Automated Deduction by Theory Resolution

Mark E. Stickel

Artificial Intelligence Center
SRI International
and
Center for the Study of Language and Information
Stanford University

Abstract

Theory resolution constitutes a set of complete procedures for incorporating theories into a resolution theorem-proving program, thereby making it unnecessary to resolve directly upon axioms of the theory. This can greatly reduce the length of proofs and the size of the search space. Theory resolution effects a beneficial division of labor, improving the performance of the theorem prover and increasing the applicability of the specialized reasoning procedures. Applications include the building in of both mathematical and special decision procedures, e.g., for the taxonomic information furnished by a knowledge representation system. Theory resolution is a generalization of numerous previously known resolution refinements. Its power is demonstrated by comparing solutions of "Schubert's Steamroller" challenge problem with and without building in axioms through theory resolution.[1]

1 Introduction

Incorporating a theory into derived inference rules so that its axioms are never resolved upon has enormous potential for reducing the size of the exponential search space commonly encountered in resolution theorem proving. Theory resolution is a method of incorporating specialized reasoning procedures in a resolution theorem prover so that the reasoning task will be effectively divided into two parts: special cases, such as reasoning about inequalities or about taxonomic information, are handled efficiently by specialized reasoning procedures, while more general reasoning is handled by resolution. The connection between the two reasoning components is made by having the resolution procedure resolve on sets of literals whose conjunction is determined to be unsatisfiable by the specialized reasoning procedure. The objective of research on theory resolution is the conceptual design of deduction systems that combine deductive specialists within the common framework of a resolution theorem prover.

Concern has often been expressed about the ineffectiveness of applying resolution theorem proving to problems in artificial intelligence. Theory resolution is designed to partly address this concern by providing a means for incorporating specialized reasoning procedures in a resolution theorem prover. The division of labor achieved in the reasoning process by theory resolution is intended to produce the dual advantages of improving the theorem prover's performance by the use of more efficient reasoning procedures for special cases and of increasing the range of applicat ion of the specialized reasoning procedures by including them in a more general reasoning system.

Past criticisms of resolution can often be characterized by their pejorative use of the terms *uniform* and *syntactic.* Theory resolution meets these objections head-on. In theory resolution, a specialized reasoning procedure may be substituted for ordinary syntactic unification to determine unsatisfiability of sets of literals. Because the implementation of this specialized reasoning procedure is unspecified—to the theorem prover it is a "black box" with prescribed behavior, namely, able to determine unsatisfiability in the theory it implements—the resulting system is nonuniform because reasoning within the theory is performed by the specialized reasoning procedure, while reasoning outside the theory is performed by resolution. Theory resolution can also be regarded as being not wholly syntactic, since the conditions for resolving on a set of literals are no longer based on their being made syntactically identical, but rather on their being unsatisfiable in a theory, and thus resolvability is partly semantic.

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 $(x < y)$ A $(y < z)$ D $(J < x)$ despite the obvious fact that a refutation based on just these two formulas is impossible. 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 can be found (total theory resolution).

An important form of reasoning in artificial intelligence applications embodied in knowledge representation systems is reasoning about taxonomic information and property inheritance. One of our goals is to be able to take advantage of the efficient reasoning provided by a knowledge representation system by using it as a taxonomy decision procedure in a larger deduction system. KRYPTON [4,13] 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.

2 Theory Resolution

We will now define the theory resolution operation and discuss various useful restrictions on theory resolution. We will limit our

[1]This research was supported by the Defense Advanced Research Projects Agency under Contract N00039-84-K-0078 with the Naval Electronic Systems Command and was also made possible in part by a gift from the System Development Foundation. 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. Approved for public release. Distribution unlimited.

discussion to the variable-free "ground" case of theory resolution, since lifting to the general case is straightforward.

We will assume the standard definitions of a *term*, an *atomic formula (atom)*, and a *literal*. We will consider a *clause* to be a disjunction of $n \geq 0$ literals. If $n = 0$, the clause is the *empty clause* $\Box$. If $n = 1$, the clause is a *unit clause*. The disjunction connective $\vee$ is assumed to be associative, commutative, and idempotent. The empty clause $\Box$ is the identity element for $\vee$. We will generally make no distinction between a unit clause and the single literal of which it is composed.

We will assume the standard definitions of an *interpretation*, an interpretation *satisfying* or *falsifying* a formula or set of formulas, and a formula or set of formulas being *satisfiable* or *unsatisfiable*.

Any satisfiable set of formulas that we wish to incorporate into the inference process can be regarded as a *theory*.

Definition 1 A *T-interpretation* is an interpretation that satisfies theory T.

For example, in a theory of partial ordering ORD consisting of $\neg(x < x)$ and $(x < y) \wedge (y < z) \supset (x < z)$, the predicate $<$ cannot be interpreted so that $a < a$ has value *true* or $a < c$ has value *false* if $a < b$ and $b < c$ both have value *true*. In a taxonomic theory TAX including $Boy(x) \supset Person(x)$, $Boy(John)$ cannot have value *true* while $Person(John)$ has value *false*.

Definition 2 A set of clauses S is *T-unsatisfiable* iff no T-interpretation satisfies S. S is *minimally T-unsatisfiable* iff S, but no proper subset of S, is T-unsatisfiable.

Definition 3 Let $C_1, \ldots, C_m$ $(m \geq 1)$ be a set of nonempty clauses, let each C_i be decomposed as $K_i \vee L_i$ where K_i is a nonempty clause, and let $R_1, \ldots, R_n$ $(n \geq 0)$ be unit clauses. Suppose the set of clauses $K_1, \ldots, K_m, R_1, \ldots, R_n$ is T-unsatisfiable. Then the clause $L_1 \vee \cdots \vee L_m \vee \neg R_1 \vee \cdots \vee \neg R_n$ is a *theory resolvent using theory T (T-resolvent)* of $C_1, \ldots, C_m$. The theory resolvent is called an *m-ary* theory resolvent. It is a *total* theory resolvent iff $n = 0$; otherwise it is *partial*. $K_1, \ldots, K_m$ is called the *key* of the theory resolution operation. For partial theory resolvents, $R_1, \ldots, R_n$ is a set of *conditions* for the T-unsatisfiability of the key. The negation $\neg R_1 \vee \cdots \vee \neg R_n$ of the conjunction of the conditions is called the *residue* of the theory resolution operation. It is a *narrow* theory resolvent iff each K_i is a unit clause; otherwise it is *wide*.

The partial theory resolution procedure permits total as well as partial theory resolution operations. Similarly, the wide theory resolution procedure permits narrow as well as wide theory resolution operations.

Example 4 A set of unit clauses is unsatisfiable in the theory of partial ordering ORD iff it contains a chain of inequalities $t_1 < \cdots < t_n (n \geq 2)$ such that either t_1 is the same as t_n or $\neg(t_1 < t_n)$ is also one of the clauses. P is a unary total narrow ORD-resolvent of $(a < a) \vee P$. $P \vee Q$ is a binary total narrow ORD-resolvent of $(a < b) \vee P$ and $(b < a) \vee Q$. $P \vee Q \vee R \vee S$ is a 4-ary total narrow ORD-resolvent of $(a < b) \vee P$, $(b < c) \vee Q$, $(c < d) \vee R$, and $\neg(a < d) \vee S$. This can also be derived incrementally through partial narrow ORD-resolution, i.e., by resolving $(a < b) \vee P$ and $(b < c) \vee Q$ to obtain $(a < c) \vee P \vee Q$, resolving that with $(c < d) \vee R$ to obtain $(a < d) \vee P \vee Q \vee R$, and resolving that with $\neg(a < d) \vee S$ to obtain $P \vee Q \vee R \vee S$.

Example 5 Suppose the taxonomic theory TAX includes a definition for fatherhood $Father(x) \equiv [Man(x) \wedge \exists y Child(x, y)]$. Then $Father(Fred)$ is a partial wide theory resolvent of $Child(Fred, Pat) \vee Child(Fred, Sandy)$ and $Man(Fred)$. Also, $\Box$ is a total wide theory resolvent of $Child(Fred, Pat) \vee Child(Fred, Sandy)$, $Man(Fred)$, and $\neg Father(Fred)$.

We will explore some possible restrictions on the definition of theory resolution that make it practical to apply while preserving completeness.

In narrow theory resolution, only T-unsatisfiability of sets of literals, not clauses, must be decided. Total and partial narrow theory resolution are both possible. In total narrow theory resolution, the resolved-upon literals (the key) must be T-unsatisfiable. In partial narrow theory resolution, the key must be T-unsatisfiable only under some conditions. The negated conditions are used as the residue in the formation of the resolvent.

We do not want to require the derivation of all partial narrow theory resolvents permitted by the definition. This would result in the derivation of obviously unnecessary resolvents. For example, we could resolve $(a < b) \vee P$ and $(c < d) \vee R$, since, under some conditions such as $(b < c) \wedge (d < a)$, $a < b$ and $c < d$ are T-unsatisfiable. If we permit inferences from $a < b$ and $c < d$, which have no terms in common, theory resolution would not be very useful. If resolving $a < b$ and $c < d$ were to actually lead to a refutation—i.e., conditions for their T-unsatisfiability do hold—then some of these conditions, e.g., $(b < c) \wedge (d < a)$, must have arguments in common with $a < b$ and $c < d$. We should restrict partial theory resolution to cases in which the literals are suitably related.

To justify such pragmatically necessary restrictions on theory resolution, we offer the following criterion for the selection of key sets of literals that provides a sufficient condition for the completeness of partial narrow theory resolution.

In essence, the key selection criterion requires that every T-unsatisfiable set of literals have one or more subset key sets of literals that can be T-resolved. For example, in theory ORD, in refuting sets of positive inequality literals, we might select only pairs of literals matching $x < y$ and $y < z$ as key sets of literals. Thus, in refuting the set $\{a < b, b < c, c < d, d < a\}$, we would be permitted, for example, to resolve upon $a < b$ and $b < c$, but not $a < b$ and $c < d$. Key sets of literals have one or more residues associated with them such that every minimally T-unsatisfiable set includes a key with a residue that can be refuted by resolving away the literals in the residue. With literals matching $x < y$ and $y < z$ selected, it is sufficient to derive T-resolvents with residue $x < z$. For example, $a < b$ and $b < c$ can be T-resolved with $a < c$ as the result. This can then be resolved with $c < d$ to derive $a < d$ that can be resolved with $d < a$ to derive $\Box$.

Key selection criterion.

- For any minimally T-unsatisfiable set of literals S, there is at least one key set of literals K such that $K \subseteq S$. K has at least two literals (one literal if S has only one literal). Each K is recognizable by the decision procedure for T and will comprise the key for possible theory resolution operations, if clauses containing the key literals are present.
- For any such key set of literals K, there is at least one, possibly empty, residue set of literals R such that $K \cup \neg R$ is minimally T-unsatisfiable, where $\neg R$ denotes the set

$\{\neg R_1, \ldots, \neg R_n\}$ when $R = \{R_1, \ldots, R_n\}$. Each $\neg R$ is a set of conditions for the T-unsatisfiability of key set K. Each R is computed from K by the decision procedure for T and is used as a residue for theory resolution operations that resolve on key K.

- It must be the case that, for some key set of literals K and associated residue set of literals R, $(S - K) \cup \{\vee R\}$ is minimally T-unsatisfiable, where $\vee R$ denotes the clause $R_1 \vee \cdots \vee R_n$ when $R = \{R_1, \ldots, R_n\}$. This ensures that key selection and residue computation will be sufficient for completeness—any T-unsatisfiable set of literals S has a T-resolvent using a key $K \subseteq S$ and residue R computed from K such that the T-resolvent is contradicted by the remaining literals $S - K$.

In total narrow theory resolution, we uniformly take the key K to be the entire minimally T-unsatisfiable set of literals S. The residue R is always empty.

In partial narrow theory resolution, we will try to minimize the number of residue sets of literals. Thus, for $K = \{a < b, b < c\}$ we might have residues $R_1 = \{a < c\}$, $R_2 = \{\neg(c < x_1), \neg(x_1 < a)\}$, $R_3 = \{\neg(c < x_1), \neg(x_1 < x_2), \neg(x_2 < a)\}$, etc. However, only R_1 need be used, since, in the theory T, R_1 implies every other R_i. R_1 can be regarded as the strongest consequence of $a < b$ and $b < c$ in theory T.

The following theorem proves the completeness of narrow theory resolution with arbitrary selection of key sets of literals satisfying the key selection criterion.

Theorem 6 *Let S be a T-unsatisfiable set of clauses. Then there is a refutation of S (derivation of $\Box$ from S) using partial narrow theory resolution with theory T for arbitrary selection of key sets satisfying the key selection criterion.*

Proof: If $\Box \in S$, then S is trivially refuted.

Otherwise we will prove the theorem by induction on complexity measure $c(S)$, where $c(S) = (|S|, k(S))$, where $|S|$ is the number of clauses in S and $k(S)$ is the *excess literal parameter* [1]. The excess literal parameter is defined to be the number of literals (i.e., literal occurrences) in S minus $|S|$. The ordering of $c(S)$ is defined by $c(S_1) \prec c(S_2)$ iff $|S_1| < |S_2|$, or $|S_1| = |S_2|$ and $k(S_1) < k(S_2)$.

Case $c(S) = (m, 0)$. Every clause must be a unit clause. Because S is T-unsatisfiable, it must include a minimally T-unsatisfiable subset S'.

> *Subcase* $|S'| \leq 2$. By the key selection criterion, S' must be selected as a key. The empty clause $\Box$ is derivable in a single unary or binary T-resolution step from S' and hence from S.
>
> *Subcase* $|S'| > 2$. By the key selection criterion, there exists a key $K \subsetneq S'$ with $|K| \geq 2$ and (possibly empty) residue R such that $S'' = (S' - K) \cup \{\vee R\}$ is minimally T-unsatisfiable. $c(S'') \prec c(S') \preceq c(S)$. Thus, by the induction hypothesis, $\Box$ is derivable from S''. Since $\vee R$ is a T-resolvent of $K \subsetneq S$, $\Box$ is derivable from S.

Case $c(S) = (m, n)$, $n > 0$. Select a nonunit clause $C \in S$. Decompose C into unit clause A and clause B, i.e., $C = A \vee B$. Because S is T-unsatisfiable, both $S_A = (S - \{C\}) \cup \{A\}$ and $S_B = (S - \{C\}) \cup \{B\}$ are T-unsatisfiable. Both $c(S_A) \prec c(S)$ and $c(S_B) \prec c(S)$. Thus, by the induction hypothesis, there must exist derivations of $\Box$ from each of S_A and S_B.

Imitate the derivation of $\Box$ from S_B, using C instead of B. The result will be a derivation of either $\Box$ or A from S. In the latter case, extend the derivation of A from S to a derivation of $\Box$ from S by appending the derivation of $\Box$ from S_A. ∎

Corollary 7 *Let S be a T-unsatisfiable set of clauses. Then there is a refutation of S (derivation of $\Box$ from S) using total narrow theory resolution with theory T.*

Although the theorem proves completeness of narrow theory resolution, its proof does not preclude the need for tautologies in a refutation. Indeed, it is the case that tautologies may have to be retained for a refutation to be found.

Example 8 Let T be the theory in which P, Q, and R are all equivalent. Let S be $\{P \vee Q \vee R, \neg P \vee \neg Q \vee \neg R\}$. There is a single-step wide T-resolution refutation of S. However, although there do exist refutations of S by narrow T-resolution, all require retention of tautologies, since all narrow T-resolvents of $P \vee Q \vee R$ and $\neg P \vee \neg Q \vee \neg R$ are tautologies.

Finally, note that heuristic restrictions of theory resolution (such as discarding all tautologies, not recognizing all cases of T-unsatisfiability, or not computing all residues), though incomplete, may be very useful in practice.

3 Examples of Theory Resolution

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 theory resolution, perhaps with additional constraints governing which theory resolvents can be inferred. We believe that the success of these specialized reasoning procedures helps to validate the concept of theory resolution.

First of all, we should note that there is a relationship between theory resolution and **hyperresolution**. Although further constraints (e.g., on the polarity of the literals) are often prescribed, the essence of hyperresolution is the derivation of $L_1 \vee \cdots \vee L_m \vee R$ from the *electron* clauses $K_i \vee L_i$, where K_i is a literal and L_i is a [possibly empty] clause and the *nucleus* clause $\neg K_1 \vee \cdots \vee \neg K_m \vee R$, where R is a [possibly empty] clause. This corresponds to a theory resolution operation using theory T, where $\neg K_1 \vee \cdots \vee \neg K_m \vee R$ is a consequence of T, $K_1, \ldots, K_m$ is the key set of literals, and R is the residue.

Theory resolution is also related to **procedural attachment**, whereby expressions are "evaluated" to produce new expressions. Ordinary procedural attachment can be regarded as unary theory resolution. Theory resolution in general can be considered as an extension of the notion of procedural attachment to sets of literals. Where ordinary procedural attachment permits the replacement of $2 < 3$ by *true*, theory resolution, in effect, can attach a procedure to the $<$ relation that permits derivation of $a < c$ from $a < b$ and $b < c$.

Two previous refinements of resolution that resemble partial theory resolution are Z-resolution and U-generalized resolution.

Dixon's **Z-resolution** [5] is essentially binary total narrow 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). Z-factoring and Z-subsumption operations are also defined.

Harrison and Rubin's U-generalized resolution [7] is essentially binary partial narrow theory resolution applied 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 [10]. The restriction to sets of clauses having unit or input refutations eliminates the need for factoring and simplifies the procedure, but otherwise limits its applicability. No effort was made in the definition of U-generalized resolution to limit the applicability of T-resolution to reasonable cases (e.g., formation of an ORD-resolvent of $a < b$ and $c < d$ is permitted by the definition).

The linked inference principle by Wos et al. [22] is related to theory resolution in concept and purpose. The linked inference principle is a somewhat more conservative extension of resolution than theory resolution, since it stipulates that the theory will be built in by means of clauses designated as linking clauses. Theory resolution, on the other hand, allows the theory to be incorporated as a "black box" that determines T-unsatisfiability questions in an unspecified manner. This facilitates the use of other systems, which do not rely upon resolution or clause representation, to build in theories. Nevertheless, many instances of theory resolution can be usefully implemented in the manner of the linked inference principle. Since the implementation proposal for the linked inference principle is more concrete, Wos et al. have expended comparatively more effort in determining how inference using the linked inference principle is to be controlled, including defining linked variants of resolution refinements such as unit-resulting resolution and hyperresolution.

We have already suggested the importance of theory resolution for taxonomlc reasoning. This is being explored in the KRYPTON knowledge representation system. Figure 1 contains a nearly verbatim transcription of a proof using KRYPTON-style reasoning. The problem is to prove that, if Chris has no sons and no daughters, then Chris has no children.

The terminological information used in this problem through theory resolution includes the statements that boys are persons whose sex is male; girls are persons whose sex is female; "no-sons* are persons all of whose children are girls; "no-daughters" are persons all of whose children are boys. Relevant portions of this information are included in Formulas 1-6, which are used to define what theory resolution operations are possible. If complements of the first two atoms of each formula can be found, they can be resolved upon, and the remaining part of the formula, if any, would be derived as the residue. Thus, Formula 1 expresses the unsatisfiability of *Boy(John)* and *-Peraon(John).* Formula 6 permits the derivation of *Girl(Sandy)* from *NoSon(Mary)* and *Child(Mory, Sandy).* These formulas behave similarly to linking clauses in linked inference [22].

The assertional information used in this problem includes the information that every person has a sex; males and females are disjoint; Chris has no sons and no daughters. From these facts, and the built in terminological information, a refutation is completed starting with the negation of the desired conclusion that Chris has no children, *sk1* and *sk2* are Skolem functions.

The following table compares the statistics for proofs com-

2-ary rule	1.	$Boy(x) \supset Person(x)$
2-ary rule	2.	$Boy(x) \wedge Sex(x,y) \supset Male(y)$
2-ary rule (not used)	3.	$Girl(x) \supset Person(x)$
2-ary rule	4.	$Girl(x) \wedge Sex(x,y) \supset Female(y)$
2-ary rule	5.	$NoSon(x) \wedge Child(x,y) \supset Girl(y)$
2-ary rule	6.	$NoDaughter(x) \wedge Child(x,y) \supset Boy(y)$
	7.	$Person(x) \supset Sex(x, sk1(x))$
	8.	$Male(x) \equiv \neg Female(x)$
	9.	$NoSon(Chris)$
	10.	$NoDaughter(Chris)$
negated conclusion	11.	$Child(Chris, sk2)$
resolve 11&9 using 5	12.	$Girl(sk2)$
resolve 11&10 using 6	13.	$Boy(sk2)$
resolve 13&7 using 1	14.	$Sex(sk2, sk1(sk2))$
resolve 13&14 using 2	15.	$Male(sk1(sk2))$
resolve 12&14 using 4	16.	$Female(sk1(sk2))$
resolve 16&8&15	20.	$\square$

Figure 1: KRYPTON-style Proof

pleted with and without Formulas 1–6 built in through theory resolution. The proof strategies used and meaning of the statistics are essentially the same as described in Section 4

Built In Axioms	Input Wffs	Der. Wffs	Ret. Wffs	Suc. Unify	Time (sec.)	Proof Length
none	11	10	20	33	1.0	9
1–6	5	9	11	24	0.5	6

There is a noticeable improvement resulting from using theory resolution, but because the problem is so small, the difference is not large. Harder problems (like the one in Section 4) can be used to demonstrate much greater improvement.

Theory resolution for taxonomic reasoning also incorporates many elements of reasoning in a many-sorted logic. For example, in Walther's ∑RP-calculus (many-sorted resolution and paramodulation) [17,19], sort declarations, subsort relationships, and sort restrictions on clauses are all incorporated into the unification procedure, and eliminated from the clauses in the statement of a problem. Thus, the ∑RP unification procedure implements a theory of sort information.

4 Experimental Results

Although the relationship of theory resolution to many other extensions of resolution and experience with numerous small examples support the practical value of theory resolution, we will not elaborate on these, but will rather bolster our claim with an examination of experimental results for "Schubert's Steamroller* challenge problem.

Schubert's steamroller problem (annotated with formula numbers) is

> (1-5) Wolves, foxes, birds, caterpillars, and snails are animals, and (7-11) there are some of each of them. Also (12) there are some grains, and (6) grains are plants. (13) Every animal either likes to eat all plants or all animals much smaller than itself that like to

eat some plants. (14) Caterpillars and snails are much smaller than birds, which are much smaller than foxes, which in turn are much smaller than wolves. (15) Wolves do not like to eat foxes or grains, while (16) birds like to eat caterpillars but (15) not snails. (17) Caterpillars and snails like to eat some plants. Therefore (18) there is an animal that likes to eat a grain-eating animal.

We present statistics on several solutions of Schubert's steamroller problem found by our theorem prover [15]. The first is a proof that does not use theory resolution; the second is a proof using theory resolution to implement the taxonomic information in the problem (Formulas 1-6); the remaining proofs show the results of using theory resolution to build in each of Formulas 14-17 successively.

The same strategy was used for all of the proofs. Nonclausal connection-graph resolution was the principal inference rule. Factoring was not employed. Pure, variant, and tautologous formulas were eliminated. Single literal formulas were used for both forward and backward demodulation.

Heuristic search, guided by a simple weighted function of the deduction level of the parents and the expected size of the resolvent, was used to decide which inference operation should be performed next. The set of support strategy (with only Formula 18 supported) and an ordering strategy that designated which atoms in a formula could be resolved upon were used to limit the number of alternative inference operations.

In using theory resolution, connection graph links were created from key sets of literals in the theory being incorporated. Formulas 1-6 and 17 were implemented by binary total narrow theory resolution links and Formulas 14-16 were implemented by 3-ary total narrow theory resolution links. For example, *Wolf(t)* and *->Animal(t)* could be linked, and *Bird*(t_1), *Snail* (t_2), and likes to eat $(t_1,t2)$ could all be linked. Theory resolution was also used in demodulation—e.g., *Wolf(t)* could be used to demodulate *Animal(t)* to *true.*

Following are the statistics for the various solutions of Schubert's steamroller problem. Included in the statistics are the number of formulas inputted to the theorem prover, the number of formulas derived in the course of searching for a proof, the number of inputted and derived formulas still present when a proof was found, the number of successful unification attempts during the search for a proof (including unification during link inheritance), the time required for the proof (on a Symbolics 3600 personal LISP machine), and the length of the proof in resolution steps.

Built In Axioms	Input Wffs	Der. Wffs	Ret. Wffs	Suc. Unify	Time (sec.)	Proof Length
none	18	2,717	595	216,987	2:53	59
1–6	12	889	246	44,928	0:20	37
1–6,14	11	408	68	5,018	0:01.3	32
1–6,14–15	10	320	63	4,555	0:01.1	32
1–6,14–16	9	212	57	3,068	0:00.7	32
1–6,14–17	8	262	24	7,711	0:01.6	24
MKRP	27	60	83		0:04.4	55
+ΣRP	12	10	13	48	0:00.2	9
ITP					0:06	

Also included in the table are statistics we know for solutions of Schubert's steamroller problem by other systems. Unfortunately, use of slightly different axiomatizations, e.g., whether "grain-eating animal" is interpreted as an animal that eats *some* grain (our work, see [16]) or *every* grain [18], makes statistics for these different solutions not strictly comparable. We will publish a more detailed comparison of solutions later.

The MKRP solution was done by Walther [20] using the Markgraf Karl Refutation Procedure [3]. This proof relied heavily on the MKRP TERMINATOR module [2], which is essentially a very fast procedure for finding unit refutations. A superior proof by Walther [18] used his ∑RP calculus [17,19] in the MKRP system to perform many-sorted resolution on a much reduced set of clauses. This proof also used the TERMINATOR module, but, given the reduction in the number of clauses and literals made possible by using many-sorted resolution and its restrictions on unification, here its use was not essential to finding a solution with reasonable effort. MKRP is written in INTERLISP and was run on a Siemens 7760 computer.

Our first theory resolution proof, in which only the taxonomic information of Formulas 16 is incorporated, has some similarity to a many-sorted resolution proof. In the MKRP ∑RP proof, *Wolf, Fox, Bird, Caterpillar,* and *Snail* were declared to be subsorts of sort *Animal and Gram* was declared to be a subsort of sort *Plant.* The unification algorithm was restricted so that a variable can be unified with a term if and only if the term is a subsort of or equals the sort of the variable. For building in just this taxonomic information, many-sorted resolution is stronger than this particular instance of theory resolution. Although theory resolution handles the sort literals more effectively than ordinary resolution, many-sorted resolution dispenses with them entirely. Also, many-sorted resolution is used to build in the sort information for Skolem constants and functions so that, in Schubert's steamroller problem, Formulas 7-12 are supplanted by type declarations.

The ITP solution was found by the automated reasoning system ITP (written in PASCAL) developed at Argonne National Laboratory [9]. This solution used qualified hyperresolution [8,21] and was completed in about six minutes on a VAX 11/780 computer [12]. Like the theory resolution and MKRP ∑RP solutions, this solution treated the taxonomic sort information in the problem specially. In qualified hyperresolution, some literals in a clause can be designated as qualifier literals that contain "conditions of definition" for terms appearing in the clause. Qualifier literals are ignored during much of the inference process e.g., a clause consisting of a single nonqualifier literal and some qualifier literals is handled as if it were a unit clause with the conditions imposed by the qualifier literals checked only after the qualified terms are instantiated. Thus, sort restrictions can be specified in qualifier literals and deductions can be performed using only the nonsort information. The deductions are then subjected to verification that terms are of the correct sort.

5 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 proofs and the size of the search space. 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 [6,15]. This system demonstrated substantial improvement in performance when theory resolution was used on Schubert's steamroller challenge problem. The KRYPTON knowledge representation system is also applying the ideas of theory resolution to combine a terminological reasoning component and an assertional reasoning component (for which they are also utilizing the KLAUS deduction system).

Theory resolution is a procedure with substantial power and generality. It is our hope that it will serve as a base for the theoretical and practical development of a methodology for combining the general reasoning capabilities of resolution theorem-proving programs with more efficient specialized reasoning procedures.

One important area for further research on theory resolution is finding restrictions on the need for retention of tautologies and determining compatibility with other resolution refinements.

Another important research question is handling combinations of theories (beyond the trivial case of totally disjoint theories). Successful combining of multiple deductive specialists within a resolution framework awaits further development in this area. The work of Nelson and Oppen [11] and Shostak [14] on combining quantifier free theories may be relevant.

Acknowledgments

The author would like to thank Mabry Tyson, Richard Waldinger, and Christoph Walther for their very useful comments on an earlier draft of this paper.

References

[1] Anderson, R. and W.W. Bledsoe. A linear format for resolution with merging and a new technique for establishing completeness. *Journal of the ACM 17,* 3 (July 1970), 525-534.

[2] Antoniou, G. and H.J. Ohlbach. TERMINATOR. *Proceedings of the Eighth International Joint Conference on Artificial Intelligence,* Karlsruhe, West Germany, August 1983, 916 919.

[3] Blasius, K., N. Eisinger, J. Siekmann, G. Smolka, A. Herold, and C. Walther. The Markgraf Karl Refutation Procedure (Fall 1981). *Proceedings of the Seventh International Joint Conference on Artificial Intelligence,* Vancouver, B.C., Canada, August 1981, 511-518.

[4] Brachman, R.J., R.E. Fikes, and H.J. Levesque. KRYPTON: a functional approach to knowledge representation. *IEEE Computer 16,* 10 (October 1983), 67-73.

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

[6] Haas, N. and G.G. Hendrix. An approach to acquiring and applying knowledge. *Proceedings of the AAAI-80 National Conference on Artificial Intelligence,* Stanford, California, August 1980, 235-239.

[7] Harrison, M.C. and N. Rubin. Another generalization of resolution. *Journal of the ACM 25,* 3 (July 1978), 341-351.

[8] Lusk, E.L. and R.A. Overbeek. Experiments with resolution-based theorem-proving algorithms. *Computers and Mathematics with Applications 8,* 3 (1982), 141-152.

[9] Lusk, E.L. and R.A. Overbeek. The automated reasoning system 1TP. Technical Report ANL-84-27, Argonne National Laboratory, Argonne, Illinois, April 1984.

[10] Morris, J.B. E-resolution: extension of resolution to include the equality relation. *Proceedings of the International Joint Conference on Artificial Intelligence,* Washington, D C , May 1969, 287-294.

[11] Nelson, G. and D.C. Oppen. Simplification by cooperating decision procedures. *ACM Transactions on Programming Languages and Systems 1,* 2 (October 1979), 245-257.

[12] Overbeek, R.A. Personal communication, September 1984.

[13] Pigman, V. The interaction between assertional and terminological knowledge in KRYPTON. *Proceedings of the IEEE Workshop on Principles of Knowledge-Based Systems,* Denver, Colorado, December 1984.

[14] Shostak, R.E. Deciding combinations of theories. *Journal of the ACM 31,* 1 (January 1984), 1-12.

[15] Stickel, M.E. A nonclausal connection-graph resolution theorem-proving program. *Proceedings of the AAAI-82 National Conference on Artificial Intelligence,* Pittsburgh, Pennsylvania, August 1982, 229-233.

[16] Stickel, M.E. Automated deduction by theory resolution. Technical Note 340, Artificial Intelligence Center, SRI International, October 1984. (An extended version of this paper.)

[17] Walther, C. A many-sorted calculus based on resolution and paramodulation. *Proceedings of the Eighth International Joint Conference on Artificial Intelligence,* Karlsruhe, West Germany, August 1983, 882-891.

[18] Walther, C. A mechanical solution of Schubert's steamroller by many-sorted resolution. *Proceedings of the AAAI-84 National Conference on Artificial Intelligence,* Austin, Texas, August 1984, 330-334.

[19] Walther, C. Unification in many-sorted theories. *Proceedings of the 6th European Conference on Artificial Intelligence,* Pisa, Italy, September 1984.

[20] Walther, C. Personal communication, October 1984.

[21] Winker, S. An evaluation of an implementation of qualified hyperresolution. *IEEE Transactions on Computers C-25,* 8 (August 1976), 835-843.

[22] Wos, L., R. Veroff, B. Smith, and W. McCune. The linked inference principle, II: the user's viewpoint. *Proceedings of the 7th International Conference on Automated Deduction,* Napa, California, May 1984, 316-332.