Notices of the AMS 24
pp. A-433 (1977)

Complexity of Superresolution
(Extended Abstract)

Karl Lieberherr, Swiss Federal Institute of Technology, Institute of Informatics, ETH-Zentrum, Clausiusstr.55, CH-8092 Zurich, Switzerland.

Abstract:
A complete proof system, called Superresolution, for unsatisfiable formulas of the propositional calculus is introduced and compared with Resolution on conjunctive normal forms (cnf´s). It is shown that Superresolution is shorter and more restrictive than Resolution, semantic Resolution, Hyperresolution, linear Resolution etc.

The length C(R(s)) of a Superresolution (Resolution) proof R(s) for the cnf s is the number of different superresolvents (resolvents) in R(s). For each cnf s (except for those with a Resolution proof of length 1) there is a Superresolution proof SR(s) such that for all Resolution proofs R(s) : C(SR(s))<C(R(s)). A decision procedure SR for Satisfiability is introduced which, by definition, only generates normal Superresolution proofs on unsatisfiable cnf´s. The concept "normal" is motivated by the fact that each Superresolution proof can be abridged to a normal Superresolution proof. Since each normal superresolvent is a resolvent and no input resolvent (except the empty clause) can be a superresolvent, normal Superresolution is a strong restriction of Resolution. Nevertheless one application of the deduction rule "normal Superresolution" and its checking only need linear time. Superresolution and Resolution are shown to be polynomially equivalent on cnf´s, and an exponential lower bound for a special kind of Superresolution is proven.

Contents:
Ø. Introduction
1. Superresolution
2. Restriction
3. Abridgement
4. Normal Superresolution
5. Lower bound

Ø. Introduction

In the field of automatic theorem proving several restrictions (refinements) of Resolution are investigated which are still complete [CH73, RO68, LOV7Ø]. These investigations are motivated by the following experience: When automatic theorem proving programs are applied unbridled to a given theorem they can not prove immediately, they produce floods of useless deductions generating useless clauses which in turn participate in yet more useless deductions (see e.g. [CH73(p100), MAR75(p137)]). If the number of possible deductions is smaller, this danger is diminished, but restrictive deduction systems may lengthen the proofs [KI72, ME71, SHO76]. Normal Superresolution has the favourable property that it allows shorter proofs than Resolution, although it is a strong restriction of Resolution. Superresolution, contrary to Resolution, is defined on all well-formed formulas of the propositional calculus. Hence transformation into normal form is not necessary for the application of Superresolution.

We shall use the following notations:
in   a in B: a is an element of B
**   exponentiation
{}   empty set
+   union of sets (and addition)
*   intersection of sets (and multiplication)
□   end of a proof
v   disjunction (or)
&   conjunction (and)
´   negation (a´ is the negation of a)
Square brackets [and] are used for subscripting.

1. Superresolution

In this section I introduce Superresolution and I prove that it is a complete proof system for proving the unsatisfiability of well-formed formulas (<u>wff</u>) of the propositional calculus. Upper bounds for the complexities of proof checking and proof generation are given.

Superresolution is an efficient realization of the generalized Resolution introduced in [RO68]. It associates a generalized resolvent with each failed model construction.

For the definition of Superresolution we need the following notions: Let F be a wff. An <u>interpretation</u> of F is described by a set I containing for each variable of F exactly one literal. If a variable x occurs positively in I (i.e. x is in I), "true" is assigned to x, otherwise, if x occurs negatively in I (i.e. x´ is in I), "false" is assigned to x. A partial interpretation of F is a subset of an interpretation of F.

Let LIT(F) be the set of literals of F and let I be a partial interpretation of F. A literal k in LIT(F) is called <u>simply determined</u> by I in the formula F, if F is unsatisfied under the interpretation I+{k´} or if k is in I (we write (F,I)->k). A formula F is said to be unsatisfied under a partial interpretation I, if the result "false" is obtained after the elimination of the constants introduced in F by I. A literal k in LIT(F) is called <u>determined</u> in F by an interpretation

I, if there are literals g[1], g[2], ... g[n]=k (in LIT(F), n>Ø) of F, such that (F,I+{g[1], g[2], ... g[i-1]}) -> g[i] for $1 \le i \le n$ (we write (F,I)=>k). Let D(F,I) be the set of all literals of F determined by I. D(F,I) may contain a complementary pair of literals, i.e. a variable v and its complement v´.

<u>Definition 1.1</u> : Let F be a wff and c a clause with literals in F. Let I be the set of complemented literals in c. Then c is a superresolvent of F, if

1. D(F,I) does not contain a complementary pair of literals and
2. there is a variable v in F, such that D(F,I+{v}) and D(F,I+{v´}) contain a complementary pair of literals.

v is called the learning variable of c.

By definition a superresolvent sr of a formula F is a clause which is satisfied by each existing model of F. Hence sr is implied by F.

Whether a clause sr is a superresolvent of a formula F depends mostly on the whole F. Therefore Superresolution, contrary to Resolution, Natural Deduction, Frege Systems etc., is a "global" proof system.

Example:
Each line of the following cnf describes one clause.

```
1. a  b
2.       c  d
3.             e  f
4. a´    c´
5. a´          e´
6.       c´    e´
7.    b´    d´
8.    b´          f´
9.          d´    f´
```

The empty clause is the only superresolvent of this cnf. An arbitrary variable is a learning variable.

<u>Definition 1.2</u> : A Superresolution proof for a formula F is a sequence of clauses c[1], c[2], ... c[m], such that c[m] equals the empty clause ({}) and for 1<i<m the clause c[i+1] is a superresolvent of F & c[1] & c[2] & ... & c[i] .

Example:
F = not[{((a & b)-->b)-->((a & b)-->c)}-->
((a & b)-->(b-->c))]
{a´,b´} is a superresolvent of this formula, because D(F,{a,b}) does not contain a complementary pair of literals. Observe that D(F,{a,b,c}) and D(F,{a,b,c´}) contain a comple-

mentary pair of literals. {} is the next superresolvent. Note that a is a learning variable in the formula F1=F&(a´∨b´). Hence this Superresolution proof for F consists of two superresolvents.

Remarks:

1. Note that for a cnf s no superresolvent of s can be equal to or subsumed by a clause in s. Hence Superresolution, contrary to Resolution, makes subsumption and equality tests automatically.
2. Note that for a satisfiable wff F a model of F is found after the generation of at most 2**n superresolvents (n is the number of variables in F).

<u>Lemma 1.1</u> : Superresolution is a complete proof system for unsatisfiable formulas of the propositional calculus.

Proof:

1. Let F be a wff and let R(F) be a Superresolution proof for F. Then F is unsatisfiable, because each superresolvent of F is implied by F.
2. Let F be an unsatisfiable wff. Then a Superresolution proof for F can be constructed by the algorithm SRØ, which is described in the following: SRØ tries to construct a model for F by constructing interpretations of F repeatedly. One interpretation construction of F corresponds to one step of SRØ. If SRØ does not find a model in the current step, a superresolvent r is added to F. r has the task to prevent that in a further step the same interpretation is constructed again. SRØ stops when r is empty. On a satisfiable wff SRØ would find a model after a finite number of steps. Now SRØ is described in detail.

   Let F be the input wff.

   Repeat the following statements 1) and 2) until the empty clause is generated:

   1) Construct a superresolvent of F: Put I={} and CHOSEN={}. repeat the following statements a), b) and c) until a superresolvent is found:
      a) choose an arbitrary variable v of F, which is not in I.
         CONTR is a predicate on the literals of v, which is defined in the following:
         If D(F,I+{v}) contains a complementary pair of literals, CONTR(v) is true, otherwise CONTR(v) is false.
         If D(F,I+{v´}) contains a complementary pair of literals, CONTR(v´) is true, otherwise CONTR(v´) is false.
      b) Four mutually excluding cases are possible:
         A) CONTR(v) and CONTR(v´) are true: Choose an arbitrary literal k of v and set CHOSEN := CHOSEN + {k}.

B) CONTR(v) and CONTR(v´) are false: v is a learning variable and the complemented literals of CHOSEN are a superresolvent sr. Choose an arbitrary literal k of v.

C) CONTR(v) is true and CONTR(v´) is false: Set k:=v´.

D) CONTR(v) is false and CONTR(v´) is true: Set k:=v.

c) I := I + D(F,I+{k})

2) Add the conjunction sr to F.

By definition the same superresolvent cannot be generated twice. For a given wff F there exists only a finite number of superresolvents and therefore algorithm SRØ has to produce the empty clause after a finite number of steps, iff F is unsatisfiable.
□

Let F be a wff and let the length L(F) be the number of occurences of literals in F.

Lemma 1.2 : Let c be a clause with literals which occur in a formula F and let v be a variable of F. Then the checking whether c is a superresolvent with learning variable v can be done in time O(L(F)) on a random access machine.

Lemma 1.3 : A superresolvent of a wff F can be generated in time O(L(F)) on a random access machine.

2. Restriction

In this section I prove that a large class of resolvents cannot be superresolvents. Later we shall see that for cnf´s each "interesting" superresolvent is a resolvent.

Given a cnf s, since s is the original input set, we shall call each member of s an input clause. An input resolvent of s is a resolvent in the set E(s), which is defined in the following:

1. All input clauses are in E(s).
2. If c1 is a clause in E(s), c2 an input clause and if the resolvent r of c1 and c2 exists, r is in E(s).
3. There are no other clauses in E(s).

Theorem 2.1 : Let s be a cnf. Then no clause in E(s), except the empty clause, can be a superresolvent of s.

Proof:
Let DS(s,I) be an arbitrary subset of D(s,I) such that each variable in D(s,I) occurs in DS(s,I), but DS(s,I) does not contain a complementary pair of literals. We prove the following with induction on the structure of E(s): Let I be a partial interpretation of s such that a clause in E(s) is unsatisfied under I. Then a clause in s is unsatisfied under each interpretation DS(s,I).

This statement is true for input clauses. Let c1 be a clause in E(s) and c2 an input clause of s such that the resolvent r of c1 and c2 exists. Let v be the variable resolved upon. Suppose that r is not empty. Let I be the partial interpretation such that r is unsatisfied. Then (s,I)->v and (s,I)->v´. Hence the following two cases are possible under the interpretation DS(s,I):

A) c1 is satisfied and c2 is unsatisfied or
B) c1 is unsatisfied and c2 is satisfied.

In case A) a clause of s is unsatisfied and hence the statement to be proven is true. In case B) we apply the induction hypotheses.

Therefore no clause c (#{}) in E(s) can be a superresolvent since a clause in s is unsatisfied under an interpretation which is determined by I={complemented literals of c}.

□

3. Abridgement

In this section it is proven that each Resolution proof can be properly abridged to a Superresolution proof (except in trivial cases). The length of the Superresolution proof may be much shorter than the length of the Resolution proof since no input resolvent can be a superresolvent. The length C(R(s)) of a Superresolution (Resolution) proof R(s) for the cnf s is the number of different superresolvents (resolvents) in R(s).

A Resolution proof is called trivial, if the empty clause is the first resolvent. Let UNSATNT be the set of unsatisfiable cnf´s for which no trivial Resolution proof exists.

Lemma 3.1 : Let s be a cnf in UNSATNT and let R(s) be a Resolution proof for s. Then there is a Superresolution proof AR(s) for s, such that C(AR(s))<C(R(s)). Furthermore AR(s) can be obtained in time O(C(R(s))**2) on a random access machine.

In the following I prove that Superresolution allows shorter proofs than semantic Resolution [SL67], Hyperresolution [ROB65] and linear Resolution [LOV70].

Lemma 3.2: Let s be a cnf which has not a semantic Resolution proof of length 1, i.e. the empty clause is not the first semantic resolvent. Let R(s) be a semantic Resoluion proof of s. Then there is a Superresolution proof of s which contains less superresolvents than R(s) contains semantic resolvents.

Note that lemma 3.2 remains true if Hyperresolution is used instead of semantic Resolution, for Hyperresolution is a special case of semantic Resolution.

Superresolution allows shorter proofs than Resolution, since the first linear resolvent of a cnf s is in E(s).

4. Normal Superresolution

It will be shown that each Superresolution proof can be translated polynomially into a Resolution proof. For this purpose we introduce a decision algorithm SR for Satisfiability, which by definition generates normal Superresolution proofs for unsatisfiable cnf´s. Each Superresolution proof can be abridged to this normal form. The normal Superresolution proofs are polynomially reducible to Resolution proofs. Since each Resolution proof can be abridged to a Superresolution proof (lemma 2.1), Superresolution and Resolution are polynomially equivalent [CO74].

We use algorithm SRØ of the proof of lemma 1.1. for the explanation of algorithm SR. SR and SRØ are essentially the same algorithms except that SR constructs superresolvents more carefully. Recall that SRØ takes the whole set of complemented literals in CHOSEN as the next superresolvent. SR determines a possibly proper subset of CHOSEN as the next superresolvent.

Now we describe how SR chooses this subset. Let s be a cnf and let I be a partial interpretation of s constructed by SRØ in statement 1). Let k be a literal of s and suppose that (s,I)->k. By definition, if (s,I)->k , a clause c of s is unsatisfied under the interpretation I+{k´}. This clause c may contain literals in CHOSEN and literals which are determined by CHOSEN. Note that CHOSEN=>k, but possibly CH=>k for a proper subset CH of CHOSEN. Therefore we assign to each literal k which is determined by CHOSEN a subset P(k) of CHOSEN, such that P(k)=>k. P(k) is called a set of preconditions for k.

1. For a literal k in CHOSEN let P(k) = k.
2. Let k be a literal which is simply determined by CHOSEN and let c be the clause which is unsatisfied under the interpretation CHOSEN + {k´}. Let COMP(c) be the set of complemented literals in c. Then P(k) = (COMP(c) * CHOSEN) - {k´}.
3. Let k be a literal which is determined by CHOSEN and let c be a clause which is unsatisfied under an interpretation DS(s,CHOSEN + {k´}). P(k) is the union of the preconditions P(h) for all h (#k´) in COMP(c).

By this recursive definition a subset P(k) of CHOSEN is defined for each k in s.

Suppose that algorithm SRØ is in statement 1)b)B), i.e. a superresolvent is found. Let v be a learning variable and let c1 be a clause which is unsatisfied under an interpretation DS(s,CHOSEN+{v}). Let c2 be a clause which is unsatisfied under an interpretation DS(s,CHOSEN + {v´}). Let Pc1 be the union of the preconditions P(h) for all h (#v´) in COMP(c1). Let Pc2 be the union of the preconditions P(h) for all h (#v) in COMP(c2). Algorithm SR chooses Pc = COMP(Pc1 + Pc2) as the next superresolvent. Note that there may exist several Pc, because c1,c2,P and DS are not unique.

Lemma 4.1: Let R(s) be an arbitrary Superresolution proof for s. R(s) can be abridged to a Superresolution proof which is generated by SR.

Because of Lemma 4.1 we call a Superresolution proof which is generated by SR a normal Superresolution proof and a a superresolvent, which is constructed by SR a normal superresolvent.

Theorem 4.2: Let s be a cnf. Each normal superresolvent of s is the result of a sequence of Resolution operations.

Lemma 4.3: Let s be an unsatisfiable cnf with m clauses and let sr be a normal superresolvent. Then sr can be obtained by the application of at most 2*m+1 Resolution operations.

Theorem 4.4: Let s be an unsatisfiable cnf and let R(s) be a normal Superresolution proof for s. Then there exists a Resolution proof R1(s) for s, such that C(R1(s)) ≤ C(R(s))**2.

Theorem 4.5: Superresolution and Resolution are polynomially equivalent.

Proof:
By lemma 3.1 each Resolution proof can be abridged to a Superresolution proof. Each Superresolution proof can be abridged to a normal Superresolution proof (lemma 4.1) and each normal Superresolution proof can be translated polynomially into a Resolution proof (theorem 4.4).
□

Lemma 4.6: A normal superresolvent of a cnf s can be generated in time O(L(s)) on a random access machine.

5. Lower bound

In this section I prove an exponential lower bound for a special kind of Superresolution, called simple Superresolution. We give a polynomial translation of simple Superresolution proofs into enumeration trees and use a result of Galil [GAL76] on enumeration procedures.

Given a set s of clauses and a partial interpretation I, we obtain another set of clauses s[I] by substituting all values given by I: s[I] is the set of clauses c´, such that c´ is a subset of a clause c in s which is not satisfied by I. c´ contains all literals of c which are not assigned by I, and all literals in c-c´ are assigned false by I.

A set s of clauses c[1], c[2], ... c[n] is said to be tree embeddable, if there is a numbering (v[1], v[2], ... v[m]) of the variables in s such that the following holds:

1. Each nonempty clause in s contains a literal of v[1].
2. Let I be an arbitrary interpretation of the variables v[1], v[2], ... v[i-1] ($1<i<n$). Every nonempty clause in s[I] contains a literal of variable v[i].

<u>Definition 5.1</u>: A Superresolution proof R(s)=sr[1], sr[2], ... sr[n] of a cnf s is simple if R(s) is tree embeddable.

A proof system BW for UNSAT is called exponential, if there is a sequence s[1], s[2], ... of cnf´s, such that for each sequence of proofs BW(s[1]), BW(s[2]), ... the lengths grow exponentially, i.e. there are constants i0 and c>0, such that for all i>i0: C(BW(s[i])) $\geq$ 2**(c*length(s[i])). C denotes the proof length.

<u>Corollary 5.3</u>: Simple Superresolution is exponential.

REFERENCES

****

CH73. CHANG C.L., LEE R.C., SYMBOLIC LOGIC AND MECHANICAL THEOREM PROVING, ACADEMIC PRESS, NEW YORK AND LONDON, 1973.

CO74. COOK S., RECKHOW R., ON THE LENGTH OF PROOFS IN THE PROPOSITIONAL CALCULUS, 6.STOC (1974), 135-148.

GAL76. GALIL Z., ON ENUMERATION PROCEDURES FOR THEOREM PROVING AND FOR INTEGER PROGRAMMING, IN AUTOMATA, LANGUAGES AND PROGRAMMING, THIRD INTERNATIONAL COLLOQUIUM, EDINBURGH, 1976.

KI72. KIEBURTZ R.B., LUCKHAM D., COMPATIBILITY AND COMPLEXITY OF REFINEMENTS OF THE RESÓLUTION PRINCIPLE, SIAM J. COMPUT., VOL. 1, 313-332.

LOV70. LOVELAND D.W., A LINEAR FORMAT FOR RESOLUTION, PROC. IRIA SYMP. AUTOMATIC DEMONSTRATION, SPRINGER-VERLAG, NEW YORK, 147-162.

MAR75. MARMIER E., AUTOMATIC VERIFICATION OF PASCAL PROGRAMS, DISS. ETH NR.5629.

ME71. MELTZER B., PROLEGOMENA TO A THEORY OF EFFICIENCY OF PROOF PROCEDURES, IN ARTIFICIAL INTELLIGENCE AND HEURISTIC PROGRAMMING, AMERICAN ELZEVIER, 1971, 15 - 33.

RO65. ROBINSON J.A., A MACHINE-ORIENTED LOGIC BASED ON THE RESOLUTION PRINCIPLE, J.ACM, VOL 12, NO 1, 23-41.

ROB65. ROBINSON J.A., AUTOMATIC DEDUCTION WITH HYPERRESOLUTION, INTERNAT. J. COMPUT. MATH., 227-234.

RO68. ROBINSON J.A., THE GENERALIZED RESOLUTION PRINCIPLE, MI 3,77-93.

SHO76. SHOSTAK R.E., ON THE COMPLEXITY OF RESOLUTION, IN ALGORITHMS AND COMPLEXITY, ED. J.F. TRAUB, ACADEMIC PRESS, P.493.

SL67. SLAGLE J.R., AUTOMATIC THEOREM PROVING WITH RENAMABLE AND SEMANTIC RESOLUTION, J.ACM 14(4), 687-697.