6

Semantic Trees in Automatic Theorem-Proving

R. Kowalski
and
P. J. Hayes
Metamathematics Unit
University of Edinburgh

INTRODUCTION

We investigate in this paper the application of a modified version of semantic trees (Robinson 1968) to the problem of finding efficient rules of proof for mechanical theorem-proving. It is not our purpose to develop the general theory of these trees. We concentrate instead on those cases of semantic tree construction where we have found improvements of existing proof strategies. The paper is virtually self-contained and to the extent that it is not, Robinson's review paper (1967) contains a clear exposition of the necessary preliminaries.

After dealing with notational matters we define a notion of semantic tree for the predicate calculus without equality. A version of Herbrand's theorem is then proved. The completeness of clash resolution (Robinson 1967) is proved and it is shown that restrictions may be placed upon the generation of all factors when resolving a latent clash. The completeness of binary resolution is proved by specializing the notion of clash, and an ordering principle is shown to be complete when used in conjunction with it. Slagle's **AM**-clashes (1967) are shown to be complete by another specialization, and some clarification is presented of the rôle of Slagle's model **M** at the general level. A further specialization of **AM**-clashes is then made to the case of hyper-resolution (Robinson 1965a) and renaming (Meltzer 1966). It is shown in this case how the restrictions on generating factors and Slagle's **A**-ordering can be combined to give a highly efficient refutation procedure. Moreover, additional restrictions on the generation of factors are obtained for all cases of **AM**-clashes by employing throughout a modified notion of **A**-ordering. In the last section we report on attempts to apply the methods of semantic trees to the construction of inference systems for the predicate calculus with equality.

PRELIMINARIES

Familiarity is assumed with the reduction of sentences to clausal form. Atomic formulae are sometimes referred to simply as *atoms*. *Literals* are atoms or their negations; *clauses* are disjunctions of literals. Disjunctions and conjunctions will often be identified with the sets of their disjuncts and conjuncts respectively. Thus one may speak of a literal L occurring in a clause C and write $L \in C$. The *null disjunct* $\square$ is always false and therefore identical to the truth value *false*.

The result of applying a substitution σ to an expression E is denoted by $E\sigma$. If $E\sigma = F$ for some σ, then F is said to be an *instance* of E. In case F contains no variables F is a *ground expression* and a *ground instance* of E. If F is an instance of E and E of F, then E and F are *variants*. If expressions E and F have a common instance G, then E and F are *unifiable* and there is a most general common instance $E\sigma = F\sigma$, where σ is the *most general unifier* (*m.g.u.*) of E and F. The m.g.u. σ of E and F is such that if μ is any unifier of E and F then there is a λ such that $\mu = \sigma\lambda$.

Constants are functions of zero arguments. The *Herbrand universe* $\mathbf{H}$ of a set S of clauses is the set of all terms constructible from the function letters appearing in S (augmented by a single constant symbol if S contains no constant symbols). The *Herbrand base* $\hat{\mathbf{H}}$ is the set of all ground instances over $\mathbf{H}$ of atoms occurring in S. If K is a set of ground atoms, then by a *complete assignment* to the set K we mean a set $\mathscr{A}$ such that for every atom $A \in K$ exactly one of the literals A or $\bar{A}$ occurs in $\mathscr{A}$ and $\mathscr{A}$ contains no other members. If $\mathscr{A}$ is a complete assignment to some subset $K' \subseteq K$, then $\mathscr{A}$ is called a *partial assignment* to K. Given a set S and its Herbrand base $\hat{\mathbf{H}}$ any complete assignment $\mathscr{A}$ to $\hat{\mathbf{H}}$ can be considered as a possible interpretation of S (i.e., the universe of the interpretation is $\mathbf{H}$; the definition of the functions over $\mathbf{H}$ is incorporated in the definition of $\mathbf{H}$; and an n-place predicate P holds for $(t_1, \ldots, t_n)$, $t_i \in \mathbf{H}$ if and only if $P(t_1, \ldots, t_n) \in \mathscr{A}$).

Every tree is a partially ordered set T whose elements are its nodes. We shall use $\leqslant$ to refer to the partial ordering of the nodes. The unique node $N \in T$ such that $N \geqslant N'$ for every node N' is the *root* of the tree. Trees will be considered as growing downward. Thus the root of a tree is the highest node in the tree, and if there are at most finitely many nodes immediately below any node then the tree is finitely branching. A *tip* of a tree T is a node N which is above no other node. A *branch* of T is a sequence of nodes beginning with the root and such that each other node in the sequence lies immediately below the preceding node in that sequence. A branch of T is *complete* if either it is infinite or else it is finite and ends in a tip.

SEMANTIC TREES FOR THE PREDICATE CALCULUS WITHOUT EQUALITY

Definitions

Let K be a set of atoms. A finitely-branching tree T is a *semantic tree for* K

when finite sets of atoms or negations of atoms from K are attached to the nodes of T in such a way that

(i) the empty set is attached to the root and to no other node;

(ii) if nodes $N_1, \ldots, N_n$ lie immediately below some node N and the sets of literals $\mathscr{B}_i$ are attached to the nodes N_i, then $\hat{\mathscr{B}}_1 \vee \ldots \vee \hat{\mathscr{B}}_n$ is a tautology, where $\hat{\mathscr{B}}_i$ is the conjunction of the literals in $\mathscr{B}_i$;

(iii) the union of the sets of literals attached to the nodes of a complete branch of T is a complete assignment to K.

Given a set S of clauses and a semantic tree T for $\hat{\mathbf{H}}$ (the Herbrand base of S), then the union of all the sets attached to any complete branch of T is a complete assignment to $\hat{\mathbf{H}}$ and therefore a possible interpretation of S. Indeed it can be easily shown from condition (ii) of the definition that every complete assignment $\mathscr{A}$ to $\hat{\mathbf{H}}$ can be obtained in this way.

The partial assignment which is the union of all the sets of literals attached to the nodes of a branch ending in a node N is written $\mathscr{A}_N$ and is termed the *assignment at* N. In this notation the set $\mathscr{B}_i$ attached to N_i, referred to in (ii) above, is just $\mathscr{A}_{N_i} - \mathscr{A}_N$.

The only case of an infinite semantic tree that we shall consider in this paper is that of a *simple binary tree*, which is used in the proof of the version of Herbrand's theorem necessary for our applications. In this tree if N_1 and N_2 lie immediately below the node N, then $\mathscr{B}_1$ and $\mathscr{B}_2$ are just $\{A\}$ and $\{\bar{A}\}$ respectively for some ground atom A in K. Every other semantic tree considered will be a finite *clash tree*. If T is a clash tree, $N \in T$ and $N_1, \ldots, N_k, N_{k+1}$ lie immediately below N, then the set $\mathscr{B}_i$ attached to N_i for $1 \leqslant i \leqslant k$ is just $\{L_i\}$ and the set $\mathscr{B}_{k+1}$ attached to N_{k+1} is $\{\bar{L}_1, \ldots, \bar{L}_k\}$, where $\{L_1, \ldots, L_k\}$ is a partial assignment to K disjoint from the partial assignment $\mathscr{A}_n$. The nodes $N_1, \ldots, N_k$ are termed *satellite nodes* and the node N_{k+1}, a *nucleus node*.

Failure

If S is a set of clauses and T a semantic tree for $\hat{\mathbf{H}}$, then T is in some sense an exhaustive survey of all possible interpretations of S. If S is in addition unsatisfiable, then S fails to hold in each of these interpretations. These considerations motivate the definitions given below.

Let T be a semantic tree and C a clause. We say that C *fails at a node* $N \in T$ when C has a ground instance $C\sigma$ such that $\mathscr{A}_N$ logically implies $\neg(C\sigma)$. (We also write $\mathscr{A}_N \models \neg(C\sigma)$, using the symbol $\models$ to denote *logical implication*). Note that if C fails at N then $\mathscr{A}_N \models \neg C$. The converse, however, is not in general true. For if $\mathscr{A}_N = \{P(a), P(f(f(a)))\}$ and $C = \bar{P}(x) \vee P(f(x))$ then $\mathscr{A}_N \models \neg C$, but C does not fail at N.

Let T be a semantic tree and S a set of clauses. A node $N \in T$ is a *failure point for* S when some clause $C \in S$ fails at N but no clause in S fails at any node $M > N$. If N is a failure point for S and $M > N$, then M is a *free node* for

S. Note that if N is free for S then any node $M>N$ is also free for S and both M and N are free for any subset of S. Also if N is a failure point for S, then no node $M<N$ is free for S and both M and N are not free for any superset of S.

A semantic tree every branch of which contains a failure point for S is said to be *closed* (for S).

Herbrand's Theorem

The following is easily shown to be equivalent to Herbrand's Theorem.

Theorem 1. If S is an unsatisfiable set of clauses then there is a finite subset $K \subseteq \hat{\mathbf{H}}$ such that every semantic tree T for K is closed for S.

Proof. Let $(A_1, \ldots, A_n, \ldots)$ be an enumeration of the Herbrand base of S and let T' be a simple binary tree for $\hat{\mathbf{H}}$ constructed as follows: the empty set ϕ is attached to the root of T'; the sets $\{A_1\}$ and $\{\bar{A}_1\}$ are attached to the two nodes immediately below the root; and if either $\{A_n\}$ or $\{\bar{A}_n\}$ is attached to the node N, then the sets $\{A_{n+1}\}$ and $\{\bar{A}_{n+1}\}$ are attached to the nodes immediately below N. Any complete branch through T' represents a complete assignment $\mathscr{A}$ to $\hat{\mathbf{H}}$ and therefore is a possible interpretation of S. Since S is unsatisfiable, $\mathscr{A}$ fails to be a model of S and some clause $C \in S$ must be false in $\mathscr{A}$. It follows that some ground instance $C\sigma$ of C must be false in $\mathscr{A}$. But for this to happen the complement of each literal in C must occur in $\mathscr{A}$, and since there are only finitely many such literals they must occur already in some partial assignment $\mathscr{A}_N$ with $C\sigma$ false in $\mathscr{A}_N$. Thus, some $M \geqslant N$ is a failure point for S and T' is closed for S.

The number of nodes of T' free for S is finite, for otherwise, by König's lemma we could find an infinite branch of free nodes containing no failure point. Let k be the length of the longest branch of T' which ends in a failure point and let $K=\{A_1, \ldots, A_k\}$. Then every branch of T' corresponding to a complete assignment to K already contains a failure point for S. Now if T is any semantic tree for K then every complete branch corresponds to a complete assignment to K and must also contain a failure point for S. Therefore T is closed for S. Q.E.D.

Note that Robinson (1967) uses essentially the same tree T' in his proof of Herbrand's theorem. The semantic trees of this paper differ, however, from those of Robinson (1968). Robinson defines failure of a clause at a node of a semantic tree for ground clauses and establishes his main results for ground clauses first. These results are then 'lifted' to the general level by applying Herbrand's Theorem. By generalizing the definition of failure and by applying Herbrand's Theorem in the form above, we establish our results for the general level directly. A principal advantage of this modification is that it becomes clear how to restrict the generation of factors of clauses.

Inference node

The concept of inference node makes it possible to transfer from the semantics of semantic trees to the syntax of inference systems. A node N of a semantic

tree T is an *inference node* for a set of clauses S if N is free for S and the nodes immediately below N are failure points for S. Note that if T is closed for S and $\square \notin S$, then T contains an inference node. For if $\square \in S$, then $\square$ fails at the root of T, and T contains neither free nodes nor inference nodes; otherwise, if T contains no inference node, then it contains free nodes and since every free node lies above another free node, we can construct a complete branch all of whose nodes are free for S, contradicting the assumption that T is closed for S.

If $\mathscr{R}$ denotes a system of valid inference rules for clauses, then by $\mathscr{R}(S)$ we denote the union of the set S with the set of all clauses which can be obtained from S by one application of one of the inference rules in $\mathscr{R}$ to clauses in the set S. Setting $\mathscr{R}^0(S)=S$ we define $\mathscr{R}^{n+1}(S)=\mathscr{R}(\mathscr{R}^n(S))$.

The following theorem provides the foundation for our use of semantic trees in automatic theorem-proving.

Theorem 2. Let $\mathscr{R}$ be a system of valid inference rules and let there be given a particular way of associating with every unsatisfiable set of clauses S a finite semantic tree T for S such that

(*) there is an inference node $N \in T$, and for some subset $S' \subseteq S$ of the set of clauses which fail immediately below N there is a clause $C \in \mathscr{R}(S')$ such that C fails at N.

Then $\square \in \mathscr{R}^n(S)$ for some $n \geqslant 0$, and consequently $\mathscr{R}$ is a complete system of refutation.

Proof: Let S be unsatisfiable, T the semantic tree associated with S. Let n be the number of nodes of T free for S (n is finite since T is finite). If $\square \in S$, then $\square \in \mathscr{R}^0(S)$. Otherwise, by (*), there is an inference node $N \in T$ and a clause $C \in \mathscr{R}(S)$ such that C fails at N. Therefore the number of nodes of T free for $\mathscr{R}(S)$ is less than or equal to $n-1$. Similarly, since T is a closed semantic tree for $\mathscr{R}^{m-1}(S)$, $m>1$, (*) applies to $\mathscr{R}^{m-1}(S)$; and consequently the number of nodes of T free for $\mathscr{R}^m(S)$ is less than or equal to $n-m$. No node of T is free for $\mathscr{R}^n(S)$, and therefore the root of T is a failure point for $\mathscr{R}^n(S)$. But then $\square \in \mathscr{R}^n(S)$, for no other clause fails at the root of a semantic tree. Q.E.D.

Theorem 1 has been used implicitly in the statement and proof of Theorem 2, because $\mathscr{R}^i(S)$ unsatisfiable implies, by Theorem 1, that T is closed, and thus T has an inference node for each $\mathscr{R}^i(S)$, $i \geqslant 0$.

Deletion strategies

A clause is a *tautology* if it contains complementary literals. A clause C *subsumes* a clause D if it has an instance $C\sigma$ which is a subclause of D (i.e. $C\sigma \subseteq D$). If $\mathscr{R}$ is a system of inference whose completeness can be justified by Theorem 2, then $\mathscr{R}$ remains a complete inference system when we allow in $\mathscr{R}$ the deletion of tautologies and of subsumed clauses.

If C is a tautology then $C\sigma$ contains complementary literals for every σ. But no $\mathscr{A}_N$ contains complementary literals and C cannot fail on any semantic tree. If $C\sigma \subseteq D$ and D fails at some node N of a semantic tree, then some ground instance $D\xi$ of D fails at N, but then $C\sigma\xi$ also fails at N. Thus tautologies and subsumed clauses need never occur in a proof of $\square$ in the system $\mathscr{R}$, for in the proof of Theorem 2 it is clear that only clauses which fail at nodes of the semantic tree T associated with the original unsatisfiable set S need ever occur in such a proof. (If S is any unsatisfiable set of clauses, then certainly S remains unsatisfiable after deleting tautologies and subsumed clauses. However, such a demonstration does not provide a proof of the compatibility of these strategies with a system of inference.)

CLASH TREES

The Latent Clash Rule

All our applications of Theorem 2 will be to inference systems $\mathscr{R}$ which consist of just one rule of inference that is in each case a specialization of Robinson's (1967) latent clash resolution rule. The corresponding tree T associated with an unsatisfiable set S will similarly be a specialization of a clash tree.

If clauses $B_1, \ldots, B_k$ fail at the satellite nodes immediately below some inference node N, then we term them *satellite clauses*. If A fails at the corresponding nucleus node, then A is a *nucleus clause*.

The following theorem and its proof provide the general setting for subsequent specializations.

Theorem 3. Let a finite clash tree T be associated with every unsatisfiable set of clauses S (where T depends on S) and let $\mathscr{R}$ consist of the single rule of inference (latent clash resolution):

(**) From the 'nucleus clause' $A = A_0 \vee D_1 \vee \ldots \vee D_m$, and the 'satellite clauses' $B_i = B_{0i} \vee E_i$, $1 \leqslant i \leqslant m$, where the complements of the literals in E_i are unifiable with the literals in D_i, and ξ is the most general simultaneous unifier of these sets of literals for all $1 \leqslant i \leqslant m$ (the variables occurring in the clauses $A, B_1, \ldots, B_m$ being standardized apart),

infer the 'resolvent' $C = A_0\xi \vee B_{01}\xi \vee \ldots \vee B_{0m}\xi$.

(Moreover we may insist that the *clash condition* be satisfied, namely that no $E_i\xi$ or complement of $E_i\xi$ occurs in any of the clauses $A\xi$, $B_1\xi, \ldots, B_m\xi$ except in $B_i\xi$ itself and in $A\xi$ as $D_i\xi$).

Then, if any clauses $A, B_1, \ldots, B_k$ fail immediately below an inference node N in T, A has the form of the nucleus clause in (**) and corresponding to A we have satellite clauses $B_1, \ldots, B_m$, $m \leqslant k$, having the form of the satellite clauses in (**) such that the resolvent C in (**) fails at N.

Remarks. (*a*) Theorems 2 and 3 combine to yield *the completeness of latent clash resolution*; for the conclusion of Theorem 3 satisfies the hypothesis of Theorem 2 and therefore the conclusion of Theorem 2 holds, namely that (**) is complete.

(*b*) The rule (**) is stated without reference to unifiable partitions and lends itself naturally to a statement in terms of factors. In either case the number

of unifiable partitions or of factors which need be generated is in general less than the total number possible. We shall return to this point after the proof of Theorem 3.

(*c*) Later we shall specialize in various ways the form of the clash tree T associated with an unsatisfiable set S. The corresponding specializations of (**) and of the proof of Theorem 3 will provide proofs of completeness for these inference systems when combined with Theorem 2.

Proof of Theorem 3. Let T be a clash tree, $N \in T$ an inference node, $N_1, \ldots, N_k$ the satellite nodes immediately below N, and N_{k+1} the corresponding nucleus node. Let $\mathscr{A}_N$ be the partial assignment at N, the singleton $\{L_i\}$, where L_i is a ground literal, the set attached to the satellite node N_i, $1 \leqslant i \leqslant k$, and $\{\bar{L}_1, \ldots, \bar{L}_k\}$ the set attached to N_{k+1}. Suppose A fails at N_{k+1} and that B_i fails at N_i, $1 \leqslant i \leqslant k$.

First we show that each B_i has the form of a satellite clause in (**). Since B_i fails at N_i but not at N there is a ground instance $B_i\sigma_i$ which is false at N_i but not at N. Thus the complements of the literals in $B_i\sigma_i$ all occur in $\mathscr{A}_N \cup \{L_i\}$ but not in $\mathscr{A}_N$. So $B_i\sigma_i = B_{0i}\sigma_i \vee \bar{L}_i$, where $B_i = B_{0i} \vee E_i$, $E_i\sigma_i = \bar{L}_i$, and $B_{0i}\sigma_i$ is false in $\mathscr{A}_N$.

Now, to show that A has the form of a nucleus clause in (**) we note that similarly, as above, A fails at N_{k+1} but not at N. Therefore some ground instance $A\sigma$ is false at N_{k+1} but not at N, and consequently the complements of the literals in $A\sigma$ all occur in $\mathscr{A}_N \cup \{\bar{L}_1, \ldots, \bar{L}_k\}$ but not in $\mathscr{A}_N$. Thus $A\sigma = A_0\sigma \vee L_1 \vee \ldots \vee L_m$, where for simplicity the nodes $N_1, \ldots, N_k$ have been reordered if necessary so that the literals L_i, $1 \leqslant i \leqslant m$, which occur in $A\sigma$, will be an initial segment of $L_1, \ldots, L_k$. Thus $A = A_0 \vee D_1 \vee \ldots \vee D_m$, where $A_0\sigma$ is false in $\mathscr{A}_N$ and $D_i\sigma = L_i$.

It only remains now to show that the inferred clause C fails at N and that the clash condition may be imposed upon the clash rule. We have already shown that the clause $C' = A_0\sigma \vee B_{01}\sigma_1 \vee \ldots \vee B_{0m}\sigma_m$ fails at N since each of $A_0\sigma$ and $B_{0i}\sigma_i$ are false in $\mathscr{A}_N$. We shall show that C fails at N by showing that C' is an instance of C. But because ξ is the most general unifier which transforms all of the literals 'resolved upon' in the inference into single literals, and because $A, B_1, \ldots, B_m$ have been standardized apart, there is a substitution λ such that $C' = C\lambda$, and therefore C fails at N.

Suppose that the clash condition is violated and that some $E_i\xi$ or complement of $E_i\xi$ occurs in C. Then $E_i\xi\lambda = L_i$ or $\overline{E_i\xi\lambda} = \bar{L}_i$ occurs in C' which fails at $\mathscr{A}_N$. But then $\bar{L}_i$ or L_i would have to occur already in $\mathscr{A}_N$, and this is impossible. The only other possibility is for $E_i\xi$ or its complement to be identical to $E_j\xi$ or $D_j\xi$ for $j \neq i$, But then L_i would be identical to L_j or $\bar{L}_i$ to $\bar{L}_j$ for $j \neq i$, which is likewise impossible. Q.E.D.

Factoring

When applying the latent clash rule (**) or some specialization of it to prove a set of clauses unsatisfiable, a single clause will normally occur as a premiss of an application of the rule many times. To avoid the duplication involved in repeatedly unifying the same groups of literals within a clause we may employ the device of factoring. The single most general simultaneous unifier (m.g.s.u.) ξ of (**), can be decomposed into a sequence of components $\xi_1, \ldots, \xi_m, \xi_{m+1}$, and ξ' such that ξ_i is the m.g.u. of E_i, $1 \leqslant i \leqslant m$, ξ_{m+1} the m.g.s.u. of $D_1, \ldots, D_m$, and ξ' the m.g.s.u. of $\overline{E_i\xi_i}$ with $\overline{D_i\xi_{m+1}}$, $1 \leqslant i \leqslant m$. Then in (**) the resolvent $C = A_0\xi \vee B_{01}\xi \vee \ldots \vee B_{0m}\xi$ equals $(A_0\xi_{m+1} \vee B_{01}\xi_1 \vee \ldots \vee B_{0m}\xi_m)\xi'$, where the unifiers ξ_i, $1 \leqslant i \leqslant m+1$, perform the necessary unifications within a single clause and ξ' only mates simultaneously single literals in the satellite clauses with the corresponding single literals in the nucleus clause. The unifier ξ' must be constructed separately for each application of the inference rule; but the unifiers ξ_i need only be constructed once when a clause is first produced, and this same substitution may be associated with its clause whenever the literals E_i, in case $1 \leqslant i \leqslant m$, or the literals $D_1, \ldots, D_m$, in case $i = m+1$, are the literals unified and resolved upon in an inference.

These considerations motivate replacing (**) by two independent operations, factoring and the resolution of factored clauses. A *factor* of a clause C is a clause $C\theta$, where θ is the m.g.u. of a single subset of literals in C in case $C\theta$ is to be used as a satellite clause, or a clause $C\theta$ where θ is the m.g.s.u. of subsets $D_1, \ldots, D_m$ of literals in C in case $C\theta$ is to be used as a nucleus clause. In addition we require that the literals in $C\theta$ which have been deliberately unified by θ be somehow *distinguished* from those which have not (this may be accomplished for programming purposes, for instance, by storing $C\theta$ with its distinguished literal or literals occurring first in $C\theta$ and separated in some way from those literals which follow and are not distinguished). The operation of factoring then consists of replacing each clause C which is not a factor by the set of all its factors. The clash rule for factored clauses then amounts to resolving a clash on its distinguished literals.

The notion of factoring defined above is an improvement on the notion one obtains by straightforward translation of unifiable partitions into terms of factors. Firstly, only the literals which are to be resolved upon are deliberately unified in a factor. Conversely, only literals deliberately unified need be resolved upon. For the case of binary resolution this version of factoring is equivalent to Robinson's (1965a) notion of 'key triple'. The remarks about factoring above are however completely general and apply to any inference rule which is a specialization of latent clash resolution.

Binary resolution and A-ordering

Given a set of clauses S we define an **A**-*ordering* for S to be a total ordering $\leqslant$ defined on some subset of the set of literals $\{L\sigma | L \in C \text{ for some clause } C \in S\}$ such that

(i) if $L_1 < L_2$ then $L_1\sigma < L_2\sigma$ for all σ;
(ii) if L_1 and L_2 are alphabetic variants or complements then $L_1 \leqslant L_2$ and $L_2 \leqslant L_1$.

This definition is similar to Slagle's (1967) but has the advantage of allowing a finer discrimination between literals.

Now let S be an unsatisfiable set of clauses and $K \subseteq \hat{\mathbf{H}}$ a finite subset such that any semantic tree for K is closed. Let $\leqslant$ be an **A**-ordering for S and $\{A_1, \ldots, A_n\} = K$ be an enumeration of K compatible with $\leqslant$; i.e., if $A_i < A_j$ then $i < j$. We associate with S the simple binary tree T for K obtained by attaching ϕ to the root of T, the sets $\{A_1\}$ and $\{\bar{A}_1\}$ immediately below the root and the sets $\{A_{i+1}\}$ and $\{\bar{A}_{i+1}\}$ immediately below any node to which $\{A_i\}$ or $\{\bar{A}_i\}$ has been attached.

Referring to the proof of Theorem 3 and using the notion of factor, we see that if a clause C fails at a failure point $N \in T$, then the distinguished literal L_1 of some factor $C\theta$ fails properly at N, while the remaining literals in $C\theta$ fail at nodes above N. Since the enumeration $\{A_1, \ldots, A_n\}$ is compatible with $\leqslant$ it follows that $L_1 < L_2$ for no $L_2 \in C\sigma$, where $L_1 \neq L_2$. For otherwise, if $L_1 < L_2$ and $(C\theta)\sigma$ is the ground instance of $C\theta$ which is false at N, then $L_1\sigma < L_2\sigma$ by (i) and yet, by the construction of T, $L_2\sigma$ is A_i or $\bar{A}_i$, and $L_1\sigma$ is A_j or $\bar{A}_j$, where $i < j$; so by (ii) $L_1\sigma < L_2\sigma$ cannot occur.

Taking into consideration the remarks above, specializing Theorem 3 appropriately and applying Theorem 2 we obtain completeness of the following version of binary resolution:

> Given a set of clauses S and an **A**-ordering $\leqslant$ for S, infer from the factors $\bar{L}_1 \vee A_0$ and $L'_1 \vee B_0$ with distinguished literals L_1 and L'_1 respectively, the clause $(A_0 \vee B_0)\xi$ where ξ is the m.g.u. of L_1 and L'_1, and where neither $L_1 < L_2$ nor $L'_1 < L_2$ for any literal L_2 in either A_0 or B_0.

As an example of the use of an **A**-ordering in conjunction with this rule, let the **A**-ordering $\leqslant$ be determined for some set of clauses by the conditions $P(f(x))\sigma < P(g(y))\sigma$ and $P(x)\sigma < Q(y)\sigma$ for all σ. Then the unfactored clause $C = P(g(a)) \vee P(f(x)) \vee Q(b)$ has only one factor $C\theta$, where θ is the identity substitution and $P(f(x))$ is the distinguished literal. In this case the **A**-ordering $\leqslant$ has eliminated the need to consider two of the three possible factors. If $C = Q(f(a)) \vee P(x) \vee P(f(a))$ then there are three factors of C compatible with $\leqslant$ and only one of the four possible factors need not be generated.

The following example shows that the rule above is compatible with neither set of support (Wos, Carson and Robinson 1965) nor P_1-deduction (Robinson 1965b): let S be the set $\{L_1 \vee L_2, L_1 \vee \bar{L}_2, \bar{L}_1 \vee L_2, \bar{L}_1 \vee \bar{L}_2\}$, and let $\leqslant$ be determined by $L_1 < L_2$. Then, although S is unsatisfiable, $\square$ can be deduced with neither P_1-deduction nor with set of support if we take $\{L_1 \vee L_2\}$ as the set of support.

In the next sections we shall see that a weaker version of the **A**-ordering restriction applies to **M**-clashes, and that in the particular case of P_p-deduction (P_1-deduction with renaming) a more restrictive ordering principle based on **A**-ordering is complete.

M-clashes

We have been unable to construct binary semantic trees to justify either the set of support strategy or P_1-deduction. However, the **M**-clash trees which we introduce below can be used to prove completeness of **M**-clashes, and by suitably choosing the interpretation **M** and by decomposing the corresponding **M**-clash rules we obtain, following Slagle (1967), the completeness of these inference systems.

Let S be a set of clauses. Define a *Herbrand interpretation* of S to be any complete assignment to $\hat{\mathbf{H}}$, the Herbrand base of S (we have already seen how any complete assignment to $\hat{\mathbf{H}}$ can be regarded as a possible interpretation of S). Assume for the moment that **M** is a Herbrand interpretation of S. Let S be unsatisfiable and $K \subseteq \hat{\mathbf{H}}$ a finite subset such that any semantic tree for K is closed. Let $M = \{A'_1, \ldots, A'_n\}$ be **M** restricted to K, where $A_i \in \hat{\mathbf{H}}$ and $A'_i = A_i$ if $A_i \in \mathbf{M}$ and $A'_i = \bar{A}_i$ if $\bar{A}_i \in \mathbf{M}$, so that $M \subseteq \mathbf{M}$ and M is a complete assignment to K. We associate with S the **M**-*clash tree* T defined as follows:

(i) ϕ is attached to the root of T;

(ii) the root of T has $n+1$ immediate descendants with $\{A'_i\}$ assigned to the ith satellite node, $1 \leqslant i \leqslant n$, and $\{\bar{A}'_1, \ldots, \bar{A}'_n\}$ to the nucleus node;

(iii) let $N \in T$, $\mathscr{A}_N$ not a complete assignment to K, $\mathscr{A}_N \subseteq M$ and $M - \mathscr{A}_N = \{A'_{j_1}, \ldots, A'_{j_k}\}$; then N has $k+1$ immediate descendants, with the singletons $\{A'_{j_1}\}, \ldots, \{A'_{j_k}\}$ attached to the k satellite nodes and the set $\{\bar{A}'_{j_1}, \ldots, \bar{A}'_{j_k}\}$ to the nucleus node.

Note that the assignment at any nucleus node is a complete assignment to K and therefore every nucleus node of T is a tip of T. Note, too, that the assignment at any satellite node is always a subset of M, and that for any such assignment containing $m \leqslant n$ literals there is a total of exactly $m!$ satellite nodes with the same assignment.

Suppose the clause B fails at a satellite node which is a failure point. Then some ground instance $B\sigma$ of B is false in M and therefore in **M**. Thus B itself is false in the Herbrand interpretation **M**.

Suppose A fails at a failure point N which is a nucleus node. Some ground instance $A\sigma$ of A must fail at N and some literals $A'_{j_1}, \ldots, A'_{j_m}$, for some $m \leqslant k$, must fail properly at N. But since these literals belong to $M \subseteq \mathbf{M}$ and therefore are true in **M** it follows that $A\sigma$ is true in **M**. Thus A has an instance which is true in **M**.

Note that, since the resolvent C of a clash fails at a satellite node, C must be false in **M**. Note also that a nucleus clause is never a resolvent and therefore must belong to the original set of clauses S.

Specializing Theorem 3 to the **M**-clash tree, keeping in mind the considerations above and applying Theorem 2, we obtain the completeness of **M**-clash resolution:

Given a set of clauses S and a Herbrand interpretation **M** of the Herbrand base **H** of S, from a factored nucleus clause $A = L'_1 \vee \ldots \vee L'_m \vee A_0$ and factored satellites $B_i = \bar{L}_i \vee B_{0i}$, $1 \leqslant i \leqslant m$, where ξ is the most general unifier such that $L_i\xi = L'_i\xi$ simultaneously for all $1 \leqslant i \leqslant m$,

infer the resolvent $C = (A_0 \vee B_{01} \vee \ldots \vee B_{0m})$,

when the clash conditions are satisfied; here each satellite B_i is false in **M**, C is false in **M**, and A has an instance true in **M**.

Remarks. (*a*) Slagle (1967) has remarked that if $T \subseteq S$ and $T - S$ is satisfiable then it is satisfied by a Herbrand interpretation **M**. It follows that no clause in $T - S$ can be a satellite of an **M**-clash. Decomposing the resulting **M**-clashes into sequences of binary resolutions we see that no two clauses are resolved which both come from $T - S$. But this is just the defining condition of the set of support strategy.

(*b*) Complications arise if we wish to use an interpretation **M** explicitly when in an application of the **M**-clash rule we need to decide the truth or falsity of clauses and their instances. Firstly, if **M** is not a Herbrand interpretation then we must extend **M** to an interpretation in which all the Skolem functions which actually occur in S are defined in some way. To **M** extended in this way there will then correspond a Herbrand interpretation which will justify the use of this extended **M**. Otherwise questions of truth or falsity for clauses whose vocabularies are not fully interpreted in **M** are literally meaningless.

A much more serious restriction on the explicit use of an interpretation **M** is that it actually admit of an algorithm for deciding truth and falsity of clauses and their instances. Otherwise there is no way in which its use can be mechanized for a computer. Interpretations containing only a finite number of elements are effective in this sense. But unless they possess other special properties the exhaustive instantiation of each clause over the domain of the interpretation for the purpose of testing it for false or true instances is likely to be prohibitive. These same considerations apply to model partitions (Luckham 1968), which can be justified as a special case of **M**-clashes.

(*c*) Slagle has also shown that hyper-resolution may be regarded as the special case of **M**-clashes where all instances of positive literals are regarded as false in **M** and all instances of negative literals as true. Then all satellites and resolvents contain only positive literals, and all the negative literals of the nucleus clause are resolved upon in the clash. The resulting clash is maximal in the sense that no subclash need ever be generated. This highly desirable property can be extended by the device of renaming. We shall show in a later section how advantage can be taken of maximality to yield a particularly efficient version of P_1-deduction.

AM-clashes

Let $\leqslant$ be an A-ordering and **M** a Herbrand interpretation for a set of clauses then the following ordering principle may be imposed upon the **M**-clash rule:

Given the factored nucleus $A = L'_1 \vee \ldots \vee L'_m \vee A_0$ and the factored satellites $B_i = \bar{L}_i \vee B_{0i}$, $1 \leqslant i \leqslant m$, with resolvent $C = (A_0 \vee B_{01} \vee \ldots \vee B_{0m})\xi$ satisfying the **M**-clash conditions, we may insist that for no literal $L_i \in B_{0i}$ do we have $L_i < L'$ for any $1 \leqslant i \leqslant m$.

This restriction improves upon Slagle's ordering principle, for the notion of **A**-ordering defined above is generally more restrictive. The proof that follows of the compatibility of the ordering principle above with the **M**-clash rule is essentially an adaptation of Slagle's argument.

Given a closed **M**-clash tree T and an **A**-ordering $\leqslant$ for the unsatisfiable set S, we shall show that there is in T an inference node N such that the factored clauses A and B_i, $1 \leqslant i \leqslant m$, which fail below the inference node and their resolvent C, satisfy the **AM**-clash conditions. It will then follow by Theorem 2 that the **AM**-clash rule is complete. We note that it is in fact only necessary to show that the satellites B_i satisfy the ordering principle above since we have already seen that the **M**-clash conditions are satisfied.

As before let $M \subseteq \mathbf{M}$ be defined as the set $\{A'_1, \ldots, A'_n\}$, where the ordering of the A'_i is compatible with the **A**-ordering $\leqslant$, i.e. if $A'_i < A'_j$, then $i < j$. We construct a subset M' of M, as follows:

(i) $M'_0 = \phi$.

(ii) If some factored clause $B_{i+1} = L_{i+1} \vee B_{0\ i+1}$ of some clause in S has a ground instance $B_{i+1}\sigma$ false in $\{A'_{i+1}\} \cup M'_i$ but no clause in S fails in M'_i, then $M'_{i+1} = M'_i$; otherwise $M'_{i+1} = \{A'_{+1}\} \cup M'_i$. In the former case we may choose the factor B_{i+1} so that $\bar{L}_{i+1}\sigma = A'_{i+1}$, and we say that B_{i+1} is associated with A'_{i+1}.

(iii) $M' = M_n$.

M' is a partial assignment to K and there is some node N such that $\mathscr{A}_N = M'$. We claim that N is an inference node. Note that the factor B_i associated with A'_i fails at the satellite node N_i immediately below N. Some nucleus clause fails at the nucleus node and no clause in S fails at N, by the construction of M'. A resolvent C from the nucleus clause and from some subset of the satellite clauses fails at N.

Suppose now that the **A**-ordering restriction is violated and that therefore for some i the distinguished literal L_i in the factor $B_i = L_i \vee B_{0i}$ associated with A'_i is less than some literal $L''_i \in B_{0i}$, i.e., $L_i < L''_i$. Then, if $B_i\sigma$ is the ground instance of B_i which fails at N_i, we have $L_i\sigma < L''_i\sigma$. But $\bar{L}_i\sigma = A'_i$ and $\bar{L}''_i\sigma = A'_j$ for some $j < i$, and since it is not true that $A'_i < A'_j$ by the compatibility of the enumeration of the A'_i with $\leqslant$, it follows that $L_i\sigma < L''_i\sigma$ does not hold; consequently it is not true that $L_i < L''_i$.

Hyper-resolution and P_1-deduction

Until other more efficient and mechanizable applications are found for **AM**-clash resolution, the two most likely candidates for an efficient proof strategy

seem to be set of support and hyper-resolution both supplemented by factoring and **A**-ordering. Meltzer (1968) has shown that renaming and P_1-deduction can often be used to sharpen a set of support strategy. Luckham (1968) has given examples of proofs obtained by what is essentially P_1-deduction with renaming, and these proofs are no less efficient than those obtained with set of support. The version of P_1-deduction stated below seems to us a distinct improvement over the existing strategy. Preliminary results obtained by programming this strategy in ATLAS-AUTOCODE on the KDF9 support this view.

Hyper-resolution has the advantage over P_1-deduction that it avoids generating the $(2^n)!$ resolvents that are produced by resolving in all possible ways among n-factored satellite clauses and a given factored nucleus clause (where each distinguished literal in the nucleus is associated with only one of the satellites). It has the disadvantage of not saving the partial hyper-resolvents that are generated on the way to producing the maximal hyper-resolvent. These partial hyper-resolvents need to be recomputed each time any of them is completed in a distinct way. In addition, the problem of searching for hyper-resolvents is more complicated than the corresponding problem for P_1-deduction. The following version of P_1-deduction incorporates the advantages of hyper-resolution over P_1-deduction without suffering from its disadvantages.

Given a set of clauses S and an **A**-ordering $\leqslant$ and after a renaming (if desired),

(i) replace each non-positive clause in S by the set of its factors (a non-positive clause is factored as a nucleus clause). Choose any total ordering of the distinguished literals in such a factor A (the ordering may be chosen independently for each A). Let $A = \bar{L}_1 \vee \ldots \vee \bar{L}_n \vee A_0$, where we agree to write negative literals $\bar{L}_i$ in order and before positive literals and where A_0 is the positive subclause of A.

(ii) replace each positive clause in S and, later, each positive resolvent by the set of its factors (positive clauses are factored as satellite clauses). We may insist that the **A**-ordering restriction is satisfied for each such factor.

(iii) resolve positive factors on their distinguished literal against the *first* negative literal of non-positive factors. If the resolvent is negative it is not factored, but a new ordering of its distinguished literals may be chosen if desired. If a resolvent is positive it is factored as in (ii) above.

It is easily seen that every hyper-resolvent is obtainable exactly once by a sequence of resolutions satisfying restriction (iii). It follows that this inference system is complete and that it satisfies the properties claimed above.

Condition (i) may be improved and replaced by

(i′) replace each non-positive clause A by its factor $A\theta = A$; where θ is the identity substitution. Choose any total ordering of the distinguished literals . . ., etc.

Condition (i′) states in effect that non-positive clauses are not factored at all. The proof that completeness is preserved when (i) is replaced by (i′) is somewhat complicated and does not lie within the scope of this paper.

Darlington (1969) shows how to exploit renaming, A-ordering, and the ordering of negative literals in non-positive clauses to avoid performing most of the resolutions excluded by set of support. He does this for the case of applications of theorem-proving to large-scale information retrieval systems where a set of support strategy seems to be highly desirable.

Other applications of semantic trees

The notion of semantic trees employed in this paper can easily be extended to the predicate calculus with equality. Indeed, Robinson's (1968) original formulation of the semantic tree construction was for this logic. None the less, we have been unable to find any binary semantic trees which yield reasonably mechanizable inference systems. It is easy to show that assignment trees (Sibert 1967) can be constructed as semantic clash trees. In this case, by exploiting the generalized notion of failure, it has been possible to impose additional restrictions on the generation of unifiable partitions. However, the basic system of three inference rules corresponding to inference nodes remains essentially that of Sibert's thesis. We are pessimistic about the possibilities of finding other semantic tree constructions which yield efficient inference systems for the predicate calculus with equality.

Hayes (1969) has applied the semantic tree method to obtain a simple mechanizable inference system for J. McCarthy's three-valued predicate calculus.

Acknowledgements

This work was supported by a grant to Dr Meltzer from the Science Research Council (Hayes) and by an I.B.M. fellowship and grant from Imperial College (Kowalski). The authors wish to thank Miss Isobel Smith for helpful discussion and Dr Meltzer for unflagging encouragement and guidance.

REFERENCES

Darlington, J. (1969) Theorem proving and information retrieval. *Machine Intelligence 4*, pp. 173–81 (eds Meltzer, B. & Michie, D.). Edinburgh: Edinburgh University Press.

Hayes, P. J. (1969) A machine-oriented formulation of the Extended Functional Calculus. *A.I. Memo*. Stanford University: Artificial Intelligence Project. (To appear.)

Luckham, D. C. (1968) Some tree-paring strategies for theorem-proving. *Machine Intelligence 3*, pp. 95–112 (ed. Michie, D.). Edinburgh: Edinburgh University Press.

Meltzer, B. (1966) Theorem-proving for computers: some results on resolution and renaming. *Comput. J.*, **8**, 341–3.

Meltzer, B. (1968) Some notes on resolution strategies. *Machine Intelligence 3*, pp. 71–5 (ed. Michie, D.). Edinburgh: Edinburgh University Press.

Robinson, J. A. (1965a) A machine-oriented logic based on the resolution principle. *J. Ass. comput. Mach.*, **12**, 23–41.

Robinson, J. A. (1965b) Automatic deduction with hyper-resolution. *Int. J. comput. Math.*, **1**, 227–34.

Robinson, J. A. (1967) A review of automatic theorem-proving. *Annual symposia in applied mathematics XIX*. Providence, Rhode Island: American Mathematical Society.

Robinson, J. A. (1968) The generalized resolution principle. *Machine Intelligence 3*, pp. 77–93 (ed. Michie, D.). Edinburgh: Edinburgh University Press.

Sibert, E. E. (1967) A machine-oriented logic incorporating the equality relation. Ph.D. thesis. Rice University.

Slagle, J. R. (1967) Automatic theorem-proving with renamable and semantic resolution. *J. Ass. comput. Mach.*, **14**, 687–97.

Wos, L. T., Carson, D. F. & Robinson, G. A. (1965) Efficiency and completeness of the set of support strategy in theorem-proving. *J. Ass. comput. Mach.*, **12**, 536–41.