :_A Z_B - \dots - Y.$$

We summarize this condition by saying that Y is dominated by the loop, which X belongs to.

(ix) $X \gg Y$ iff $X \gg^1 Y$ and not $Y \gg^1 X$.

Remark. Let \mathcal{L} be a loop with exit Z and let C be a pure chain starting with Z . If $C \cap \mathcal{L} \neq \{Z\}$ then there must be a cycle.

Indeed, suppose

$$\mathcal{L} : Z_A - \dots -_C U_D - \dots -_A Z$$

and

$$\mathcal{C} :_A Z_B - \dots -_E U - \dots - Y.$$

Now either

$${}_E U_D - \dots -_A Z_B - \dots -_E U \quad \text{or} \quad {}_E U_C - \dots -_A Z_B - \dots -_E U$$

or both are cycles, depending on whether $E = C$, $E = D$ or $C \neq E \neq D$, respectively. ■

4.2. Basic Properties of Chains

Lemma 1. *Suppose no chain is a cycle. Then \gg is a strict partial ordering.*

Proof. To show that $X \gg Y$ and $Y \gg Z$ implies $X \gg Z$, we need only to show that for all U, V, W , $U \gg V$ and $V \gg^1 W$ implies $U \gg^1 W$. Indeed, assume $X \gg Y$ and $Y \gg Z$. Then certainly $X \gg^1 Z$. Moreover, given $Y \gg Z$, if $Z \gg^1 X$, then $Y \gg^1 X$, a contradiction.

Assume there are loops $\mathcal{L}_1, \mathcal{L}_2$ and pure chains $\mathcal{C}_1, \mathcal{C}_2$

$$\mathcal{L}_1 : \quad U_A - X_1 - \dots - X - \dots - X_n -_A U$$

$$\mathcal{C}_1 : \quad {}_A U_B - U_1 - U_2 - \dots - U_i -_E Y$$

$$\mathcal{L}_2 : \quad V_C - X'_1 - \dots -_F Y_G - \dots - X'_m -_C V$$

$$\mathcal{C}_2 : \quad {}_C V_D - V_1 - V_2 - \dots - Z$$

Clearly, either ${}_E Y_G$ or ${}_E Y_F$ or both, depending on whether $E = F$, $E = G$ or $F \neq E \neq G$, say the latter.

Then we claim that

$$U_B - \dots -_E Y_G - \dots -_C V_D - \dots - Z$$

is a pure chain. Since there are no cycles, $\mathcal{C}_2 \cap \mathcal{L}_2 = \{V\}$, as remarked above. On the other hand, if $U^* \in \mathcal{C}_1 \cap \mathcal{C}_2$, then either we have a cycle

$${}_H U_L^* - \dots - Y - \dots - V - \dots -_H U^*$$

or

$$V_D - \dots -_H U_L^* - \dots - U - \dots - X$$

is a pure chain and $Y \gg^1 X$, against the hypothesis. ■

Let \mathbf{T} be a subset of \mathbf{C} such that for each $X, Y \in \mathbf{T}$ if $X \neq Y$, then $X \cap Y = \emptyset$ and for each $A \in \mathbf{C}$, if $X \cap A \neq \emptyset$ then $A \subset X$. The intended interpretation of \mathbf{T} is the set of *top level* (i.e., outermost) conjunctive subformulas. Consider the set **Chain** of chains of elements of \mathbf{T} .

Lemma 2. *If **Chain** contains an infinite chain \mathcal{C}^∞ but no cycle, then the relation \gg is nonempty and there is $X_0 \in \mathbf{T}$ such that for no $Y \in \mathbf{T}$, $X_0 \gg Y$.*

Proof. Since there is no cycle, for some $\mathcal{C} \subset \mathcal{C}^\infty$ there is a $\mathcal{C}' \in \mathbf{Chain}$ such that

$$\mathcal{C}' = \mathcal{L}' - \dots - \mathcal{C} - \dots - \mathcal{L}''$$

where \mathcal{L}' and \mathcal{L}'' are loops.

Let W be the exit of \mathcal{L}' and X an element of \mathcal{L}' different from W . Suppose \mathcal{L}^* is another loop, $W \in \mathcal{L}^*$ and X is dominated by \mathcal{L}^* , then it is immediate to see that X and W belong to a cycle. Hence $X \gg W$ and \gg is nonempty.

An X_0 minimal with respect to \gg exists by finiteness of \mathbf{V} and Lemma 1. ■

5. Proof of the Main Theorem

This proof details the procedures by which we can compute the proof data structure from the representing chain and vice versa.

5.1. From Proofs to Chains

Consider first an application \mathcal{R}_1 of Weakening in \mathcal{D} , let A be the principal formula, and \mathcal{R}_2 the inference immediately below \mathcal{R}_1 .

If A is passive in \mathcal{R}_2 , then we can certainly permute \mathcal{R}_1 and \mathcal{R}_2 . Thus we may assume that for every Weakening \mathcal{R}_1 , the principal formula A of \mathcal{R}_1 is either

- (i) active in the inference \mathcal{R}_2 immediately below, or
- (ii) that \mathcal{R}_2 is also a Weakening with principal formula B and that the next inference \mathcal{R}_3 is a one-premise rule, say \vee -Right, with B and the descendant of A as active formulas.

If (i) and \mathcal{R}_2 is a two-premises rule, then we can delete the entire branch ending with the other premise of \mathcal{R}_2 and replace \mathcal{R}_1 and \mathcal{R}_2 with a sequence of Weakenings.

In case (ii) we can replace \mathcal{R}_1 , \mathcal{R}_2 and \mathcal{R}_3 by a unique application of Weakening with principal formula $A \vee B$. And so on.

The point of this standard fact is that, by applying Weakening *as low as possible* we obtain a derivation in which every active formula in a two premise rule has some ancestor in an axiom. We assume that \mathcal{D} has this property.

Now we construct the desired path \mathcal{P} by induction on the length of the derivation \mathcal{D} of S . We let \mathcal{P} be the set of (P, P') such that $\vdash P, \neg P'$ (or $\vdash \neg P', P$) is an axiom of \mathcal{D} . If \mathcal{D} consists of an axiom or if the last rule of \mathcal{D} has one premise only, then the proof is trivial.

Suppose it has two premises S_1 and S_2 , with subderivations \mathcal{D}_1 and \mathcal{D}_2 , active formulas A , B and principal formula, say $A \wedge B$. By induction hypothesis there are paths \mathcal{P}_1 and \mathcal{P}_2 for S_1 and S_2 , respectively, satisfying the required conditions. The pairs of formulas in \mathcal{P}_1 and \mathcal{P}_2 correspond to axioms of \mathcal{D}_1 and \mathcal{D}_2 . Construct S^\prec by adding the appropriate conjunctive link below the subformulas-trees of A and B , etc.

Let $\mathcal{P} = \mathcal{P}_1 \cup \mathcal{P}_2$. Since \mathcal{D} is separated, \mathcal{P} satisfies part (a) in the definition of path (4.2). \mathcal{P} obviously satisfies parts (b) and (d). By the argument above $\mathcal{P}_1 \mapsto A$ and $\mathcal{P}_2 \mapsto B$, so \mathcal{P} satisfies also (c), the relevance condition for conjunctive subformulas. The fact that \mathcal{P} is minimal is immediate from the inductive hypothesis. Condition (α) is trivial. The axioms of \mathcal{D}_1 and \mathcal{D}_2 are distinct, so \mathcal{P} satisfies condition (β) .

To check condition (γ) , since a cycle cannot occur only inside S_1^\prec or S_2^\prec , we need to consider only infinite conjunctive chains containing $A \wedge B$, say

$$\mathcal{C} :_F X_C - \dots -_D A \wedge B_E - \dots -_F X.$$

Every chain containing conjunctive subformulas of both $S_1^<$ and $S_2^<$ must contain $A \wedge B$.
If

$$C_1 : X_C - Y_n - \dots -_D A \wedge B$$

and

$$C_2 : A \wedge B_E - \dots - Z_m -_F X$$

are both pure, then all $Y_i \in S_1^<$ and all $Z_j \in S_2^<$. But X cannot belong both to $S_1^<$ and to $S_2^<$.

Thus we may assume that X is $A \wedge B$, $C \prec A$ and $F \prec B$. If, however, there was a conjunctive subformula, say $C \wedge D \prec A$, then we would have a cycle in $S_1^<$, against the hypothesis. Thus $C = D = A$, i.e. C_1 is a loop (and similarly for C_2).

5.2. From Chains to Proofs

We have $S^<$ and \mathcal{P} satisfying the conditions. The proof is by induction on the number of logical symbols in S , plus the number of atoms in S which do not occur in any $(P, P') \in \mathcal{P}$.

The case of all formulas in S that are not satisfied by \mathcal{P} is clearly handled using Weakening.

If S contains a disjunctive formula, say $S = A \supset B, C_1, \dots, C_n$, then the result is immediate from the induction hypothesis applied to $S' = \neg A, B, C_1, \dots, C_n$, using \supset -right.

Now we assume that $S = \Gamma \cup \Pi \cup \Sigma$, where Γ contains only negations of atoms, Π only atoms and Σ only conjunctive formulas.

Case 1. The case of two atoms in S connected by \mathcal{P} corresponds to an Axiom, possibly followed by a sequence of Weakenings.

Now we assume that $\mathcal{P} \mapsto C$, for some conjunctive subformula C .

Case 2. There is a conjunctive formula X , in Σ , say $X = \neg(A_1 \vee A_2)$, such that for every $(P, P') \in \mathcal{P}$, if $P \prec A_i$, then also $P' \prec A_i$, say $i = 2$.

By condition (β) on \mathcal{P} , the inductive hypothesis is satisfied by the proof structures $((\neg A_1, \Phi_1)^<, \mathcal{P}_1)$ and $(\neg A_2^<, \mathcal{P}_2)$. Here Φ_1 is $S \setminus \{X\}$ and where the path \mathcal{P}_1 is \mathcal{P} restricted to $(A_1, \Phi_1)^<$ and \mathcal{P}_2 is \mathcal{P} restricted to $A_2^<$. Therefore $\vdash \neg A_1, \Phi_1$ and $\vdash \neg A_2$ are derivable. The claim follows by \vee -left.

Case 3. There is a conjunctive formula X in Σ , say $X = \neg(A \supset B)$, such that one immediate subformula, say $\neg B$, is connected only to $\Pi \cup \Gamma$.

Let Γ_1, Π_1 be the list of atoms or negations of atoms in S connected to $\neg B$ and let $\Gamma_0 = \Gamma \setminus \Gamma_1, \Pi_0 = \Pi \setminus \Pi_1, \Sigma_0 = \Sigma \setminus \{X\}$. By definition of path (4.2.(a)), these lists are disjoint. The inductive hypothesis holds again for $(\Gamma_1 \cup \Pi_1 \cup \{\neg B\})^<$, on one hand, and $(\Gamma_0 \cup \Pi_0 \cup \Sigma_0)^<$, on the other, with the set \mathcal{P} appropriately restricted. Our claim follows by \supset -left.

Case 4. none of the above.

Now $\mathcal{P} \mapsto C$, for some (conjunctive) subformula $C \in \Sigma$, and since cases 2 and 3 do not apply, for each $X \in \Sigma$ there is a (conjunctive) $Y \in \Sigma$ such that $\dots - X - Y$. Let **Chain** be the set of all conjunctive chains on S . Thus **Chain** is nonempty, and it contains an infinite chain C^∞ . Furthermore, **Chain** contains no terminal chain, since cases 2 and 3 do not apply.

Let $X_0 \in \Sigma$ be minimal with respect to \gg . Such a X_0 exists by Lemma 2, say $X_0 = A_1 \wedge A_2$. For $i = 1, 2$ let

$$\Sigma_i = \{Y \in \Sigma \setminus \{X_0\} : Y \parallel A_i \text{ or } A_i - C - Y, \text{ for some pure chain } C \text{ with } X_0 \notin C\}.$$

Since there is no cycle and X_0 is minimal with respect to \gg ,[†]

$$\Sigma_1 \cap \Sigma_2 = \emptyset$$

Let

$$\Gamma_i = \{P \in \Gamma : P \parallel \Sigma_i \cup \{A_i\}\} \quad \Pi_i = \{P \in \Pi : P \parallel \Sigma_i \cup \{A_i\}\}$$

By the property (a) of paths,

$$\Gamma_1 \cap \Gamma_2 = \emptyset = \Pi_1 \cap \Pi_2$$

Let $\Phi_i = \Gamma_i \cup \Pi_i \cup \Sigma_i \cup \{A_i\}$ and let

$$\mathcal{P}_i = \{(P, Q) \in \mathcal{P} : \text{for some } Z, Z' \in \Phi_i, \text{ we have } P \prec Z \text{ and } Q \prec Z'\}.$$

We would like to apply the induction hypothesis to (Φ_i, \mathcal{P}_i) and then conclude by using \wedge -right. We certainly can do so if we show that

For each i , \mathcal{P}_i is a path for Φ_i .

The fact that the \mathcal{P}_i are minimal and satisfy (α) , (β) and (γ) follows immediately from the same fact for \mathcal{P} .

Lemma 3. *If **Chain** contains an infinite chain but no cycle, and $X \in S$, $X \neq X_0$, then*

$$X \parallel Y, \quad Y \in \Sigma_i \cup \{A_i\} \Rightarrow X \in \Phi_i.$$

Proof. If X is atomic, then by definition $X \in \Gamma \cup \Pi$. Suppose X is conjunctive and not $X = A_i$.

Let Y be A_i . Since in S , **Chain** has no terminal chain, certainly $\dots - X' - X - A_i - X_0$, i.e., $X \in \Sigma$, and thus $X \in \Sigma_i$.

[†] This is crucial: for a partition of $\Sigma \setminus A \wedge B$ to exist, the conjunctive chains reaching out from A and B must not join, not only in a cycle but also in the exit of a loop: here we need the minimality condition.

Assume now that $Y \in \Sigma_i$, say $Y \in \Sigma_1$ and, moreover, that $X \notin \Sigma_1$. We show that this contradicts the assumption that \mathcal{P} is a minimal path for S ; we conclude that $X \in \Sigma_1$.

Fact. *Let X, Y, Y' be conjunctive, with $Y' \parallel Y \parallel X$; suppose $Y, Y' \in \Sigma_1$ but $X \notin \Sigma_1$. If $C \circ D \prec Y$ is such that $Y' \parallel C$ and $X \parallel D$, then $C \circ D$ is disjunctive.*

Proof. Let $A_1 - C - Y' - Y$ be a pure chain, where we may suppose that $Y \not\prec Z$ for all $Z \in \mathcal{C}$. If $C \circ D$ is conjunctive, then $A_1 - C - Y' - C \circ D - X$ is a pure chain, i.e., $C \in \Sigma_1$. ■

Returning to the refutation of the assumption $X \notin \Sigma_1$, first notice that this implies $X \notin \Sigma_2$ too. Otherwise, given pure chains $Y - C_1 - A_1$ and $A_2 - C_2 - X$ we conclude that $Y - C_1 - X_0 - C_2 - X - Y$ is a loop, thus X_0 is not minimal with respect to \gg .

Now let Φ_0 be $S \setminus (\Phi_1 \cup \Phi_2)$. If $U \in \Phi_0$ and $U \parallel V$ with $V \in \Phi_j$, for $j = 1, 2$, then U and V are conjunctive. Let $(Q, Q') \in \mathcal{P}$ be such that $Q' \prec U$ and $Q \prec V$. If $A_i - \dots - V' - V$ is any pure chain and (P, P') is such that $P' \prec V'$ and $P \prec V$, then there must be a *disjunctive* subformula $C \circ D$ of V such that, say, $P \prec C$ and $Q \prec D$, by the above Fact. Therefore, if we drop (Q, Q') from \mathcal{P} , then the resulting path still satisfies the relevance condition on conjunctive subformulas, *relatively to* Σ_j . In conclusion, let

$$\mathcal{P}_0 = \{(P, Q) \in \mathcal{P} : \text{for some } Z \in \Phi_0, P \prec Z \text{ or } Q \prec Z\}$$

and let

$$\mathcal{P}' = \mathcal{P} \setminus \mathcal{P}_0.$$

It is easy to check that \mathcal{P}' is a path not only for $\Phi_1 \cup \Phi_2$, but also for S . But \mathcal{P}' is a proper subset of \mathcal{P} , and this contradicts the minimality of the path \mathcal{P} . ■

Now we can conclude the proof of the Theorem and check the conditions of Section 3.1. for \mathcal{P}_i . Of these, (a) and (d) are immediate and (b) follows from relevance for A_i and Lemma 3.

We need to check the relevance condition (c). Assume $\mathcal{P}_i \mapsto C \wedge D \prec \Phi_i$ via (P, P') , say $P \prec C$ and $P, P' \prec \Phi_i$. Then there is a $(Q, Q') \in \mathcal{P}$ such that $\mathcal{P} \mapsto D$ via (Q, Q') , say $Q \prec D$. Let $X \in S$ such that $Q' \prec X$. Since $C \wedge D \prec \Phi_i$, Lemma 3 implies that $X \in \Phi_i$. This means that $(Q, Q') \in \mathcal{P}_i$, namely $\mathcal{P}_i \mapsto D$. ■

6. Equivalence with Girard's Proof Nets

Of course, the equivalence of the decision procedure with Girard's Proof Nets for the Multiplicative Fragment with box for Weakening follows from their equivalence with Sequent Calculus for Direct Logic. But there may be some interest in seeing a direct, graph-theoretic proof.

Lemma. *If β is a Proof-Net with conclusions S and without Cut-links in propositional Multiplicative Linear Logic without \perp , 1 , then there is a minimal path \mathcal{P} for S satisfying conditions (α) , (β) and (γ) .*

Proof. Let \mathcal{P} be the set of axiom links of the Proof-Net β . Suppose \mathcal{P} is not minimal, and let \mathcal{P}' be a proper subset of \mathcal{P} and also a path for S . Then either (1) there is a proper subset S' of S such that $\mathcal{P}' \not\vdash S \setminus S'$, or (2) there is a nonempty set of subformulas $C \circ D$ in β such that, say, $\mathcal{P}' \vdash C$ and $\mathcal{P}' \not\vdash D$. In case (1) no trip starting in S' will ever reach $S \setminus S'$. In case (2) “ \circ ” in $C \circ D$ must be a “*par*”, since \mathcal{P}' satisfies the relevance condition for conjunctive subformulas. We obtain a short trip as follows: start with an X such that $\mathcal{P}' \vdash X$; if and when a trip reaches C downwards, continue the trip upwards on C , etc. Thus in both cases, we contradict the fact that β is a Proof-Net.

Suppose \mathcal{P} does not satisfy condition (β) . It is easy to see that to (S^\prec, \mathcal{P}) there corresponds a Girard proof structure of the form

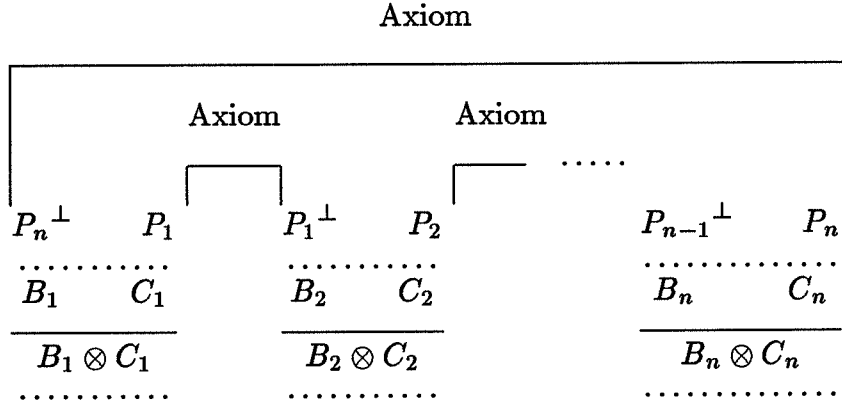
$$\begin{array}{c} \text{Axiom} \\ \boxed{} \\ P_1 \quad P_2 \\ \dots \quad \dots \\ \overline{A \quad B} \\ A \otimes B \\ \dots \end{array}$$

Notice that the switches can always be arranged in such a way that from A^\wedge the trip reaches directly P_1^\wedge and from $P_{2\vee}$ reaches directly B_\vee . Then set the switch on L at the indicated *times* link: the trip

$$A^\wedge, \dots, P_1^\wedge, P_{2\vee}, \dots, B_\vee, A^\wedge$$

never reaches $A \otimes B$, and so is short.

Suppose \mathcal{P} does not satisfy condition (γ) . Then there is a Girard Proof Structure of the form



Again we can set the switch to obtain a short trip:

$$B_1^\wedge, \dots, P_n^{\perp\wedge}, P_{n\vee}, \dots, C_{n\vee}, B_n^\wedge, \dots, C_{2\vee}, B_2^\wedge, \dots, P_1^{\perp\wedge}, P_{1\vee}, \dots, C_{1\vee}, B_1^\wedge. \blacksquare$$

Lemma. *If (S^\prec, \mathcal{P}) satisfies (α) , (β) and (γ) , then there is a Proof Net β (with boxes)*

The proof follows that of Main Theorem (\Leftarrow): in *Case 1* we construct a Proof Net consisting of just an axiom, possibly inside several boxes. In *Cases 2, 3* the result is given by the induction hypothesis and in *Case 4* Lemma 3 of 5.1 guarantees that we can split the structure and apply the induction hypothesis. ■

7. Implementation of the Decision Procedure for DPC

We shall sketch the definition of a decision procedure for Direct Predicate Calculus. No extensions – natural or not – will be discussed.

We may assume, for the sake of exposition, that we are considering a formula F involving only \wedge, \vee, \neg with all negations pushed in, Skolemized and all unifiable (existential) variables noted. As shown in [KETONEN and WEYHRAUCH 1984] an arbitrary formula can be gotten into this form by linear time transformations. Of course, the procedures defined below can be applied with minimal change to the more general case.

We will use symbols p, q, r, \dots to denote atoms or negated atoms, U, V, \dots for *unifiers*: unifiers are viewed as finite sequences of pairs (x, t) , where x is a variable and t is term.

Let us define a *match* as a triple (p, q, U) , where we have a positively occurring term p together with a negatively occurring formula q with a unifier U such that

$$(M.a) : \quad p[U] = q[U].$$

and

$$(M.b) : \quad \text{there is no conjunction } A \wedge B \text{ such that } p \prec A \text{ and } q \prec B \text{ or vice versa.}$$

We will re-define the notion of a path slightly: A path is a finite sequence of matches paired with a unifier containing all the unifiers in the matches such that

$$(P.a) : \quad \text{no literal occurs more than once.}$$

We shall use symbols $\mathcal{P}, \mathcal{Q}, \mathcal{R}, \dots$ to denote paths, and symbols $\mathcal{X}, \mathcal{Y}, \dots$ to denote sets of paths.

We also need to define functions that combine paths:

COMBINE-PATHS(\mathcal{P}, \mathcal{Q}) returns the least path \mathcal{R} containing all the matches in \mathcal{P} and \mathcal{Q} . If no such path exists, it returns ERROR. We will use the notation $\mathcal{P} * \mathcal{Q}$ for the resulting path.

For any formula A and set of paths \mathcal{X} , let $\mathcal{X} \setminus A$ be the set of paths in \mathcal{X} containing no literals $\prec A$.

7.1. The Goal

Our objective is to decide whether there exists a path \mathcal{P} of matches such that

- (a) \mathcal{P} satisfies F ($\mathcal{P} \mapsto F$): there is a triple (p, q, U) in \mathcal{P} such that $p \prec F$ or $q \prec F$.
- (b) if $\mathcal{P} \mapsto A \wedge B$, then $\mathcal{P} \mapsto A$ and $\mathcal{P} \mapsto B$.
- (c) \mathcal{P} contains no cycles; there is no sequence $A_1 - A_2 - \dots - A_n = A_1$ of distinct conjunctions appropriately connected to each other via conjunctive components.

As we have shown, such paths correspond to proofs; our problem is equivalent to finding a proof of F . Each element of such a path corresponds to an axiom in the sense of Gentzen calculi; a sequent of the form $p \rightarrow q$.

Thus the critical issue consists of defining an appropriate search strategy for paths with all of the above properties. Note that the search space size is roughly exponential in size with respect to the size of F . Thus we need limit the search in smart ways.

7.2. Finding All Matches

The first phase of the process consist of finding the set \mathcal{M} of all matches (p, q, U) that can occur in a path described above.

Let \mathcal{L} be the set of all literals in F . Then $\mathcal{L} = \text{POS} \cup \text{NEG}$, the union of all positive and negative literals in F .

Construct the set of all such matches, \mathcal{M} , by unifying each p in POS against each q in NEG s.t. condition (b) for matches is satisfied.

This process is roughly quadratic in the size of F . In practice, the use of suitable indexing methods will make this pass very fast.

Define a function MATCHES on \mathcal{L} as follows: For any p in \mathcal{L} , $\text{MATCHES}(p)$ is the set of all matches associated to p in \mathcal{M} :

$$\text{MATCHES}(p) = \{(p, q, U) \mid (p, q, U) \in \mathcal{M}\} \quad (p \in \text{POS}),$$

or

$$\text{MATCHES}(p) = \{(q, p, U) \mid (q, p, U) \in \mathcal{M}\} \quad (p \in \text{NEG}).$$

7.3. Simple Depth First Search

Let's look at the problem of finding paths with (a) and (b): Any non-empty path \mathcal{P} automatically satisfies (a). If (b) is not true, then the set

$$\text{UNCOVERED-CONJUNCTS}(\mathcal{P}) = \{A \mid A \wedge B \prec F \wedge \mathcal{P} \mapsto B \wedge \neg \mathcal{P} \mapsto A\}$$

is non-empty. For any A in this set, we can look at the set of all possible extensions to \mathcal{P} :

$$\text{EXTENSIONS}(\mathcal{P}, A) = \{Q * \mathcal{P} \mid Q \in \text{PATHS}(p), p \prec A\},$$

and then repeating this process for the members of this set until we have constructed a path with no uncovered conjuncts.

This allows us to find a path satisfying (a) and (b) without having to construct all alternatives at the same time. This can be time consuming, since the suggested algorithm will do an enormous amount of re-computation in situations where an entire branch terminates in failure or no proof exists. In fact, in a typical "first time around" situation the "fact" to be proved is often invalid — the user in question forgot to include all the relevant assumptions.

7.4. Breadth First Search: Pre-Computing Valid Path Combinations

The simplest algorithm using breadth first search can be expressed in terms of computing the function PATHS on all subformulas of F :

$$\begin{aligned} \text{PATHS}(p) &= \{ \{((q, r, U), U)\} \mid (q, r, U) \in \text{MATCHES}(p) \} & (p \in \mathcal{L}) \\ \text{PATHS}(A \wedge B) &= \{ \mathcal{P} * \mathcal{Q} \mid \mathcal{P} \in \text{PATHS}(A), \mathcal{Q} \in \text{PATHS}(B) \} \\ \text{PATHS}(A \vee B) &= (\text{PATHS}(A) \setminus B) \cup (\text{PATHS}(B) \setminus A) \cup \\ &\quad \{ \mathcal{P} * \mathcal{Q} \mid \mathcal{P} \in \text{PATHS}(A) - \text{PATHS}(A) \setminus B, \mathcal{Q} \in \text{PATHS}(B) - \text{PATHS}(B) \setminus A \}. \end{aligned}$$

This, of course, has the effect of computing all possible proofs for F . $\text{PATHS}(F)$ then results in all paths with properties (a) – (b). What remains is to find a non-cyclic path. More precisely, every valid path contains a path from $\text{PATHS}(F)$.

Of course, the algorithm suggested by the above equations is impractical. We propose an alternative; an algorithm that combines aspects of depth and breadth first search in order to prune the search space down as much as possible.

7.5. Static Irrelevance Elimination: the First Refinement

One of the primary causes for failure is irrelevance; a subformula may have no connections with any other fact simply because some critical assumptions were omitted. Let's call a subformula *weakly irrelevant* if none of its subformulas can occur as a subformula of an axiom in any Direct Predicate Calculus proof of F . In the language of Gentzen calculi, this means that the only way it can be introduced into a proof is through the rule of weakening. In practice, weak irrelevancy is hard to compute. Instead, it is better to return to our stated goals; namely reduction in the size of the search space for valid paths or proofs. One step towards this goal is to eliminate in advance those literals that provably cannot occur as a part of any valid path for the entire formula. We can make a few observations:

Lemma 1: *If $\text{MATCHES}(p)$ is empty, then p cannot occur as a part of any valid path.*

Proof: Obvious.

Lemma 2: *If no literal of a formula A occurs as a part of any valid path, then the same holds for any formula of the form $A \wedge B$.*

Proof: If not, take a valid path $\mathcal{P} \mapsto A \wedge B$. But then $\mathcal{P} \mapsto A$ by property (b) of paths.

Lemma 3: *If all literals of $\text{MATCHES}(p)$ other than p cannot occur as a part of any valid path, then p cannot occur as a part of any valid path.*

Proof: Obvious.

Definition The set of *irrelevant* formulas is the smallest set of formulas closed under the following rules:

- If $\text{MATCHES}(p)$ is empty or contains matches of p to only irrelevant literals, p is irrelevant.
- If A and B are irrelevant, then $A \vee B$ is irrelevant.
- If A or B is irrelevant, then $A \wedge B$ is irrelevant.
- If A is irrelevant and $B \prec A$, then B is irrelevant.

Theorem 4. *No subliteral of an irrelevant formula can occur as a part of a valid path.*

Proof: Obvious, from above.

It is useful to observe the interaction of the last two rules; if any part of a conjunction is irrelevant, then all other parts may be declared irrelevant. Irrelevancy elimination can be simply implemented as a relaxation algorithm; mark formulas irrelevant until no more irrelevance can be found.

In practice, irrelevance elimination significantly reduces the search space for valid paths: We may from now on assume that all literals in \mathcal{L} are not irrelevant.

7.6. Modified Breadth First Search: Partial Pre-Computation

As pointed out above, a full breadth first algorithm is impractical; too much information is kept and computed. Our second improvement involves only partially computing the PATHS function on the set of all subformulas of F . First of all, we wish to separate all potential paths into strongly connected components; if F is conjunctive, they are the conjunctive components of F . Thus, in this case we apply our procedure separately to each component. We may assume that F is disjunctive, of the form $C_1 \vee C_2 \vee \dots \vee C_n$. In fact, we expect that in the most typical instance F represents a query of the form

$$A_1 \wedge A_2 \wedge \dots \supset B;$$

i.e., to conclude fact B from assumptions A_1, A_2, \dots

We compute the function PATHS for all elements of Dom , the set of all disjunctive sub-components of F .

Given the function PATHS on Dom , we can extend our irrelevance algorithm: Remove from the set \mathcal{M} all matches not occurring in any $\text{PATHS}(A)$ for $A \in Dom$. Declare any A irrelevant for which $\text{PATHS}(A) = 0$. We can then apply the methods defined in the previous section iteratively in order to further reduce the search space.

7.7. Searching for Paths

We can now describe our algorithm in full: Consider the function PATHS on Dom , where

$$Dom = \{C_1, C_2, \dots, C_n\},$$

such that

$$0 < \text{card}(\text{PATHS}(C_1)) \leq \text{card}(\text{PATHS}(C_2)) \leq \dots$$

We search in order through Dom , trying to find an element of $\mathcal{P} \in \text{PATHS}(C_i)$ that can be extended to a complete path. If no such path can be found in C_i , we declare the subformula C_i irrelevant, compute the propagated irrelevancies, and move onto the next element in Dom . If no such element in Dom can be found, we return failure.

The general step in completing a path is as follows: Consider a path \mathcal{P} , and the set

$$\text{UNCOVERED}(\mathcal{P}) = \{C \in Dom \mid \mathcal{P} \mapsto C, \mathcal{P} \text{ contains no path from } \text{PATHS}(C)\}.$$

Pick a C in this set with the lowest index, and consider all extensions of \mathcal{P} that are complete with respect to C :

$$\text{EXTENSIONS}(\mathcal{P}, C) = \{Q * \mathcal{P} \mid Q \in \text{PATHS}(C)\};$$

repeat this process for the members of this set until we have constructed a path with no uncovered elements of Dom . This method will search through all paths satisfying (a) and (b).

7.8. Dealing with Cycles

It remains to consider the issue of detecting cycles in paths; at first blush, this seems time-consuming and complicated. But again there are many situations where it is not necessary to perform such a check and by a relaxation algorithm we can extend these situations iteratively much further.

Let us call a subformula non-cyclic if it cannot possibly contribute to a cyclic path. We can make the following observations:

A atom is non-cyclic if it is not contained in any conjunction.

If a formula is non-cyclic, so are all of its subformulas.

A conjunctive formula C is non-cyclic if all but one of the conjuncts connected to a conjunction distinct from C is non-cyclic.

The iterative application of these rules drastically reduces our need for cycle checking.

For any match (p, q, U) we construct the set of all possible triples (A, B, C) such that A, B, C are non-cyclic conjuncts occurring as a part of a path because of this match; i.e. $A -_D B_E - C$ with $p \prec D$ and $q \prec E$ or vice versa. These triples will be stored in a table.

Then, for any path \mathcal{P} it remains to check all of the triples in this table that are active for this path and verify that there is no cycle; i.e. a sequence of active triples

$$(A_1, A_2, A_3), (A_2, A_3, A_4), \dots, (A_n, A_{n+1}, A_1), (A_{n+1}, A_1, A_2),$$

where no A_i is a subexpression of another A_j for $j \neq i$. Note that n has to be ≥ 2 for such a cycle. This is easily accomplished by a depth first search through the table.

8. Bibliography

[AVRON 1987] :

A. Avron, *The Semantics and the Proof Theory of Linear Logic*, Laboratory for Foundations of Computer Science, University of Edinburgh, ECS-LFCS-87-27.

[DREBEN and GOLDFARB 1979] :

B. Dreben and W.D. Goldfarb, *Solvable Classes of Quantificational Formulas*, Reading Mass. Addison-Wesley, 1979.

[GIRARD 1987] :

J.Y. Girard, *Linear Logic*, Theoretical Computer Science 50 (1987).

[GIRARD 1988 A] :

J.Y. Girard, *Geometry of Interaction*, preprint, including:

Towards a Geometry of Interaction, forthcoming in *AMS Proceedings of the Conference "Category Theory and Computer Science"*, Boulder, June 1987;

Quantifiers in Linear Logic, forthcoming in *Proceedings of the Conference SILFS*, Cesena, January 1987;

Multiplicatives, forthcoming in *Rendiconti del seminario matematico dell' Università e Politecnico di Torino*, 1988;

Linear Logic and Parallelism, forthcoming in *Proceedings of the School on "Semantics of Parallelism"* held in IAC, Roma, September 1986.

[GIRARD 1988 B] :

J.Y. Girard, *Geometry of Interaction I, Interpretation of the System F*, preprint, Équipe de Logique, UA 753 du CNRS mathématiques, Université Paris VII, October 1988.

[KETONEN and WEYHRAUCH 1984] :

Jussi Ketonen and Richard Weyhrauch, *A Decidable Fragment of Predicate Calculus* Theoretical Computer Science 32 (1984) pp. 297-307

[O. KETONEN 1944] :

O. Ketonen, *Untersuchungen zum Prädikatenkalkül*, Ann. Acad. Sci. Fennicae Series A I, Mathematica-Physica 23, (Springer Berlin, 1944).

[LAFONT 1988] :

Y. Lafont: *From Linear Algebra to Linear Logic*, preprint, Laboratoire d'Informatique, École Normale Supérieure, Paris, October 1988.

[LEWIS 1978] :

H. Lewis: *Complexity of Solvable Cases of the Decision Problem for the Predicate Calculus*, in: *Proceedings of the IEEE Symposium on "Foundations of Computer Science"* (1978), 35-47.

[ONO 1988] :

Hiroakira Ono, *Structural Rules and a Logical Hierarchy*, forthcoming in *Proceedings of Heyting 1988 Conference*, Plenum Press.

[PRAWITZ 1965]:

D.Prawitz, *Natural Deduction, A Proof Theoretical Study*, Almqvist and Wicksell, Stockholm, 1965.

[SAMBIN 1988]:

G.Sambin, *Intuitionistic Formal Spaces and their Neighborhood*, forthcoming in *Proceeding of the Logic Colloquium 1988, Padua*.