A LINEAR FORMAT FOR RESOLUTION

D. W. Loveland
Carnegie-Mellon University
Pittsburgh, Pennsylvania
December, 1968

This work was supported by the Advanced Research Projects Agency of the Office of the Secretary of Defense (F44620-67-C-0058) and is monitored by the Air Force Office of Scientific Research. This document has been approved for public release and sale; its distribution is unlimited.

This research was also partially supported by NSF Grant GP-7064.

ABSTRACT

The Resolution procedure of J. A. Robinson is shown to remain a complete proof procedure when the refutations permitted are restricted so that clauses C and D and resolvent R of clauses C and D meet the following conditions: (1) C is the resolvent immediately preceding R in the refutation if any resolvent precedes R, (2) either D is a member of the given set S of clauses or D precedes C in the refutation and R subsumes an instance of C or R is the empty clause, and (3) R is not a tautology.

A Linear Format for Resolution*

D. W. Loveland**

Following the introduction of the Resolution principle as a complete strategy for demonstration of the inconsistency of an unsatisfiable set of first order clauses in Robinson [1965a], there have been several papers demonstrating restrictions on the generation of resolvent clauses while maintaining the completeness condition. Papers of this type include Robinson [1965b], Wos, Robinson, Carson [1965], and Andrews [1968]. In this paper also a restricted format for resolution is shown to be a complete strategy.

We assume familiarity with the notation and results of Robinson [1965a], in particular sections 2 and 5. Our concern is to deduce a contradiction from a finite set S of clauses. Each clause is itself a set of literals. 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 $L_1 \in B$ 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 in general suitable instantiations of clauses B and C followed by the operation shown 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.

A particular distinguished clause is the empty clause, denoted by $\square$. A deduction of clause C (from the set S) 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 in S or a resolvent of B_j and B_k, $1 \le j, k < i$ and (ii) B_n is C. A refutation of the set S of clauses is a deduction of $\square$ from S. We define a linear deduction of C from the set S of clauses as a deduction of C from S such that $B_1, \ldots, B_k$ are in S and every B_i, $k+1 \le i \le n$ is a resolvent with B_{i-1} as one parent clause of the resolution. Each B_i, $i=k, \ldots, n-1$, is called a near parent clause. The other parent clause for B_{i+1} may be any B_j, $j \leqslant i$. The subsequence $B_1, \ldots, B_k$, which serves to introduce the needed members of S

*This research was partially supported by NSF Grant GP-7064, ARPA #F44620-67-C-0058.
**Carnegie-Mellon University, Pittsburgh.

into the deduction, is called the prefix of the linear deduction. A linear refutation of the set S of clauses is a linear deduction of □ from S.

In section 7 of Robinson [1965a] the notion of subsumption is introduced. We state the definition here so as to include subsumption by the empty clause: given two distinct clauses B and C, B subsumes C precisely if an instance of B is a subset of C, i.e. $B\sigma \subseteq C$ for some substitution σ.

An s-linear deduction of clause D from the set S of clauses is a finite sequence $B_1, B_2, \ldots, B_n$ of clauses such that

(i) the sequence is a linear deduction of D;

(ii) if $B_1, \ldots, B_k$ is the prefix of the deduction and if $k+1 \le i \le n$ then one parent clause of B_i is either (a) from S or (b) a clause B_j, $j < i-1$, chosen so that the resolvent B_i subsumes an instance of B_{i-1}.

(iii) no tautology occurs in the sequence of clauses.

(A clause is a tautology if it contains complementary literals.) We shall prove the following theorem.

Theorem. The set S of clauses is unsatisfiable if and only if there is an s-linear refutation of S (i.e. an s-linear deduction of □ from S).

In an s-linear deduction of D, if $i > k$ then we shall call the parent clause of B_i which is constrained by condition (ii) an imported (parent) clause. We may slightly weaken condition (ii) to make more explicit the nature of the constraint on the imported clause. We note that for k as above, if $i > k$, then the imported clause C for B_i is either a member of S or has the property that there exists instances $C\sigma$ and $B_{i-1}\gamma$ such that for each literal L of C not appearing in the key triple $L\sigma \in B_{i-1}\gamma$. For a deduction consisting of ground clauses (a ground deduction) condition (ii) requires that the imported clause C is either in S or if L_1 is the literal in C that "disappears" in the resolution of B_{i-1} and C, then $C - \{L_1\} \subseteq B_{i-1}$.

The reader should note that it is not always possible to deduce □ from a given unsatisfiable set S of clauses if resolution is restricted by the requirement that one parent clause always be from S. If S is formed from the full conjunctive form on two predicates (i.e. $S=\{\{P,Q\}, \{-P,Q\},\{P,-Q\},\{-P,-Q\}\}$) we observe the only new clauses generated under the above constraint are four one-literal clauses plus two tautologies.

(If S is formed from the full conjunctive form on three predicates, then not even complementary one-literal clauses are derivable from S under the above constraint.) Restriction of one parent clause to membership in S hence does not produce a complete refutation strategy for resolution. Condition (ii) is a slight weakening of the "one parent from S" restriction, a weakening that is sufficient to allow completeness.

What is the purpose of studying such restrictions on the resolution operation? One reason, of course, is to obtain a better understanding of the concept of resolution. More practically, it is hoped that restrictions will trim the number of resolutions performed in the search for a refutation when attempted by hand or by computer. Unfortunately, it seems that with at least some of the restrictions already tested that the shortest refutation is often eliminated by the given restriction. Then the search for the longer refutations usually proves nearly or totally as big as the original search in spite of the reduced number of resolutions needed to consider all required deductions of a fixed length. Establishing the completeness of a restricted form of resolution is useful, however, in that any relaxation of the restriction need be considered only if it justifies itself by frequently realizing sufficiently shorter refutations. For example, it might develop in practice that neglecting the linearity condition is better than using it. That is, perhaps in practice one obtains a good strategy by insisting that every resolution have one parent clause taken from S or else that one parent clause "subsumes" the other parent clause as stated in the weaker version of condition (ii). Although such a strategy is complete because all s-linear deductions may be developed, it might happen that few of the refutations which appear first in a computer realization of the strategy happen to be linear.

Another strategy which is shown to be complete by the theorem is one closely related to that given in Andrews [1968]. Following Andrews, we say a merge of clauses B and C exists if there exists an instantiation $B\gamma$ of B and $C\delta$ of C such that a resolvent exists and $B\gamma \cap C\delta$ is non-empty. From the theorem stated earlier, it follows that S is unsatisfiable if and only if there exists a refutation including only resolvents with one parent clause either in S or a one-literal clause or the resolvent itself is a resolvent with a merge. It should be noted that this strategy differs somewhat from that of Andrews [1968] largely in that Andrews uses a merged resolvent as one criterion for a parent of an acceptable resolvent.

Hand calculation of a few simple examples leads one to surmise that when s-linear deductions are employed "depth-first" rather than "breadth-first" searches may be desirable. The s-linear deductions obtained on the attempted examples were in general longer than the unrestricted deductions, but were also easily discovered. This suggests the possibility that good planning heuristics can estimate the clauses in S likely to be needed so that few attempts (of quite some depth) are needed before an s-linear refutation is found. Question-answer systems seem one area where this approach may be desirable.

We turn our attention to the proof of the theorem. We make use of the basic Lemma of Robinson [1965a]. We paraphrase the summary statement of Robinson [1965b]. _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'. By induction it follows that if S is a set of clauses, if S' is a set of ground clauses, each clause of which is an instance of S, and if there exists a deduction of ground clause D' from S', then there exists a deduction of a clause D from S where D' is an instance of D. If D' is the empty clause, then D must also be the empty clause. Thus to show the existence of a refutation of S, it suffices to show the existence of a ground refutation from a suitable S'. Moreover, in section 2 of Robinson [1965a] it is shown that precisely if S is unsatisfiable, there exists a finite set S' of instances of S for which a ground refutation of S' exists. (Also see summary in Robinson [1965b]). These results allow us to establish the theorem at the ground level and obtain the full theorem by appeal to the stated results. (Care must be taken that the necessary distinctions in the definition of an s-linear deduction in the ground and general cases are correctly drawn. This will be left to the reader to verify; the translation is quite direct.)

It is immediate that if there is an s-linear refutation of S then S is unsatisfiable, due to the soundness of resolution. We must establish the converse. From the preceding paragraph it is clear we may assume from the unsatisfiability of S that a ground refutation of S' exists where S' is a finite set of instances of clauses of S. We need show the existence of a ground s-linear refutation of S'. For convenience we identify S' with S hereafter and consider all clauses of S to be ground clauses. We shall let $A, A_i, i=1,2,\ldots,$ denote atoms and $L, L_i, i=1,2,3,\ldots,$ denote literals. Certain early alphabet capital letters, perhaps with subscripts, shall denote clauses; occasionally $S_i, i=1,\ldots,m$ shall denote the m (ground) clauses comprising S. A ground resolution is conveniently pictured by use of a directed graph consisting of a one node tree. For example, if B and C are clauses and $L_1 \in B$ and $L_2 \in C$ are complementary literals with common atom A, a graph representing the resolution of B and C is given in Figure 1. We associate each parent clause with an incoming directed line segment and associate the resolvent clause D with the outgoing directed line segment. We associate the atom A with the node itself and label A the _canceled atom_ of the node and of the resolution. The clause D, i.e. the set $(B - \{L_1\}) \cup (C - \{L_2\})$, does not have a literal with atom A if neither B nor C is a tautology.

Using the one node graph as a building block, we can display a refutation of S by a tree structure. Those clauses which are both resolvent clauses and parent clauses will label directed line segments passing from the node of the resolvent which formed the clause to the node of which it is one parent. The one outgoing line segment not pointing to a node, the _final_ segment, is labeled by the empty clause; each incoming directed line segment not coming from a node, an _initial_ segment, is labeled with a clause from S. Our assumption asserts the existence of such a tree. Figure 2 illustrates the tree giving the refutation of the set $S = \{\{P,Q\}, \{-P,Q\}, \{-Q\}\}$. Similarly, we can associate a tree structure with a deduction of clause D from S. Such a tree is called a _deduction tree of D from S_ (or a _refutation tree of S_ if D is the empty set). We shall often use the phrase "deduction tree of D" when S is determined by context. A _minimal deduction tree_ of D is a deduction tree of D for which no collection of directed line segments and nodes can be removed so that (perhaps with relabeling) a new deduction tree of D from S is formed.

The directed line segments and nodes (and their labels) on a path from an initial segment to the final segment is called a _branch_ of the tree. A branch is considered an _ordered_ collection of directed line segments, nodes, clauses and canceled atoms with the order coinciding with the direction of the directed line segments, e.g., clause D (and the associated final segment) is last in the ordering. Phrases such as "node N_1 precedes node N_2 on the branch" refer to this ordering. There will be occasions when a

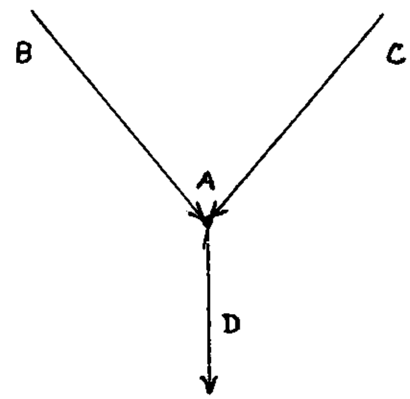

Figure 1

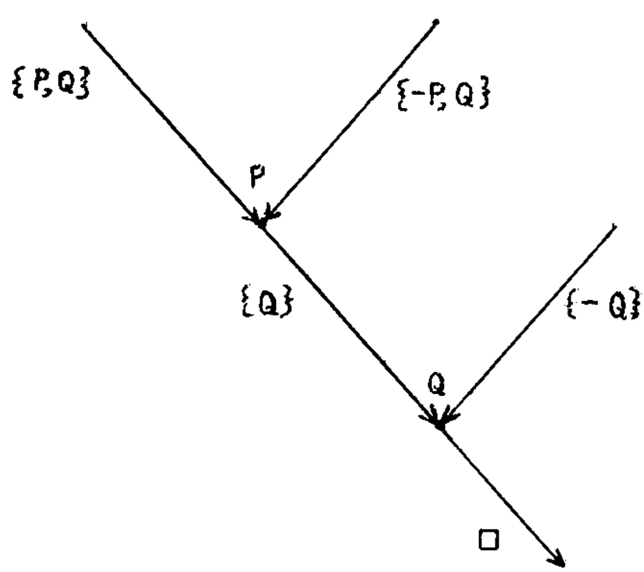

Figure 2

distinguished primary branch is indicated by specifying the initial segment. At a node N on the primary branch, a primary node, the parent clause of the resolution associated with N which lies on the primary branch is called the primary parent clause at N. The other parent clause is the secondary parent clause at N. The deduction tree of the secondary parent clause is the secondary deduction tree at N. The two complementary literals which allow the resolution at N, one of which is in the primary parent clause and the other which is in the secondary parent clause, are called canceled literals at N, one literal called the primary canceled literal and the other the secondary canceled literal. Clearly, both literals contain the canceled atom at N.

It is often useful to view a given deduction tree of D from S with a given primary branch as a sequence of primary clauses, the succeeding primary clause arising from a given primary clause by removal of one literal (the primary canceled literal) and the possible addition of new literals (from the secondary parent clause). In this regard we note the following fact.

Fact. If C is a given primary clause in a deduction tree of D from S and $L \in C$, then either L is a primary canceled literal in some following primary clause or $L \in D$.

The fact follows as there is a finite sequence of primary clauses between C and D, the last primary clause. If $L \notin D$ then there exists adjacent primary clauses C_1 and C_2 such that $L \in C_1$ but $L \notin C_2$. By the remark above, L must be the primary canceled literal of the node between C_1 and C_2.

The proof of the first lemma below proceeds by showing that a certain deduction tree is not minimal. We pass to a smaller tree structure by the operation of "removing a node N". The phrase remove (primary) node N shall imply the removal of all parts of the tree associated with primary node N, i.e., the secondary deduction tree at node N, the node N itself and the directed line segment of the resolvent at node N (with all associated labels). The primary parent clause G of node N becomes the primary parent clause of the following primary node N'.

The succeeding directed line segments on the primary branch must be relabeled with the correct resolvents of the indicated parent clauses from node N' to the final segment. It may well be impossible to perform a full relabeling in this manner because at some node the primary canceled literal is not present in the relabeled primary parent clause. The relabeling then halts and the tree structure left as is. In this case the tree is not a deduction tree.

To illustrate the above definition with Figure 2, we let the branch specified by {-P,Q} be the primary branch and remove the node with canceled atom P. The result is a tree such as given in Figure 1 where B is {-P,Q}, C is {-Q}, A is Q and D is {-P}.

Let N be a node with canceled atom A. A _positive_ (respectively, _negative_) _parent clause at_ N is a parent clause containing literal A (respectively, -A) and not containing literal -A (respectively, A). Clearly, a node need not possess a positive, or negative, parent clause. If node N has a positive (respectively, negative) parent clause, the _positive_ (respectively, _negative_) _subtree for_ N is the deduction tree of the positive (respectively, negative) parent clause at N.

Lemma 1. Let N be a node with canceled atom A in a minimal refutation tree of S. Let N have a positive and a negative parent clause. Then the positive (respectively, negative) subtree for N has no clause containing literal -A (respectively, A).

Proof. We give the proof for the positive subtree; the other case follows analogously. Let B denote the positive parent clause at N. Suppose literal -A appears in the positive subtree for N. Then it must appear in some clause $S_1 \in S$ where S_1 appears in the positive subtree for N at an initial segment. S_1 determines (temporarily) a primary branch of the positive subtree for N. There must be a primary node N* in the positive subtree for N with canceled atom A for otherwise, by the Fact stated earlier, B would contain literal -A. Therefore, the secondary parent clause F at N* must contain the literal A. The deduction tree of F must then have an initial segment labeled by $S_2 \in S$ with $A \in S_2$. We now choose the branch from S_2 of the refutation tree as the primary branch of the refutation tree. This branch contains nodes N* and N as primary nodes and hence specifies primary branches for the deduction tree of F and also the positive subtree for N. Now remove node N*. Further, remove any following primary node(s) which prevents the relabeling of the primary branch from being completed because of a missing primary canceled literal. The result of this modification is a smaller deduction tree. If it is a refutation tree, the original tree is not minimal, contradicting the hypothesis. Hence literal ~A cannot appear in any clause of the positive subtree for N and the lemma will be proved. We now show that the resulting tree is a refutation tree.

Because of the removal of node N*, the "new" primary clauses following F in the new deduction tree may contain A, where their counterparts in the given refutation tree may not. However, in the given refutation tree clause B contains literal A and, indeed, A is the primary canceled literal at N, where N follows N* on the primary branch. Node N, which appears in the new deduction tree unless the primary parent clause does not contain A, will remove A. This assures us that the literal A, though retained in primary clauses of the new deduction tree longer than for the original refutation tree, is eliminated not to appear in the final (primary) clause of the new deduction tree. Other than this addition of a literal, each new primary clause is a (perhaps proper) subset of its counterpart primary clause in the given refutation tree. (Recall in this regard that removal of any node other than node N* occurs only when the primary canceled literal is "already" missing from the primary clause. Literals appearing in a primary clause of the given refutation tree may be missing in the counterpart new primary clause, of course, because they were introduced by a secondary parent clause of a node deleted in the new deduction tree.) But the empty clause is the only subset of itself so the final primary clause of the new deduction tree must be the empty clause. Thus the new deduction tree is a refutation tree. The lemma is proved.

<u>Corollary 1</u>. A minimal refutation tree of S contains no tautologies.

<u>Proof</u>. Suppose the tree possesses a tautology B with complementary literals A and ~A. Choose as primary branch some branch containing B. Consider the last primary node N which has canceled atom A. Because the

tree is a refutation tree, N must have a positive and a negative parent clause, for otherwise the extra literal with atom A in one of the parent clauses must be present in the resolvent and hence, by Fact, in the empty clause. Contradiction. But then Lemma 1 is valid at node N. But B must be in either the positive or the negative subtree for N so cannot contain both A and -A.

Remark. By Corollary 1, in a minimal refutation tree every node has a positive and a negative parent clause.

Corollary 2. If two nodes lie on the same branch of a minimal refutation tree, then they do not have the same canceled atoms.

Proof. Suppose nodes N_1 and N_2 both have canceled atom A and suppose N_1 precedes N_2 on some branch of the refutation tree. By the preceding remark, N_2 has a positive and negative parent clause, hence a positive and negative subtree. N_1 must be in either the positive or negative subtree for N_2. But then either the primary or secondary canceled literal at N_1 must be missing by definition of positive (negative) subtree. Contradiction. The corollary is proved.

Suppose we are given a refutation tree of S with a designated primary branch and primary node N. We say a set J of literals satisfies the * condition (at N) if every literal in J is the primary canceled literal of a node following node N on the primary branch.

Lemma 2. Given a minimal refutation tree of S with a designated primary branch and designated primary node N, if D denotes the resolvent of primary parent clause B and secondary parent clause C, if B' is a subset of B containing the primary canceled literal and if J is a set of literals satisfying the * condition and disjoint from B', then there exists an s-linear deduction of a set $J \cup D'$ from $\{J \cup B'\} \cup S$, where $D' \subseteq D$. Moreover, the clause $J \cup B'$ need appear only as the first near parent clause of the s-linear deduction.

Before giving the proof of Lemma 2, let us see how it yields a proof of the Theorem.

<u>Proof of Theorem</u> (assuming Lemma 2). From remarks made earlier in the paper, we recall it suffices to prove the existence of a (ground) s-linear refutation from the existence of a (ground) refutation of S. Clearly, a minimal refutation tree of S exists if a refutation of S exists, so we may assume the given refutation tree of S is minimal. We are free to choose any branch as primary branch; we may base our selection on which clause we wish as the first near parent clause of our s-linear deduction. Our choice for first near parent clause must be an initial clause of some minimal refutation tree. The choice determines the primary branch. (This freedom allows us to assert about the general procedure that if a clause of S has an instance in a minimal refutation tree of S then there exists an s-linear refutation of S with the clause as the first near parent clause). We assume now a primary branch has been selected.

Let $E_1, E_2, \ldots, E_n$ be the sequence of clauses of the primary branch. In particular, $E_1 \in S$ and $E_n = \Box$. The s-linear deduction we now define has E_1 as the first near parent clause. A sequence of the members of S which appear on the refutation tree of S (with E_1 last) forms the prefix of the deduction. It suffices to show for i=1,2,...,n-1 how to obtain an s-linear deduction of some set E'_{i+1}, where $E'_{i+1} \subseteq E_{i+1}$, from $S \cup \{E_i'\}$ where $E_i' \subseteq E_i$ if we demand the s-linear deduction contain E_i' only as the first near parent clause. The juxtaposition of these deductions (with prefixes removed) appended to the above-mentioned prefix forms the desired s-linear deduction.

If $E_i' \subseteq E_{i+1}$, let $E'_{i+1} = E_i'$ and the required s-linear deduction is the empty sequence. If $E_i' \nsubseteq E_{i+1}$, it must be because E_i' contains the primary canceled literal of node N separating E_i from E_{i+1} in the refutation tree. But then we apply Lemma 2 with J taken as the empty set. This yields immediately the set E_{i+1}' and the (existence of the) required s-linear deduction. The theorem is proved.

We now give the proof of Lemma 2.

<u>Proof of Lemma 2</u>. The proof is by induction on the size n of the secondary subtree at N. Size of a subtree is measured by the number of directed line segments (or number of clauses counting duplications) in the subtree.

Case n=1. The secondary clause C is a clause in S as it must label an initial segment of the refutation tree. The resolvent of $J \cup B'$ and C is of form $J \cup D'$ where $D' \subseteq D$ and J and D' are disjoint. The desired s-linear deduction is the sequence C, $J \cup B'$, $J \cup D'$. We must show that none of these clauses is a tautology. By Corollary 1, C is not a tautology. Let L be a literal of J with atom A. Let N' be the last primary node with canceled atom A. Such a node exists as J satisfies the * condition. J cannot also contain complementary literal $\overline{L}$ for then it is a primary canceled literal at a node N" which must precede N' on the primary branch. But both N' and N" have the same canceled atom, violating Corollary 2. Thus J is not a tautology. Also clauses B and D are in the same subtree of node N' as the primary parent clause of N' which contains literal L. Thus neither clauses B or D contain $\overline{L}$ so neither $J \cup B'$ or $J \cup D'$ is a tautology. (Recall we know B and D are not tautologies by Corollary 1).

Case n=k, assuming the result true for $n < k$. Let L denote the primary canceled literal at node N. Because C contains $\overline{L}$ there is a clause $S_1 \in S$ within the deduction tree of C such that $\overline{L} \in S_1$. $J \cup (B' - \{L\}) \cup S_1'$, with $J \cup (B' - \{L\})$ and S_1' disjoint, is the resolvent of $J \cup B'$ and S_1. Here $S_1' \subseteq S_1$. The s-linear deduction begins with $B_1, B_2 \ldots, B_m$, $J \cup B'$, $J \cup (B' - \{L\}) \cup S_1'$ where $B_1, \ldots, B_m$ lists the members of S. These clauses are shown to be non-tautologous in the same manner as the clauses in Case 1. $J \cup B'$ is the first near parent clause of the s-linear deduction.

It is convenient to represent these clauses in a different notation. Define J* as the set $J \cup B' - \{L\}$. Then we may write $J \cup (B' - \{L\}) \cup S_1'$ as $J^* \cup E_1'$ where $S_1' = E_1' \subseteq E_1 = S_1$ (so E_1' and J* are disjoint). Thus the first two near parent clauses of the s-linear deduction desired are $J^* \cup \{L\}$ and $J^* \cup E_1'$. We now choose a new primary branch for the refutation tree, namely, that branch which begins with S_1. Note that the branch passes through node N but that C is now the primary parent clause and B the secondary parent clause at N. All terms hereafter refer to this new choice of primary branch. We let the sequence $E_1, E_2, \ldots, E_m$ denote the primary clauses through C, e.g. $E_1 = S_1$ and $E_m = C$. The primary clauses after E_m were the primary clauses following B under the choice of primary branch given by statement of the Lemma. All the literals of $B - \{L\}$ hence are primary canceled literals of nodes following N in the new primary branch as well as in the "old" primary branch. Hence J* satisfies the * condition with the new primary branch at any node preceding and including node N. We develop the s-linear deduction sequence after $J^* \cup E_1'$ to $J^* \cup E_m'$ in the same manner as we proved the Theorem using this Lemma. Note that the secondary deduction trees at the nodes preceding N are smaller than the deduction tree for C so the induction hypothesis may be invoked to use the Lemma. We recall the manner of obtaining an s-linear deduction of $J^* \cup E_{i+1}'$, for a suitable E_{i+1}', from $\{J^* \cup E_i'\} \cup S$ for $i=1,2,\ldots,m-1$. If $E_i' \subseteq E_{i+1}$, let $E_{i+1}'=E_i'$ and the desired deduction is the empty sequence. Otherwise, E_i' contains the primary canceled literal of the node N' between E_i and E_{i+1} so by induction hypothesis we have a clause $E_{i+1}' \subseteq E_{i+1}$, which we may also assume is disjoint from J^*, and an s-linear deduction of $J^* \cup E_{i+1}'$ with $J^* \cup E_i'$ as first near parent clause. Each of these deductions (minus their prefixes) for $i=1,2,\ldots,m-1$ are fitted together in sequence and appended to the beginning sequence of clauses named above to give an s-linear deduction of $J^* \cup C'$ from $\{J \cup B'\} \cup S$. The Lemma assures us no tautologies appear in the deduction. If $\bar{L} \notin C'$ then $J^* \cup C'$ may be written as $J \cup D'$ for a $D' \subseteq D$ with D' disjoint from J because $C' \cup B'-\{L\} \subseteq D$. However, $\bar{L}$ may appear in C'. In this case, we use the subsumption option of condition (ii) of the definition of an s-linear deduction. We resolve $J^* \cup \{L\}$ with $J^* \cup C'$ to obtain $J^* \cup C'-\{\bar{L}\}$ which meets the condition that the resolvent subsume its near parent clause. $J^* \cup C' -\{\bar{L}\}$, which may be written as $J \cup D'$ for a suitable D' as above, becomes the final clause in the s-linear deduction. This clause is certainly not a tautology if its predecessor is not. The Lemma is proved.

Suppose we remove from the definition of s-linear deduction the requirement that no tautology appear in the deduction. Then Lemma 2 can be proved as stated except that "a minimum refutation tree" may be replaced by simply "any refutation tree". The proof is as given with the sections concerning tautologies removed. The "practical" significance is that by making less strict the requirements for an acceptable deduction, one does obtain refutations "beginning with" (i.e. having as first near parent clause) members of S for which no true s-linear deduction exists. Indeed, by allowing tautologies, one may begin with any clause of S which appears in some refutation tree of S. A simple example shows that we cannot disallow tautologies and still maintain this freedom of choice of members of S for first near parent clause.

Let $S = \{\{P,Q\},\{-P,-Q\},\{P\},\{Q\}\}$. No s-linear refutation exists with $\{P,Q\}$ as first near parent clause although such a refutation exists if tautologies are allowed.

Finally, we note that from the Theorem (and its manner of proof) the completeness of the set of support strategy of Wos, Robinson, Carson [1965] is obtained. A refutation is a refutation of S with set of support $T \subseteq S$ if every clause of the refutation of S which is a resolvent has at least one parent clause either a resolvent itself or a member of T.

Corollary (Wos, Robinson, Carson). If S is a finite unsatisfiable set of clauses and if $T \subseteq S$ is chosen such that S - T is satisfiable, then there is a refutation of S with set of support T.

Proof. There must exist a (ground) minimal refutation tree of a finite set of ground instances of S with an occurrence of some $T_1 \in T$ as a label for some initial segment of the refutation tree. This is true because the set of ground instances of S-T is a satisfiable set. As we noted in the proof of the Theorem from Lemma 2, it follows from the proof of the Theorem that there exists an s-linear refutation of S with T_1 as first near parent clause. The first resolvent of this s-linear deduction has T_1 as one parent clause; all other resolvents have resolvents as one parent clause. The Corollary follows.

The author would like to thank Peter Andrews, whose comments have led to corrections of several shortcomings of the original paper.

BIBLIOGRAPHY

[1968] Andrews, P. B. "Resolution with merging", J.ACM, 15,3 (July 1968), 367-381.

[1965a] Robinson, J. A. "A machine-oriented logic based on the resolution principle," J.ACM, 12, 1 (Jan. 1965), 23-41.

[1965b] Robinson, J. A. "Automatic deduction with hyper-resolution", Int. J. Computer Math. 1 (1965), 227-234.

[1965] Wos, L., G. A. Robinson and D. F. Carson. "Efficiency and completeness of the set of support strategy in theorem proving", J.ACM 12, 4 (Oct. 1965), 536-541.

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

A LINEAR FORMAT FOR RESOLUTION

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, 1968	15	

8a. CONTRACT OR GRANT NO. F44620-67-C-0058 b. PROJECT NO. 9718 c. 6154501R 681304 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(SRMA) 1400 Wilson Boulevard Arlington, Virginia 22209

13. ABSTRACT

The Resolution procedure of J.A. Robinson is shown to remain a complete proof procedure when the refutations permitted are restricted so that clauses C and D and resolvent R of clauses C and D meet the following conditions: (1) C is the resolvent immediately preceding R in the refutation if any resolvent precedes R, (2) either D is a member of the given set S of clauses or D precedes C in the refutation and R subsumes an instance of C or R is the empty clause, and (3) R is not a tautology.

DD FORM 1 NOV 65 1473

Security Classification

Security Classification

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

Security Classification