

ON THE COMPLETENESS OF LINEAR STRATEGIES
IN AUTOMATIC CONSEQUENCE FINDING

by

ELIANA MINICOZZI

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

in the Department
of
Computer Science

We accept this thesis as conforming
to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA

April, 1972

In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the Head of my Department or by his representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of Computer Science

The University of British Columbia
Vancouver 8, Canada

Date April 14, 1972

ABSTRACT

The problem of the automatic generation of logical consequences of a set of axioms is examined. The merging subsumption linear strategy has been shown complete with respect to that problem. A generalization of a set of support strategy is given, and the completeness of merging subsumption linear strategy with set of support is proved. The merging-linear-A-ordered strategy and the merging linear-C-ordered strategy have been shown to be incomplete. Relations between unit strategy and input strategy have been examined. A little review of the 'interesting theorem' is given.

TABLE OF CONTENTSPage No.

INTRODUCTION.....	1
CHAPTER 1: FORMAL AXIOMATIC THEOREIS AND THE RESOLUTION PRINCIPLE.....	4
Section 1: Formal axiomatic theory.....	4
Section 2: Set of well formed formulas as set of clauses.....	9
Section 3: Herbrand's theorems.....	13
Section 4: Completeness definition.....	14
Section 5: The Resolution Principle.....	15
CHAPTER 2: MERGING-SUBSUMPTION-LINEAR RESOLUTION AND SET OF SUPPORT STRATEGY.....	19
Section 1: Preliminary definitions.....	19
Section 2: The completeness theorems for consequence finding of merging-linear-subsumption strategy.....	20
Section 3: The extension of the set of support strategy to the consequence finding case.....	24
CHAPTER 3: SOME DIFFERENCES BETWEEN THEOREM PROVING AND CONSEQUENCE FINDING.....	27
Section 1: Unit and input strategies....	27
Section 2: Merge-linear-A-ordered Resolution.....	29
Section 3: Merge-linear-C-ordered Resolution.....	35
CHAPTER 4: INTERESTING THEOREMS.....	40
Section 1: Short review on the interest- ing theorem problem.....	40
Section 2: Observations and suggestions..	42
CONCLUSIONS.....	45
BIBLIOGRAPHY	48

ACKNOWLEDGEMENTS

I wish to warmly thank all the members of the Computer Science Department at the University of British Columbia for their friendly and helpful attitude towards me. In particular, I have enjoyed discussions on all possible subjects with Professors Abbe Mowshowitz, Frieder Nake, Raymond Reiter, Richard Rosenberg, Mrs. Vanni Criscuolo and Mrs. Peter Rowat.

I have been lead to work in the area of this thesis by an inspiring course given by Professor Raymond Reiter and this thesis derives from a close collaboration with him. To him my hearty acknowledgement for continuous guidance and help, and for his patience during endless hours of discussion.

My thanks to Miss Cathy Smith, not only for the typing of this thesis, but also for helping me to survive and go through the language barrier during my first period in Vancouver.

INTRODUCTION

The problem studied in this thesis is the automatic (i.e. by computer) generation of theorems from a given set of axioms S . This problem, called 'consequence finding', has been introduced by R.C.T. Lee in [5].

Loosely speaking, we can say that Lee shows that using a particular rule of inference, the Resolution Principle [9], it is possible to deduce all the theorems of S . Later on Slagle, Chang and Lee [12] have shown that with respect to the consequence finding problem, the Semantic Resolution, another rule of inference, has the same property of the Resolution Principle.

In both of the works that we have mentioned (particularly in the first) the authors also try to formalize what is meant by an 'interesting theorem'. Moreover, they try, less successfully, to find mechanical procedures for selecting the 'interesting' theorems from the 'uninteresting' ones.

Our interest in this problem is based mainly on two reasons. One is that it can lead, as was the case for theorem proving, to a deeper understanding of the laws of the deduction in mathematical logic; for example one can find rules of inference simpler but equivalent to the ones traditionally used.

The second reason is that to study automatic consequences finding can help the development of Artificial Intelligence topics, not necessarily related to mathematics. In any case, we believe that this problem must be seen as a topic belonging to the field of Artificial Intelligence.

Of course, given the tight relationship between our topic and theorem proving, a much more developed subject, the question arises as to how worthwhile it is to develop it as an independent subject of research. In order to answer this question we wish here to stress the following two points:

- 1) Theorem proving is actually developing as a study of a particular type of proof, the proof of a contradiction. Of course, one can prove that a proposition T is a theorem of a given set of axioms not only by contradiction, (i.e. showing that from S and the negation of T one can deduce a contradiction), but also by the deduction of T itself from S .

Then the research on consequences finding can be seen as the study on a kind of proof different from the one by contradiction. In this regard we can think the relationship between theorem proving and consequence finding as the one between problem of generation and problem of recognition:

- 2) Consequence finding differs from theorem proving in the fact that it does not need previous knowledge of the theorems. So a consequence finding program can lead the computer to find new theorems, an activity that should be considered as intelligent as the one of proving theorems.

The approach we follow is essentially the same as the one of Lee, Chang and Slagle who have studied the applicability to consequence finding of some strategies first introduced in theorem proving.

We have taken into consideration strategies different from those and our technique for proving theorems is also different

from the one used by them.

The main results we have found in this thesis are the following:

The merging linear subsumption strategy is complete for consequence finding; it is possible to generalize the set of support strategy to consequence finding and the merging linear subsumption strategy with set of support is complete for it; the merging A-ordered linear strategy and the merging C-ordered linear strategy are not complete for consequence finding.

We have also found under which hypothesis on the set of axioms S it is possible to use the merging A-ordered linear and the merging C-ordered linear strategies for deducing theorems from S , and we have shown that for consequence finding the unit strategy is not equivalent to the input strategy.

We do not think that the approach we have chosen is the only possible one, but we believe that it can help in better understanding both the peculiarity of theorem proving itself and the difference between theorem proving and consequence finding, as they have been historically developing.

CHAPTER 1: FORMAL AXIOMATIC THEORIES AND THE RESOLUTION PRINCIPLE.

In this chapter a rather rigorous description of what a formal axiomatic theory is will be given; it will be, also, clarified under which limits we will speak about axioms and theorems; a particular rule of inference, that is the resolution principle, will be introduced.

Section 1: Formal Axiomatic Theories.

Definition [7]: A formal axiomatic theory K is defined when the following conditions are satisfied:

- 1) A countable set of symbols is given as symbols of K . A finite sequence of symbols of K is called an expression of K .
- 2) There is a subset of the expressions of K called the set of well formed formulas (wffs) of K (we will assume that there exists an effective procedure to determine whether a given expression is a wff).
- 3) A set of wffs is set aside and called the set of axioms of K . There exists an effective procedure to determine if a wff is an axiom or not.
- 4) There is a finite set $I_1 \dots I_n$ of relations among wffs, called rules of inference. For each I_i there is a unique positive integer J such that, for every set of J wffs and each wff A , one can effectively decide whether the given J wffs are in the relation I_i to A , and if so, A is called a direct consequence of the given wffs by virtue of I_i .

(L1) Definition [7]: A wff A is said to be a consequence in K of a set S of wffs. if and only if there is a sequence $A_1 A_2 \dots A_n$ of wffs such that $A_n = A$ and for each i , either A_i is an axiom or A_i is in S , or A_i is a direct consequence by some rule of inference of some of the preceding wffs in the sequence. Such a sequence is called a proof (or deduction) of A from S . The members of S are called the hypothesis or premises of the proof.

If S is just the set of the axioms of K , then the consequences in K of S are called theorems.

Before giving an equivalent definition of 'wff is a consequence in K of a set S of wffs' let us give some other specification of our theory K :

1) The set of symbols of K will be a subset of the following set of symbols:

a) Punctuation symbols:

.,; (;);

b) Logical symbols:

\forall ; $\&$; $-$; \vee ; \exists ;

c) Individual variables:

\bar{x} ; y ; z ; x_1 ; y_1 ; z_1 ; etc.

d) Individual constants:

a ; b ; c ; a_1 ; b_1 ; c_1 ; etc.

e) Predicate letters:

P ; Q ; R ; S ; T ; P_1 ; Q_1 ; R_1 ; S_1 ; T_1 ; etc.

Where any symbol represents an n -ary predicate letter with $n \geq 0$.

f) Function letters:

$f; g; h; f_1; g_1; h_1; \text{etc.}$

Where any symbol represents an n -ary function letter with $n \geq 0$.

2) Set of terms:

- a) Variables and individual constants are terms.
- b) If \mathcal{L} is an n -ary function letter and $n_1 \dots n_n$ are terms, then $\mathcal{L}(n_1, \dots, n_n)$ is a term.
- c) An expression is a term only if it can be shown to be a term on the basis of a) and b).

3) Set of atomic formulas:

If n_1, \dots, n_n , are terms and L is an n -ary predicate then $L(n_1, \dots, n_n)$ is an atomic formula.

4) Set of wffs:

- a) Every atomic formula is a wff.
- b) If A and B are wffs, and w is a variable then:
 $A; A \& B; A \vee B; \forall w(A); \exists w(A)$ are wffs.
- c) An expression is a wff only if it can be shown to be a wff on the basis of a) and b).

- Definitions [7]:
- 1) An expression is said to be quantifier free if the symbols \forall, \exists do not appear in it.
 - 2) In $\forall w(\text{or } \exists w)(A)$ A is called the scope of the universal (or existential) quantifier.
 - 3) An occurrence of a variable w is bound in a wff, if either it is the variable of a universal quantifier or it is

the variable of an existential quantifier in the wff, or it is within the scope of either ' $\forall w$ ' or ' $\exists w$ ' in the formula. Otherwise the occurrence is said to be free in the wff.

- 4) A variable is said to be free (bound) in a wff if it has a free (bound) occurrence in the wff.

Definition [7]: An interpretation consists of a not-empty set D , called the domain of the interpretation, and an assignment to each n -ary function letter \mathcal{L} of an n -place operation in D (i.e. a function from D^n into D), to each n -ary predicate letter L of an n -place relation in D , whose values are 'True' or 'False' (that is a function $D^n \rightarrow (\text{True}, \text{False})$)).

Given such an interpretation variables are thought of as ranging over the set D , each constant is thought of as a single point in D , ' \forall ', ' $\&$ ' ' \forall ' ' \exists ' are given their usual meaning.

For a given interpretation, a wff without free variables (closed wff) represents a proposition which is true or false, whereas a wff with free variables stands for a relation on the domain of the interpretation which may be satisfied (true) for some values in the domain of the free variable and for the values of the constants, and not satisfied (false) for other values of the free variables and for the values of the constants⁽¹⁾.

⁽¹⁾We will not go into any definition of truth and satisfiability more formal than the one we have given before.

- Definitions:
- 1) A wff is called 'satisfiable' if there is some interpretation in which it is satisfied.
 - 2) A set of wffs is satisfiable if there is an interpretation in which all the wffs in the set are satisfied.
 - 3) A wff is called 'unsatisfiable' if it is not satisfiable.
 - 4) A set of wffs is 'unsatisfiable' if every wff in the set is not satisfiable.

(L2) Definition 7 : A wff A is said to be logical consequence in K of a set S of wffs if the set $S' = S \cup \bar{A}$ is unsatisfiable. We will say S implies A for 'A is a logical consequence of S'.

Definition 7 : A wff A is said to be logically valid if and only if A is true for every interpretation.

Definition 7 : A theory K is said to be a first order logic theory if:

- 1) The axioms of K are divided into two classes: the logical axioms and the proper axioms.
- 2) The rules of inference are:
 Modus ponens and generalization⁽²⁾.

⁽²⁾For the definition of these inference rules, see the reference given in Chapter 2.

The logical axioms are the same for every first order logic theory and they are logically valid wffs. The proper axioms vary from theory to theory.

A first order logic theory in which there are not proper axioms is called a first order predicate calculus.

The logical axioms are so designed that the logical consequences of the closure of the axioms of K are precisely the theorems of K . The theories we will consider will be first order logic theories.

Section 2: Set of wffs as set of clauses.

Definition: A literal is an atomic formula or its negation.

Definition: A clause is a set of literals. We will indicate them with the following symbols:

$C; T; C'; T'; C''; T''; \text{etc.}$

$C_0; T_0; C_1; T_1; \text{etc.}$ $R_0; R_1; \text{etc.}$

$R_0'; R_1'; \text{etc.}$

Definition [7]: A wff B such that $B \equiv Q_1 w_1 \dots Q_K w_K (A)$ ⁽³⁾

Where: Q_i is either the universal quantifier

or the existential quantifier, w_i is

a variable and $w_i \neq w_j$ for $i \neq j$,

A is a quantifier free wff

is said to be in prenex normal form; A is

called the matrix and the sequence $Q_1 w_1 \dots Q_K w_K$

is called the prefix.

⁽³⁾ B can be also a quantifier free wff.

Definition: A wff is said to be in prenex conjunctive form if it is in prenex normal form and its matrix is a conjunction of wffs.

Definition: A wff A is said to be equivalent to a wff B if B is a consequence of A and A is a consequence of B.

Th. [7] : There is an effective procedure for transforming any wff A into a wff B in prenex normal form such that $\vdash A \equiv B$ ⁽⁴⁾.

Given a closed wff A in prenex normal form: $A = Q_1 W_1 \dots Q_K W_K (C)$ we will say that:

Definition: B is the functional normal form of A if B is obtained from A using the following procedure:
 0) If A is a quantifier free wff, B is just A.
 1) Starting from the left of A let Q_r be the first existential quantifier we meet, and let W_r be the variable which is quantified by Q_r , construct

$$B = Q_1 W_1 \dots Q_{r-1} W_{r-1} Q_{r+1} W_{r+1} \dots Q_K W_K (C')$$

where C' is C except that any occurrence of W_r in C is substituted either with $\mathcal{L}(W_1 \dots W_{r-1})$, where \mathcal{L} is a functional symbol which does not occur in our theory K, if $r > 1$, or with a constant symbol, which does not occur in our theory K, if $r=1$;

⁽⁴⁾ $\vdash A \equiv B$ means that 'A equivalent to B' is a theorem of the first order predicate calculus.

- 2) If either there is not such Q_r , or $r=K$, then stop: B is a functional normal form of the starting wff A, otherwise call B, A and C', C and go to step 1.

Because this procedure substitutes for a variable bound by an existential quantifier a particular term, intuitively it is clear that the ~~prenex~~ normal form of a wff is a consequence of its functional normal form, but the opposite is not true; let us give an example:

$$A = \forall x \forall y \exists z C(x, y, z)$$

$$B = \forall x \forall y C(x, y, f(x, y)),$$

Clearly, whenever B is satisfied also A is satisfied.

Now we are ready to perform the following transformation of our theory K:

- 1) We consider the closure of the wffs of our theory K,
- 2) Every wff so obtained is transformed into its prenex conjunctive normal form.
- 3) Every wff so obtained is transformed in its functional normal form.
- 4) Every wff so obtained is transformed into a corresponding set of clauses.

Let us give an example of the algorithm just given:

Example: Consider a wff

$$A = \forall x \exists y ((p(x, y) \vee q(x, c)) \& (p(x, b) \vee q(x, y)))$$

By steps 1 and 2 A is transformed into A, by step 3 A is transformed into B such that

$$B = (p(x, f(x)) \vee q(x, c)) \& (p(x, b) \vee q(x, f(x)))$$

where f is a new functional symbol.

By step 4 B is transformed into a set of clauses

$$S = (p(x, f(x))q(x, c), p(x, b)q(x, f(x)))$$

(Note that from the fact that our wffs are closed formulas and from the logical soundness⁽⁵⁾ of the rules of inference we will use, it follows easily that the definitions (L1) and (L2) are equivalent, that is a wff A is a logical consequence of a set of wffs if and only if A is a consequence in K of S. It seems natural to us to look at definitions (L1) and (L2) as characteristic respectively of the consequence finding approach and the theorem proving approach).

From now on we will regard the axioms of our theory as a set of clauses⁽⁶⁾ and our theorem as a clause⁽⁷⁾.

-
- (5) Logical soundness is an elementary property that every set of rules of inference must satisfy. A set of rules of inference is logically sound if, when it is applied to some satisfiable set of wffs, the resulting set of wffs is still satisfiable.
- (6) For an intuitive understanding of some notion we will introduce later, it is enough to remember that each clause represents the disjunction of its literals, and a set of clauses represents the conjunction of its clause.
- (7) Actually, since the rules of inference we are going to use are such that they transform a couple of clauses into a single clause, a clause will be treated as an axiom.

Section 3: Herbrand's theorems.

- Definitions [9]: 1) A literal which does not contain variables is called a ground literal.
- 2) A clause each member of which is ground is called a ground clause.

A particular ground clause is the empty clause. By general clause or simply clause we will mean any clause (not necessarily ground).

Definition: (8): With any set S of clauses can be associated a set of ground terms called the Herbrand Universe of S:

$$H_S = \bigcup_{n=0}^{\infty} H_{n,s}$$

Where: $H_{0,s}$ = the set of all constant symbols belonging to S if there are any, otherwise a generic constant symbol,
 $H_{n,s} = H_{n-1,s} \cup$ set of all terms obtained from all the functional symbols of S in which the variables are substituted by terms of H_{n-1} .

- Definitions [9]: 1) If S is any set of clauses and P is any set of terms then by $P(S)$ we denote the saturation of S over P, that is the set of all ground clauses obtainable from members of S by replacing variable with member of P, occurrences of the same variable in any one of clause being

(8) Definition of the Herbrand Universe equivalent to this one can be found, for example, in [9].

replaced by occurrences of the same term.

- 2) A set of ground literals which does not contain a literal and its negation is called a model.
- 3) A clause C belonging to a set S of clauses is satisfied by a model M if for every member C' of $H_S(C)$, $C' \wedge M$ is not empty.

A set S of clauses is satisfied by a model M if every clause in S is satisfied by M . Such M is said to be a model for S .

First Herbrand's theorem⁽⁹⁾: If S is any finite set of clauses, then S is unsatisfiable if and only if there is a finite subset of $H_S(S)$ which is unsatisfiable.

Generalized Herbrand's theorem [12]: If S is any finite set of clauses and C is a ground clause then S implies⁽¹⁰⁾ C if and only if there is a finite subset S' of $H_{SUC}(S)$ such that S' implies C .

Section 4: Completeness.

Let S be any finite set of clauses, let I be a set of rules of inference, let C be a clause.

Definitions: 1) I is refutation complete if:

S is unsatisfiable⁽¹¹⁾ if and only if there is a deduction of the empty clause from S by I .

(9) Formulation of the Herbrand theorem equivalent to this one can be found in [4], [9], [10].

(10) S implies C is equivalent to ' C is a logical sequence of S '.

(11) Obviously S is unsatisfiable if there is no model for it.

2) \mathcal{I} is complete for consequence finding if:

S implies C if and only if there is a deduction of a clause C' from S by \mathcal{I} and C' implies C .

Section 5: Resolution principle.

Definitions [9] : 1) Any expression of the form n/w where w is any variable and n is any term different from w is called substitution component.

2) Any finite set of substitution components (possibly empty), none of the variables of which are the same, is called a substitution. We write the substitution whose components are $n_1/w_1, \dots, n_k/w_k$ as $(n_1/w_1, \dots, n_k/w_k)$ with the understanding that the order of components is immaterial.

We use lower case Greek letters to denote substitution. In particular ϵ is the empty substitution.

3) If $\theta = (n_1/w_1, \dots, n_k/w_k)$ and λ are two substitutions, then $\theta' \vee \lambda'$ where λ' is the set of all components of λ whose variables are among $w_1 \dots w_k$ and θ' is the set of all components $n_1 \lambda/w_1$ such that $n_1 \lambda$ is different from w_1 is called the composition of θ by λ and is denoted by $\theta \lambda$.

- 4) If E is any string of symbols and $\theta = (n_1/w_1, \dots, n_k/w_k)$ is any substitution, then the instantiation of E by θ is the operation of replacing each occurrence of w_i in E by an occurrence of the term n_i , $E\theta$ is called the instance of E by θ . (As a result in [7] we want to remember that $E(\sigma\lambda) = (E\sigma)\lambda$.)
- 5) If C is any set of strings and θ is a substitution then the instance of C by θ is the set of all strings $E\theta$ where E is in C . We denote this set by $C\theta$ and we say that it is an instance of C .

Definition: Given a string of symbols E (or a set of strings C), the substitution σ is a most general substitution for E (or for C) if for any substitution θ there is a substitution λ such that $E\theta = E(\sigma\lambda)$ (or $C\theta = C(\sigma\lambda)$), that is for any θ , $E\theta$ (or $C\theta$) is an instance of $E\sigma$ (or $C\sigma$).

Definitions 9 : 1) Given two clauses C and C' such that there are two literals m and m' , m belonging to C , m' belonging to C' and $m\theta = \overline{m'\sigma}$ ⁽¹²⁾ where θ and σ are the most general substitutions with

(12)

For avoiding confusion, given a set S of clauses, we rename variables so that all variables in one clause are distinct from all the variables in the other clause. (Standardization process [9]).

respect to making m and m' the complement of each other, the clause $R_i = (C-m) \cup (C'-m')$ is called a resolvent of C and C' . Clearly two clauses can have only a finite number of resolvents.

- 2) If S is any set of clauses the set of all clauses which are members of S or resolvents of members of S is denoted by $R(S)$;
 $R(R^{n-1}(S))$ with $R^0(S) = S$ and $n \geq 1$ is called the n -th resolution of S .

Let us give an example of the last definitions:

Ex:

Let S be $(prq, \bar{p}q, \bar{q}\bar{r})$

$R^1(S) = R(S)$ will be $rq, pr\bar{r}, pq\bar{q}, \bar{p}\bar{r}$

Definition [9] : The Resolution Principle is the following rule of inference:

From any pair of clauses C and C' , one may infer a resolvent of C and C' . C and C' are called parent clauses of their resolvent.

It is clear from the definition that any model for the parent clauses of a resolution satisfies the resolvent, that is the resolvent is a logical consequence of its parents.

Intuitively we can say that the resolution principle involves, as ideas, the syllogism principle of the propositional calculus, that is from $p \vee q$ and $\bar{p} \vee r$ we obtain $q \vee r$, and the

instantiation of predicate calculus, that is from $\forall w_1 \dots \forall w_n$
 $A(w_1, \dots, w_n)$ we obtain $A(n_1, \dots, n_n)$ where w_1, \dots, w_n are variables,
 A is a wff, $n_1 \dots n_n$ are terms.

Results: For the resolution principle both refutation
completeness and completeness for consequence
finding have been proved [9,5].

CHAPTER 2: MERGING SUBSUMPTION LINEAR RESOLUTION AND SET OF SUPPORT STRATEGY.

In this chapter we will prove the completeness for consequences finding of the merging linear subsumption strategy. A generalization to consequence finding of the set of support strategy will be given and its completeness will be proved.

Section 1: Preliminary definitions.

Deduction as a tree: We will represent a deduction of a clause from a set of clauses as a tree performing the following elementary correspondence: if a clause R_i is inferred from C_j and C_k by I_m then the tree shown in Figure 1 will represent this deduction. When there is no possibility of confusion about the rule of inference used the tree of Figure 1 will be substituted by the tree Figure 1a.

Let us give an example of a not elementary deduction:

Let S be $(prq, p\bar{q}l, \bar{l})$. The clause pr can be deduced from S by resolution, one of these deductions can be:

From prq and $p\bar{q}l$ by resolution prl ,

From prl and \bar{l} by resolution pr , the deduction tree corresponding to this deduction of pr will be the tree of Figure 1b.

Definition: If C and C' are two clauses we say that

C subsumes C' if there is a substitution σ such that $C\sigma$ is a subset of C' (15).

(15)

Note that if C subsumes C' then C implies C' , the converse is not true.

Definition 2 : A ground clause C'' is called a merge resolvent of two ground clauses C and C' if C'' is a resolvent of C and C' and there is some literal K such that K is an element of C , C' and C'' . K is called a merge literal.

Definition: A clause T is a prime implicate of a set of clauses S , if S implies T and every T' such that $T \not\subseteq T'$ and S implies T' , does not subsume T .

Definition: A set of clauses S is a minimum deduction set for a clause T if S implies T and $S-C$, where C is any element of S , does not imply T . If S is a minimum deduction set for T we will say also that S minimally implies T .

Section 2: Completeness theorems for consequence finding for the merging-linear-subsumption strategy.

Definition: A merging-subsumption-linear deduction (m-s-l) $[2,6,1]$ of a clause is a deduction represented by the tree in Figure 2 where:

- 1) C_1 is in S .
- 2) R_i for $1 \leq i \leq n$ is a resolvent of the two clauses immediately above it.
- 3) C_i for $0 \leq i \leq n-1$ is either in S or is one R_j for some $j < i$.
- 4) If C_i is not in S then:
 - a) R_{i+1} subsumes some instance of R_i
 - b) C_i is a merge resolvent
 - c) The literal resolved upon in C_i is a merge literal of C_i

5) No clause in the deduction is a tautology.

Result: The refutation completeness of the m.s.l. strategy with top clause ⁽¹⁶⁾ any clause in a minimally unsatisfiable set of clauses has been proved [1].

Lemma 1: Let S be a finite set of ground clauses, let K be a set of literals, let S' be the set $(C'/C \in S \ \& \ C' = C - K)$ ⁽¹⁷⁾

let T be a clause then:

if S implies T , S' implies T .

Proof: Every model for S' is a model for S . Then because S implies T , S' implies T .

Corollary 1: Let S be a finite set of ground clauses let K be a set of literals, let S' be the set $(C'/C \in S \ \& \ C' = C - K)$ then if S is unsatisfiable, S' is unsatisfiable.

Proof: Directly from Lemma 1 when T is the empty clause.

Th. 1: Let S be a finite set of ground clauses, let T be a clause, let C be a clause in S then:

if S minimally implies T ⁽¹⁸⁾ then there is a m.s.l. deduction with top clause C of a clause T' such that T' subsumes T .

Proof: Since S implies T , the set $S_1 = S \cup \bar{T}$ is unsatisfiable.

Consider $S'_1 = (C'/C \in S_1 \ \& \ C' = C - T)$

from corollary 1 S'_1 is unsatisfiable.

(16)

An m.s.l. deduction with top clause C is a m.s.l. deduction in which the left top clause is C .

(17)

Note that if some C in S is just K the empty clause will be in S' .

(18)

The hypothesis that S minimally implies T can be substituted by the hypothesis S implies T if the deduction of T in which we are interested is just a m.s.l. deduction of T from S .

Consider S'_2 a subset of S'_1 such that S'_2 is minimally unsatisfiable. From the refutation completeness of the m.s.l. strategy with top clause any clause in a minimally unsatisfiable set of clauses, for any clause C' in S'_2 there is a m.s.l. deduction of the empty clause from S'_2 with top clause C' . In particular, because S minimally implies T , there will be a m.s.l. deduction of the empty clause from S'_2 with top clause $C' \equiv C-T$ where C is the given clause of S . Let D (Fig.3) be such a deduction. Starting from the top of D , apply the following algorithm:

Algorithm 'Reconstruction'.

- 1) Substitute for C' the corresponding clause in S_1 (19)
- 2) For $0 \leq i \leq n-1$ if C'_i is in S'_2 substitute for it the corresponding clause in S_1 .
- 3) Substitute for each resolvent appearing in D as R'_i or C'_h , the resolvent obtained under the previous substitution.

Let the deduction D' in Fig.4 be such deduction.

Observe now that no t such that t is in T , appears in D' because otherwise t would be in some clauses of D and then in the empty clause; this means that: the no-tautologies condition holds for D' because it holds for D and D' is a deduction from S more than from S_1 .

Trivially the last resolvent of $D'T'$ subsumes T (20)

Trivially the merging and subsumption conditions hold for D' .

Th.2: Let S be a set of general clauses, let T be a general clause then: if S minimally implies $T(x_1, \dots, x_n)$ then for every

(19) as in

(20) This means to add T'' to $C' \equiv C-T$ if T'' is a subset of T and of C .

Remember that the empty clause subsumes every clause.

clause in S , there is a m.s.l. deduction from S with top clause C of a clause T' such that T' subsumes T .

Proof: Let us write T as $T(x_1, \dots, x_n)$ to emphasise the variables of T .

- 1) Let us introduce the individual symbols b_1, \dots, b_n such that for $1 \leq i \leq n$ b_i does not appear in S or in T .
- 2) Consider $T(b_1, \dots, b_n)$, since S implies $T(x_1, \dots, x_n)$ represents a closed wff, $T(x_1, \dots, x_n)$ implies $T(b_1, \dots, b_n)$ and S implies $T(b_1, \dots, b_n)$.
- 3) Let us construct the Herbrand universe of S and $T(b_1, \dots, b_n)$, that is H_{SUT} .
- 4) From the generalised Herbrand theorem there is a finite subset S' of $H_{SUT}(S)$ such that S' implies $T(b_1, \dots, b_n)$.
- 5) Consider the subset S'' of S' such that S'' minimally implies $T(b_1, \dots, b_n)$.
- 6) From Th.1, for every clause C' in S'' there is a m.s.l. deduction with top clause C' from S'' of a clause T'' such that T'' subsumes $T(b_1, b_2, \dots, b_n)$.

Let D , (Fig.5) be the m.s.l. deduction of T'' , with top clause C' , which is an instance of the given clause C of S .

Starting from the top of D do the following operations:

- a) Substitute for C' the clause C of which C' is an instance,
- b) if C'_i belongs to S'' , substitute for it the clause in S of which C'_i is an instance,
- c) substitute for each resolvent appearing in D as R_i' or C'_i the clause obtained under the previous substitution. (21)

(21) In [7] we can find the following result: 'If clauses T and C have as instances respectively T' and C' with resolvent R_i' then there exists a resolvent R_i of T and C with instance R_i' '.

Let D' in Fig.5a be the deduction so obtained and let T' be the result of the deduction D' .

Note that:

- a) T' does not contain b_1, \dots, b_n , because b_1, \dots, b_n are not individual symbols in S ,
- b) because T' subsumes T'' there is a substitution $(b_1/x_1, \dots, b_n/x_n)$ such that T' subsumes T'' .

From the fact that T'' subsumes $T(b_1, \dots, b_n)$ it follows that T' subsumes $T(x_1, \dots, x_n)$.

It is easy to see that the top clause condition holds for D' as well as the merging subsumption and no tautologies conditions.

Corollary 2:: Let S be a set of clauses,

let C be a clause in S ,

let T be a clause then

if S minimally implies T and T is prime implicate of S then there is a m.s.l. deduction with top clause C , from S , of T .

Proof: From Th.2 and from the definition of prime implicate.

Section 3: The extension of set of support strategy to the consequence finding case.

The set of support strategy was introduced by Wos, Robinson, Carson [13] in order to speed up the search for the deduction of the empty clause from an unsatisfiable set of clauses, when the used rules of inference are resolution [13] and the paramodulation [14]. The intuitive idea on which the set of support strategy is based is essentially this one: Given an unsatisfiable set of clauses S , if our goal is the deduction of

the empty clause from S , it is not necessary to resolve (or to paramodulate) two clauses in a satisfiable subset of S . However, since the definitions are more general than this intuitive idea we will give them⁽²²⁾.

Definition [13]: Given a set S of clauses and a subset S' of S , a clause C has S' -support (with respect to S) if either C is in S_1 or C is a resolvent and at least one of its parent clauses has S' -support.

Definition [13]: Given a set S of clauses and a subset S' of S , a S' -supported deduction is a sequence of clauses such that every clause in the sequence either is in $S-S'$ or has S' support.

Results: Given an unsatisfiable set of clauses S , for every S' subset of S , such that $S-S'$ is satisfiable the refutation completeness of both resolution with set of support S' [13,14] and m.s.l. resolution with set of support S' [1] has been proved.

The way of generalizing the set of support strategy to consequence finding comes to us naturally from observing that the set S' mentioned in the results just given is such that $S-S'$ does not imply the empty clause.

Th.3: Let S be a set of clauses,

let T be a clause then

if S implies T , then for every S' subset of S such that $S-S'$ does not imply T , there is a m.s.l. S' -supported deduction of a clause T' such that T' implies T .

(22) Because we are interested in strategies based on resolution we will not mention further paramodulation.

Proof: If T is the empty clause, Th.3 is just the refutation completeness of m.s.l. resolution with S' -support.

Otherwise, since $S-S'$ does not imply T , the minimum deduction set for T is such that its intersection S'' with S' is not empty.

Let C be a clause in S'' . From theorem 2 we have that there is a m.s.l. deduction with top clause C of a clause T' such that T' implies T . This one is just a m.s.l. deduction with set support S' .

CHAPTER 3: SOME DIFFERENCES BETWEEN THEOREM PROVING AND
CONSEQUENCE FINDING.

In this chapter we will examine unit resolution, input resolution, A-ordered resolution, C-ordered resolution with respect to the consequence finding problem.

Section 1: Unit and input strategies.

Definitions: Given a set S of clauses we will say that:

- 1) C is an input clause if C is an element of S .
- 2) C is a unit clause if C is a set consisting of a single literal.
- 3) A deduction D is a unit proof of a clause C if D is a deduction of C by resolution, and every resolution in D is such that at least one of the two parent clauses is unit.
- 4) A deduction D is an input proof of a clause C , if D is a deduction of C and every resolution in D is such that at least one of the two parent clauses is an input clause.

Result: Chang [3] : If S is an unsatisfiable set of clauses then there is an input proof of the empty clause from S if and only if there is a unit proof of the empty clause from S .

The above result was important in theorem proving because of the efficiency (time and storage used) of the unit strategy (i.e. the only allowed resolutions are unit resolutions).

Unfortunately, in consequence finding (Th.4) we loose this equivalence, trivially because for having a unit proof we must have at least some unit clause in S. The following example shows this loss of equivalence between unit and input proof:

Ex: Let the following clauses be our axioms:

$$1) \quad p \bar{q}$$

$$2) \quad p \quad q$$

Resolving 1 and 2 we have an input proof of p, and obviously there is not unit proof of p from 1 and 2.

Th.4 : Let S be a set of ground clauses, let T

let T be a clause,

let S imply T then

if there is a unit proof from S of some clause T' such that T' implies T, then there is an input proof from S of some T'' such that T'' implies T.

Proof: Since there is a deduction by resolution from S of T' then S implies T' and $S \vee \bar{T}'$ is unsatisfiable.

Since T' is a set of unit clauses and there is a unit proof of T' from S then there is a unit proof D from S' of the empty clause (Fig.6).

By Chang's theorem there is an input proof from S of the empty clause.

In this proof remove every T'_i element of T' on the right as shown in Fig.7 and in Fig.7a; the deduction so obtained is an input proof from S of T'' such that T'' implies T.

Th.5: Let S be a set of general clauses,
 let T be a clause then
 if there is a unit proof from S of some
 clause T' such that T' implies T , then
 there is an input proof from S of some
 clause T'' such that T'' implies T .

Proof: It follows from Th.4 using the same method has
 been used for proving Th.2.

We must observe that the loss of equivalence between
 input and unit resolution for consequence finding causes some
 loss of the relevance of Th.4.

Section 2: Merge linear A-order resolution:

Definitions: Let S be a set of clauses, we will say that:

- 1) S is an A-ordered set of clauses if a
 sequence $O = A^1A^2\dots A^n$ of all atoms in
 S without repetitions is given,
- 2) A^j is greater than A^i if j is greater than
 i .

Definition: An atom and its negation are equal under
 any A-ordering.

Definition: Given an A-ordered set of atoms R , then
 if R' is any subset of R , the A-ordering
 of R' induced by an A-ordering O of R , is
 the sequence O' obtained from O by deleting
 the atoms of $R-R'$ from O .

Definition: A merge A-ordered linear resolution (m.a.l.)
 with top clause C from an A-ordered set of
 clauses S is a deduction D (Fig.8) where:

- a) C is in S ,
- b) each resolvent is formed by resolving upon the largest literal under the A-ordering of S in the clause on the left in D ,
- c) for $0 \leq i \leq n - 1$ either C_i belongs to S or C_i is some R_j for $j \leq i$, in which case C_i is a merge A-ordered resolvent and the literal resolved upon in C_i is a merge literal (as well as being the largest literal in R_i).

4) No clause in the deduction D is a tautology.

Result: The refutation completeness of m.a.l. resolution has been proved [8].

Unfortunately, as the following example will show, the completeness for consequence finding of the m.a.l. resolution does not hold:

Ex: Consider the set of axioms S ($pq, \bar{q}rp, \bar{p}qs$).

We can easily see that pr and qs are logical consequences of S , and moreover, that they are prime implicates of S . By observing that r and s do not appear negated in S , we see that pr is a prime implicate of pq and $\bar{q}rp$ and qs is prime implicate of pq and $\bar{p}qs$; then the only possible deductions of pr and qs are those ones shown in Figs: 9, 10, 11 and 12. Just looking into deductions we convince ourselves that no A-ordering of S is possible, if we wish to deduce both pr and qs . In fact, for deducing qs we need $p q$, for deducing pr we need $q p$.

Th.6: The m.a.l. strategy is not complete for consequences finding. Moreover there are axiom sets for which there is not A-ordering such that we can find all prime implicates of S using A-ordered resolution.

Proof: The previous example proves this theorem.

Th.7: Let S be a set of ground clauses, let T be a clause, let C be a clause in S , then

if S minimally implies T there is an A-ordering of S , such that there is a m.a.l. deduction with top clause C from S of some clause T' such that T' subsumes T .

Proof: Since S implies T , $S \cup T$ is unsatisfiable. Let S_1 be $S \cup T$.

Consider $S_1' = (C'/C \in S_1 \ \& \ C' = C - T)$, from corollary 1 S_1' is unsatisfiable.

Consider S_2' subset of S_1' such that S_2' is minimally unsatisfiable.

From the refutation completeness of the m.a.l. strategy with top clause any clause in a minimally unsatisfiable set of clauses considered any clause C' in S_2' and for any A-ordering O of S_2' , there is a m.a.l. deduction with top clause C' from S_2' of the empty clause. In particular, because S minimally implies T there will be a m.a.l. deduction of the empty clause from S_2' with top clause $C' = C - T$ where C is the given clause of S . Let D be such deduction (Fig.13).

Clearly O' is an A-ordering of S since no clause in \bar{T} is some clause of S_2' because S_2' minimally

implies the empty clause and all clauses of S are clauses of S'_2 because S minimally implies T .

Apply now to the deduction D the reconstruction algorithm. Let D' be the deduction so obtained (Fig.13a).

Clearly D' is a m.a.l. ordered deduction with top clause C from S of T' , such that T' subsumes T , with O' the A-ordering of S .

Note that the A-ordering O' for which Theorem 6 holds is arbitrary except for the fact that the literals belonging to T must be less than every literal in S under O' . The extension of Theorem 6 to a case in which S is a set of general clauses follows from Theorem 1.

Th.8: Given a set of ground ⁽²³⁾ clauses S , let T_1, \dots, T_n be prime implicates of S , let S_1, \dots, S_n be the minimum deduction sets respectively for T_1, \dots, T_n such that $S_1 \subseteq S_2 \subseteq \dots \subseteq S_n$ then

there is an A-ordering such that there is a m.a.l. deduction of T_i from S for every $1 \leq i \leq n$.

Proof: We give the algorithm for constructing this A-ordering:

Let O be the ordered sequence of atoms in S already constructed:

Stage 1:

1) Read a new literal $K \in S_1$

(23)

For simplicity we speak about a set of ground clauses, the extension of Th.8 to the general case follows as Th.2 follows from Th.11.

- 2) If K belongs to T_1 and either K or \bar{K} belongs to O as $A^{(j)}$, then drop $A^{(j)}$ from O and change every $A^{(h)}$ in O to $A^{(h+1)}$ for $0 \leq h < j$, go to 6.
- 3) If K belongs to T_1 but K and \bar{K} do not belong to O then change every $A^{(h)}$ in O to $A^{(h+1)}$, go to 6.
- 4) If K does not belong to T_1 and either K or \bar{K} belongs to O then go to 7.
- 5) If K does not belong to the prime implicate T_1 but K and \bar{K} do not belong to O , if $A^{(j)}$ is the greatest atom in O , then put $A^{(j+1)} = K$, go to 7.
- 6) Put $A^{(0)} = K$
- 7) Call O the new sequence of atoms obtained. If there is some literal K in S_1 not already read, get it and go to 1, otherwise $i = i + 1$, if $i < n + 1$ go to 1, otherwise stop.

With theorem 6 in mind, we can understand, from the following observations, that the A-ordering constructed satisfies the condition of Th.8.

Observations:

- 1) Substeps 2, 3 and 6 of the given algorithm are based on the fact that if K belongs to T_1 then, because S_1 minimally implies T_1 and because $S_1 \subseteq S_2$ for $i \leq l$, K does not appear negated in every S_1 with $i \leq l$ (i.e. K is never resolved in any deduction of T_1 from S_1).
- 2) Step 4 and 5 of the given algorithm are based on the fact that the only condition we have on the literals in

S_i , but not in T_i , is that they must be greater than every literal in T_i .

Lemma 2: If a set of clauses S minimally implies a clause T and T is a prime implicate of S , then there is not a clause $Q \neq T$ such that S minimally implies Q and Q is prime implicate of S .

Proof: Suppose the lemma is false then we have:

- a) S minimally implies T and
- b) S minimally implies Q and
- c) both T and Q are prime implicate of S and
- d) $T \neq Q$

From corollary 2 there will be a m.s.l. deduction from S as of T as of Q . Let D and D' be such deductions.

From a) there is no \bar{t} such that t is in T and \bar{t} is in some clauses of S ,

from b) all clauses C in S such that some t is in C and t is in T appear somewhere in D' ,

then T subsumes Q ; but if T subsumes Q because for hypothesis Q is prime implicate of S , T must be equal to Q contradicting the hypothesis that $T \neq Q$, therefore the lemma holds.

From theorem 8, lemma 2, and from the fact that an unsatisfiable set of clauses has as unique prime implicate, the empty clause, it is intuitively enough clear why the m.a.l. strategy is complete for refutation. (We could say just for the same reasons for which it is not complete for consequence finding).

Section 3: Merge linear C-ordered resolution.

Definition: An ordered clause (O-clause) is a sequence of literals with no two literals the same.

If C is a clause we denote by (C) an O-clause obtained from C by some arbitrary but fixed ordering of its literals.

If S is a set of clauses we denote by (S) the set S in which all clauses are ordered.

Definition: Given two C-ordered ground clauses (C) and (C') , the clause R_i will be the C-ordered resolvent between (C) and (C') if

- a) R_i is a resolvent between C and C' .
- b) The literal resolved upon is the rightmost literal in (C) if (C) is the left clause in the deduction tree of R_i from (C) and (C') , otherwise it is the rightmost literal in (C') .
- c) If (C) is the left clause in the deduction tree of R_i from (C) and (C') , and $C = c_1 \dots c_n$ and $C' = c'_1 \dots c'_k$ so that $c_n = c'_s$ for some $s \leq k$, then R_i is the sequence $c_1 \dots c_{n-1}$ followed by the sequence obtained from $c'_1 \dots c'_k$ by deleting c'_s together with any literal c'_i which also occurs in C . If such c'_i exists, it is a merge literal and R_i is an ordered merge resolvent.

We will write R_i as (R_i) . (Fig.14 is an example of the previous definition).

Definition: Let (S) be a set of ground 0-clauses, a merge linear C-ordered (m.c.l) deduction from S is a deduction like the deduction in Fig.15

where:

1) (C) is in S

2) For each i , $0 \leq i \leq n-1$

Either (C_i) is in (S) or (C_i) is some R_j for some $j < i$ in which case

a) (C_i) is an ordered merge resolvent,

b) the literal resolved upon in (C_i) is a merge literal, so that the rightmost literal of (R_j) is its complement.

3) No clause in the deduction is a tautology.

With m.c.l. deduction with top clause C , we mean a m.c.l. deduction with top left clause just C .

Result: The refutation completeness for m.c.l. resolution with top clause any clause in a minimally unsatisfiable set of clauses has been proved [8].

Th.9: The m.c.l. resolution is not complete for consequence finding. Moreover, there are some set of clauses S and some prime implicate T of it such that there is no C-ordering for which it is possible to have a m.c.l. deduction of T from S .

Proof: Consider the set of ground clauses

$$S = (t_1p, t_2\bar{p}, t_3\bar{q}, \bar{p}qr),$$

clearly $t_1t_2t_3$ is a prime implicate of S , but it is easy to see that there does not exist a C-ordering of S , which allows us to have a m.c.l. deduction of $t_1t_2t_3$ from S .

As an example of the possible deductions of $t_1 t_2 t_3$ from S see Fig.16. There the resolvent $t_1 q t_2$ does not allow us to define a C-ordering for that deduction.

Th.10: Let S be a set of ground clauses,

let T be a clause,

let C be a clause in S such that T is a subset of C , then:

if S minimally implies T and T is prime implicate of S , then there is a C-ordering of clauses in S for which exist a m.c.l. deduction of T from S .

Proof: Because S implies T , $S \cup T$ is unsatisfiable.

Consider the set S' such that $S' = \{C' / C \in S \cup T \quad C' = C - T\}$

From corollary 1 S' is unsatisfiable. Let S'' be a subset of S' minimally unsatisfiable.

Since S minimally implies T , there will be in S'' some clause C' , such that $C' = C - T$ and T is a subset of C .

From the refutation completeness of a m.c.l. resolution with top clause any clause in a minimally unsatisfiable set of clauses, for any C-ordering of S'' , there is a m.c.l. deduction of the empty clause from S'' with top clause $(C') = (C - T)$ where C has as its subset T . Let D in Fig.17 be such deduction.

Consider now the C-ordering of S such that under it:

- 1) If C in S corresponds to the top clause (C') in D then the literals in T are the leftmost literals in C and the literals in $C - T$ are ordered, as the literals in the top clause C' of D .
- 2) All the other clauses C in S are ordered as the corresponding (C') in D , and literals contained in both C and T are inserted anywhere.

Starting from the top of D , apply the following algorithm:

- 1) Substitute to (C') the corresponding clause (C) in (S)
- 2) For $0 \leq i \leq n-1$ if (C'_i) is in S' substitute for it the corresponding clause in (S) .
- 3) Substitute for each resolvent appearing in D as either (R_1) or (C_n) the ordered resolvent obtained under the previous substitution.

Let D' be such deduction, then from the following facts:

- a) T is a subset of the top clause in D' ,
- b) We merge left,
- c) S minimally implies T (that is no \bar{t}_i such that t_i is in T , is in some clause of S),

it follows that D' is a m.c.l. ordered deduction of T from S .

Before going to theorem 11, let us make some observations about theorem 10 to indicate the difference between the consequence finding case and the theorem proving case:

- 1) the top clause in the deduction satisfying theorem 10 is not arbitrary,
- 2) the C-ordering satisfying theorem 10 is arbitrary for all clauses in S except for the top clause of D' .

Th.11: Let S be a set of ground clauses,

let $T_1 \dots T_n$ be prime implicates of S ,

let $C_1 \dots C_n$ be clauses in S having the following properties:

a) $T_i \leq C_i$ & $C_i \not\leq C_j$ for $i \neq j$

b) $C_i \in S_i$ where S_i minimally implies T_i , then

there is a C-ordering of the clauses in S , such that there is a m.c.l. deduction from S of every T_i for $1 \leq i \leq n$.

Proof: Choose a C-ordering of S arbitrary except that under it for any i , $1 \leq i \leq n$ T_i is the leftmost subset of the corresponding C_i .

From theorem 10 and the second observation above, it is clear that the theorem holds.

Obviously, theorems 10 and 11 hold also in the case in which S is a set of general clauses, their extension to a general case follows from them, as theorem 2 follows from theorem 1.

CHAPTER 4: INTERESTING THEOREMS

As we are interested in consequence finding as a sub-field of Artificial Intelligence, we have to deal with the problem of how to single interesting theorems out of all the theorems of a given set of axioms.

The aim of this chapter is to give a short review of the result obtained up to now on this problem, and make some suggestions on possible developments.

Section 1: Short review on the interesting theorems problem.

We will report some of the work done by R.C.T. Lee [5] to give an idea on what is known up to now on how to single interesting theorems out of all the theorems following from a given set of axioms.

Indeed, most of the basic ideas developed up to now on this subject are contained in this work. The following definition of non trivial theorem (i.e. interesting theorem) is given:

Definitions: 1) A theorem T is trivial if it is implied by another theorem T' such that T' is not equivalent to T.

Given this definition it is easy to see that:

- 1) The only uniform way to decide that a theorem T is not trivial is to prove that there is not any theorem T' such that T' implies T (this statement is undecidable for meaningful first order theories).
- 2) In order to single out interesting theorems, one needs a theorem proving program or a consequence finding program (in both cases one is faced with new problems, for example in the last case, still with the problem

of the generation of trivial theorems).

Faced with these difficulties, Lee tries to give algorithms able to select prime implicates of a given set of axioms rather than interesting theorems. We notice that, although an interesting theorem is prime implicate, the converse is not necessarily true⁽²⁴⁾.

As an example of the algorithms we can find in his work we will mention the following two:

- 1) Unit resolution principle and Bound of Clause, that is:
 - a) clauses containing more than k literals are not generated for some fixed k .
 - b) the triviality is checked only for unit clauses.
 - c) the unit resolution strategy is used.
- 2) Axiom resolution strategy, that is:
 - a) axioms are resolved with axioms or with theorems,
 - b) resolutions between theorems are forbidden.

Obviously, both of these heuristics, as well as the other ones introduced by Lee, do not make us sure to obtain all the prime implicates as well as only the prime implicates of a given set of axioms.

Since Lee's work is the only one in which this topic is presented extensively, it is worthwhile to mention his opinion as a fair description of the stage of the research on how to single out interesting theorems:

(24)

Remember that: if a clause C subsumes a clause D then C implies D , but the converse is not necessarily true.

The greatest failure of this research is that the author did not succeed in obtaining a general theory on how an important theorem is derived.

Section 2: Observations and suggestions.

To begin this discussion, it seems useful to us to stress the existence of an ambiguity in the way the problem of how to single out interesting theorems is treated. This ambiguity arises from the fact that, although one would like to single out theorems which are interesting with respect to their logical properties, necessarily one has to take into account the computational power of the actual computers. For example, it seems clear to us that the proposal of the strategies reported in the last section is suggested much more by considerations on the computational power of computers rather than by the original purpose. In fact, it is possible to understand from the discussion in Chapter 3 of this thesis, on the unit and input strategies (i.e. the Lee's axiom resolution), that both of them do not give us any assurance even on the problem of finding prime implicates.

Of course this does not mean that these two strategies are completely useless for finding interesting theorems. In particular the unit strategy can be an efficient procedure because of our definition of theorems as disjunction of literals.

With regard to the relation between results found in our thesis and the problem in discussion here, we first remember that an obvious consequence of the completeness of the m.s.l. strategy is that by it we can generate all prime implicates of a given set of axioms (Corollary 2).

However, at the beginning of our work we thought that the m.s.l. strategy would restrict the number of uninteresting theorems more than we have actually found. To give an idea of its lack of efficiency with respect to this problem we give the following example:

Let us consider the set of axioms $S = (q\bar{p}, \bar{p}\bar{q}, p\bar{r}, qr)$.

Clearly the clause \bar{r} is a prime implicate of S and the subset S_1 of S such that $S_1 = (q\bar{p}, \bar{p}\bar{q}, p\bar{r})$, is the minimum deduction set for \bar{r} . Let us consider the following deductions of \bar{r} from S_1 by m.s.l. resolution:

Deduction D1 (Fig.18):

from $q\bar{p}$ and $\bar{p}\bar{q}$, \bar{p} ;

from \bar{p} and $p\bar{r}$, \bar{r} ;

Deduction D2 (Fig.19):

from $p\bar{r}$ and $q\bar{p}$, $q\bar{r}$;

from $q\bar{r}$ and $\bar{p}\bar{q}$, $\bar{p}\bar{r}$;

from $\bar{p}\bar{r}$ and $p\bar{r}$, \bar{r} ;

Deduction D3 (Fig.20):

from $p\bar{r}$ and $q\bar{p}$, $\bar{r}\bar{q}$;

from $\bar{r}\bar{q}$ and $q\bar{p}$, $\bar{r}\bar{p}$;

from $\bar{r}\bar{p}$ and $p\bar{r}$, \bar{r} ;

We see that D2 and D3 generate a relatively large number of trivial theorems, namely: $q\bar{r}$, $\bar{p}\bar{r}$, $\bar{r}\bar{q}$, and D1 generates only prime implicates, namely: \bar{p} , \bar{r} .

The previous example shows that the freedom with respect to the top clause that we have in m.s.l. strategy is paid with a loss of efficiency.

However, there is an obvious difference between D1 and the deductions D2 and D3, namely that in D1 we introduce our theorem literals as late as is possible; it is easy to see from the characteristics of the minimum deduction set that this fact cut-off the generation of clauses subsumed by our theorem.

We have mentioned this difference between D1 and D2 and D3 because we hope that it could suggest some possible implementation of the m.s.l. strategy with respect to finding interesting theorems.

Furthermore, we suggest using the m.s.l. strategy supplemented by the rule of deleting every left theorem that in a m.s.l. deduction we resolve with theorem, instead of the Lee's axiom resolution.

As far as the other strategies discussed in our thesis are concerned, we stress that their applicability to the interesting theorems problem presents different but not minor difficulties to those of the m.s.l.

The m.a.l. and the m.c.l. strategies are almost useless for finding interesting theorems, since they are applicable to consequence finding only in very peculiar cases.

The m.s.l. strategy with set of support presents the obvious difficulty that is is undecidable whether a given set of axioms is a minimum deduction set for a theorem T.

However, in this case, as a possible suggestion for developing efficient strategies for finding prime implicates we give the following procedure: Let S be a set of ground clauses,

- 1) Consider one literal K contained in some clause of S .
- 2) Consider the subset S' of S such that \bar{K} does not appear in any clause of S' and \bar{K} appears in all the clauses of $S-S'$.
- 3) If K is prime implicate of S , we will find it from S' by m.s.l. In this case we choose another literal in S and we repeat the procedure applied to K .

If K is not prime implicate of S we certainly will deduce all prime implicate in K of S . Let T be one of them, we consider the set of clauses $S \cup T$ and for all of the literals contained in it we apply the same procedure we applied to K .

We notice that this procedure as it stands is not applicable to the case of a set of general clauses.

Conclusions

We think that the completeness for consequence finding of the m.s.l. strategy, proved in this thesis, its completeness for refutation already well known, and its simplicity can help in better understanding the laws of the logical deduction.

It is already well known that the m.s.l. resolution has a higher efficiency than simple resolution, because of the restrictions it puts on the deduction. The completeness theorem that we have proved allows us to use it in consequence finding instead of ordinary resolution.

Furthermore, it seems that the m.s.l. strategy, when supplemented with a certain heuristic procedure can usefully supplement Lee's axiom resolution with respect to the interesting theorems problem.

We have also seen in this thesis that the m.a.l. and the m.c.l., although complete for theorem proving, are not complete for consequence finding and that the m.s.l. strategy with set of support although complete is difficult to use. This seems to support our initial feeling on the difference between theorem proving and consequence finding, and, at the same time, provide some insight in both fields.

In particular, it seems to us that our results suggest the following conclusions:

- 1) The completeness for consequence finding and the degree of applicability to consequence finding of a given strategy can be taken as criterion of its efficiency for theorem proving. We guess that the less a strategy is applicable to consequence finding the more efficient it is for theorem proving.
- 2) The generality of a strategy, for example its completeness for both consequence finding and theorem proving, is paid in its efficiency. We believe in fact that the more efficient a strategy is the more it is applicable to the deduction of peculiar classes of theorems.

Support for this idea comes to us, not only from the results mentioned above, but also from the following reflection on the nature of theorem proving. We can in fact regard theorem proving as the problem of finding prime implicates of an unsatisfiable set of clauses; then the compatibility between the completeness of a strategy and its efficiency seems to us related to the lucky situation that for an unsatisfiable set of clauses the set of prime implicates reduces to a single element, the empty clause.

On the basis of these considerations it seems to us that a possible line of development of the field of consequence finding is to search strategies related to the logical properties of theorems and axioms, rather than complete strategies.

An amusing example of theorems the computer can derive, when the strategy used completely disregards the semantics, is the following given by R.C.T. Lee 5 : Consider the case in which we have the following statements as axioms:

- 1) If x has y z 's and z has v w 's, then x has $y \cdot v \cdot w$'s
- 2) Man has 2 hands;
- 3) Hand has 5 fingers.

These three axioms can be put into three clauses as follows:

- 1) $\bar{p}(x,y,z) \bar{p}(z,v,w) p(x,f(y,v)w)$
- 2) $p(\text{man},2,\text{hand})$
- 3) $p(\text{hand},5,\text{finger})$.

By m.s.l. we can deduce from 1 and 2 $\bar{p}(x,y,\text{man})p(x,y \cdot 2,\text{hand})$.

We can easily see that the above clause, although it is a theorem, does not have an acceptable meaning; then its deduction is completely useless.

We think that a point of view very similar to the one we have just discussed for consequence finding should also be taken when dealing with the problem of how to single out interesting theorems. Moreover, in this case we want also to notice that, although important semantic and syntactic properties differentiate the class of prime implicate from the other theorems, further subdivisions of this class could be useful for finding more efficient strategies and getting more insight into what is meant by 'interesting theorems'.

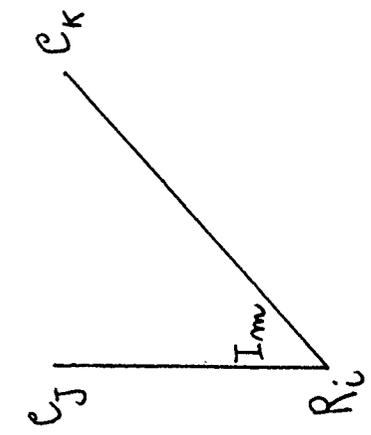


Fig: 1

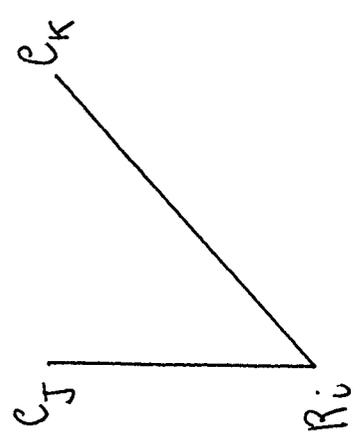


Fig: 1a

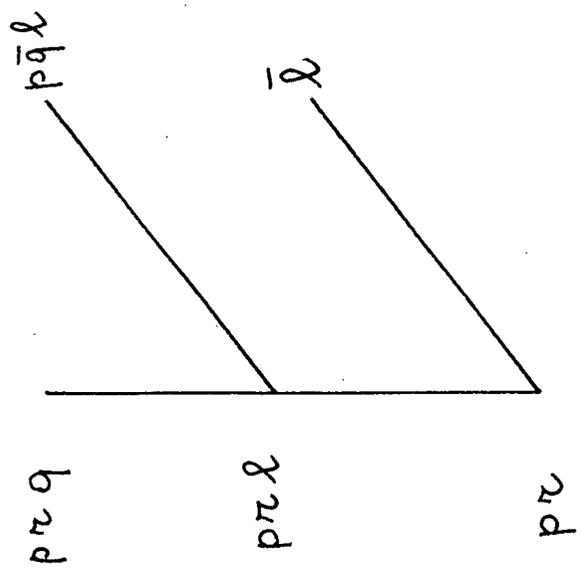


Fig: 1b

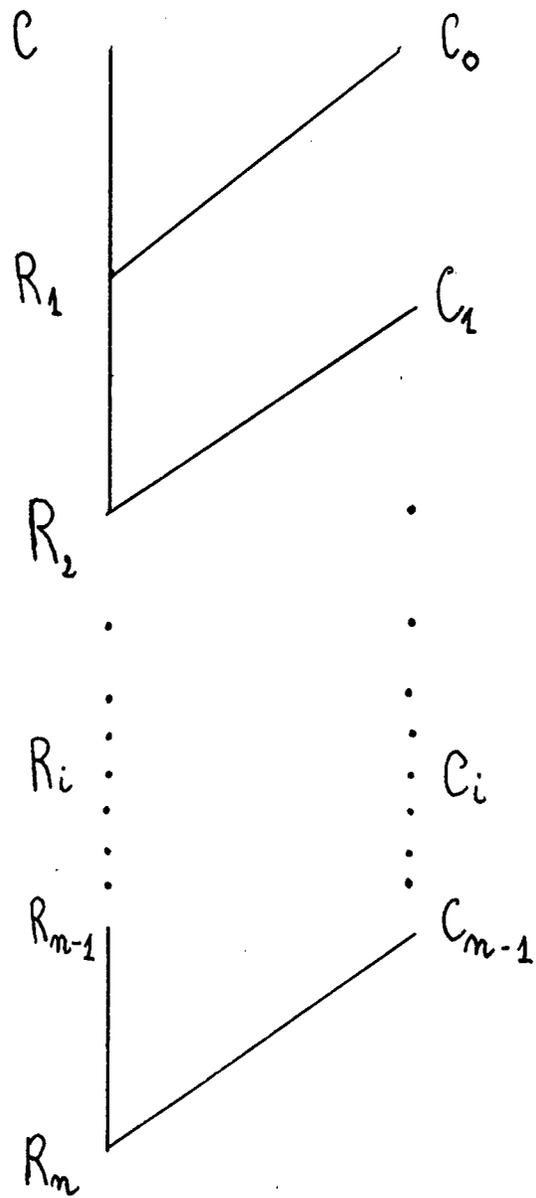


Fig: 2

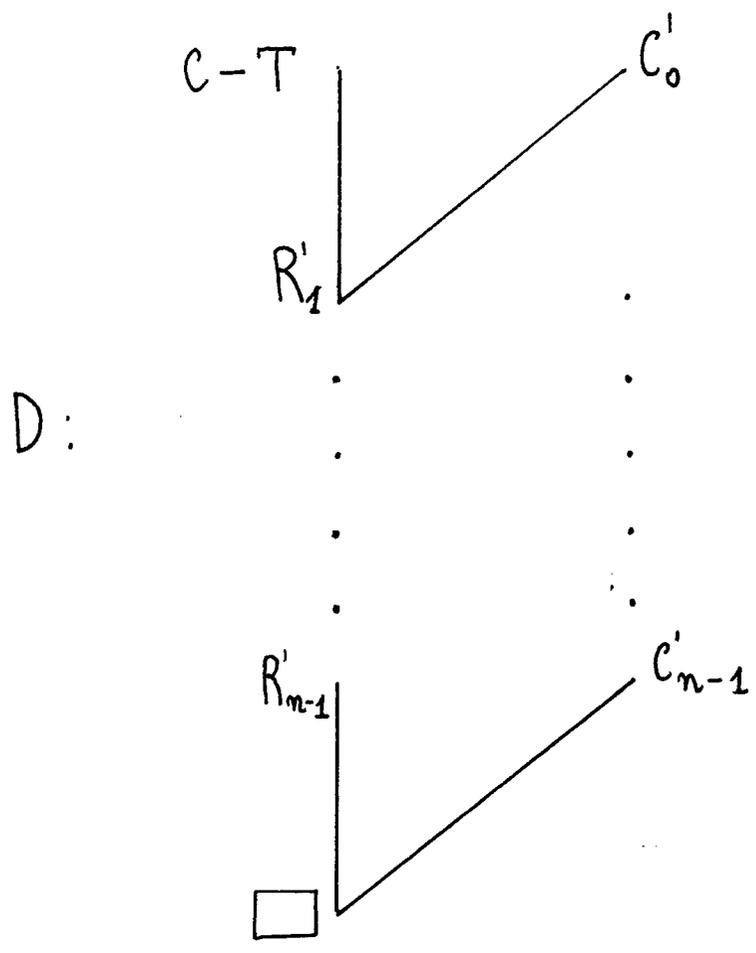
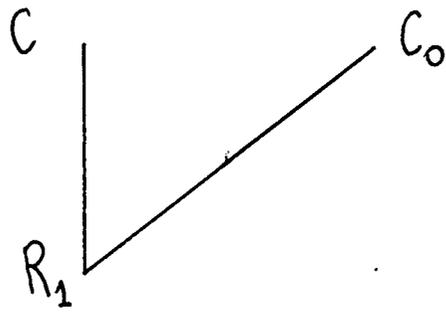


Fig: 3



D' :

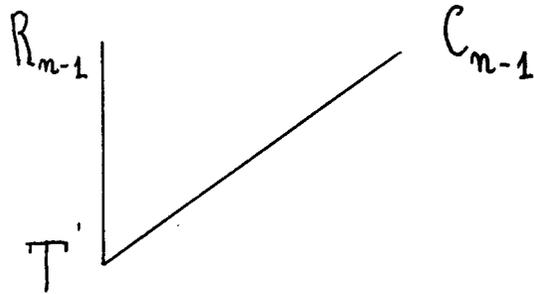


Fig: 4

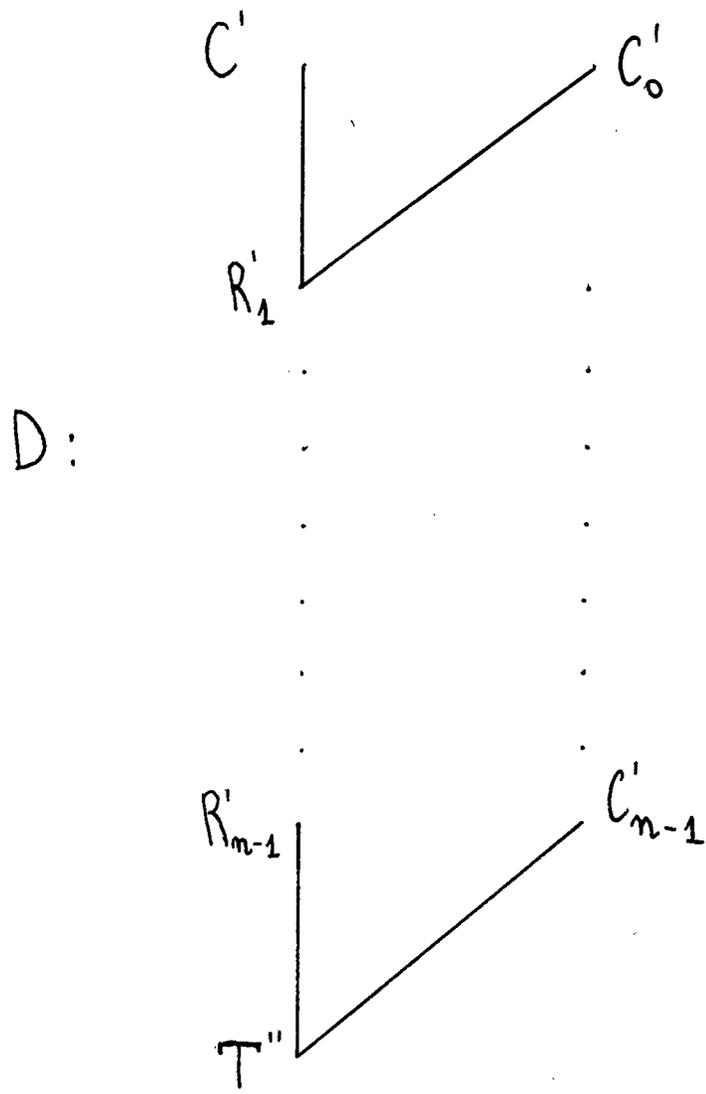


Fig: 5

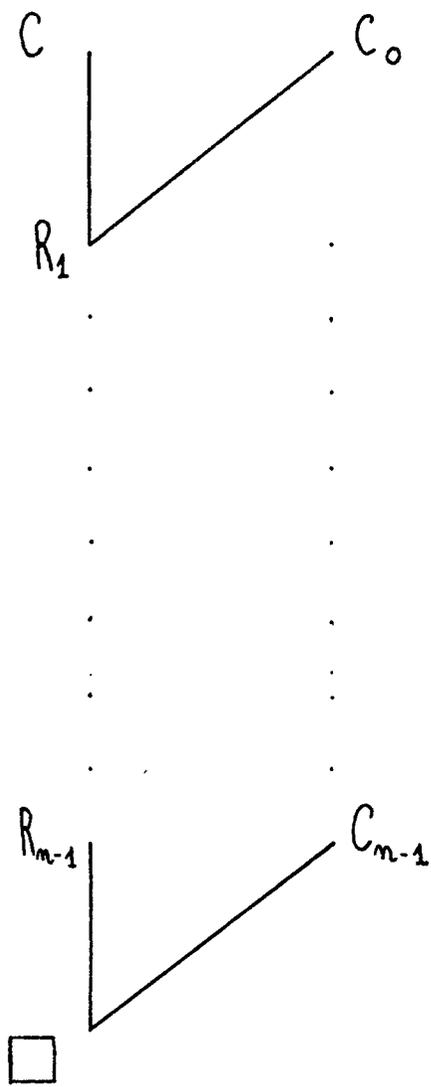


Fig: 6

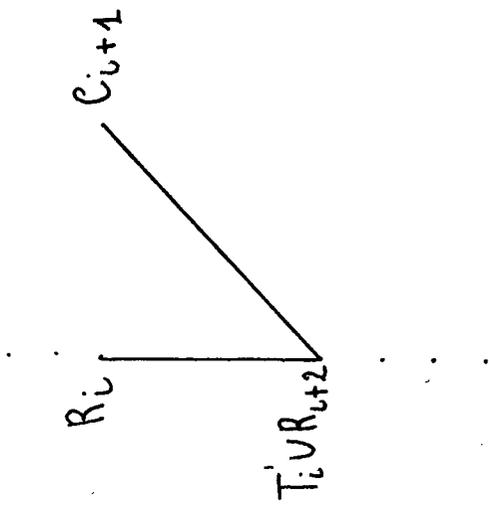


Fig: 7a

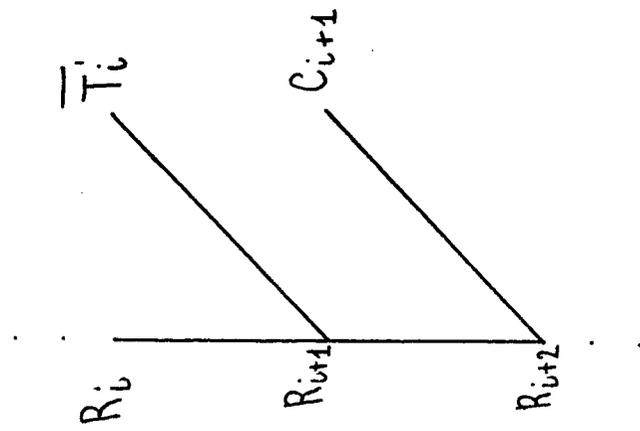


Fig: 7

D:

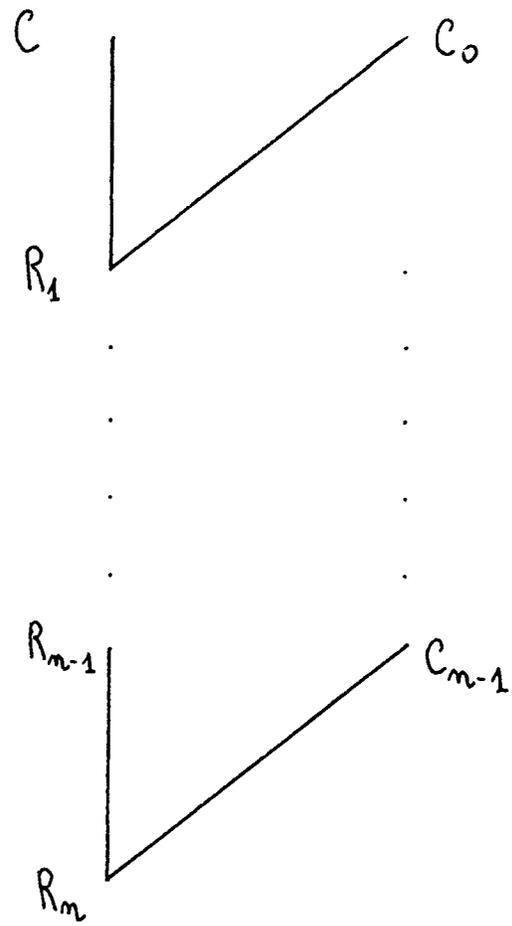


Fig: 8

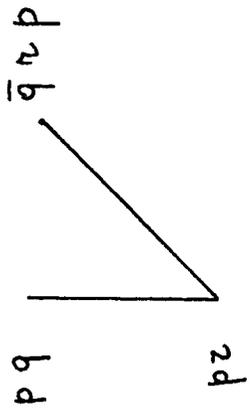


Fig: 9

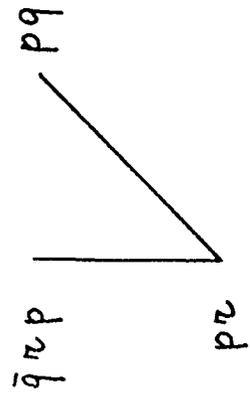


Fig: 10

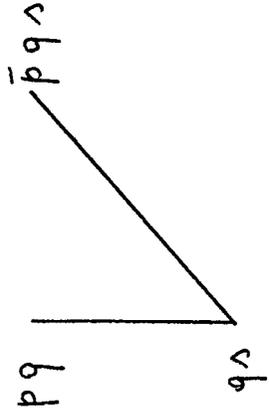


Fig: 12

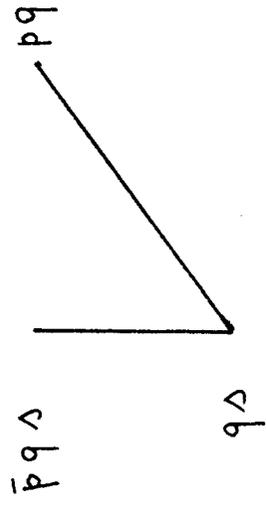


Fig: 11

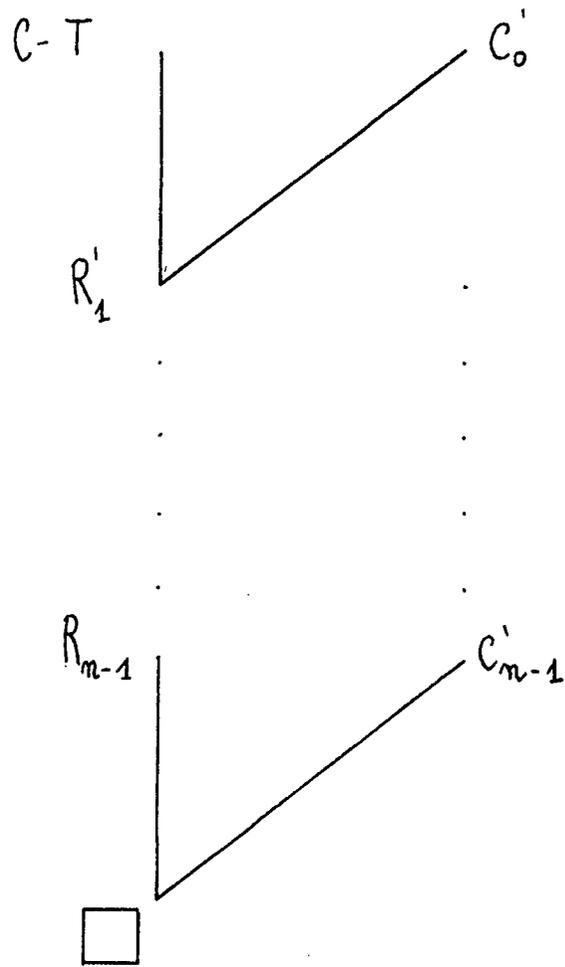


Fig: 13

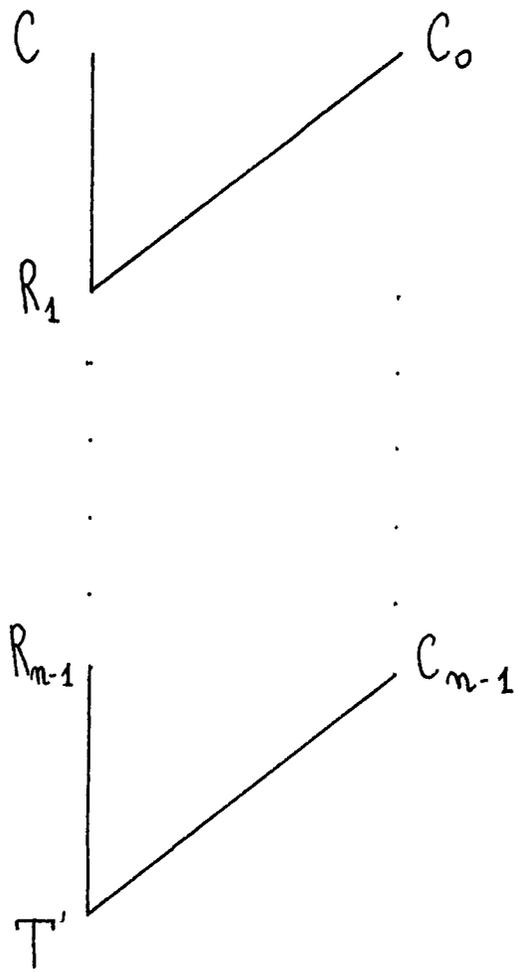


Fig: 13 a

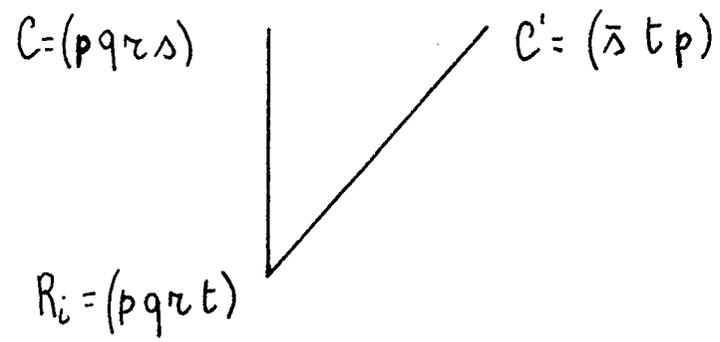


Fig: 14

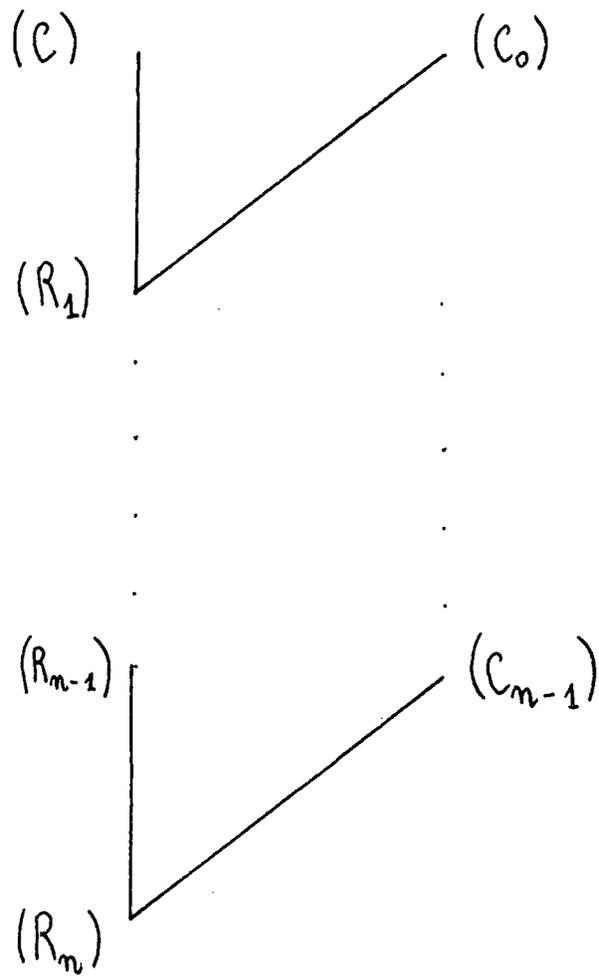


Fig: 15

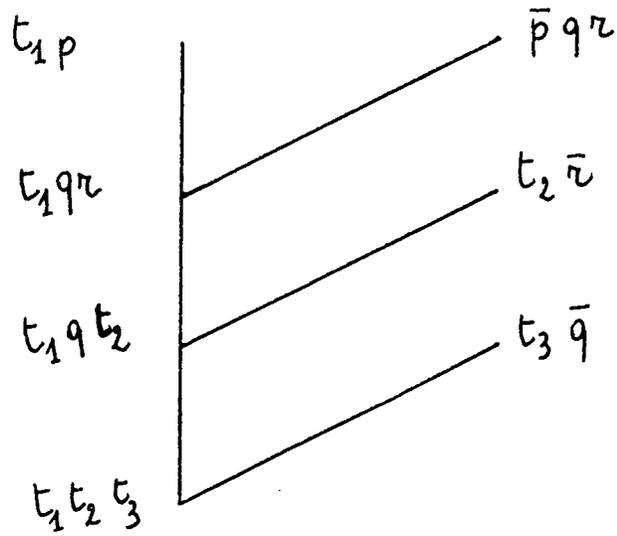


Fig: 16

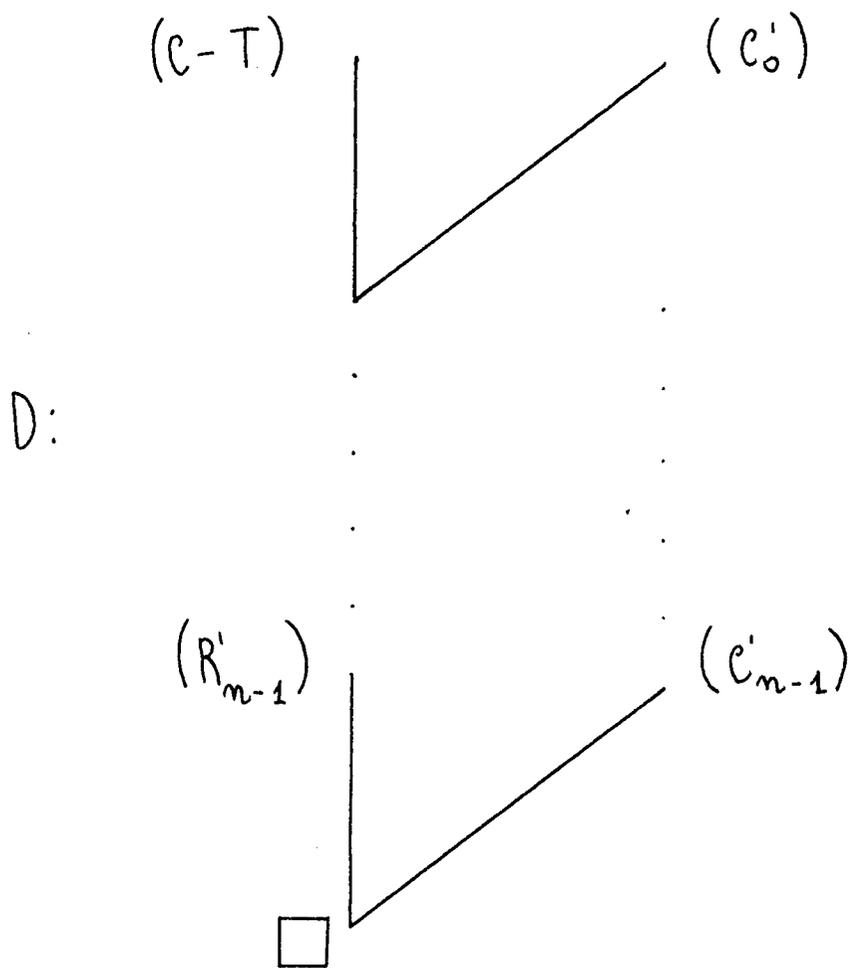


Fig: 17

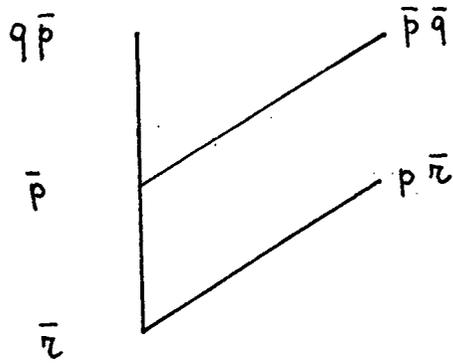


Fig: 18

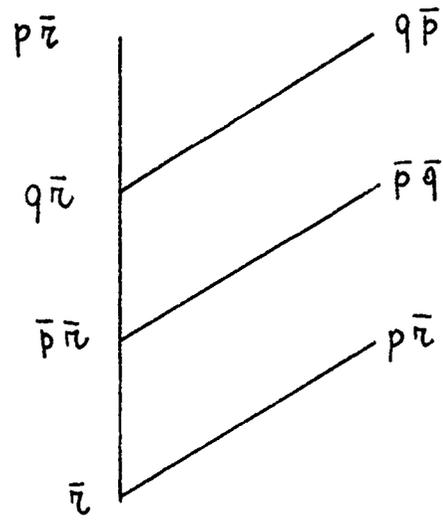


Fig: 19

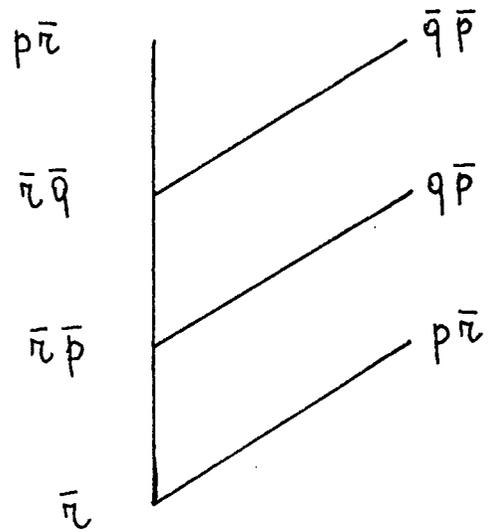


Fig: 20

BIBLIOGRAPHY

1. Anderson, R., Bledsoe, W.W. A Next Technique for Establishing Completeness. J. ACM, Vol.17, 3 (July 1970) pp.525-534.
2. Andrews, P.B. Resolution With Merging. J. ACM, Vol.15, 3(July 1968)pp.367-381.
3. Chang, C.L. The Unit Proof and the Input Proof in Theorem Proving. J.ACM, 17,4 (October 1970) pp.698-707.
4. Davis, M. Eliminating the Irrelevant from Mechanical Proofs. Proc. of Symposia in Applied Mathematics. American Mathematical Society (1963) pp.15-30.
5. Lee, R.C.T. A Completeness Theorem and a Computer Program for Finding Theorems Derivable from Given Axioms. Doctoral Dissertation, Department of Electrical Engineering and Computer Science, University of California, Berkeley, 1967.
6. Loveland, D.W. A Linear Format for Resolution. Carnegie Mellon University, Pittsburg, Pa., December 1968.
7. Mendelson, E. Introduction to Mathematical Logic. Princeton N.J., Van Nostrand 1964.
8. Reiter, R. Two Results on Ordering for Resolution with Merging and Linear Format. J. ACM, 18, 4(October 1971), pp.630-646.
9. Robinson, J.A. A Machine Oriented Logic Based on the Resolution Principle. J. ACM, Vol.7 (January 1965) pp.23-41.
10. Robinson, J.A. A Review of Automatic Theorem Proving. Annual Symposia in Applied Mathematics, Vol.XIX (1965).
11. Slagle, J.R. Automatic Theorem Proving with Renamable and Semantic Resolution. J. ACM, Vol.12, 4(October 1967), pp.687-697.
12. Slagle, J.R., Chang, C.L., Lee, R.C.T. Completeness Theorems for Semantic Resolution for Consequence Finding. Div. of Computer Res. and Technol. National Institute of Health, Bethesda, Md., 1969.
13. Wos, L., Robinson, G.A., Carson, D.F. Efficiency and Completeness of the Set of Support Strategy in Theorem Proving. J. ACM, Vol.12, 4 (October 1965), pp.536-541.
14. Wos, L., Robinson, G.A. Paramodulation and Set of Support. Symposium on Automatic Demonstration. Versailles (France), December 1968.