THEORY RESOLUTION: BUILDING IN NONEQUATIONAL THEORIES

Mark E. Stickel

Artificial Intelligence Center
SRI International, Menlo Park, CA 94025

ABSTRACT

Theory resolution constitutes a set of complete procedures for building nonequational theories into a resolution theorem-proving program so that axioms of the theory need never be resolved upon. Total theory resolution uses a decision procedure that is capable of determining inconsistency of any set of clauses using predicates in the theory. Partial theory resolution employs a weaker decision procedure that can determine potential inconsistency of a pair of literals. Applications include the building in of both mathematical and special decision procedures, such as for the taxonomic information furnished by a knowledge representation system.

I INTRODUCTION

Building theories into derived inference rules so that axioms of the theory are never resolved upon has enormous potential for reducing the size of the exponential search space commonly encountered in resolution theorem proving [5,9]. Plotkin's work on equational theories [15] was concerned with general methods for building in theories that are equational (i.e., theories that can be expressed as a set of either equalities or, by slight extension, equivalences of a pair of literals). This building in of equational theories consisted of using special unification algorithms and reducing terms to normal form. This work has been extended substantially, particularly in the area of development of special unification algorithms for various equational theories [16].

Not all theories that it would be useful to build in are equational. For example, reasoning about orderings and other transitive relations is often necessary, but using ordinary resolution for this is quite inefficient. It is possible to derive an infinite number of consequences from $a < b$ and $(x < y) \wedge (y < z) \supset (x < z)$ despite the obvious fact that no refutation based on just these two clauses is possible. A solution to this problem is to require that use of the transitivity axiom be restricted to occasions when either there are matches for two of its literals (partial theory resolution) or a complete refutation of the ordering part of the clauses to be refuted can be found (total theory resolution).

Another important form of reasoning in artificial intelligence applications addressed by knowledge representation systems [4] is reasoning about taxonomic information and property inheritance. One of our goals is to be able take advantage of the efficient reasoning provided by knowledge representation systems in this area by using the knowledge representation system as a taxonomy decision procedure in a larger deduction system. Combining such systems makes sense, since it relieves the general-purpose deduction system of the need to do taxonomic reasoning and, in addition, extends the power of the knowledge representation system towards greater logical completeness. Other researchers have also cited advantages of integrating knowledge representation systems with more general deductive systems [3,18]. KRYPTON [2] represents an approach to constructing a knowledge representation system composed of two parts: a terminological component (the TBox) and an assertional component (the ABox). For such systems, theory resolution indicates in general how information can be provided to the ABox by the TBox and how it can be used by the ABox.

Building in nonequational theories differs from building in equational theories. Because equational theories are defined by equalities or equivalences, a term or literal can always be replaced by an equal term or equivalent literal. This is not the case for nonequational theories that are expressed in terms of implication rather than equivalence. If we build in a nonequational theory of taxonomic information that includes $Man(x) \supset Person(x)$, we would expect to be able to infer $Person(John)$ from $Man(John)$, but not $\neg Person(Mary)$ from $\neg Man(Mary)$ (i.e., replacement of $Man(x)$ by $Person(x)$ is permitted only in those cases in which Man occurs unnegated).

Nonequational theories may express conditional inconsistency of a pair of literals. For instance, $(x < y) \wedge$

This research was supported by the Defense Advanced Research Projects Agency under Contract N00039-80-C-0575 with the Naval Electronic Systems Command. The views and conclusions contained in this document are those of the author and should not be interpreted as representative of the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the United States government.

$(y < z) \supset (x < z)$ expresses the fact that $y < z$ and $\neg(x < z)$ are inconsistent only in the presence of $x < y$.

Theory resolution is a set of complete procedures for building in nonequational theories using decision procedures as components of a more general deduction system. Two forms are described. Total theory resolution employs a decision procedure that can determine inconsistency of any set of clauses using predicates of the theory and is quite restricted as to what inferences it will make. Partial theory resolution is less restricted as to what inferences it will make but requires much less of the decision procedure—making it more feasible, for example, to use knowledge representation systems as the decision procedure.

We will give definitions and completeness proofs for the ground case of theory resolution and definitions for the general case. Completeness for the general case follows directly from ground case completeness. We consider only clausal resolution here, but these results should be easily extendable to nonclausal resolution [10,13,22].

II TOTAL THEORY RESOLUTION

In building in a theory T, we are interested in ascertaining whether a set of clauses S is T-inconsistent (i.e., whether $S \cup T$ is inconsistent). If we have a decision procedure for T that is capable of finding minimally T-inconsistent subsets of clauses from any set of clauses using only predicates in T, then it can be applied to S with all literals having predicates not in T removed to create an inference rule (total theory resolution) that derives clauses containing no occurrrences of predicates that are referred to in T.

A theorem justifying such a rule of inference follows. In it, P is the set of predicates in T, S corresponds to $S \cup T$ above, and T is some subset of S_P. The decision procedure for T must determine T-inconsistency of sets of clauses from $S_P - T$ and W_P.

Theorem. Let S be a set of ground clauses and P be a set of predicates. Let S_P be the set of all clauses of S containing only predicate symbols in P. Let $S_{\overline{P}}$ be the set of all clauses of S containing only predicate symbols not in P. Let W be $S - S_P - S_{\overline{P}}$. Let W_P be the list of clauses C_i formed by restricting each clause in W to just the predicates in P. Let $W_{\overline{P}}$ be the list of clauses D_i formed by restricting each clause in W to just the predicates not in P. $W = \{ C_i \vee D_i \mid 1 \leq i \leq n \}$. Let X be the set of all clauses of the form $D_{i_1} \vee \cdots \vee D_{i_m}$ where $C_{i_1}, \ldots, C_{i_m}$ are all the clauses of W_P in a minimally inconsistent set of clauses from S_P and W_P. Then S is inconsistent if and only if $S_{\overline{P}} \cup X$ is inconsistent.

Proof: *If part. This proves the soundness of the rule.* Assume $S_{\overline{P}} \cup X$ is inconsistent. For every element of X, *false* is a logical consequence of S_P and some $C_{i_1}, \ldots, C_{i_m}$ from W_P. Therefore $D_{i_1} \vee \cdots \vee D_{i_m}$ is a logical consequence of S_P and W (derived, for example, by imitating a ground resolution derivation of *false* from S_P and $C_{i_1}, \ldots, C_{i_m}$ using $C_{i_1} \vee D_{i_1}, \ldots, C_{i_m} \vee D_{i_m}$ instead of $C_{i_1}, \ldots, C_{i_m}$). Because $S_{\overline{P}} \subseteq S$ and every element of X is a logical consequence of $S_P \cup W$ and $S_P \cup W \subseteq S$, if $S_{\overline{P}} \cup X$ is inconsistent, then so is S.

Only if part. This proves the completness of the rule. Assume S is inconsistent. Then either S_P is inconsistent, $S_{\overline{P}}$ is inconsistent, or the inconsistency of S depends at least partially on W.

Case 1. S_P is inconsistent. Then *false* $\in X$ and $S_{\overline{P}} \cup X$ is inconsistent.

Case 2. $S_{\overline{P}}$ is inconsistent. Then $S_{\overline{P}} \cup X$ is inconsistent.

Case 3. S_P and $S_{\overline{P}}$ are consistent. Because they have disjoint sets of predicates, $S_P \cup S_{\overline{P}}$ is also consistent. Then there is a minimally inconsistent set of clauses $S'_P \cup S'_{\overline{P}} \cup W'$ such that $S'_P \subseteq S_P$, $S'_{\overline{P}} \subseteq S_{\overline{P}}$, and $\emptyset \subset W' \subseteq W$. By completeness of A-ordered resolution [17,20,5,9], there exists an A-ordered resolution refutation of this set with predicates in P preceding predicates not in P in the A-ordering. Included in the refutation is a set of clauses X' containing no predicates in P derived entirely from $S'_P \cup W'$. $S'_{\overline{P}} \cup X'$ is clearly inconsistent. When we look at the A-ordered derivations, it is apparent that each element of X' is of the form $D_{i_1} \vee \cdots \vee D_{i_m}$ derived from $S'_P \cup W'$ such that a subset of $S'_P \cup \{ C_{i_1}, \ldots, C_{i_m} \}$ is inconsistent where $\{ C_{i_j} \vee D_{i_j} \} \subseteq W'$. If this subset is minimally inconsistent then $D_{i_1} \vee \cdots \vee D_{i_m} \in X$. Otherwise $D_{i_1} \vee \cdots \vee D_{i_m}$ is still subsumed by (possibly identical to) an element of X. Because X contains each element of X' or an element that subsumes it, the inconsistency of $S_{\overline{P}} \cup X$ follows from the inconsistency of $S'_{\overline{P}} \cup X'$. ∎

Definition. Let $C_1, \ldots, C_m$ be nonempty clauses and $D_1, \ldots, D_m$ be clauses such that each $C_i \vee D_i$ is in S and every predicate in C_i is in theory T and no predicate in D_i is in theory T. Let $\sigma_{11}, \ldots, \sigma_{1n_1}, \ldots, \sigma_{m1}, \ldots, \sigma_{mn_m}$ be substitutions such that $\{ C_1\sigma_{11}, \ldots, C_1\sigma_{1n_1}, \ldots, C_m\sigma_{m1}, \ldots, C_m\sigma_{mn_m} \}$ is minimally T-inconsistent. Then $D_1\sigma_{11} \vee \cdots \vee D_1\sigma_{1n_1} \vee \cdots \vee D_m\sigma_{m1} \vee \cdots \vee D_m\sigma_{mn_m}$ is a *total theory resolvent* from S, using theory T.

Total theory resolution, plus ordinary resolution or some other semidecision procedure for first-order predicate calculus operating on clauses that do not contain predicates in T, is complete.

Example. Consider a theory of partial ordering *ORD* consisting of $\neg(x < x)$ and $(x < y) \wedge (y < z) \supset (x < z)$. A set of unit clauses in this theory is inconsistent if and only if it contains a chain of inequalities $t_1 < \cdots < t_n (n \geq 1)$ such that either $t_1 = t_n$ or $\neg(t_n < t_1)$ is one of the clauses. Total theory resolvents would include $F(b)$ from $(x < b) \vee$

$F(x)$ and $F(a) \vee G(c)$ from $(x < b) \vee F(x)$, $(b < y) \vee G(y)$, and either $c < a$ or $\neg(a < c)$.

Thus the types of reasoning that are employable in the decision procedure can be quite different from and more effective (in its domain) than resolution.

There are limitations to the use of total theory resolution. The requirement that the decision procedure for the theory be capable of determining inconsistency of any set of clauses using predicates in the theory is quite strict. Reasoning about sets of clauses is probably an unreasonable requirement for such purposes as using a knowledge representation system as a decision procedure for taxonomic information, since such systems are often weak in handling disjunction. This tends to limit total resolution's applicability to building in mathematical decision procedures that handle disjunction. Incomplete restrictions of total theory resolution could still be usefully employed. For example, it may be easy for a system to decide inconsistency of sets of single literals (unit clauses), as above for *ORD* and maybe also in the case of taxonomic reasoning (note that taxonomic hierarchies can be expressed in the monadic predicate calculus for which there exists a decision procedure that could possibly be used in complete or incomplete theory resolution). Total theory resolution could then be used to resolve on only one literal from each clause.

Some care must be taken in deciding what theory T to build in so that the decison procedure does not have to decide too much. The theory must be capable of deciding sets of clauses that are constructed by using any predicates appearing in T. Thus, if we try to use total theory resolution to build in the equality relation with equality substitutivity (i.e., $x = y \supset (P(\cdots x \cdots) \supset P(\cdots y \cdots))$ for each predicate P), the decision procedure will have to decide all of S.

There may be a large number of T-inconsistencies that do not result in useful T-resolvents. It would be a worthwhile refinement to monitor the finding of T-inconsistent sets of clauses to verify that the substitutions applied do not preclude future use of the T-resolvent. This is like applying a purity check in A-ordered resolution.

A-ordered resolution slightly resembles total theory resolution. It permits resolution operations only on the atoms of each clause that occur earliest in a fixed ordering of predicates (the A-ordering). The A-ordering could place predicates of T before all others. A-ordered resolution differs from total theory resolution in that it assumes resolution (or hyperresolution) is to be used as the inference operation. It is thus inflexible, since it does not permit T to be built in except by resolution. Furthermore, total theory resolution creates a resolvent only from inconsistent set of clauses using predicates of T. A-ordered resolution is not so restrictive.

There is probably a useful relationship to be discovered between total theory resolution and the work on combining decision procedures [14,19]. So far we have discussed only building in a single decision procedure, though the procedure could be repeated as long as the sets of predicates do not overlap. It is likely that we would want an extension of theory resolution that permits the sets of predicates to overlap at least in the case of the equality predicate. A difference between total theory resolution and the work on combining decision procedures is that the latter has been concerned primarily with decision procedures that do not have to instantiate their inputs, unlike our requirements for finding substitutions to make a set of clauses inconsistent.

III PARTIAL THEORY RESOLUTION

Partial theory resolution is a procedure for building in theories that requires a less complex decision procedure than total theory resolution; all it needs is a decision procedure that determines for any pair of literals a complete set of substitutions and conditions for the inconsistency of the literals.

Partial theory resolution will first be defined and proved complete for ground clauses. We will define a T-resolution operation that resolves on one or more literals from each input clause, like ordinary resolution without a separate factoring operation. We will then extend it to the general case, showing how T-resolvents can be computed by using only one literal at a time from each input clause.

There are two types of T-resolvents in partial theory resolution. If some set of literals of a clause is inconsistent with T, those literals can be removed from the clause to form a T-resolvent:

Definition. Let A be a nonempty ground clause and let C be a ground clause. Then C is a ***ground T-resolvent*** of $A \vee C$ if and only if $T \vdash \neg A$.

If, with T assumed, a set of literals of one clause is inconsistent with a set of literals of another clause under certain conditions, then T-resolvents can be formed as the disjunction of the other literals of the clauses and negated conditions for the inconsistency:

Definition. Let A and B be nonempty ground clauses and let C and D be ground clauses. Then $C \vee D \vee E$ where E is a ground clause is a ***ground T-resolvent*** of $A \vee C$ and $B \vee D$ if and only if $T \vdash \neg A \vee \neg B \vee E$ but not $T \vdash \neg A_i \vee E$ for any literal A_i in A or $T \vdash \neg B_j \vee E$ for any literal B_j in B. E is called the ***residue*** of matching A and B.

The residue E is a negated condition for the inconsistency of A and B because $T \vdash \neg A \vee \neg B \vee E$ is equivalent to $T \vdash \neg E \supset ((A \wedge B) \equiv \mathit{false})$. The restriction that neither $T \vdash \neg A_i \vee E$ nor $T \vdash \neg B_j \vee E$ assures us that all the literals of both of A and B are essential to the possible T-inconsistency of $A \wedge B$.. T-resolution includes ordinary resolution because it is always the case that $T \vdash \neg A \vee \neg B \vee E$ if B is $\neg A$.

The soundness of ground T-resolution is obvious. T-semantic trees will be used to prove its completeness.

Definition. Let S be a set of ground clauses with set of atoms $\{A_1, \ldots, A_k\}$. Then a ***semantic tree*** for S is a binary tree with height k such that for each node n with depth i $(0 \leq i < k)$, n has two child nodes n_1 and n_2 at level $i+1$, the arc to n_1 is labeled by the literal A_{i+1}, and the arc to n_2 is labeled by the literal $\neg A_{i+1}$.

Each node n in a semantic tree provides a partial (or, in the case of terminal nodes, total) interpretation I_n for the atoms of S, assigning ***true*** to each atom A that labels an arc on the path from the root to n and assigning *false* to each atom A where $\neg A$ labels an arc on the path from the root to n.

Definition. Node n in a semantic tree for S ***falsifies*** clause $C \in S$ if and only if C is *false* in interpretation I_n.

Definition. A ***T-semantic tree*** is a semantic tree from which all nodes representing T-inconsistent truth assignments are removed.

Definition. Node n in a T-semantic tree for S ***T-falsifies*** clause $C \in S$ if and only if C is *false* in interpretation I_n taking account of T.

Definition. Node n is a ***failure node*** in a T-semantic tree for set of ground clauses S if and only if n T-falsifies a clause in S and no ancestor of n is a failure node.

Definition. Node n is an ***inference node*** in a T-semantic tree for set of ground clauses S if and only if both of n's child nodes are failure nodes.

Note that although in a T-semantic tree each nonterminal node may have either one or two child nodes, an inference node will always have two. If node n has only a single child node n_1 and n_1 T-falsifies a clause (and thus might be a failure node), then n also T-falsifies the clause. Thus, a failure node will never be the single child node of its parent.

Theorem. A set of ground clauses S is T-inconsistent if and only if every branch of a semantic tree $\mathcal{T}$ for S contains a failure node. Either the root node of $\mathcal{T}$ is a failure node or there is at least one inference node in $\mathcal{T}$.

Theorem. Let S be a T-inconsistent set of ground clauses and let $\mathcal{T}$ be a T-semantic tree for S. Then if the root node of $\mathcal{T}$ is not a failure node, there is a T-inconsistent set of ground clauses S' derivable from S by ground T-resolution such that $\mathcal{T}$ is a semantic tree for S' but has fewer failure nodes for S' than for S.

Proof: Since S is T-inconsistent and the root node of $\mathcal{T}$ is not a failure node, there must be at least one inference node n in $\mathcal{T}$. Then n's child nodes, n_1 and n_2, are both failure nodes. Let the clause T-falsified at n_1 be $A \vee C$ where nonempty clause A consists of all the literals not already T-falsified at n. Let the clause T-falsified at n_2 be $B \vee D$ where nonempty clause B consists of all the literals not already T-falsified at n. Then n, or an ancestor of n if n is the single child node of its parent, is a failure node for $S' = S \cup \{C \vee D \vee E\}$ where E is a clause consisting of the negations of the literals labeling arcs above node n that were used in T-falsifying A and B. Because $T \vdash \neg E \wedge \neg A_i \supset \neg A$ and $T \vdash \neg E \wedge A_i \supset \neg B$ where A_i and $\neg A_i$ label the arcs to n_1 and n_2 respectively, $T \vdash \neg A \vee \neg B \vee E$ and $C \vee D \vee E$ is a ground clause T-resolvent of $A \vee C$ and $B \vee D$. $\mathcal{T}$ contains fewer failure nodes for S' than for S because n (or an ancestor of n) is a failure node instead of n_1 and n_2. ∎

Theorem. Ground clause T-resolution is complete.

Proof: Let S be a T-inconsistent set of clauses. Let $\mathcal{T}$ be a T-semantic tree for S. Then either the root node of $\mathcal{T}$ is a failure node, in which case the empty clause is derivable by ground T-resolution from the clause T-falsified at the root, or $\mathcal{T}$ contains an inference node, in which case completeness is assured by induction on the number of failure nodes, applying the previous theorem that uses ground T-resolution to add a clause that makes the inference node (or an ancestor of it) be a failure node. ∎

The ground T-resolution operation as defined above shares somewhat an undesirable feature of total theory resolution, i.e., that it demands too much of the decision procedure—in this case, requiring it to determine the possible inconsistency of two sets of literals (i.e., two clauses) instead of just two literals. This is easily remedied.

In the definition of ground T-resolvents with two parent clauses, let A be $A_1 \vee \cdots \vee A_m$ and let B be $B_1 \vee \cdots \vee B_n$. Then $T \vdash \neg A \vee \neg B \vee E$ is equivalent to all of $T \vdash \neg A_1 \vee \neg B_1 \vee E, \ldots,$ and $T \vdash \neg A_m \vee \neg B_n \vee E$ being true. Therefore, in computing T-resolvents, it is sufficient to determine possible T-inconsistency of single pairs of literals A_i and B_j with negated condition (residue) E_{ij} (i.e., $T \vdash \neg A_i \vee \neg B_j \vee E_{ij}$) and form T-resolvents of $A \vee C$ and $B \vee D$ as $C \vee D \vee E_{11} \vee \cdots \vee E_{mn}$. E_{ij} is a ***ground T-match*** of literals A_i and B_j. Note that this multiple pairwise matching is a substitute for a separate factoring operation.

We now extend T-matches and T-resolvents to the general (nonground) case.

Definition. Let A and B be two literals. Then $\langle E, \sigma \rangle$ where E is a clause and σ is a substitution is a ***T-match*** of A and B if and only if $T \vdash \neg A\sigma \vee \neg B\sigma \vee E$ but not $T \vdash \neg A\sigma \vee E$ or $T \vdash \neg B\sigma \vee E$.

A T-match of literals A and B specifies a substitution σ and condition $\neg E$ that make A and B be T-inconsistent.

We will give examples based on two theories:

- A taxonomic hierarchy theory TAX, including $Man(x) \supset Person(x)$.
- The partial-ordering theory ORD (defined previously).

Example. $\langle false, \{w \leftarrow John\}\rangle$ is a TAX-match of $Man(John)$ and $\neg Person(w)$.

Example. $\langle a < c, \emptyset\rangle$ is an ORD-match of $a < b$ and $b < c$. But $\langle \neg(c < x) \vee (a < x), \emptyset\rangle, \langle \neg(c < x) \vee \neg(x < y) \vee (a < y), \emptyset\rangle, \ldots$ are also ORD-matches of $a < b$ and $b < c$. The notion of minimal complete sets of T-matches is defined to exclude these additional T-matches.

Definition. Let $M = \{\langle E_1, \sigma_1\rangle, \ldots, \langle E_n, \sigma_n\rangle\}$ $(n \geq 0)$ be a set of T-matches of literals A and B. Then M is a *complete set of T-matches* of A and B if and only if for every T-match $\langle E, \sigma\rangle$ of A and B there is some T-match $\langle E_i, \sigma_i\rangle \in M$ and substitution θ such that $\sigma = \sigma_i\theta$ and $T \vdash E_i\theta \supset E$. M is a ***minimal complete set of T-matches*** of A and B if and only if M, but no proper subset of M, is a complete set of T-matches of A and B.

Example. $\{\langle a < c, \emptyset\rangle\}$ is a minimal complete set of ORD-matches of $a < b$ and $b < c$. ORD-matches of the form $\langle \neg(c < x) \vee (a < x), \emptyset\rangle, \ldots$ have the property that $ORD \vdash (a < c) \supset [\neg(c < x) \vee (c < a)], \ldots$, and are therefore not in the minimal complete set of ORD-matches.

As in the ground case, single-parent and double-parent T-resolution operations are defined:

Definition. Let A be a literal and $A \vee C$ be a clause and let σ be a substitution such that $T \vdash \neg A\sigma$. Then $C\sigma$ is a *T-resolvent* of $A \vee C$.

Example. $Positive(1)$ is an ORD-resolvent of $(x < 1) \vee Positive(x)$.

More than one T-inconsistent literal can be removed from a clause by performing single parent T-resolution repeatedly.

Definition. Let A and B be the nonempty clauses $A_1 \vee \cdots \vee A_m$ and $B_1 \vee \cdots \vee B_n$, let $A \vee C$ and $B \vee D$ be clauses, and let $\langle E_{ij}, \sigma_{ij}\rangle$ be T-matches of A_i and B_j. Then $C\sigma \vee D\sigma \vee E\sigma$ is a ***T-resolvent*** of $A \vee C$ and $B \vee D$ where σ is the most general combined substitution of $\sigma_{11}, \ldots, \sigma_{mn}$ and E is $E_{11} \vee \cdots \vee E_{mn}$.

Example. $\neg Robot(John)$ is a TAX-resolvent of $Man(John)$ and $\neg Person(w) \vee \neg Robot(w)$.

Example. $C(u) \vee D(u) \vee (a < c)$ is an ORD-resolvent of $(a < u) \vee C(u)$ and $(v < c) \vee D(v)$.

Further constraints on what T-resolvents can be inferred may be required for partial theory resolution to be really effective. For example, $a < b$ and $c < d$, which have no terms in common, have ORD-resolvents $\neg(b < c) \vee (a < d)$, $\neg(b < c) \vee (b < d)$, $\neg(b < c) \vee (a < c)$, $\neg(d < a) \vee (c < b)$, etc. If the first of these is actually used in a refutation, there must exist matches $b < c$ and $\neg(a < d)$ for its literals. It would be preferable to T-resolve these literals with $a < b$ and $c < d$ (e.g., T-resolve $a < b$ and $b < c$ deriving $a < c$, T-resolve $a < c$ and $c < d$ deriving $a < d$, and resolve that with $\neg(a < d)$) instead of directly T-resolving $a < b$ and $c < d$. We would impose the restriction that ORD-resolvents be derived only by resolving on pairs of literals that have a term in common.

There are two previous resolution refinements that resemble partial theory resolution: Z-resolution and U-generalized resolution.

Dixon's Z-resolution [6] is essentially partial theory resolution with the restriction that T must consist of a finite deductively closed set of 2-clauses (clauses with length 2). This restriction does not permit inclusion of assertions like $\neg Q(x) \vee Q(f(x))$, $\neg(x < x)$, or $(x < y) \wedge (y < z) \supset (x < z)$, but does permit efficient computation of T-resolvents (even allowing the possibility of compiling T to Lisp code and thence to machine code).

Harrison and Rubin's U-generalized resolution [8] is essentially partial theory resolution restricted to sets of clauses that have a unit or input refutation. They apply it to building in the equality relation, developing a procedure similar to Morris's E-resolution [12]. The restriction to sets of clauses having unit or input refutations eliminates the need for factoring and simplifies the procedure (only a single literal of each parent must be used to create a T-resolvent), but otherwise seriously limits its applicability. No effort was made in the definition of U-generalized resolution to limit T-resolution by using minimal complete sets of T-matches.

Partial theory resolution is a procedure with substantial generality and power. Thus, it is not surprising that many specialized reasoning procedures can be viewed as instances of partial theory resolution, perhaps with additional constraints governing which partial theory resolvents can be inferred:

Where T consists of the equality axioms, T-resolution operations include paramodulation [23] (e.g., $P(b) \vee C \vee D$ can be inferred from $P(a) \vee C$ and $a = b \vee D$) and E-resolution [12] (e.g., $\neg a = b \vee C \vee D$ can be inferred from $P(a) \vee C$ and $\neg P(b) \vee D$).

Where T consists of ordering axioms, including axioms that show how ordering is preserved (such as $(x < y) \supset (P(x) \supset P(y))$ and $(x < y) \supset (x + z < y + z)$), T-resolution operations include Manna and Waldinger's program-synthetic special relation substitution rule (e.g., $P(b) \vee C \vee D$ can be inferred from $P(a) \vee C$ and $(a < b) \vee D$) and relation matching rule [11] (e.g., $\neg(a < b) \vee C \vee D$ can be inferred from $P(a) \vee C$ and $\neg P(b) \vee D$), which are extensions of paramodulation and E-resolution. *T*-resolution with ordering axioms is also similar to Slagle and Norton's reasoning about partial ordering [21]. Bledsoe and Hines's variable elimination [1] is a very refined method for reasoning about inequalities that can be viewed partly as partial theory resolution for inequality with added constraints on partial theory resolution operations. The *ORD*-resolvent $a < c$ of $a < b$ and $b < c$ is a variable-elimination-procedure chain resolvent only if b is a shielding term (nonground term headed by an uninterpreted function symbol). The variable elimination rule allows inferring *ORD*-resolvent $(a < b) \vee C$ from clause $(a < x) \vee (x < b) \vee C$ only if x does not occur in a, b, or C. The variable elimination rule more generally allows replacement of multiple literals $a_i < x$ and $x < b_j$ in a clause by literals $a_i < b_j$. This result is obtainable by partial theory resolution if we include the axiom $\neg(x < min(x,y))$ and a rule to transform $min(a_{i_1}, a_{i_2}) < b_j$ to $(a_{i_1} < b_j) \vee (a_{i_2} < b_j)$.

IV CONCLUSION

Theory resolution is a set of complete procedures for incorporating decision procedures into resolution theorem proving in first-order predicate calculus. Theory resolution can greatly decrease the length of refutations and the size of the search space, for example, by hiding lengthy taxonomic derivations in single *TAX*-matches and by restricting use of ordering axioms in *ORD*-matches. Total theory resolution can be used when there exists a decision procedure for the theory that is capable of determining inconsistency of any set of clauses using predicates of the theory. This may be a realistic requirement in some mathematical theorem proving. For example, a decision procedure for Presburger arithmetic (integer addition and inequality) might be adapted to meet the requirements for total theory resolution.

Partial theory resolution requires much less of the decision procedure. It requires only that conditions and substitutions for inconsistency of a single pair of literals be determinable by the decision procedure for the theory. This makes it feasible, for example, to consider use of a knowledge representation system as the decision procedure for taxonomic information. Partial theory resolution is also a generalization of several other approaches to building in nonequational theories.

We are implementing and testing forms of theory resolution in the deduction system component of the KLAUS natural-language-understanding system [7,22].

ACKNOWLEDGMENTS

The author would like to thank Richard Fikes, Mabry Tyson, and Richard Waldinger for their helpful comments on an earlier draft of this paper.

REFERENCES

[1] Bledsoe, W.W. and L.M. Hines. Variable elimination and chaining in a resolution-based prover for inequalities. *Proc. 5th Conf. on Automated Deduction,* Les Arcs, France, July 1980, 70–87.

[2] Brachman, R.J., R.E. Fikes, and H.J. Levesque. KRYPTON: a functional approach to knowledge representation. To appear in *IEEE Computer 16,* 9 (September 1983).

[3] Brachman, R.J. and H.J. Levesque. Competence in knowledge representation. *Proc. AAAI-82 Nat. Conf. on Artificial Intelligence,* Pittsburgh, Pennsylvania, August 1982, 189–192.

[4] Brachman, R.J. and B.C. Smith (eds). Special Issue on Knowledge Representation. *SIGART Newsletter 70* (February 1980).

[5] Chang, C.L. and R.C.T. Lee. *Symbolic Logic and Mechanical Theorem Proving.* Academic Press, New York, New York, 1973.

[6] Dixon, J.K. Z-resolution: theorem-proving with compiled axioms. *J. ACM 20,* 1 (January 1973), 127–147.

[7] Haas, N. and G.G. Hendrix. An approach to acquiring and applying knowledge. *Proc. AAAI-80 Nat. Conf. on Artificial Intelligence,* Stanford, California, August 1980, 235–239.

[8] Harrison, M.C. and N. Rubin. Another generalization of resolution. *J. ACM 25,* 3 (July 1978), 341–351.

[9] Loveland, D.W. *Automated Theorem Proving: A Logical Basis.* North-Holland, Amsterdam, The Netherlands, 1978.

[10] Manna, Z. and R. Waldinger. A deductive approach to program synthesis. *ACM Transactions on Programming Languages and Systems 2,* 1 (January 1980), 90–121.

[11] Manna, Z. and R. Waldinger. Special relations in program-synthetic deduction (a summary). To appear in *J. ACM.*

[12] Morris, J.B. E-resolution: extension of resolution to include the equality relation. *Proc. Int. Joint Conf. on Artificial Intelligence,* Washington, D.C., May 1969, 287–294.

[13] Murray, N.V. Completely non-clausal theorem proving. *Artificial Intelligence 18,* 1 (January 1982), 67–85.

[14] Nelson, G. and D.C. Oppen. Simplification by cooperating decision procedures. *ACM Trans. Program. Lang. Syst. 1,* 2 (October 1979), 245–257.

[15] Plotkin, G.D. Building-in equational theories. In Meltzer, B. and Michie, D. (eds.), *Machine Intelligence 7,* Halsted Press, 1972, pp. 73–90.

[16] Raulefs, P., J. Siekmann, P. Szabo, and E. Unvericht. A short survey on the state of the art in matching and unification problems. *SIGSAM Bulletin 13,* 2 (May 1979), 14–20.

[17] Reynolds, J. Unpublished seminar notes. Stanford University, Palo Alto, California, Fall 1965.

[18] Rich, C. Knowledge representation languages and predicate calculus: how to have your cake and eat it too. *Proc. AAAI-82 Nat. Conf. on Artificial Intelligence,* Pittsburgh, Pennsylvania, August 1982, 193–196.

[19] Shostak, R.E. Deciding combinations of theories. *Proc. Sixth Conf. on Automated Deduction,* New York, New York, June 1982, 209–222.

[20] Slagle, J.R. Automatic theorem proving with renamable and semantic resolution. *J. ACM 14,* 4 (October 1967), 687–697.

[21] Slagle, J.R. and L.M. Norton. Experiments with an automatic theorem-prover having partial ordering inference rules. *Communications of the ACM 16,* 11 (November 1973), 682–688.

[22] Stickel, M.E. A nonclausal connection-graph resolution theorem-proving program. *Proc. AAAI-82 Nat. Conf. on Artificial Intelligence,* Pittsburgh, Pennsylvania, August 1982, 229–233.

[23] Wos, L. and G.A. Robinson. Paramodulation and set of support. *Proc. Symp. on Automatic Demonstration,* Versailles, France, 1968, *Lecture Notes in Mathematics 125,* Springer-Verlag, Berlin, West Germany (1970), 276–310.