SOME LINEAR HERBRAND PROOF PROCEDURES:
AN ANALYSIS

D. W. Loveland

December, 1970

This work was supported by the Advanced Research Projects Agency of the Office of the Secretary of Defense (F44620-70-C-0107) and is monitored by the Air Force Office of Scientific Research. Support was also received through NSF grants GJ-381 and GJ-580.

ABSTRACT

Several Herbrand proof procedures proposed during the 1960 decade are shown to be related in varying degrees. Most of the paper deals with a relationship between s-linear resolution and model elimination. Refinements of each are proposed and the spaces of ground deductions are shown to be isomorphic in a suitable sense. The two refined procedures are then studied at the general level where they are no longer isomorphic and do not always relate to a natural ground level counterpart. Other topics considered are the introduction of an added merge condition to model elimination and also an expanded possible use of lemmas. Finally, the model elimination procedure is interpreted in the linked conjunct procedure of Davis and the matrix reduction procedure of Prawitz.

Some Linear Herbrand Proof Procedures: An Analysis

D. W. Loveland

1. Introduction. Among the several Herbrand proof procedures introduced during the 1960 decade, resolution [16] has been the procedure most widely studied and implemented. A natural question is whether or not other procedures should be pursued more vigorously because they would lead to the most successful future procedures if properly developed. Results of implementation are very useful in this regard (and ultimately the only test) but implementations are rare and nonuniform in quality (and power of machine), and report only present abilities. One would wish for a general theory of efficiency applicable to actual procedures and hopefully some progress will be made in this direction, but presently there is none. A useful alternative is to relate procedures so that estimates of relative performance are possible.

This paper relates certain strategies of the resolution, linked conjunct [4] and matrix reduction [14] procedures to model elimination [8]. Because a very close tie, indeed, an "isomorphism" of ground level, exists between a refinement of model elimination (ME) and a refinement of s-linear resolution [10] which we explicitly spell out, one can see a (weaker) relationship between resolution and the remaining two procedures.

*The author has just learned of the following report which also establishes the basic result given here: Kowalski, R. and Kuehner, D., "Linear resolution with selection function", Memo 34 (Oct., 1970), Mètamathematics unit, Edinburgh University, Scotland. Commendably, they also tackle a notion of efficiency for proof procedures.

By a linear procedure, we mean a procedure whose trial deductions proceed on a line-by-line basis, where a given entry is always used in the rule-of-inference yielding the next entry. S-linear resolution and ME are inherently of this form. The other two are not inherently linear but we show a natural strategy in each which is linear. We offer no great defense of linear procedures here. We do remark that they easily adapt to the General Problem Solver [12] format where one always tries to reduce the difference between the last line and a goal. Linear procedures are also particularly attractive for man-machine interactive programs because man tends to think linearly at least over small segments of a problem. Specific "defense" of the alleged desirability of each procedure is given in the paper introducing the procedure (e.g., see [7]).

Most of the paper involves the relationship between ME and s-linear resolution because of the accumulating knowledge concerning linear resolution (e.g. [11], [3], [1], [20]). The relationship between ME and the other procedures is handled much more succinctly. In the ME/resolution portion we try to emphasize behavior at the so-called general level which differs somewhat from the ground level (propositional) behavior, although for the most part determined by the ground level behavior, of course. This seems important because the general level is the level at which theorem provers really operate. We do not know how to be precise in discussing the general level, but we spend some effort considering distinctions with the ground level situation and with variations of the related procedures. A better understanding of procedures at the general level would be most helpful.

The paper is intended to be read in either of two ways. The studious reader may go through in order, thus understanding in detail the ground level case before going on. Others may prefer to skip the proofs in sections 3, 4, and 7 and read sections 2, 5, 6, and 8 first. Proofs have been included that the author feels are elegant, such as the Anderson-Bledsoe Technique [1] of proving completeness (theorem 1), and/or are especially informative in helping understand the procedures; the rest are omitted.

2. Preliminaries and basic result. We assume that the wff to be tested for unsatisfiability is in the form of a set S of clauses, each clause being a set of literals. The set S is the usual abstraction from a wff in prenex normal form with its matrix in conjunctive normal form and with Skolem function instances replacing existentially quantified variables. For a clear account of the preparation of such a set from a given wff see [4]. Familiarity with the procedures of s-linear resolution [10] and model elimination, [8] or [9], will be useful, although the definitions will be reviewed here. The procedures of [4] and [14] appear only in the last section. Standard terms from [16] and [19] are used without further reference.

Resolution may be taken as an operation mapping two parent clauses B and C into a resolvent clause D. If B and C are ground clauses and if $L_1 \in B$ and $L_2 \in C$ are complementary literals then the ground resolvent of B and C is the set $(B-\{L_1\} \cup (C-\{L_2\})$. The resolvent of arbitrary clauses B and C requires suitable instantiation followed by the operation shown above for ground resolution. The literals of B and C which under instantiation form the complementary literals are recorded in the key triple defined by Robinson. An alternate way of treating the general resolution operation on two clauses is to form all factors of clauses and then only one literal from each of the two parent clauses need appear in the complementary pair of literals for resolution. The literals which go into the complementary pair upon appropriate instantiation by a most general unifier substitution are called the literals resolved upon (as may their instances which form the complementary pair). The definitions given here will be based on the use of factoring rather than the key triple device. This will make the formulation of s-linear resolution differ technically from the presentation in [10] where the key triple approach is used. However, the factoring approach greatly simplifies the presentation of the additional constraints on s-linear resolution with which we will be concerned here.

A deduction of clause C from S, where S is a set of clauses, is a finite sequence $B_1, B_2, \ldots, B_n$ of clauses such that

(i) B_i, $1 \le i \le n$, is either

(a) in S,

(b) a factor of (a clause of) S,

(c) a factor of B_j, $j < i$,

(d) a resolvent of two clauses each either in S, a factor of S, a B_j or a factor of B_j, $j < i$;

(ii) B_n is C

It is somewhat inelegant that a factor of a clause need not appear explicitly to be used as a parent of a resolvent. If this were not permitted, then the definition of linear deduction (see below) given here would differ in substance from the definition of linear deduction in terms of the key triple (see [10]). By "in substance" we mean that there would exist a linear deduction in the "key triple" sense which could not be directly converted to a linear deduction with factors. A minor distinction with the definition of deduction in [10] is that all members of S to be used in the deduction need not appear explicitly in the deduction as defined here.

A refutation of the set S is a deduction of the <u>empty clause</u> □ .

A <u>linear deduction of</u> (clause) C <u>from</u> S, where S is a set of clauses, is a deduction $B_1, \ldots, B_n$ such that B_1 is in S and for $1 \le i < n$ either B_{i+1} is a resolvent with B_i as one parent, or B_{i+1} is a factor of B_i. In a linear deduction, if B_{i+1} is a resolvent, then B_i is the <u>near parent</u> of B_{i+1} and the other parent is the <u>far parent</u>. Note that the far parent may be a factor of a clause appearing earlier but the near parent must be the appropriate factor.

Given two distinct clauses B and C, B <u>subsumes</u> C precisely if an instance of B is a subset of C, i.e. $B\sigma \subseteq C$ for some substitution σ. Here B may be the empty clause.

An <u>s-linear deduction of</u> (clause) C <u>from</u> S is a sequence $B_1, \ldots, B_n$ of clauses such that

(i) the sequence is a linear deduction of C from S;

(ii) the far parent of B_{i+1}, if B_{i+1} is a resolvent, is either

(a) a clause, or a factor of a clause, from S, or

(b) a B_j, $j < i$, such that B_{i+1} subsumes an instance of B_i (we then say B_{i+1} is obtained by an <u>s-resolution</u> operation);

(iii) no tautology occurs in the deduction.

When viewing an s-linear deduction, it is sometimes convenient to work with the set S^f consisting of S and all factors of clauses of S for then all factors concerning the deduction appear explicitly, either in S^f or in the deduction. For later convenience we have chosen to disallow in this definition the use of an unstated factor of an earlier clause for the far parent even though the resulting definition is slightly more restrictive than that of [10]. Because of the strong constraint on the far parent for s-linear deductions, there are probably few deductions that arise in practice that meet one definition and not the other. The ground deduction proof of completeness in [10] still is applicable to this definition by an appropriate "lifting" argument. This distinction is discussed further in section 5. It is also a consequence of theorem 1 of this paper that s-linear resolution (in either form) is a complete procedure. (Luckham independently showed in [11] that linear resolution, which he called "ancestry filter form," is a complete procedure.)

When reference is made to a <u>ground literal</u>, <u>ground clause</u>, <u>ground deduction</u>, etc., this implies that all the <u>terms</u> of the formal language appearing in the entity under discussion are members of the <u>Herbrand universe</u> H (S) formed from the given set S of clauses. We take the Herbrand universe to be the set of all (well-defined) terms composed from the constants of S where an individual constant is supplied if none appears in S.

Let D be a given deduction $B_1, \ldots, B_n$ of B_n from S. D' is an <u>image ground deduction</u> of D if D' is a ground deduction $B'_{r(1)}, \ldots, B'_{r(n)}$ of $B'_{r(n)}$ where (i) B'_i is a ground instance of B_i, i.e. a ground clause obtained from B_i by appropriate substitution and with redundant literals removed, and (ii) $r(i) < r(i+1)$ such that if B_j is not a factor of B_k, $k < j$, then $j=r(i)$ for some i, $1 \leq i \leq m$. Intuitively, an image ground deduction of deduction $B_1, \ldots, B_n$ is a deduction formed from $B'_1, B'_2, \ldots, B'_n$, where B'_i is a ground instance of B_i, by removing some (usually all) of the redundant clauses. Factoring is a vacuous operation in a ground deduction where no instantiation is possible. D' is an <u>image s-linear ground deduction</u> of s-linear deduction D if D' is an image ground deduction of D and D' is s-linear. Image ground deductions with other modifiers are defined in a similar manner. Not every s-linear deduction has an image s-linear ground deduction; this is explicitly considered in section 5.

An <u>ordered clause</u> is a set of literals with a (total) ordering relation over these literals. We denote the order relation by $\leq$ and speak of a literal being less than, less than or equal to, etc., another literal of the clause. It is convenient to set the convention that left-right positioning of literals indicates the ordering with literal L_1 less than ($<$) literal L_2 if and only if L_1 is to the left of L_2. Thus for a written ordered clause the rightmost literal is the greatest literal of the clause.

For an ordinary, i.e. unordered, clause the set notation dictates that only one occurrence of any literal appears in a clause. In an ordered clause there might be several occurrences of a given literal. Although this is in general true there is a particular convention for eliminating multiple occurrences of a literal in an ordered clause that is appropriate for the procedures studied here. The <u>left occurrence convention</u> (or <u>least occurrence convention</u>) makes each ordered clause with a literal of multiple occurrences equivalent to the clause containing only the leftmost (least) occurrence of any multiple occurrence literal, the other occurrences being deleted. The latter clause replaces the former clause which is not considered well-formed under the convention. Thus PxPuPx is ill-formed, and is really PxPu under this convention. For brevity we omit brackets, parentheses and commas when writing clauses; thus PxPuPx is a shorthand for (Px,Pu,Px) where parentheses replace brackets to indicate an ordered clause. With this shorthand, "set" vs. "set with order relation" is determined by context.

An <u>ordered clause deduction</u> is an s-linear deduction $B_1,\ldots,B_n$ with all clauses ordered and such that for all i, $1 \leq i < n$,

(i) the literals of B_{i+1} derived by instantiation from B_i are ordered as determined by B_i and if B_{i+1} is a resolvent, all literals of the far parent appear in some chosen arbitrary order to the right of the literals derived from B_i;

(ii) the left occurrence convention is applicable to all clauses and their instances;

(iii) all factoring must include the rightmost literal of B_i and all resolutions have the rightmost literal of B_i as the literal resolved upon.

By (iii) we note that all unification concerning B_i involves the rightmost literal of B_i. Note there is no added constraint on the far parent even if it is a B_j for $j < i$. In particular, there is no requirement that the literal resolved upon in B_j be the rightmost in B_j although it will be seen in section 4 that there is in no sense a loss in demanding this. It is important to observe the effect of the left occurrence convention on the resolution operation. If B_i is PaQyPx then -Pa is not a suitable far parent because unification of {Px,Pa} results in near parent clause instance PaQyPa. This is really PaQy which cannot be ground resolved with -Pa as only the rightmost literal, here Qy, may be resolved upon. Examples of ordered clause refutation are given below.

A deduction is <u>tight</u> if no B_i subsumes B_j for $i < j$. If the deduction is an ordered clause deduction the subsumption test is applied to the clauses disregarding order. Thus the ordered clause deduction QR,Q,PRT,PRQ from set S = {QR,-R,-QPRT,-TQ} is not tight as QR subsumes PRQ. An s-linear deduction satisfies the <u>subsumption rule</u> if and only if every permitted s-resolution of the s-linear deduction is performed. For an ordered clause deduction the permitted resolutions must resolve on the rightmost literal of the near parent clause. The ordered clause deduction PT,PR,P-T,PQ does not satisfy the subsumption rule as PT is an appropriate far parent for near parent P-T with resolvent P instead of PQ as given. Here the set S is {PT,-TR,-R-T,TQ}.

We now give an example of a tight ordered clause refutation satisfying the subsumption rule (abbreviated as a TOCS refutation). This refutation has image TOCS ground refutations, one of which we give immediately afterwards.

Example:	S :	Px Qy	I
		Px-Qy	II
		-Px Qy	III
		-Px-Qy	IV

Refutation:	1.	Px Qy	I
	2.	Px Pu	resolvent using II
	3.	Px	factor of line 2
	4.	Qy	resolvent using III
	5.	-Px	" " IV
	6.	□	s-resolvent using line 3

Image ground refutation:

1.	Pa Qa	I'
2.	Pa	resolvent using II'
3.	Qa	" " III'
4.	-Pa	" " IV'
5.	$\square$	s-resolvent using line 2.

The ground image of S is:

Pa Qa	I'
Pa-Qa	II'
-Pa Qa	III'
-Pa-Qa	IV'

Because upon instantiation of the constant a for each variable in the first refutation, we get a ground deduction with lines 2 and 3 identical (by the left occurrence convention), the r function of the definition of image ground deduction is here given by r(1)=1, r(2)=2, r(3)=4, r(4)=5, r(5)=6. Recall the r function gives the correspondence of ground deduction lines to original deduction lines so as to omit redundant lines in the former deduction.

In section 5 we shall see that the ground TOCS deduction doesn't always have a general TOCS deduction for which it is an image ground deduction. This occurs because of our definitions of the tightness condition and the subsumption rule. A variation of TOCS deduction is introduced in section 5 which overcomes the situation.

We now turn to model elimination. As for s-linear resolution we suppress in our review the details on finding the appropriate instantiation and similar matters which are treated in detail in [8] or [9]. In this section we present ME without the lemma device.

The basic element of ME is a <u>chain</u> which is a finite sequence of literals. Literals in chains are of two types, A-literals or B-literals. The simplest chain is the <u>elementary chain</u> consisting of only B-literals. The <u>matrix set</u> (or <u>initial set</u>) M of chains is the set of elementary chains formed from the given set S of clauses by creating one chain for each literal of each clause of S. For each literal the associated chain has that literal as first literal with the remaining members of the clause following in some chosen order. A member of the matrix set is a <u>matrix</u> (or <u>initial</u>) <u>chain</u>.

It is convenient to adopt again the left-right ordering notation to reflect the ordering of literals in a chain. Thus, for example, the last element of the chain is the rightmost element. We use "last element" rather than "greatest element" as it is the more common notation for finite sequences and is in keeping with [8], [9].

Not all chains are considered well-formed; we are interested (only momentarily in this paper) in the class of preadmissible chains. A chain is preadmissible precisely if

(i) any two complementary B-literals are separated by an A-literal;

(ii) no B-literal identical to an A-literal appears to the right of the A-literal;

(iii) no two A-literals are identical or complementary. A chain is admissible if it is preadmissible and the rightmost element is a B-literal. The empty chain $\emptyset$, the chain with no members, is by definition admissible.

There are three operations which we outline here.

(Basic) extension. The extension operation has two inputs, an admissible chain K, called the parent chain and as the second chain, an elementary chain K_2. If by appropriate substitutions θ_1, θ_2 the rightmost literal of $K_1\theta_1$ is made complementary to the leftmost literal of $K_2\theta_2$, a new chain K_3 is formed by placing the chain $K_2\theta_2$-, i.e. $K_2\theta_2$ minus the leftmost literal, to the right of the chain $K_1\theta_1$. In K_3 the rightmost literal of $K_2\theta_2$ is made an A-literal; all the other literals in K_3 retain the classification of the literals from which they were derived.

(Basic) reduction. The reduction operation has one input, an admissible chain K as parent clause. If by appropriate substitution θ the rightmost literal of $K\theta$ is made complementary to an A-literal of $K\theta$, then a new chain K_1 is formed by simply removing the rightmost literal from $K\theta$. All literals of K_1 retain the classification of the literals from which they are derived. This reduction operation differs from that of [8] and [9] in that only the rightmost B-literal, i.e. the rightmost literal, is removed.

(Basic) contraction. The contraction operation has one input, a preadmissible nonadmissible chain K as parent chain. A new chain K_1 is formed exactly like K except the A-literals to the right of the rightmost B-literal are deleted.

In the extension and reduction operations the appropriate substitutions involve most general unifying substitutions as for resolution.

For the purposes of this paper we wish to imbed the contraction operation in each of the other operations. To this end we define the c-extension and c-reduction operations as follows:

> The c-extension (resp. c-reduction) operation is exactly like the extension (resp. reduction) operations except that if a preadmissible nonadmissible chain is formed by the extension (resp. reduction) operation, then the contraction operation is performed on this chain and the resultant admissible chain is the output of the operation.

This is a natural alteration as the contraction operation is the mandatory operation on a preadmissible nonadmissible chain and thus is not a branch point on the search tree as is the case for extension and reduction. (Contraction was originally introduced as a separate operation simply for expository reasons.)

An operation not part of the usual ME procedure, but of use in this paper, is the c-factoring operation. This operation has as input an admissible chain K as parent chain. If θ is a most general unifier of the rightmost literal (a B-literal) with another B-literal of K then $K\theta$ minus the last B-literal is the output chain unless contraction is then applicable, in which case the output of the operation is the output of the contraction operation. Note the similarity with the c-reduction operation; the distinction is whether the non-rightmost literal involved is an A-literal and potentially complementary or a B-literal and potentially identical. As an example, if underlining indicates A-literals then the chain Px$\underline{Qy}$Pa yields Pa under the c-factoring operation; also PaQyQaPy yields PaQaQa under the operation.

An ME deduction with factoring of K from S, where K is a chain and S is a set of clauses, is a sequence $K_1, \ldots, K_n$ of chains such that

(i) K_1 is a matrix chain formed from S;

(ii) K_{i+1} is formed by c-extension, c-reduction or c-factoring with K_i as parent chain, $1 \leq i < n$;

(iii) K_n is K.

An ME refutation with factoring of S is an ME deduction with factoring of the empty chain $\emptyset$ from S. Note that all chains, except perhaps K, must be admissible in an ME deduction as defined above.

A *standard ME deduction of K from S*, where K is a chain and S is a set of clauses, will mean an ME deduction with factoring of K from S with the c-factoring operation deleted. A standard ME deduction of K from S is essentially the procedure of [8] and [9]. [8] contains a proof that the standard ME procedure (and hence the ME procedure with factoring) is complete.

We want to consider certain further modifications of ME. In this direction we consider a left occurrence convention for chains. The *ME left occurrence convention* defines an equivalence between any chain K with multiple B-literal occurrences of literal L and a chain K_1 like K except every occurrence of L but the leftmost has been deleted. The chain K is considered ill-defined and is replaced by K_1 in any deduction. Also, a chain K_2 with a B-literal occurrence of literal L and an A-literal occurrence of L to its right is ill-defined by this convention and has no replacement. That is, no deduction contains a clause of form K_2 nor any equivalent. In this case we say the clause is *deleted* from a deduction, in contrast to being replaced by a well-defined equivalent. We recall that if the leftmost occurrence of a multiple occurrence literal is an A-literal, then the chain is deleted because it is inadmissible. Thus if the left occurrence convention is invoked we see that no chain of a deduction has more than one occurrence of any literal.

Recall that we underline literals in a chain to indicate those literals are A-literals. Then under the ME left occurrence convention the chain $P\underline{Q}P\text{-}Q$ is ill-defined and should be written $P\underline{Q}\text{-}Q$. We make a remark concerning implementation here. It is possible and seemingly convenient to regard ME as a procedure where one works on only the rightmost literal of the last chain (just as we have organized a TOCS deduction). When the left occurrence convention is employed, removal of a literal L not rightmost in a chain of a deduction can be postponed until L becomes a rightmost literal for it may never need be removed if the deduction is rejected. However, the full convention always applies to the elementary chain used by c-extension. For the remainder of the paper this modification will be assumed to be part of the ME left occurrence convention. This preserves the property that only rightmost literals are deleted from chains in the deduction. For more discussion of this convention see the first paragraph of section 7.

[Note: To establish the isomorphism with the TOCS deduction as we do later, we must also choose to delay removal of non-rightmost identical literals for TOCS deductions. Unless explicitly mentioned, comments pertaining to the ME left occurrence convention option apply to the TOCS left occurrence convention also. We choose to detail the ME option as the ME format seems more desirable to implement than the TOCS format. Besides some implementation advantages, this option does simplify slightly some proofs but no proof really depends on the option. If one wishes to avoid this option in implementation, one should also perform every c-reduction immediately without delaying until the B-literal is rightmost. Although one loses the property that chains change only at their righthand members, one simplifies the notion of an acceptable chain in strong ME, defined below, to a chain with no two literals having identical atoms.]

An ME deduction satisfies the *tautology rule* precisely if it contains no chains with either complementary B-literals or a B-literal with a complementary A-literal to its right. For example, the chains $P\underline{Q}{-P}$ and $P\underline{Q}{-\underline{P}}$ cannot appear in any deduction satisfying the tautology rule.

A *strong ME deduction of* K *from* S is an ME deduction with factoring satisfying the ME left occurrence convention and the tautology rule.

Given a strong ME deduction we shall wish to refer to an *image strong ME* ground deduction which is the obvious extension of image ground deduction to the strong ME procedure. Again the r function may not be the identity function if c-factoring occurs in the given deduction.

We now give as an example of a strong ME refutation, the refutation of the set used to illustrate a TOCS refutation. Again, we give an image strong ME ground refutation immediately afterwards.

Example:	S:	Px Qy	I	,
		Px -Qy	II	,
		-Px Qy	III	,
		-Px -Qy	IV	.

		Chain	How obtained	
Refutation	1.	Px Qy	I	,
	2.	$Px\ \underline{Qy}\ Pu$	c-extension using II	,
	3.	Px	c-factor	,
	4.	$\underline{Px}\ Qy$	c-extension using III	,
	5.	$\underline{Px}\ \underline{Qy}\ {-Pu}$	" " IV	,
	6.	$\emptyset$	c-reduction	.

Image ground refutation

	Chain	How obtained	
1.	Pa Qa	I'	,
2.	Pa	c-extension using II'	,
3.	<u>Pa</u> Qa	c-extension " III'	,
4.	<u>Pa</u> <u>Qa</u> -Pa	c-extension " IV'	,
5.	$\emptyset$	c-reduction	

The ground image of S is:		
Pa Qa	I'	,
Pa -Qa	II'	,
-Pa Qa	III'	,
-Pa -Qa	IV'	.

We summarize the processing of chains with identical or complementary literals within the strong ME procedure. The rules have been added piecemeal which obscures the fact that with this procedure any occurrence of an identical or complementary pair of literals provokes a fixed action that eventually eliminates one of the pair. A chain meeting these conditions is called <u>acceptable</u>.

The table below indicates the situation with regard to a pair of literals within an acceptable chain. If a chain has more than one applicable pairing, the processing proceeds by identifying one pair of literals at a time.

	Type (class) of left literal of pair	Type (class) of right literal of pair	Relationship of literals	Appropriate Action
1.	A	A or B	identical	discard chain
2.	A	A	comple.	discard chain
3.	A	B	comple.	c-reduction forced[1] when B-literal becomes rightmost literal
4.	B	A or B	comple.	discard chain
5.	B	B	identical	replace by chain with right occurrence of literal deleted[2]
6.	B	A	identical	discard chain

Table 1.

[1] C-reduction is forced when the appropriate B-literal is the rightmost literal, for c-extension would create complementary A-literals which are discarded.

[2] By a convention set earlier this deletion is postponed until the literal to be deleted is the rightmost literal.

We now state two theorems concerning the foregoing procedures, of which the second theorem is our key result. The theorems concern ground deductions; the implications for the general setting are considered in section 5.

<u>Theorem 1</u>. If S is a set of ground clauses then S is unsatisfiable if and only if there is a TOCS ground refutation of S. If S is a minimally unsatisfiable set then any clause C of S may be the first clause of the deduction with any desired ordering placed on C.

If K is a chain, let f(K) denote the ordered clause consisting of the B-literals of K in the order given by K. The following theorem is an isomorphism theorem in that it gives a 1-1 correspondence between deductions preserving the length property.

<u>Theorem 2</u> (the Isomorphism theorem). If $K_1, \ldots, K_n$ is a strong ME ground deduction of K_n from a set S of clauses then $f(K_1), \ldots, f(K_n)$ is a unique TOCS ground deduction of $f(K_n)$ from S. Conversely, if $C_1, \ldots, C_n$ is a TOCS ground deduction of C_n from S then there is a unique strong ME ground deduction $K_1, \ldots, K_n$ such that $f(K_i) = C_i$, $1 \le i \le n$.

If the reader returns to the example at the end of each procedure description, he will observe the relationship between procedures given by theorem 2. Also consider the following propositional example.

<u>Example</u>: $S = \{ABD, -BC, -B-CD, -D, -A-C, -AC\}$

A TOCS refutation:

1.	ABD	member of S			,
2.	AB	resolvent using		-D	,
3.	AC	"	"	-BC	,
4.	A -BD	"	"	-B-CD	,
5.	A -B	"	"	-D	,
6.	A	s-resolvent	"	line 2	,
7.	-C	resolvent	"	-A -C	,
8.	-A	"	"	-AC	,
9.	$\square$	s-resolvent	"	line 6	.

A strong ME refutation:

1.	ABD	chain from S			,
2.	AB	c-extension using		-D	,
3.	$A\underline{B}C$	"	"	-BC	,
4.	$A\underline{BC}$ -BD	"	"	-C -BD	,
5.	$A\underline{BC}$ -B	"	"	-D	,
6.	A	c-reduction			,
7.	$\underline{A}$ -C	c-extension using		-A -C	,
8.	$\underline{A}$ -$\underline{C}$ -A	"	"	C -A	,
9.	$\emptyset$	c-reduction			.

Note that at lines 4 and 5 of the TOCS deduction, if resolution was not confined to the last literal of line 4, the subsumption condition would be applicable to yield AD at line 5. This provides an alternate route to step 6 which is voided by the ordered clause strategy. The value of this strategy is in voiding many redundant paths in the search space. This phenomenon has been noted by several investigators (e.g., see [18], [6]).

The isomophism theorem yields a number of potentially useful observations, not all evident from our examples. One evident property is that the ME notation provides a fast subsumption check; c-reduction corresponds to s-resolution but only the A-literals of the present chain are searched and each check is a check for a complementary pair as opposed to a search over the history of the deduction with sometimes a multiple literal matching to complete the subsumption test. For example, in step 8 of the strong ME refutation of the last example only two "easily located" A-literals needed to be checked to reveal the c-reduction rather than a check back over seven previous steps. The contrast is sharper when a subsumption does not exist for that is when the whole list is checked before a negative answer is obtained. This is the more usual situation as can be surmised from the examples given. This suggests that the strong ME procedure is the appropriate way to implement s-linear resolution. Also some of the discard conditions (see previous table) involving A-literals have no easily implemented counterpart within s-linear resolution.

The modifications already described that sharpen both the s-linear resolution and the ME procedures is another dividend of the isomorphism theorem. For example, having shown the TOCS procedure complete via theorem 1, we have shown the tautology rule does not destroy the completeness of strong ME. This is not true of standard ME (as pointed out to the author by Henry Goldberg). Also, suggested by theorem 2, the strong ME procedure acquires new lemma making machinery not directly recognized as applicable to the procedure (see section 6).

It should be remarked that we are not gathering all the best that is known about the s-linear resolution and ME in these improvements. For example, c-factoring has been added as an operation to ME in formulating strong ME. Thus c-factors of a chain need be considered as well as the chain itself which gives a branching of possibilities not present in standard ME. However, there are situations where having c-factoring speeds things up.

Another example of known restrictions not incorporated applies to s-linear resolution. It has been shown that a merge condition is compatible with s-linear resolution. This is considered in the extensive remark below.

Remark. It has been shown by Anderson and Bledsoe [1], and independently by Yates, Raphael and Hart [20], that resolution with merging, introduced in [2] and sharpened in [15], is compatible with s-linear resolution. The notion of merging is defined for ground clauses as follows: if C_1 and C_2 are (non-tautologous) ground clauses with a resolvent C_3 then C_3 is a merge resolvent with merge literals $L_1, \ldots, L_n$, $n \geq 1$ if and only if there exists literals $L_1, \ldots, L_n$, which are in both C_1 and C_2. For non-ground clauses a factoring of C_3 is usually needed to unify two literal occurrences in C_3 to obtain the actual merge resolvent. It can be shown (see "Remark" in section 3) that the following restriction is compatible with the TOCS procedure:

If the far parent is not a member of S then the far parent is a merge resolvent and the literal to be resolved upon is a merge literal of the far parent.

For convenience we call the augmented procedure the MTOCS procedure. It is natural to ask if there is a corresponding restriction or alteration to ME so that an isomorphism of deductions exists. Then the benefits (and costs) of the ME notation and this more restricted resolution procedure could be shared. Such a modification to strong ME is possible; we discuss it below for ground deductions and in section 5 for the general form.

We will state here the version of strong ME which is related by an isomorphism theorem to MTOCS deduction as strong ME relates to TOCS deduction. (The isomorphism theorem involves ground deductions as before.) We do not state formally or prove this version of the isomorphism theorem which is really an extension of theorem 2. The justification of the claim of isomorphism is left as an exercise to the interested reader; hopefully, useful comments will be included as an aid and also to help understand how this sharpened set of procedures compares to the TOCS/strong ME procedures. We call the revised strong ME version below the m-strong ME procedure. We include a modified form of m-strong ME procedures for reasons considered below and in section 5. The modifications are placed in brackets. When we say that a class X literal becomes a class Y literal in the definition below, we mean that the literal whose parent in the preceding chain is a class X literal is a class Y literal. Otherwise class is unaffected.

An ME deduction $K_1, \ldots, K_n$ is a [modified] m-strong ME deduction of K_n if it is like a strong ME deduction except for the following changes.

(1) There are four classes of literals: A, Am, B, Bm. The Am-literal is an A-literal, the Bm-literal is a B-literal.

(2) A Bm-literal becomes an Am-literal (in the following chain) upon performing a c-extension when this literal is rightmost, in the same manner as a B-literal becomes an A-literal.

(3) A B-literal becomes a Bm-literal precisely if:

(a) a c-reduction is performed and the B-literal is to the left of the A-literal involved in the c-reduction;

(b) the literal in the new chain is a unification of two or more literals by the left occurrence convention or c-factorization;

(4) A Bm-literal becomes a B-literal precisely if a c-reduction or c-extension is performed and the literal does not "become" a Bm-literal by case 3 or become an Am-literal by case 2.

(5) C-reduction is permitted only with an Am-literal [unless some A-literal is already complementary to the rightmost literal and then that c-reduction is required];

(6) If two complementary A-literals occur in a chain the chain is not discarded unless the left A-literal of the complementary pair is an Am-literal.

(7) [If two complementary A-literals occur in a chain and the left A-literal is not an Am-literal, the chain is replaced by a chain having the rightmost of the complementary A-literals removed along with all literals to its right.]

Although this appears somewhat involved, the above is a direct incorporation of the merge concept as translated through the isomorphism. A key point in understanding the conversion is the following fact about TOCS deduction. Although not required by definition of a TOCS deduction, all s-resolutions actually can resolve on only the rightmost literal of the earlier clause. This is shown in the proof of theorem 2. The Bm-literals in essence mark the merged literals of a resolvent and the Am-literals encode earlier merge resolvents available for resolution. For ground deductions the MTOCS (or m-strong ME) procedure has no advantage over the TOCS (or strong ME) procedure, except possibly fewer entries (A-literals) to check for subsumption (c-reduction), because every subsumption resolution (c-reduction) is locally desirable and executing each such opportunity is all right, i.e. completeness is preserved. On the other hand, as only merge resolvents are allowed as candidates for resolution with later clauses in a deduction, it is easy to construct examples (see section 5) where the deduction is longer than for a TOCS deduction because some subsumption resolutions are missed. The modification to the m-strong ME procedure corrects the fault at the ground deduction level and, in particular, makes the modified m-strong ME procedure behave as the strong ME procedure for ground deductions. The general situation is quite different; see section 5. Examples of MTOCS deduction appear in section 5.

3. Proof of theorem 1. Because the proof of theorem 1 follows closely the manner of the elegant proof of Anderson and Bledsoe's theorem 5 [1], we give a somewhat brief description of the proof. The soundness of the TOCS procedure is a consequence of the soundness of resolution. The proof of completeness is by induction on the number k of "excess literals" of S, i.e. k is the number of literals in clauses of S minus the number of clauses of S. The precise statement we prove is: if S is a minimally unsatisfiable set of ground clauses and C is an ordered clause of S then there exists a TOCS ground refutation of S with C as first clause.

Induction base, k=0. The only possibility for S is two one-literal (unit) clauses, giving a TOCS refutation of length two starting with either clause.

Induction step, assume true for $k < n$, show true for n, $n > 0$.

Case 1. There is a unit clause $\{L\}$ in S.

Subcase a: $C \neq \{L\}$, i.e. $\{L\}$ is not chosen for first clause. Let us denote the complement of literal L by L^c. We note a fact: if S is a minimally unsatisfiable set containing unit clause $\{L\}$ and S' is a set formed from S by removing clause $\{L\}$ and deleting L^c from all other clauses of S, then S' is minimally unsatisfiable. For if S' were satisfiable, clearly also S would be by including L in the model, and if a proper subset of S' is unsatisfiable, so would be the proper subset of S consisting of the unsatisfiable subset of S' with L^c returned to the clauses plus $\{L\}$. Note that no other clause of S contains L if $\{L\} \in S$ by minimality. Let $C' \in S'$ be the ordered clause C with L^c deleted if $L^c \in C$. S' has fewer than n excess literals because L^c must always be an excess literal (as $n > 0$ so $\{L^c\} \notin S$). Let D' be a TOCS refutation of S' with C' as first clause as given by induction hypothesis. Alter D' to form D, a refutation of S, by inserting L^c at the appropriate places throughout and whenever L^c is a rightmost literal insert a resolution with $\{L\}$ as far parent. D is a TOCS refutation as desired.

Subcase b: $C = \{L\}$. Choose a C' in S with $L^c \in C'$ and form a refutation D of S from C' as done in subcase a. We may choose to have L^c the rightmost literal of C' in the first step of this refutation. Thus in D a resolution with $\{L\}$ as far parent yields $C'-\{L^c\}$ as the second ordered clause of the deduction. We now alter D choosing (L) as the first ordered clause and C' as far parent which yields the same second clause. This is the desired TOCS refutation of S.

Case 2. There is no unit clause in S.

Let L denote the leftmost literal of C. Let S' denote the set derived from S by removing all clauses containing L^c and deleting L from all clauses of S containing L. If C' denotes C-{L} under the same ordering as C then C'$\in$ S'. We assume for now that C' is in a minimally unsatisfiable subset of S'. We know S' is unsatisfiable as otherwise S is satisfiable, so S' has a minimally unsatisfiable subset. As S' has less than n excess literals, there exists a TOCS refutation D' of S' from C'. Now add L as leftmost literal of each clause of D'. This is a TOCS deduction D of (L) from S. (Recall the left occurrence convention and the absence of clauses containing L^c in S'.) Let S* denote the set S with all clauses containing L removed and the unit clause {L} added. S* is an unsatisfiable set. Let S" denote a minimally unsatisfiable subset of S*. {L} $\in$ S" as S-{{L}} is satisfiable. By induction hypothesis there is a TOCS refutation D" of S" with first clause (L). We now append deduction D" to the end of D removing the first clause (L) of D" as this appears as the last clause of D. The result is a TOCS refutation of S from C as is easily checked. For example, tightness is preserved across the boundary of deductions D and D" as all clauses of D contain L and none of the clauses of D" except its first clause contain L.

We must now observe that C' is in a minimally unsatisfiable subset T of S' to remove this assumption. If C' is not in T then there exists a deduction D* (similar to D') of (L) from S without use of C. Note also that it is not used in refutation D" as C $\notin$ S". Then C is not in the unsatisfiable subset of S found by the new refutation of S formed by combining D* and D". This contradicts the minimality condition of S. The theorem is proved.

Remark: To make the above proof applicable to MTOCS ground refutation rather than TOCS ground refutation, it suffices to add the observation in case 2 that (L), the last ordered clause of deduction D, is a merge resolvent. This follows as the far parent of (L) must either occur earlier in the deduction or be a 2-literal clause from S. Either way the far parent must contain L. This observation is due to Anderson and Bledsoe. More detail appears in [1].

4. Proof of theorem 2. We show first that a strong ME ground deduction implies a corresponding unique TOCS ground deduction. The uniqueness is immediate by the nature of the mapping function f. The proof is by induction on the length k of the strong ME deduction.

Induction base, k=1. K_1 must be an elementary chain formed from a clause of S. $f(K_1)$ is the same string of literals so is a member of S and hence a TOCS deduction of $f(K_1)$.

Induction step, assume true for $k < n$ and show true for $k=n$, $n > 1$.

Case 1. K_n is obtained from K_{n-1} by c-extension. C-extension adjoins a chain K from S, with the leftmost literal missing, to the chain K_{n-1} to form K_n. Also the rightmost literal of K_{n-1} becomes an A-literal in K_n. Thus $f(K_n)$ is the resolvent of $f(K_{n-1})$ with f(K) if K is ordered to correspond to the ordering of the literals in $f(K_n)$ and the literal resolved upon in f(K) is the leftmost literal. Contraction does not influence B-literals so is not of concern should it occur at the end of the extension. The left occurrence conventions of both procedures assure that multiple occurrences of literals are handled similarly by each procedure.

Case 2. K_n is obtained from K_{n-1} by c-reduction. Let K_s denote the subchain of K_{n-1} beginning with the leftmost literal of K_{n-1} and including all literals through the A-literal involved in the c-reduction. The key observation is that given any initial subchain of K_{n-1} with an A-literal as rightmost literal (e.g., K_s) there exists a chain preceding K_{n-1} in the deduction exactly like the subchain except the rightmost literal is a B-literal. This follows by noting how A-literals are created and removed; nothing occurs to the left of that A-literal between creation and removal. Let K_a be the earlier chain corresponding to K_s in the manner described. Then $f(K_a)$ occurs in the deduction prior to $f(K_{n-1})$ by induction hypothesis, and is capable of resolution with $f(K_{n-1})$ under the s-linear resolution rules, and indeed must resolve with $f(K_{n-1})$ to satisfy the subsumption rule. The resolvent is clearly $f(K_n)$. Again contraction has no effect should it occur.

We do not need to consider the effect of factoring in ground deductions. We need to observe that no tautologies occur among the $f(K_i)$. This follows directly from the ME tautology rule.

It remains to demonstrate that the deduction $f(K_1), \ldots, f(K_k)$ is tight and also satisfies the subsumption rule.

Claim: it suffices to show for each chain K_i in the strong ME deduction that for all K_j, $j > i$, either K_j has as an A-literal some B-literal of K_i where the initial subchain of K_j ending in this A-literal is initial in K_i or else K_j is a proper initial subchain of K_i (i.e., an initial subchain minus at least the last B-literal). This means directly that $f(K_i)$ isn't contained in $f(K_j)$ so tightness holds. As regard the subsumption rule, for s-resolution to be possible using $f(K_i)$ and $f(K_j)$ when the B-literal L of K_i is also an A-literal of K_j, L must either be a B-literal of K_j in addition or have its complement a B-literal of K_j. This is by definition of s-resolution. The only permitted case for acceptable chains is the A-literal L followed by the complementary B-literal L^c which means that L^c is the last literal of K_j. Otherwise, s-resolution is impossible at this step. But this is the condition for c-reduction of K_j so $f(K_{j+1})$ is obtained by s-resolution and the subsumption rule is seen to hold. (Clearly K_j cannot be a proper initial subchain of K_i and have $f(K_i)$ and $f(K_j)$ s-resolve.) We shall have cause to refer to this last argument later.

We establish our claim by induction on the difference $j-i$. For $j = i+1$, if K_{i+1} arises through c-extension, K_{i+1} has the rightmost B-literal of K_i as an A-literal unless there is contraction, in which case K_{i+1} is a proper initial subchain of K_i. C-reduction gives K_{i+1} as a proper subchain of K_i. For $j - i > 1$, we assume the result for K_{j-1}. One has to check for each of the two possibilities for K_{j-1} deriving K_j by c-extension or c-reduction. Unless contraction occurs, the pertinent A-literal of K_{j-1} and all literals to its left survive. If the A-literal is removed by contraction then by hypothesis the remaining chain is a proper initial subchain of K_i.

We now show that a TOCS ground deduction implies a corresponding unique strong ME ground deduction. We first prove by induction on the length of a TOCS deduction that there exists such a strong ME deduction. For convenience, we delay the check that each chain defined is acceptable, i.e., a chain contains two literals with identical atoms only if the left literal of the pair is an A-literal and the right literal is a B-literal not the rightmost. These conditions are part of the induction hypothesis, however. Let $C_1, \ldots, C_n$ be the given TOCS deduction.

Induction base, k=1. C_1 by definition of a TOCS deduction must be a member of S. We choose K_1 to be a chain from the same clause under the same ordering. Then $f(K_1) = C_1$.

Induction step, assume true for $k < n$ and show true for $k = n$, $n > 1$.

Case 1. C_n is derived from C_{n-1} by resolution with clause $B \in S$. We define a chain K_n such that $f(K_n) = C_n$ by c-extension. The matrix chain B* used is clause B ordered so that the literal to be resolved upon is leftmost and the remaining order is determined by C_n. It follows immediately (modulo the check on literal conditions) that the c-extension of parent chain C_{n-1} by B* yields a K_n such that $f(K_n) = C_n$.

Case 2. C_n is derived from C_{n-1} by s-resolution with C_m, $m < n - 1$. In this case we know that C_n is simply C_{n-1} with the rightmost literal removed. Also, if we apply c-reduction to K_{n-1} we get a chain with the rightmost B-literal removed. We would be done if we knew that c-reduction is applicable to K_{n-1} whenever C_n is obtained by s-resolution with C_m, $m < n - 1$. The argument that this is the case has already been presented in the proof that the subsumption rule holds: see the proof of the converse implication above. The K_i of that argument is the K_m here and the K_j of that argument is K_{n-1} here. The claim used in the argument is valid for deduction $K_1, \ldots, K_{n-1}$ as this deduction is strong ME by induction hypothesis.

This completes the induction proof except for the check that the chains are acceptable.

We know that corresponding to TOCS deduction $C_1, \ldots, C_{n-1}, C_n$ there is a deduction $K_1, \ldots K_{n-1}, K_n$ where $K_1, \ldots, K_{n-1}$ is a strong ME deduction by induction hypothesis and K_n is defined from C_n as above. If by case 1 above identical B-literals do potentially occur in K_n by c-extension, the ME left occurrence convention allows for the removal of the right identical literal. By agreement this operation is considered delayed until the right identical literal is the rightmost literal.

This option holds for both strong ME and TOCS deductions or neither, so the correspondence holds. Complementary B-literals do not occur in K_n for otherwise C_n would be a tautology. K_n does not have a B-literal with an identical or complementary A-literal to its right for by the key observation of case 2 in the earlier proof of the converse result, an earlier chain would violate one of the two statements immediately above. An A-literal with a complementary B-literal to its right is permitted in K_n. K_n cannot have an A-literal with an identical B-literal to its right because the key observation then guarantees an earlier chain K exists such that f(K) subsumes K_n. This violates the tightness condition on the TOCS deduction. The case of K_n possessing identical A-literals is ruled out by use of the key observation to produce an earlier chain with a left A-literal and right B-literal case just shown impossible. Finally, K_n cannot have two complementary A-literals for there then is an earlier chain with a B-literal to the right of a complementary A-literal. In the converse proof it was shown such a chain corresponded to a clause C_k with C_{k+1} as a s-resolvent. This requires that K_{k+1} be obtained by c-reduction eliminating the B-literal from which the right A-literal arose. Thus all the literal conditions on K_n are checked.

We now consider uniqueness of the strong ME deduction given a TOCS deduction. Recall that we have seen that c-reduction is mandatory when possible (if the deduction contains a next chain) for a chain with complementary A-literals is created by c-extension. Using this fact and other similar facts, it is easy to establish by induction on the length of the TOCS deduction the following: if $K_1, \ldots, K_n$ and $K_1', \ldots, K_n'$ are strong ME deductions such that $f(K_i) = C_i = f(K_i')$ where $C_1, \ldots, C_n$ is a given TOCS deduction, then $K_i = K_i'$, $1 \le i \le n$. Details are left to the reader.

We point out that although the s-resolution operation within the TOCS deduction permits any literal of the far parent to be resolved upon, in fact only the rightmost literal need be resolved upon. This follows by considering the related strong ME deduction. We showed earlier that a c-reduction occurs in ME whenever a s-resolution occurs in the equivalent step of the corresponding TOCS deduction. Using the key observation again, let K be the chain earlier in the deduction having the A-literal L of the c-reduction as its rightmost B-literal (plus having the initial subchain property). Then f(K) is a clause which can s-resolve with the given near parent and the literal of K resolved upon, i.e. L, is the rightmost literal. Thus there is no gain (but a great cost) in searching non-rightmost literals for potential far parents.

5. The general form setting. In this section the general form (i.e. free variable form with substitution permitted) of the TOCS and strong ME procedures is considered. Theorem 2 of section 2 shows that there is a natural mapping from ground deductions of one procedure to the ground deductions of the other procedure. We see in this section that this fails for the general forms of the procedures. This occurs because the general form of a procedure does not always mirror the properties of the associated ground deductions. Indeed there are several ways to define a general procedure given a class of ground deductions. These points are illustrated through a series of examples in this section. The MTOCS and m-strong ME procedures are considered at the end of this section.

The technique for generalizing a ground resolution procedure, dubbed "the lifting lemma" in the literature, is given in [16]. We paraphrase the summary given in [17]: if clauses B and C have instances B' and C' with resolvent D' then there exists a resolvent D of B and C with instance D'. We use this to obtain the general form for the TOCS procedure. Given a set S of unsatisfiable clauses, we have a minimally unsatisfiable set S_{gr} of ground clauses by Herbrand's theorem. By theorem 1 there is a TOCS refutation D_{gr} of S_{gr}. Using the lifting lemma stated above, we develop a deduction D_1 of clauses of S such that D_{gr} is the image ground deduction of D_1 with r the identity function. Thus D_1 is a refutation. Is D_1 necessarily a TOCS refutation? No, for several reasons.

The first reason involves factoring. The lifting lemma is based on the key triple device which adds no steps simply to remove what is a redundant occurrence of a literal under the appropriate substitution. We therefore consider a deduction D derived from D_1 with the factoring added to conform with the requirements of an ordered clause refutation. As factoring is permitted only when it involves the rightmost literal, two non-rightmost literals in a clause of D may represent the same literal of the corresponding clause in D_{gr}. However, a D exists conforming to the factoring requirements of an ordered clause deduction such that D_{gr} is an image ground deduction of D with r no longer the identity function in general. This is easily checked with the possible exception of the restriction on the far parent when it is not a clause from S. The restriction came from our present definition of s-linear deduction. In that case no factor not in the deduction itself can be used for the far parent. However, in section 4 it is shown that when the far parent is an earlier clause of a TOCS ground deduction the literal resolved upon may be restricted to be the rightmost literal. This fact applies to D_{gr}. If C_{gr} denotes the "earlier clause" in D_{gr} and C is the corresponding clause to C_{gr} in D_1, and if the rightmost literal L_{gr} of C_{gr} corresponds to literal L of C, then any literals to the right of L may be "factored out" by adding appropriate clauses following C in defining deduction D. In D there then exists a clause which is itself an appropriate far parent at the later stage. We now note that these clauses were necessary for D anyway to allow the operation on C_{gr} which yields the following clause in D_{gr} to be imitated in D. Thus for deductions "lifted" from TOCS deductions the disallowance of nonexplicit factors of earlier clauses for far parents is no restriction whatsoever.

We use an <u>ad hoc</u> example to illustrate the above point. Let $S = \{Px, -Px\ Qx\ Ry\ Qy, -Rx\ Tx, -Tx\ -Rx, -Qx\}$ be the given unsatisfiable set of clauses. We give refutations D_{gr}, D_1 and D below. D_1 is not an ordered clause refutation but is a suitable "lifting" from D_{gr} using key-triple resolution. D is a TOCS refutation. Step 3 of D is needed to allow step 4 of D to correspond to step 3 of D_{gr}. It also supplies the correct factor as far parent to obtain step 6 in D. In D_{gr}, a is the constant added to build H(S).

D_{gr}: (5.1)

1. Pa ,
2. Qa Ra ,
3. Qa Ta ,
4. Qa -Ra ,
5. Qa ,
6. $\square$.

D_1: (5.2)

1. Px ,
2. Qx Ry Qy ,
3. Qx Ty Qy ,
4. Qx -Ry Qy ,
5. Qx Qy ,
6. $\square$.

D : (5.3)

1. Px ,
2. Qx Ry Qy ,
3. Qx Rx ,
4. Qx Tx ,
5. Qx -Rx ,
6. Qx ,
7. $\square$.

A second reason D_1 is not necessarily a TOCS refutation although D_{gr} is such is that the subsumption rule can be violated. This holds also for D. We illustrate this with another example. Let $S = \{Pa, -Pa\ Rx\ -Px, Pb, -Rb\}$. After D_{gr} is given, a corresponding TOC (tight ordered clause) refutation $D(=D_1)$ is given. No TOCS refutation with D_{gr} as an image ground refutation is possible, indeed a TOCS refutation exists beginning with Pa only if Rx is made rightmost literal at line 2, violating the free choice of order condition of ordered clause deductions.

$$
\begin{array}{llll}
D_{gr}: & 1. & Pa & , \\
 & 2. & Rb\ -Pb & , \\
 & 3. & Rb & , \\
 & 4. & \square & .
\end{array}
\tag{5.4}
$$

$$
\begin{array}{llll}
D\ : & 1. & Pa & , \\
 & 2. & Rx\ -Px & , \\
 & 3. & Rb & , \\
 & 4. & \square & .
\end{array}
\tag{5.5}
$$

The subsumption rule is violated in D as lines 1 and 2 are not resolved.

Another reason why D_1, and D, are not necessarily TOCS refutations although D_{gr} is such is that tightness also can be violated. We also illustrate this. Let $S = \{Pa, -Py\ Pf(x), -Pf(f(x))\}$. Note that S is a minimally unsatisfiable set of clauses. We give D_{gr} and D $(=D_1)$.

$$
\begin{array}{llll}
D_{gr}: & 1. & Pa & , \\
 & 2. & Pf(a) & , \\
 & 3. & Pf(f(a)) & , \\
 & 4. & \square & .
\end{array}
\tag{5.6}
$$

$$
\begin{array}{llll}
D\ : & 1. & Pa & , \\
 & 2. & Pf(x) & , \\
 & 3. & Pf(y) & , \\
 & 4. & \square & .
\end{array}
\tag{5.7}
$$

It should be noted that $S_{gr} = \{Pa, -Pa\ Pf(a), -Pf(a)\ Pf(f(a)), -Pf(f(a))\}$ used in D_{gr} is a minimally unsatisfiable set of clauses.

Clearly a (more natural) TOCS refutation exists, namely D': Pa, Pf(x), □. The point is that "bad" TOCS ground refutations exist. Incidentally, we conjecture that given a clause C in a minimally unsatisfiable set S of clauses, there is a TOC refutation of S with first clause C. However, we have just seen that not all TOCS (or TOC) ground refutations lift naturally to TOC refutations.

We have been considering lifting from ground deductions. We now consider mapping down to ground deductions. We show that not all TOCS refutations have image TOCS ground refutations, indeed, not even image s-linear ground deductions. This was mentioned in section 2. Let S = {Px Qx, Px -Qx, -Pf(x) Qx, -Px -Qx}. We give D first and then a non-image TOCS ground refutation D'. The reader can check that there is no image s-linear ground refutation associated with D. We will return to this example later.

D :	1. Px Qx ,		(5.8)
	2. Px ,		
	3. Qy ,		
	4. -Py ,		
	5. □ .		
D':	1. Pf(a) Qf(a) ,		(5.9)
	2. Pf(a) ,		
	3. Qa ,		
	4. -Pa ,		
	5. -Qa ,		
	6. □ .		

We observe from these examples that there is little direct tie between TOCS ground deductions and general TOCS deductions. It is not hard to show that TOCS ground deductions can always be lifted to ordered clause deductions. The prescription for doing this is the process of defining D from D_{gr} given earlier. This yields the following theorem. A formal proof is omitted.

Theorem 3. If C is a member of a minimally unsatisfiable set S of clauses then there exists an ordered clause refutation of S with first clause C.

Let us call a linear deduction system strongly complete if given an arbitrary clause C of a minimally unsatisfiable set S of clauses a refutation of

S with C as first clause exists; moreover, if the deduction system employs ordering, any ordering of literals permitted by the system must allow a refutation of S. To summarize, then, TOCS deduction is not strongly complete, TOC deduction may be strongly complete and ordered clause deduction is strongly complete. The examples of TOCS refutations (of the general form) given in this paper show that many TOCS refutations exist and some are shorter than any ordered clause refutation lifted from a TOCS ground refutation (e.g. the last example). The gain is realized when a clause is successfully used as a lemma; this is considered further in the next section.

The definition of a TOCS procedure can be modified slightly so that every TOCS ground refutation D_{gr} has a refutation, defined as D is above, within the class. We define the <u>weak TOCS deduction</u> to be as the TOCS deduction except the tightness condition and subsumption rule apply only propositionally, i.e., if no substitution is needed to meet the conditions. Here we no longer consider the same variable name as distinct in separate clauses of a deduction. The class of weak TOCS ground deductions is the class of TOCS ground deductions, but the general class of weak TOCS deductions is much broader. In particular, the deductions serving as counterexamples for tightness and the subsumption rule given in this section are weak TOCS deductions. This fits our assertion above that if D_{gr} is a TOCS ground refutation and D is defined from D_{gr} as described above, then D is a weak TOCS refutation. The proof of this is left to the reader. A consequence of the statement is the following theorem.

<u>Theorem 4.</u> The weak TOCS procedure is strongly complete.

Of course, it still holds that not every weak TOCS refutation has an image s-linear ground refutation. We will see also that as for the TOCS procedure, it is not isomorphic to strong ME in the general form.

Like weak TOCS deductions, strong ME ground deductions lift in a natural way to strong ME deductions in the general form in all cases. As a consequence the following theorem holds.

Theorem 5. Strong ME is strongly complete.

Again, we do not give a formal proof of this theorem but do consider the lifting process and give an example.

Lemma 2 of [8] replaces the lifting lemma as the underlying justification for associating certain deductions and ground deductions. In essence, lemma 2 states: given a standard ME ground deduction using ground instances of chains from a given set S of clauses, by performing the same operations in the same order using the corresponding chains from S, one obtains a standard ME deduction of a chain having as an instance the last chain of the ground deduction. This is what we mean by saying a standard ME ground deduction lifts to a standard ME deduction. Clearly the ground deduction is an image ground deduction in this case. As strong ME is standard ME with c-factoring, the ME left occurrence convention and the tautology rule added, these must be checked to assert that strong ME ground deductions lift in the natural way. (The alteration induced by using c-extension and c-reduction here rather than basic extension, reduction and contraction as in [8] produces no problem in adapting lemma 2 of [8]). Such a check reveals that no problem is encountered in lifting strong ME.

Given a strong ME ground deduction D_{gr} one obtains a strong ME deduction D with D_{gr} as an image ground deduction as follows. Imitate the development of D_{gr} as stated in the lemma 2 paraphrase unless an ME left occurrence convention application has occurred before or at the corresponding step in D_{gr} which has removed the occurrence of the present rightmost literal in its ground instance. Then apply the c-factoring operation to the rightmost literal and the appropriate B-literal to its left. This suffices to determine D. To illustrate this we use a set S considered earlier in this section, i.e. S = {Px, -Px Qx Ry Qy, -Rx Tx, -Tx -Rx, -Qx}. We give D_{gr} then D. Note step 3 of D is obtained by c-factoring.

$$
D_{gr}: \quad
\begin{array}{ll}
1. & Pa \quad , \\
2. & \underline{Pa}\ Qa\ Ra \quad , \\
3. & \underline{Pa}\ Qa\ \underline{Ra}\ Ta \quad , \\
4. & \underline{Pa}\ Qa\ \underline{Ra}\ \underline{Ta}\ {-Ra} \quad , \\
5. & \underline{Pa}\ Qa \quad , \\
6. & \emptyset \quad .
\end{array}
\qquad (5.10)
$$

$$
D: \quad
\begin{array}{ll}
1. & Px \quad , \\
2. & \underline{Px}\ Qx\ Ry\ Qy \quad , \\
3. & \underline{Px}\ Qx\ Rx \quad , \\
4. & \underline{Px}\ Qx\ \underline{Rx}\ Tx \quad , \\
5. & \underline{Px}\ Qx\ \underline{Rx}\ \underline{Tx}\ {-Rx} \quad , \\
6. & \underline{Px}\ Qx \quad , \\
7. & \emptyset \quad .
\end{array}
\qquad (5.11)
$$

It might be enlightening to the reader to develop the strong ME refutations of the examples which illustrated the violation of tightness and the subsumption rule in the general form. Considering these examples, one sees that the strong ME procedure behaves like the weak TOCS procedure. However, the fourth example with S= {Px Qx, Px -Qx, -Pf(x) Qx, -Px -Qx} which produced a TOCS (hence weak TOCS) refutation with no image s-linear ground refutation demonstrates that the weak TOCS and strong ME procedures are not isomorphic. The TOCS refutation of S given previously, see (5.8), has no counterpart strong ME refutation. The strong ME refutation D starting with Px Qx, which we give below, does have a weak TOCS refutation counterpart which the reader may find. D has an image strong ME ground refutation from which a TOCS ground refutation is immediately obtained. We omit the ground refutation. Notice that step 2 involves the left occurrence convention and c-factoring is not used.

$$
D: \quad
\begin{array}{ll}
1. & Px\ Qx \quad , \\
2. & Px \quad , \\
3. & \underline{Pf(x)}\ Qx \quad , \\
4. & \underline{Pf(x)}\ \underline{Qx}\ {-Px} \quad , \\
5. & \underline{Pf(x)}\ \underline{Qx}\ \underline{Px}\ {-Qx} \quad , \\
6. & \emptyset \quad .
\end{array}
\qquad (5.12)
$$

In the next section where explicit "lemma" devices are considered, we see that there exists a strong ME refutation with lemmas that corresponds to the TOCS refutation (5.8).

Finally, a strong ME deduction D is given which has no image ground deduction. Let S = {Qx Pxy, -Pxy -Pyx, Pxy Rx}. Note that H(S) = {a}.

D : 1. Qx Pxy , (5.13)
2. Qx $\underline{\text{Pxy}}$ -Pyx ,
3. Qx $\underline{\text{Pxy}}$ -$\underline{\text{Pyx}}$ Rx .

Remark. We give an example to illustrate the MTOCS and m-strong ME procedures and to aid understanding of the relationship of these procedures to the TOCS and strong ME procedures. Again, let S = {Px Qx, Px -Qx, -Pf(x) Qx, -Px -Qx}. Refutations D_1 and D_2 are the MTOCS and m-strong ME ground refutations respectively. A pair of brackets around a literal denotes a merge literal in MTOCS procedures and an Am-literal or a Bm-literal in m-strong ME procedures. If D_1 is compared to (5.9) it is seen that the restriction on earlier clause resolution has cost an extra step. However, the general MTOCS refutation D_3 below is the same as (5.8) and has no image MTOCS ground refutation. The m-strong ME refutation is structurally the same as its ground image D_2 and is not presented. The modified m-strong ME refutation is structurally like (5.12) as its ground image so is not presented. The modified m-strong ME refutation is an actual saving over the m-strong ME refutation even at the general level here because in each clause every literal shares the same variables. This "ties" the clauses together upon c-extension as regards free variables. Such deductions behave as ground deductions. For this reason, the modified m-strong ME procedure seems the actual procedure one would wish to program if the merge condition is considered a desirable addition to s-linear procedures. After the refutations are presented the merits and costs of adding the merge strategy are considered briefly.

An MTOCS ground refutation of S.

D_1: 1. Pf(a) Qf(a) , (5.14)
2. [Pf(a)] ,
3. Qa ,
4. -Pa ,
5. -Qa ,
6. -Pf(a) ,
7. □ .

An m-strong ME ground refutation of S.

D_2: 1. Pf(a) Qf(a) , (5.15)
2. [Pf(a)] ,
3. [$\underline{\text{Pf(a)}}$] Qa ,
4. [$\underline{\text{Pf(a)}}$] $\underline{\text{Qa}}$ -Pa ,
5. [$\underline{\text{Pf(a)}}$] $\underline{\text{Qa}}$ -$\underline{\text{Pa}}$ -Qa ,
6. [$\underline{\text{Pf(a)}}$] $\underline{\text{Qa}}$ -$\underline{\text{Pa}}$ -$\underline{\text{Qa}}$ -Pf(a) ,
7. $\emptyset$.

An MTOCS refutation of S.

D_3:	1.	Px Qx	,	(5.16)
	2.	[Px]	,	
	3.	Qx	,	
	4.	-Px	,	
	5.	$\square$	.	

The disadvantage of adopting the modified m-strong ME procedure, for example, over the strong ME procedure is the possibility of lengthening a refutation and, most likely, the search effort . This has been illustrated. (The added programming, etc., is also a disadvantage.) Much harder to illustrate in this setting is the advantage of attempting fewer c-reductions in the general form. Experience with a standard ME procedure reveals that on problems with clauses whose literals share few variables, a frequent occurrence in mathematical problems, c-reduction is tried many times when it is not an appropriate action. This occurs because other substitutions than the appropriate one are possible. Reducing the number of c-reductions should cut down on the proof search by restricting this type of error.

6. Lemmas. For the purpose of this paper, a lemma in deduction D is a clause obtainable during the development of D which is used, or retained for possible use, later in the deduction to shorten the proof or proof search but which is not needed to satisfy the completeness conditions of the procedure being used. Also each lemma must be satisfied by any model of the given set S of clauses. Usually a lemma is a far parent not normally allowed under the restrictions on far parents. However, we formally regard lemmas as previously unrecognized members of S (so S changes in practice) so that the previous definitions are still satisfied.

There are two parts to consider concerning lemmas, their generation and their selection and use. In this section we say something new about their generation; their selection and use receives little attention here. Some comments will be made concerning use based on the author's experience with an implementation of standard ME using lemmas.

The lemmas of the TOC(S) or weak TOCS procedures are the deduction clauses themselves. Because resolution is a valid inference rule, any direct or indirect resolvent of members of S is satisfied by any model of S. We restrict our attention to a narrow subcase, a lemma use built into the (weak) TOCS procedure. This case is well illustrated by deduction (5.8) of section 5. We consider this a lemma situation for shortening the proof even though it falls within the (weak) TOCS deduction rules because the two uses of (the clause of) line 2, to obtain line 3 and line 5, employ different substitution instances. This use of line 2 with line 4 to obtain line 5 is not needed for completeness; it actually occurs in such a manner as to permit no image ground deduction as noted in section 5. One can regard (5.8) as a lifting of deduction (5.9) to the general form except that at line 5 one deviates to maintain a valid TOCS deduction rather than arbitrarily following the lifting procedure. But lifting the isomorphic strong ME deduction to (5.9) results in deduction (5.12) where the "lemma device" is excluded from use by the format. This "gain" at line 5 of (5.8) we attribute to a use of a lemma.

The importance of this subcase is that this advantage of the TOCS procedure can be carried over to strong ME. Although the isomorphism theorem doesn't hold at the general level, any strong ME deduction is mapped by the correspondence f with $f(K) = C$, where K is a chain and C is an ordered clause consisting of the B-literals of K with the ordering of K, to a weak TOCS deduction. We state in theorem 6 below only what we need here, that if there is a strong ME deduction of chain K, there is a resolution deduction of f(K) as an unordered clause. This means that any chain K in a strong ME deduction can be used to define a clause f(K) which may be added to the given set S for use by the c-extension operation later in the deduction. When such is done within the strong ME setting we call the procedure the <u>strong ME procedure with lemmas</u>.

We state theorem 6. The proof is discussed in section 7. (Section 7 should be read before undertaking implementation of lemmas.) Clause $f(K_n)$ below is viewed as an unordered clause.

<u>Theorem 6.</u> If D is a strong ME deduction $K_1, \ldots, K_n$ of K_n from S, then there exists a deduction D' of $f(K_n)$ from S by resolution.

In illustration of the strong ME procedure with lemmas we note that deduction (5.12) can be shortened a line by using (the B-literal of) line 2 as the elementary chain with c-extension at line 4. Line 5 is then the empty chain. This, of course, parallels (5.8).

Nearly unrestricted use of B-literal clauses amounts to imbedding a general resolution procedure in the strong ME procedure, which defeats the purpose of carefully restricting the deductions allowed. Some comments on selection of lemmas will be made later.

The standard ME procedure has provision for creating lemmas. This is discussed in detail in [8] and summarized with an example of lemma use in [9]. In the ME procedures lemmas come only from certain lines of the deduction and some processing is required to obtain each lemma. The lemmas are generally fewer in number than for the related (weak) TOCS procedure and often distinct from those of the TOCS procedure.

The modifications to standard ME that produces strong ME slightly alter the method of lemma production. The lemma mechanism for the strong ME procedure is presented here by describing the necessary alterations to the operations of strong ME. With each literal is associated a non-negative integer called the scope of the literal which is zero unless explicitly changed.

C-extension. Each literal of the new chain has the scope of its parent in the preceding chain. Literals from the appended elementary chain have scope 0. If contraction occurs, a lemma is formed for each A-literal removed proceeding from the rightmost A-literal. The lemma consists of the complement of the A-literal removed, the complement of any A-literal L whose scope exceeds the number of A-literals strictly between L and the A-literal removed, and any B-literal L' whose scope also exceeds the number of A-literals strictly between L' and the A-literal removed. At the end of a contraction any literal L whose scope is larger than the number n(L) of A-literals to the right of that literal is reduced to n(L).

C-reduction (c-factorization). The A-literal (B-literal) L which complements (is identical to) the B-literal L' that is removed has its scope increased if necessary to the number of A-literals strictly between L and L'. All other scopes in the new chain are those of the parent literal of the previous chain. Contraction is handled as for c-extension.

The left occurrence convention which removes B-literals is handled exactly as for c-factorization.

For an example of the use of lemmas in an ME deduction see [9]. A use of lemma also occurs in some refutations of the set S = {Px Qx, Px -Qx, -Pf(x) Qx, -Px -Qx} beginning with -Pf(x) Qx. A refutation D simply to illustrate creation of lemmas where c-factorization occurs is given here. Let S = {Qy Ry, -Rx Sa, -Sa Qb -Rx, -Qb}. The scope, when non-zero, is placed in brackets preceding the predicate letter.

D:			
	1.	Qy Ry	
	2.	Qy <u>Ry</u> Sa	
	3.	Qy <u>Ry</u> <u>Sa</u> Qb -Rx	
	4.	Qy [1] <u>Ry</u> <u>Sa</u> Qb	
	5.	a. [1] Qb [1] <u>Rb</u> <u>Sa</u>	
		b. Qb	Lemmas formed: -Sa -Rb Qb, -Rb Qb
	6.	∅	Lemma formed: -Qb

Two of the three lemmas are instances of members of S and thus certainly uninteresting. By not retaining the first lemma produced after a c-extension many of the trivial lemmas would be suppressed.

A strong ME procedure which includes the above mechanism for producing lemmas for use, under control, with c-extension is called a <u>strong ME procedure with lemmas</u>. This label is used also for a strong ME procedure using this lemma device in conjunction with the B-literal lemmas previously discussed.

The critical problem in a strong ME procedure with lemmas is the intelligent selection of lemmas for use. Experience with an implementation of a standard ME procedure with lemmas has lead the author to the following tentative conclusion: although lemma use has sped up the proof search or given a shorter proof in some instances, a search without lemma use often will be more efficient. This seems true even if only unit clause lemmas are retained, although this is certainly a good first rule. The reason lemmas hurt more than help is that these procedures presently have no contextual control over the lemma use so even though some refutations are shorter the number of deductions of the size of the shortest refutation has increased greatly. As problems get bigger, lemmas may be necessary, but they may be best selected by humans who survey the list of lemmas produced to date and select one or two lemmas that look particularly appropriate. These lemmas are then added to S (with perhaps additional constraints) for further attempts at the refutation.

7. Proof of theorem 6. The proof of theorem 6 is considered here simply because it is dependent on a careful formulation of the left occurrence conventions option when adopted as it is in this paper. The theorem is immediate at the ground level by the isomorphism theorem, of course. As we have seen, this does not immediately guarantee the result for the general setting. The potential trouble occurs when substitutions in elementary chains used for c-extension identify two previously unidentified literals. As stated for ME, the ME left occurrence convention option works as follows when c-extension using chains K and K' is contemplated: assuming the rightmost B-literal of K complements the leftmost literal of K' after suitable substitution θ, we remove the rightmost literal of $K\theta$ if it agrees with a B-literal to its left (in which case the c-extension will most likely be inapplicable) and remove all literals of $K'\theta$ which agree with a literal to its left in $K'\theta$. If still possible, the remainder of the c-extension operation is then performed. In particular, chain Pa when c-extended by -Px -Pa Qx yields Pa Qa, not Pa -Pa Qa.

Notice that f(Pa Qa) = Qa while f(Pa -Pa Qa) = -Pa Qa. The latter chain does not yield the former under the left occurrence convention. If only the rightmost literal of the elementary chain were checked for redundancy during c-extension, theorem 6 would have to be amended to assert that $f(K_n)$ is subsumed by some clause obtainable by resolution.

The proof of theorem 6 follows the appropriate portion of the proof in section 4 that each strong ME deduction has a TOCS deduction counterpart. The appropriate instantiations are seen to be obtained and the discussion above shows that the ME left occurrence convention option handles satisfactorily the removal of superfluous literals after instantiation.

8. Other procedures. In this section the model elimination procedure is interpreted in the linked conjunct procedure of Davis [4] and the matrix reduction procedure of Prawitz [13], [14].

We first consider the linked conjunct procedure. A set S of clauses is a linked conjunct if each literal in each clause of S is the complement of another literal from another clause of S. The linked conjunct procedure applied to a given set S of clauses consists of a systematic search for a set of substitution instances of clauses of S which are linked conjuncts. Each linked conjunct is tested to see if it is a contradictory set by use of the Davis-Putnam procedure [4], [5] which provides a fast test of truth-functional unsatisfiability for finite sets of clauses. This testing procedure does not concern us as ME can be viewed as a way of restricting the scope of the linked conjunct check and also dispensing with the final test. That is, each linked conjunct found using the ME procedure will be known to be a contradictory set.

The linked conjunct given by ME is formed as follows. A clause is chosen from given set S and its rightmost literal (under some ordering) is made complementary, i.e. matched, to some other literal of a clause of S via a suitable substitution in each clause (c-extension). The new clause minus matching literal is attached to the right end of the existing chain as is usual for c-extension. This operation may be iterated. A linking of a rightmost literal with one of a certain subset of the literals of already selected clauses, i.e. the A-literals, is also permitted (c-reduction). If the linking of complementary pairs is done in the manner dictated by the ME procedure then a linked conjunct is obtained precisely when the empty chain is reached. Of course, to explicitly obtain the clauses of the linked conjunct some bookkeeping is necessary. The clause instances of S that are used must be recorded and then modified because new substitutions, those needed for matching two literals, influence through shared variables other clauses already selected. It is known from the completeness of ME that the linked conjunct so obtained is contradictory. In contrast, resolution does not provide such a direct scheme for finding linked conjuncts for it is not always possible to find a linked conjunct for an unsatisfiable set S by matching a literal "at hand" with a literal of a new clause instance of S (the resolution operation) or with an eligible existing literal (factoring). This is simply the statement that linear resolution is not complete when the far parent is always from the given set S. Indeed, one could say that just the literals already present to which one wants to restrict the linking, i.e. the A-literals, are the ones discarded by the resolution operation.

For illustration, let S = {PQ, -PQ, P-Q, -P-Q}. The following list of three clauses of S forms a linked conjunct as seen immediately by viewing the columns.

```
 P  Q
 P -Q
-P -Q
```

This is not a contradictory set, however, so it is clear an auxiliary test is necessary to differentiate this set of clauses from the set S which is contradictory. The reader is encouraged to write out an ME refutation of S with an extra step for each contraction and an auxiliary list of the clauses indicating the matching positions. One sees that one ME refutation suggests the following listing of clauses where columns again indicate the linking intended.

```
 P  Q
 P -Q
-P     -Q
-P      Q
```

(The author was first made aware that a relationship between ME and the linked conjunct procedure exists by J. A. Robinson.)

The matrix reduction procedure of Prawitz was proposed about five years after the linked conjunct procedure so it is not surprising that it is a more sophisticated procedure at least in terms of the amount of mechanism in the ME procedure used in the interpretation. This does not necessarily mean it is a more efficient procedure but should have the potential to be stronger. Prawitz in [14] presents a case for a possible increase in efficiency for matrix reduction over resolution based on the retention of more information in the end product of a basic operation. The quite natural interpretation of (one form of) ME within matrix reduction suggests this is correct if the comparison is against basic resolution. Prawitz also makes the statement in [14], pp. 212-3, that many resolution strategies are probably capable of being superimposed upon matrix reduction also. This suggests that if the matrix reduction procedure essentially as presented is superior to basic resolution, this advantage might be held over any resolution strategy by some similar improvement to matrix reduction. We might call this the "one step ahead" conjecture. However, the presentation below allows one to view a natural treatment of matrix reduction as a linear resolution strategy. This means the resolution strategies may be able to mimic the matrix reduction strategies of importance, making the "one step ahead" hopes for matrix reduction quite improbable.

We first outline the matrix reduction procedure and then present an example of its execution on an unsatisfiable set. As before, we practically ignore the aspect of finding appropriate substitutions to make literals complementary which exists in both matrix reduction and ME. The example is used to demonstrate the relation between (the particular implementation of) matrix reduction and ME. A summary of the structure of the interpretation follows.

Let S be a set of clauses. By a <u>variant</u> of clause C we mean a clause C' defined from C by replacing the set of variables of C by a disjoint set; we will assume that the new variables have not occurred anywhere within the setting of concern. The basic <u>reduction</u> <u>operation</u> applied to a set M of clauses is performed as follows.

> First choose a clause C of M and a literal L of C, then choose either a variant C_1 of a clause of S or a clause C_1 from M and then choose a literal L_1 of C_1. Form sets M_L, the left set, and M_R, the right set, as below if the literals L and L_1 can be made to match. If a match is possible, perform the required substitutions throughout a copy of M, with C_1 attached if it is not in M. Call the instantiated M set M' and the associated named entities C',L', C_1' and L_1'. M_L is M' except that C' is replaced by $\{L'\}$ and C_1' is replaced by $C_1' - \{L_1'\}$; if $C_1' - \{L_1'\}$ is empty then M_L is the empty set, marked "void". M_R is M' except that C' is replaced by $C' - \{L'\}$ and C_1' is replaced by $\{L_1'\}$; if $C' - \{L'\}$ is empty then M_R is the empty set, marked "void."

A set M which has some reduction operation which makes M_L and M_R void is called pre-void. A set M which has some reduction operation which makes M_L and M_R pre-void or void is also pre-void. The procedure consists of choosing a clause C of S and setting $M_0 = \{C\}$ then applying the reduction operation to M_0, then to the resultant sets $M_{0,L}$,$M_{0,R}$ etc. until M_0 is demonstrated to be pre-void or the procedure is instructed to halt. If M_0 is shown to be pre-void then S is unsatisfiable. One way to instruct the procedure to halt is to limit the number of variants of each clause that can be used. For example, we can allow first one variant per clause of S, then two variants per clause, etc. each time increasing the variant count by one (or n) when the procedure halts until successful or bored. This is the method mentioned by Prawitz [14].

Figure 1 contains a refuation of the set S given by

1. PQR ,
2. PQ-R ,
3. P-Q ,
4. -P-R ,
5. -PR .

The initial set M_0 is {1.}. For each M, M_L and M_R appear below M and are connected by lines to M. M_L appears to the left of M_R. For each set M, represented as a list of clauses, the number to the left of a clause designates the source clause in S from which the clause originated. Any clause derived from a clause of S introduced as C_1 by the reduction operation applied to M appears in the last line of the listing in M_L and M_R. The brackets and parentheses as well as the parenthesized number "(n)" beside a set are for aid in interpreting the ME refutation below in terms of the matrix reduction refutation. The clause **C** and literal L of C chosen for the reduction operation applied to M is the clause not enclosed by parentheses or brackets and the rightmost literal of that clause respectively.

We now give an ME refutation for the set S just listed. The form of ME we use is the standard ME procedure defined in section 2 with the addition of lemmas. This is the procedure of [8] but with c-extension and c-reduction rather than extension, reduction and contraction as operations. For clarity we explicitly demonstrate the contraction by presenting certain lines of the deduction in two parts. Only lemmas that differ from clauses of S or previous lemmas are recorded. The scope of an A-literal is placed in brackets just preceding the literal if the scope is non-zero.

1.	PQR		clause 1 ,
2.	PQRPQ		c-extension using clause 2 ,
3.	PQRPQP		" " " 3 ,
4.	PQRPQP-R		" " " 4 ,
5. a.	PQ[1]RPQP		c-reduction
b.	PQRP		
	new lemma formed:	-Q-R	(i) ,
6.	PQRP-R		c-extension using clause 4 ,
7. a.	PQ [1] RP		c-reduction
b.	PQ		
	new lemma formed:	-R	(ii) ,
8.	PQP		c-extension using clause 3 ,
9.	PQPR		" " " 5 ,
10. a.	PQPR		" " lemma (ii),
b.	P		
	new lemma formed:	-P	(iii)
		-Q	(iv) ,
11.	PR		c-extension using clause 5 ,
12. a.	PR		" " lemma (ii),
b.	$\emptyset$		.

The lines of this deduction correspond to sets (lists) in the matrix reduction procedure indicated by the number in parentheses beside the set (Fig. 1). The translation rule is given from a line in the ME refutation to the corresponding set in the matrix reduction refutation. The rule is that the A-literals map to one-literal entries enclosed in brackets and the B-literal string following the last A-literal is on a following line. The set may contain other one-literal lines (those in parentheses); for each of these there is a lemma formed at the corresponding line or earlier only one literal of which is not complementary to an A-literal of the chain (in the general case only after a suitable matching procedure is undertaken). The one literal is that listed in parentheses.

This example illustrates several points but not all possibilities. There is not always a 1-1 correspondence between matrix reduction sets and lines in the ME deduction when the matrix reduction is performed as indicated above and summarized below. We now consider briefly why the 1-1 correspondence exists in the example and where it is possible to deviate from this correspondence. We restrict our attention to the ground level case.

The matrix reduction refutations (we shall restrict our attention to such deductions) are in the form of a binary tree with the initial set M_0 as root. It is easily proven by induction that any non-void set in the tree, where a set represents a node, has precisely one clause not enclosed by brackets or parentheses. We call such a clause free. The definition of M_L and M_R from a set M in the form (strategy) of matrix reduction studied here can be summarized as follows using the notation of the reduction operation. The rightmost literal of the free clause C of M is chosen as L. First we consider C_1 as a new variant of S. M_L is a copy of M except "[L]" is entered in the line for C and $C_1 - \{L_1\}$ is a new last line and is the free clause. M_R is a copy of M except $C - \{L\}$ is the new free clause entered in the line for C and "(L_1)" appears as a new last line. If C_1 is a clause already in M, it must be a unit (one-literal) clause enclosed either in brackets or parentheses. In this case M_L is always void and M_R is like M except the line for C is replaced by $C - \{L\}$ as the free clause. Some of the above mentioned sets may be void, of course. This summary can be verified in Fig. 1 by the reader.

We now consider the nature of the corresponding ME refutation. From any node on the matrix reduction refutation tree a subtree is defined with the node as root. The leftmost branch from the root is of primary importance; we really only look at leftmost branches of various subtrees. In Fig. 1 the leftmost branch for set (1), which is the initial set M_0 here, consists of sets (2), (3), (4) and the leftmost void set. Proceeding down a leftmost branch corresponds to c-extension except perhaps when the void set is produced in the final step. Except in the final step if one regards the bracketed literals as A-literals and the free clause (which always follows the bracketed literals) as the B-literal string to the right of the last A-literal in the chain, then passing from an M to a non-void M_L is indeed the c-extension operation. Here literals in parentheses are always ignored. In the final step, c-extension with a new unit clause from S is one possibility. Another possibility is c-reduction which occurs when the set M with the void M_L is reduced by choosing C_1 as a bracketed literal of M. In illustration, note set (4) of Fig. 1 is reduced in this manner, and in the corresponding ME refutation line 5 is

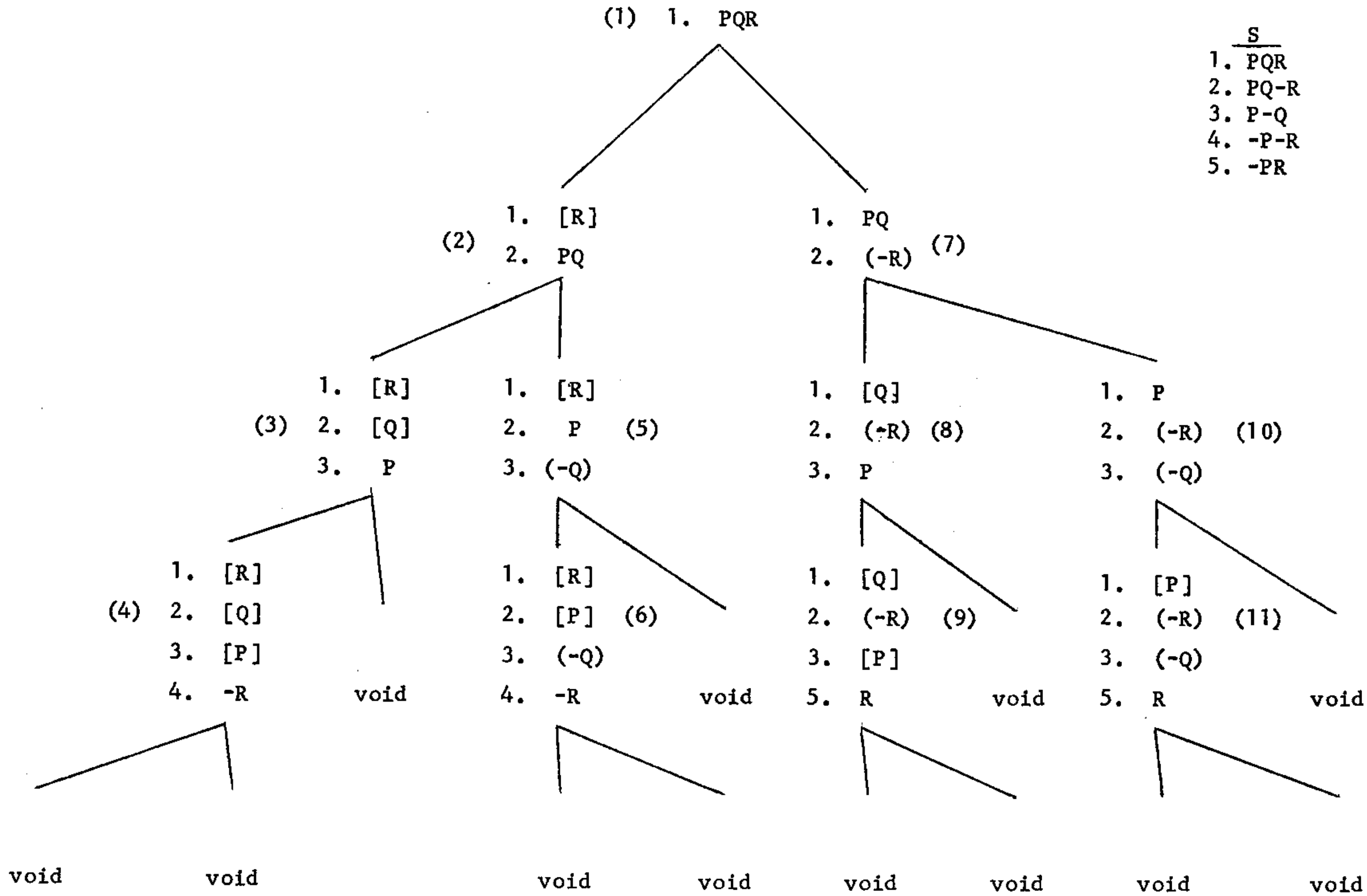

Figure 1

A MATRIX REDUCTION REFUTATION TREE

obtained from line 4 by c-reduction. A third possibility is the selection of a parenthesized literal for C_1 as in the reduction of set (9) in Fig. 1. The corresponding operation in ME is c-extension by a lemma one of whose literals agrees with the parenthesized literal for C_1 and the other literals are each a complement of an A-literal of the chain. Each such A-literal also appears as a bracketed literal of M. Such a lemma can be shown always to exist by the stage it is needed due to the order of processing. The fourth possibility of two bracketed literals being complementary can be ruled out as this means the third possibility was possible earlier in analogy with the inadmissibility of chains with complementary A-literals. The other possibilities arise first for sets of type M_R so are considered in that setting.

As is apparent from Fig. 1, the order of scan of a (sub)tree is that of processing the leftmost branch and then backing up and scanning each subtree one encounters as one backs up the branch. By "encountering a subtree" we mean selecting a node which is the M_R set for a set M at some node on the branch. This selected node is the root of a subtree not yet scanned. It suffices to consider the possibilities of such roots of subtrees for all other nodes mark M_L type sets and have been considered.

Let us label the set of concern as M and the set from which it is derived as M'. Thus M'_R is M. By the convention adopted for this form of matrix reduction, the reduction of M calls for choosing the free clause for C. Note that C still corresponds to the string of B-literals following the A-literals of the ME chain. For illustration see sets (5), (7) and (10) of Fig. 1 and chains 5, 7 and 10 in the ME refutation. This can be a poor strategy on occasion, however, especially when two unit clauses in M complement each other but neither is the free clause. As this leads directly to M_L and M_R void, we wish to consider modifying the given matrix reduction strategy to exploit this and ask if there is a counterpart in ME. In fact there is a counterpart. If two parenthesized literals are complementary, then in the corresponding ME deduction there exists two lemmas at this point, each containing one of the literals in parentheses and otherwise containing literals complementary to the A-literals of the chain K corresponding to M. If these lemmas are resolved together a clause of complements of A-literals of K is obtained. It is then possible to use this clause as a lemma to remove a rightmost B-literal earlier in the ME deduction. This is one place where the 1-1 correspondence may break down for the ME deduction may circumvent a whole subdeduction in this manner whereas the matrix reduction need need not back up to a corresponding position. This analogy does call for utilizing resolution of lemmas as mentioned in [9] but not mentioned in the present paper until now. There is also a possibility of a parenthesized literal opposing a bracketed literal in M. In ME this corresponds to a lemma all of whose literals complement A-literals of the corresponding chain K and allows the deduction to be shortened in the same manner as mentioned above. One other new possibility exists, that is when a parenthesized literal complements a literal of the free clause. If it complements the rightmost literal then this is a case previously discussed. If it complements a literal not rightmost it is reasonable to delay reduction based on that complementary pair until the literal is rightmost, similar to only working on rightmost literals in ME. No work is saved by earlier processing, just a reordering. All other possibilities for M do not involve a parenthesized literal and are similar to those given earlier.

This completes the demonstration that ME can be interpreted in matrix reduction. The form of matrix reduction imposed is natural, the only limitation being the requirement that the rightmost literal of the free clause be one of the literals made complementary. This choice was not required if the two derived sets were void under some other reduction. The matrix reduction strategy studied actually should be somewhat more complex than recorded as some possibilities allow inadmissible chains in the corresponding ME refutation which is unacceptable in a translation of ME to matrix reduction. We have omitted most such considerations for simplicity but point out that this should help the matrix reduction strategy by showing search paths which do not need to be followed.

Again, we remark that the consideration of the general form is easily superimposed on that presented here.

If one is interested in studying the relationship of matrix reduction and linear resolution in detail, the linear resolution considered must be modified to account for the difference between a strong ME deduction format and the standard ME deduction format used here.

References

[1] Anderson, R. and Bledsoe, W. W. A linear format for resolution with merging and a new technique for establishing completeness. J.ACM 17 (July 1970), 525-34.

[2] Andrews, P. Resolution with merging. J.ACM 15 (July 1968), 367-81.

[3] Chang, C. L. The unit proof and the input proof in theorem proving. J.ACM 17 (October 1970), 698-707.

[4] Davis, M. Eliminating the irrelevant from mechanical proofs. Proceedings of Symposia in Applied Mathematics XV, pp. 15-30, Providence: American Math. Soc., 1963.

[5] Davis, M. and Putnam, H. A computing procedure for quantification theory. J.ACM 7 (July 1960), 201-15.

[6] Hayes, P. and Kowalski, R. Semantic trees in automatic theorem-proving. In (eds. B. Meltzer and D. Michie) Machine Intelligence 4, pp. 87-102, Edinburgh: Edinburgh University Press, 1969.

[7] Loveland, D. W. Mechanical theorem proving by model elimination. J.ACM 15 (April 1968), 236-51.

[8] Loveland, D. W. A simplified format for the model elimination theorem-proving procedure. J.ACM 16 (July 1969), 349-63.

[9] Loveland, D. W. Theorem provers combining model elimination and resolution. In (eds. B. Meltzer and D. Michie) Machine Intelligence 4, pp. 73-86, Edinburgh: Edinburgh University Press, 1969.

[10] Loveland, D. W. A linear format for resolution. Lecture Notes in Mathematics 125 (Symposium on Automatic Demonstration), pp. 147-62, Berlin: Springer-Verlag, 1970.

[11] Luckham, D. Refinement theorems in resolution theory. Lecture Notes in Mathematics 125 (Symposium on Automatic Demonstration), pp. 163-90, Berlin: Springer-Verlag, 1970.

[12] Newell, A. and Simon, H. GPS, a program that simulates human thought. Lernonde Automater, Munich, 1961.Reproduced in (eds. E. Feigenbaum and J. Feldman) Computers and Thought, pp. 279-96, New York: McGraw-Hill, 1963.

[13] Prawitz, D. Advances and problems in mechanical proof procedures. In (eds. B. Meltzer and D. Michie) Machine Intelligence 4, pp. 59-71, Edinburgh: Edinburgh University Press, 1969.

[14] Prawitz, D. A proof procedure with matrix reduction. Lecture Notes in Mathematics 125 (Symposium on Automatic Demonstration) pp. 207-14, Berlin: Springer-Verlag, 1970.

[15] Raphael, B. Some results about proof by resolution. SIGART Newsletter 14 (February 1969).

[16] Robinson, J. A. A machine-oriented logic based on the resolution principle. J.ACM 12 (January 1965), 23-41.

[17] Robinson, J. A. Automatic deduction with hyper-resolution. International Journal of Computer Mathematics 1 (July 1965), 227-34.

[18] Slagle, J. Automatic theorem proving with renamable and semantic resolution. J.ACM 14 (October 1967), 687-97.

[19] Wos, L. and Robinson, G. The unit preference strategy in theorem proving. Proc. AFIPS 1964 Fall Joint Computer Conference 26, pp. 615-21, Washington, D C.: Spartan Books.

[20] Yates, R., Raphael, B. and Hart, T. Resolution graphs. Artificial Intelligence Group Technical Note 24, Stanford Research Institute, 1970.

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Carnegie-Mellon University Department of Computer Science Pittsburgh, Pennsylvania 15213	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP

3. REPORT TITLE

Some Linear Herbrand Proof Procedures: An Analysis

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Scientific Interim

5. AUTHOR(S) (First name, middle initial, last name)

D. W. Loveland

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
December, 1970	54	20

8a. CONTRACT OR GRANT NO. F44620-70-C-0107 b. PROJECT NO. A0827-5 c. 61101D d.	9a. ORIGINATOR'S REPORT NUMBER(S) 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. DISTRIBUTION STATEMENT

1. This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
TECH, OTHER	Air Force Office of Scientific Research 1400 Wilson Boulevard (SRMA) Arlington, Virginia 22209

13. ABSTRACT

Several Herbrand proof procedures proposed during the 1960 decade are shown to be related in varying degrees. Most of the paper deals with a relationship between s-linear resolution and model elimination. Refinements of each are proposed and the spaces of ground deductions are shown to be isomorphic in a suitable sense. The two refined procedures are then studied at the general level where they are no longer isomorphic and do not always relate to a natural ground level counterpart. Other topics considered are the introduction of an added merge condition to model elimination and also an expanded possible use of lemmas. Finally, the model elimination procedure is interpreted in the linked conjunct procedure of Davis and the matrix reduction procedure of Prawitz.

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT

Security Classification

Corrections and Improvements for

Some Linear Herbrand Proof Procedures: An Analysis

Note: "p. 2, par. 1, line -2" indicates the second page, first full paragraph, and the second from last line of that paragraph.

1. p. 1, Contents table

   line 3., change "21" to "20"
   line 4., " "23" to "22"
   line 5., " "28" to "27"
   line 6., " "37" to "36"
   line 7., " "41" to "40"
   line 8., " "42" to "41"

2. p. 1, Add the following paragraphs to the * footnote:

   The following report shows the completeness of certain strategies combining linear resolution, ordering and the merge condition: Reiter, R., "Two results on ordering for resolution with merging and linear format," Department of Computer Science, University of British Columbia (July 1970). The procedures developed in the report do not utilize the subsumption condition.

   The author is indebted to Raymond Reiter for the discovery of an error in the first version of the MTOCS procedure defined in this paper.

3. p. 4, line -2. Insert the following note after line -2.

   Note: In (ii)b, the resolvent B' of B_i and B_j, $j < i$, may have to be factored several times to unify literals of B' to obtain a B_{i+1} satisfying the subsumption condition. This is considered an inherent part of the definition of the s-resolution operation. This factoring is completely unrelated to any factoring in anticipation of the resolution operation.

4. p. 5, par. 2, line 2. Change "$B'_{r(n)}$" to "$B'_{r(m)}$" at two occurrences.

5. p. 6, line -7. Change "literals of the far parent" to "literals in B_{i+1} of the far parent".

6. p. 6, line -1. Add the following note at the bottom of page 6:

Note: The factoring restriction here does not apply to any factoring necessary as part of the s-resolution operation to satisfy the subsumption condition.

7. p. 8, par. 1, line 2. Delete comma between "refutation" and "we".

8. p. 9, line 12. Begin a new, standard margin, paragraph for "A chain is <u>admissible</u> ..."

9. p. 17, line -2. See "entry 9" given at the end of the list.

10. p. 18, par. 1, line 6. Remove comma after "hopefully".

11. p. 19, lines 1-3. Delete the entry marked (4); i.e. lines 1-3.

12. p. 19, line -11. Change "subsumption resolution" to "s-resolution".

13. p. 19, line -9. Alter the sentence beginning on this line to read:

On the other hand, as only merge literals are to be resolved upon, it is easy to construct examples... (as before).

14. p. 19, lines -6, -5. "subsumption resolution" to "s-resolution".

15. p. 21, line -1. Add the following sentences after the last sentence on p. 21.

For both the TOCS and MTOCS procedures s-resolution must be checked carefully in subcase 1a. The argument changes little going from the TOCS case to the MTOCS case, but one observes here why the far parent for s-resolution cannot be restricted to the merge resolvent to produce a stronger MTOCS procedure. (This fact was pointed out to the author by Raymond Reiter).

16. p. 27, par. 2, line 10. Insert the following sentence preceding the sentence "Thus D_1 is a refutation":
Factoring within the s-resolution operation to satisfy the subsumption condition is permitted in D_1, of course.

17. p. 27, line -2. Replace the last full sentence of p. 27 by the following sentences.

We now consider how D is obtained. Two ways exist of obtaining B_{i+1} from B_i in D_1, or D_{gr}: (1) resolution using B_i and a clause of S, or S_{gr}, and (2) s-resolution using B_i and a B_j, $i < j$. We consider the latter case first. Recall that there is a factoring restriction on the far parent in this case.

18. p. 28, line -1. See "entry 28" given at the end of this list.

19. p. 34, Remark, line 7, 8. Change "earlier clause resolution" to "s-resolution".

We now give the major alterations.

Entry 9. Delete the last sentence on page 17, the indented sentence and the following sentence at the top of page 18. Insert the following paragraph there.

To superimpose a merge condition on a TOCS deduction we must deviate from the usual form of employing merging, introduced in [2]. However, the effect will not be very different from the original intent. First, the notion of a "descendent" literal is needed. If C is a resolvent of clauses A and B, and literal L' occurs in C because it is a substitution instance of L of clause A or B, the L' is called a descendent of L. A descendent literal in a factor of a given clause is defined analogously. Also, we define a literal to be a descendent of itself and a descendent of a descendent of L to be a descendent of

L. For example, RaPxQa and -QyRy yield resolvent RaPz, where Ra is a descendent of Ry and Ra, and Pz is a descendent of Px.

It can be shown (see "Remark" in section 3) that the following restriction is compatible with the TOCS procedure:

> if the far parent C is not from S, then C must be a clause of the deduction containing descendents of merge literals and one of these descendents must be the literal resolved upon when C is used as far parent.

For convenience we call the augmented procedure the MTOCS (merge-TOCS) procedure. As we note below, the literal resolved upon in the far parent of an s-resolution operation can be shown to be the rightmost literal. Therefore, although missing here is the usual restriction of using only a merge resolvent as the far parent clause for s-resolution, there is still only one clause in the deduction where a given merge literal occurrence, or a descendent, is actually eligible to be resolved upon for s-resolution (not counting a clause to which c-factoring is applied).

Entry 18. Enter the following paragraphs after the last line on p. 28.

We now consider the remaining case, i.e., that B_{i+1} is obtained from B_i and a clause in S, or S_{gr}. Let B_{gr} denote such a B_i in D_{gr} and and B denote the corresponding clause in D_1. Let C_{gr} of S_{gr} denote the far parent in D_{gr} applied to B_{gr}, and let C be the corresponding clause of S. The obvious problem that may arise in deduction D_1 at such a step is that the rightmost literal of B is not the literal associated with the rightmost literal of image ground clause B_{gr}. For example, consider line 2 in D_{gr} and D_1 directly above, taking those clauses for B_{gr} and B respectively. If such a situation arises in D, as many c-factoring operations as needed may be applied to obtain the desired literal as rightmost literal, and then the resolution indicated by B_{gr} and C_{gr} may be executed at the general level. See lines 3 and 4 of deduction D above.

A less obvious situation can also arise at this step in the deductions of D_{gr} and D_1. Let L_{gr} denote the literal of C_{gr} which is resolved upon with L^c_{gr} of B_{gr} at this step. It may happen that there are two or more literals of C associated with L_{gr} of C_{gr} and the substitution which unifies the atoms of the single literals of C and B does not unify the several literals of C having L_{gr} as ground image. If this arises in deduction D then the resolvent B* in D does not have clause B'_{gr}, the resolvent of B_{gr} and C_{gr}, as ground image for there would be an occurrence of L_{gr} in B^*_{gr}. However, the extra occurrences can be removed by successive s-resolutions using B as far parent in deduction D. The final s-resolvent will have B'_{gr} as ground image.

We consider an example illustrating this point. Let $S_{gr} = \{QaPa, -Pa, -Qa\}$ and $S = \{QxPy, -Px-Py, -Qa\}$. Below, D has D_{gr} as one image ground deduction with r(1)=1, r(2)=3, and r(3)=4. Note that line 3 of D is an s-resolvent which required an additional factoring. Deduction D_1 is omitted as it is quite similar to D_{gr}. (Of course, there exists a key-triple deduction that appears like D, also).

D_{gr}: 1. QaPa ,
2. Qa ,
3. □ .

D : 1. QxPy ,
2. Qu-Pv ,
3. Qu ,
4. □ .