

THE MANAGEMENT OF HEURISTIC SEARCH
IN BOOLEAN EXPERIMENTS WITH RUE RESOLUTION

Vincent J. Digricoli

Hofstra University

166-11 17th Road, Whitestone, N.Y. 11357

ABSTRACT

In assessing the power of a theorem prover, we should select a theorem difficult to prove, compare the quality of proof with the published work of mathematicians, and most important determine whether cpu time used to find the proof is economically acceptable.

In this paper we apply the above criteria to RUE resolution, equality-based binary resolution which incorporates the axioms of equality into the definition of resolution. We select a theorem in Boolean algebra, show the published proof of George and Garret Birkhoff side by side with the computer deduced proof achieved in less than 30 seconds of cpu time. The proof is quite long requiring the derivation of four lemmas and is proven by two RUE refutations of 16 and 18 steps respectively. The same refutations with the equality axioms and unification resolution are 38 and more than 40 steps. Hence, the power of RUE resolution is shown by the brevity of proof compared to using the equality axioms.

The primary pragmatic issue in theorem proving is the effective management of heuristic search to find proofs in acceptable computer time. Whether an inference system supports or obstructs this objective is a crucial property and in this paper we explain in detail the heuristics applied to find proofs. These heuristics are RUE specific and dependent, and cannot be applied in the context of unification resolution.

*This research was supported by a sabbatical grant from the IBM Systems Research Institute in New York.

I RESOLUTION BY UNIFICATION AND EQUALITY

RUE resolution is equality-based binary resolution in that it incorporates the axioms of equality into the definition of resolution, making refutations less than half as long as compared to using the equality axioms. Furthermore, it establishes a context for heuristics leading to more efficient searches for proofs. RUE resolution is based on the following rule of inference:

1. RUE Rule of Inference:

"The RUE resolvent of $P(s_1, \dots, s_n) \vee A$ and $\bar{P}(t_1, \dots, t_n) \vee B$, is $\sigma A \vee \sigma B \vee D$, where σ is a substitution and D is a disjunction of inequalities specified by a disagreement set of the complementary literals, $\sigma P(s_1, \dots, s_n)$ and $\sigma \bar{P}(t_1, \dots, t_n)$."

The above inference rule is in open form since we are free to choose the substitution and disagreement set to be used. Let us now define the notion of a disagreement set:

2. Disagreement Set of a Pair of Terms:

"If s, t are non-identical terms, the set of one element, the pair $s:t$, is the origin disagreement set. If s, t have the form, $f(a_1, \dots, a_k), f(b_1, \dots, b_k)$, then the set of pairs of nonidentical, corresponding arguments is the topmost disagreement set. Furthermore, if D is a disagreement set, then D' formed by replacing any member of D by the elements of its disagreement set, is also a disagreement set. If s, t are identical terms, the empty set is the sole disagreement set."

For example, the pair of terms:

$f(a, h(b, g(c))) : f(b, h(c, g(d)))$ has the disagreement sets:

D1: { $f(a, h(b, g(c))) : f(b, h(c, g(d)))$ }
 D2: { $a : b, h(b, g(c)) : h(c, g(d))$ }
 D3: { $a : b, b : c, g(c) : g(d)$ }
 D4: { $a : b, b : c, c : d$ }

or reduce to the inequalities of any disagreement set. We call the above the Negative Reflexive Function rule.

In (Digricoli, 1983) we prove:

4. Completeness of RUE Resolution:

We now define a disagreement of complementary literals, $P(s_1, \dots, s_n), \bar{P}(t_1, \dots, t_n)$ as the union:

$$D = \bigcup_{i=1}^n D_i$$

where D_i is a disagreement set of the corresponding arguments s_i, t_i . The topmost disagreement set of $P(s_1, \dots, s_n), \bar{P}(t_1, \dots, t_n)$ is the set of pairs of corresponding arguments which are not identical.

Using the substitution axiom for predicates, we can now state:

$$P(s_1, \dots, s_n) \wedge \bar{P}(t_1, \dots, t_n) \rightarrow D$$

where D now represents a disjunction of inequalities specified by any disagreement set of $P, \bar{P}$. In resolution by unification and equality, we can resolve P and $\bar{P}$ immediately to D . For example:

$$\begin{array}{c} P(f(a, h(b, g(c)))) \\ \vdash \\ P(f(b, h(c, g(d)))) \\ D \end{array}$$

resolves in four distinct ways depending on our choice of D . We may resolve to $f(a, h(b, g(c))) \neq f(b, h(c, g(d)))$, to $a \neq b \vee b \neq c \vee c \neq d$, or to an intermediate level D . These are logically distinct deductions since an input set may imply $f(a, h(b, g(c))) = f(b, h(c, g(d)))$ without implying $a = b \wedge b = c \wedge c = d$, so that the former participates in a refutation but the latter does not.

We now define a second inference rule similar to the above applying directly to an inequality:

3. NRF Rule of Inference:

"The NRF resolvent of the clause $t_1 \neq t_2 \vee A$ is $\sigma A \vee D$, where σ is a substitution and D is a disjunction of inequalities specified by a disagreement set of $\sigma t_1, \sigma t_2$."

For example we may deduce by NRF:

$$\begin{array}{l} f(a, h(b, g(c))) \neq f(b, h(c, g(d))) \\ \rightarrow a \neq b \vee b \neq c \vee c \neq d \end{array}$$

"If S is an E-unsatisfiable set of clauses, there exists an RUE-NRF deduction of the empty clause from S ."

This theorem establishes that resolving to inequalities is complete without introducing the axioms of equality (for substitution, transitivity and reflexivity) or paramodulation. Apart from symmetry, the axioms of equality are applied implicitly by the RUE-NRF rules of inference.

We describe the above as completeness in 'open form' since we have not specified either the substitution or disagreement set to be used. In (Digricoli, 1983), we deal with this issue and define the RUE unifier and the topmost viable disagreement set as part of a completeness theory stated in strong form. However, in this paper we will use RUE resolution in open form as defined above, heuristically exploiting this form and making efficiency of proof search our primary goal.

II OUR PRIMARY EXPERIMENT

Our case study deals with proving the following theorem:

Given: the axioms of Boolean algebra:

1. $x \vee y = y \vee x$ commutativity
2. $x \wedge y = y \wedge x$ "
3. $x \vee 0 = x$
4. $x \wedge 1 = x$
5. $x \vee \bar{x} = 1$
6. $x \wedge \bar{x} = 0$
7. $x(y \vee z) = xy \vee xz$ distributivity
8. $x \vee yz = (x \vee y)(x \vee z)$ "

Prove: $x \vee (y \vee z) = (x \vee y) \vee z$
 associativity of logical or

This is a fairly complex theorem for a human to prove and we have the following proof published in the Transactions of the American Mathematical Society [2] by George and Garret Birkhoff:

Theorem: $a \vee (b \vee c) = (a \vee b) \vee c$

Proof (Birkhoff):

We first prove lemmas: L1, L2, L3 and L4.

L1: $a = aa$ since: $a = a1 = a(av\bar{a}) = aava\bar{a}$
 $= aav0 = aa$

L2: $av1 = 1$ since: $av1 = (av1)1 = (av1)(av\bar{a})$
 $= av(1a) = av\bar{a} = 1$

L3: $a = avab$ since: $a = a1 = a(bv1) = abval$
 $= abva = avab$

L4: $a(avb) = a$ since: $a(avb) = aa \vee ab$
 $= a \vee ab = a$

From lemmas L1, L3, L4 and axiom 7, it follows that:

$$\begin{aligned} a &= a((avb)vc) \\ b &= b((avb)vc) \\ c &= c((avb)vc) \end{aligned}$$

Substitute the above for a, b, c in $av(bvc)$:

$$av(bvc) = a((avb)vc) \vee (b((avb)vc) \vee c((avb)vc)).$$

On the right side, factor out the expression $(avb)vc$ to the right:

$$av(bvc) = (av(bvc))((avb)vc).$$

On the right side, distribute $(av(bvc))$ across $((avb)vc)$:

$$av(bvc) = ((av(bvc))a \vee (av(bvc))b) \vee (av(bvc))c.$$

Applying L1, L3, L4 and axiom 7 to each member of the right side, we obtain:

$$a \vee (bvc) = (avb) \vee c. \quad \square$$

It is evident that especially the latter part of the above proof will be difficult for a human to deduce. In fact in human experiments with three mathematically astute university students, one could not prove associativity after six hours of work, a second proved associativity in nine hours and the third proved the theorem in three hours. Hence, we are asking the RUE theorem prover to prove a theorem which humans find quite difficult.

The following is the proof deduced by the RUE theorem prover in 24 seconds of cpu time (IBM 370/3081), using a total 7572 unifications in its proof search. We first ask the theorem prover to prove two lemmas, $xy \vee x = x$ and $(x \vee y)x = x$, and then augment the input axiom set with these lemmas to prove associativity.

2. Refutation to Prove: $xy \vee x = x$

$$\begin{array}{l} ab \vee a \neq a \\ \quad \vdash xy \vee xz = x(y \vee z) \\ \quad \quad a/x, b/y \\ az \neq a \vee a(b \vee z) \neq a \\ \quad \vdash x\wedge 1 = x \quad a/x, 1/z \\ a(b \vee 1) \neq a \\ \quad \vdash x\wedge 1 = x \quad a/x \\ 1 \neq b \vee 1 \\ \quad \vdash 1 = x \vee \bar{x} \quad b/x \\ b \vee \bar{b} \neq b \vee 1 \\ \quad \vdash x \vee yz = (xvy)(xvz) \\ \quad \quad b/x \\ yz \neq \bar{b} \vee (bvy)(bvz) \neq bvl \\ \quad \vdash x\wedge 1 = x \quad \bar{b}/x, \bar{b}/y, 1/z \\ (b \vee \bar{b})(b \vee 1) \neq b \vee 1 \\ \quad \vdash 1\wedge x = x \quad (bvl)/x \\ 1 \neq b \vee \bar{b} \\ \quad \vdash 1 = x \vee \bar{x} \quad b/x \\ \square \end{array}$$

In the above refutation, we are uniformly applying the following substitution:

"In a left-to-right scan of complementary literals, first unify at the highest argument level, and then in a second scan unify at all lower levels."

The refutation is a succinct 8 step RUE proof of the lemma, $xy \vee x = x$, and within this proof the sub-lemma, $x \vee 1 = 1$, is proven beginning at step 4. The same refutation performed with the equality axioms and standard unification resolution would be 19 steps.

3. Refutation to Prove: $(x \vee y)x = x$

$$\begin{array}{lcl}
 (a \vee b)a \neq a & & \\
 \mid & & (xvy)(xvz) = x \vee yz \\
 \mid & & a/x, b/y \\
 a \vee z \neq a \quad \vee \quad a \vee bz \neq a & & \\
 \mid & & x \vee 0 = x \quad a/x, 0/z \\
 a \vee b0 \neq a & & \\
 \mid & & x \vee 0 = x \quad a/x \\
 0 \neq b0 & & \\
 \mid & & 0 = x\bar{x} \quad b/x \\
 b\bar{b} \neq b0 & & \\
 \mid & & x(yvz) = xy \vee xz \quad b/x \\
 y \vee z \neq \bar{b} \quad \vee \quad by \vee bz \neq b0 & & \\
 \mid & & x \vee 0 = x \quad \bar{b}/x, \bar{b}/y \\
 & & 0/z \\
 b\bar{b} \vee b0 \neq b0 & & \\
 \mid & & 0 \vee x = x \quad b0/x \\
 0 \neq b\bar{b} & & \\
 \mid & & 0 = x\bar{x} \quad b/x \\
 \square & &
 \end{array}$$

We have a succinct RUE proof of 8 steps and an equivalent proof with the equality axioms and unification resolution would be 19 steps. Note that the sublemma $0 = x\bar{0}$ is proven beginning at step 4.

When the theorem prover is given the negated dual lemma:

$$ab \vee a \neq a \quad \vee \quad (avb)a \neq a$$

it produced a 16 step refutation proving both lemmas in 1.4 seconds with a total 1485 unifications in the proof search. This refutation is simply the concatenation of the above two refutations and counting sublemmas, four lemmas are being

proven in a single run of the theorem prover. It corresponds to the work of the Birkhoff paper proving lemmas L1, L2, L3, L4 and is a substantial piece of work.

4. Refutation to Prove:

$$x \vee (y \vee z) = (x \vee y) \vee z$$

The 18 step refutation may be summarized as follows:

suppose:

$$a \vee (bvc) \neq (avb) \vee c$$

$$\text{then: } (av(bvc))1 \neq ((avb)vc)1$$

$$(av(bvc))(cv\bar{c}) \neq ((avb)vc)(cv\bar{c})$$

$$(av(bvc))c \vee (av(bvc))\bar{c} \neq *$$

$$\begin{array}{ccc}
 \mid & & \mid \\
 c & \vee & (avb)\bar{c} \neq *
 \end{array}$$

$$(Ax7, L3, L4) \quad (Ax7, 6, 3)$$

where * is:

$$((avb)vc)c \vee ((avb)vc)\bar{c}$$

$$\begin{array}{ccc}
 \mid & & \mid \\
 c & \vee & (avb)\bar{c}
 \end{array}$$

$$(L4) \quad (Ax7, 6, 3)$$

which completes the proof by contradiction. We see that the computer deduced proof is perhaps simpler and more elegant than that stated in the Birkhoff paper. The actual refutation is given in Appendix I.

The theorem prover found the 18 step refutation proving associativity in 22.4 seconds using 6087 unifications in the proof search. An equivalent proof with the equality axioms and unification resolution would be more than 40 steps. Altogether 23.8 seconds with 7572 unifications were used for the entire proof. Cpu time would be substantially reduced by a more efficient implementation of the theorem prover in assembly language in place of PL/1. This is a very fine result compared to other published work (McCharen, Overbeek, Wos, 1976) and our purpose in this paper is to study the heuristic management of proof search which led to the above result.

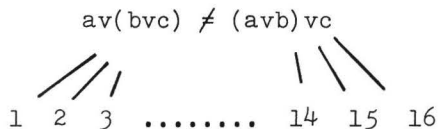
III FINDING A NEEDLE IN A HAYSTACK

Let us examine the enormity of the task of search which confronts a theorem prover seeking to find the refutations we have stated. The difficulty of proving theorems in boolean algebra is compounded by the commutivity of the boolean functions ($\vee, \wedge$) and the symmetry of equality, so that the axiom $x \vee 0 = x$, has 4 variants based on these properties. We can incorporate the entire effect of commutivity and symmetry by stating for the remaining boolean axioms all variants based on these properties:

- | | |
|---------------------------------------|------------|
| 1. $x \vee 0 = x$ | 4 variants |
| 2. $x \wedge 1 = x$ | 4 " |
| 3. $x \vee \bar{x} = 1$ | 4 " |
| 4. $x \wedge \bar{x} = 0$ | 4 " |
| 5. $x(y \vee z) = xy \vee xz$ | 64 " |
| 6. $x \vee yz = (x \vee y)(x \vee z)$ | 64 " |

Hence, in boolean algebra we are really dealing with an input set of 144 axioms when we drop the axioms for boolean commutivity and equality symmetry. Furthermore, the RUE inference rules implicitly incorporate the axioms of equality for substitution, transitivity and reflexivity and these axioms also do not appear in the input set. Let us assume that we will limit the input set to n clauses selected from the above variants. Let us assess the magnitude of search for the 18 step refutation which proves associativity when n is as low as 16.

The theorem prover begins with the negated theorem with skolem constants, $av(bvc) \neq (avb)vc$, and must find the refutation sequence we have stated. The refutation search is represented by a tree:



Since in RUE resolution complementary literals always resolve, each of the 16 clauses in the input set resolves immediately with the inequality of the negated theorem and in a breadthfirst search 16 resolvents appear at level one of the search tree. The complete breadthfirst expansion of the search tree to level 18 has 16-to-18th leaf nodes. The situation is actually worse if we take into account resolving to different disagreement sets making the expansion factor at a node higher than 16. Even with the input set reduced to 16 clauses, a breadthfirst search for a refutation is out of the

question. The refutation we seek will be a linear input refutation represented by a path in the above tree. A miniscule number of paths from the root will culminate in the empty clause but at least one will when the input set is E-unsatisfiable.

We have stated that the refutation for associativity was actually attained by the RUE theorem prover in 6087 unifications, i.e., the theorem prover heuristically developed a subtree of 6087 nodes in place of the breadthfirst expansion. We wish now to explain precisely what heuristics were applied to define this subtree of search.

IV A HEURISTICALLY CONTROLLED PROOF SEARCH

In order to find a refutation in acceptable computer time, we must drastically prune the breadthfirst search tree and furthermore order the search in the remaining subtree. We will define the components of an evaluation function which heuristically determine which search paths to abort and what is the best leaf node to expand in the search tree. Furthermore, in applying an axiom to a node, we must heuristically select a disagreement set apt to participate in a refutation. The following principles were applied with these objectives in mind:

- (1) heuristic ordering by degree of unification,
- (2) selection of the lowest level disagreement set not containing an irreducible literal,
- (3) heuristic substitution selection,
- (4) complexity bounds relating to:
 - (a) argument nesting
 - (b) number of distinct variables in a clause
 - (c) number of occurrences of the same constant or function symbol in a clause
 - (d) maximum number of literals in a clause
 - (e) maximum character length of a clause
- (5) removing redundant resolvents,
- (6) frequency bounds for the use of individual axioms in a ref. path.

All of the above principles are syntactic in nature and apply generically to experiments performed. (1) through (3), which were crucial in our work, are RUE specific and cannot be applied in standard unification resolution with the equality axioms. The remaining principles have been commonly used in resolution theorem proving.

A. Heuristic Ordering by Degree of Unification

If we wish to erase the literal, $tl \neq t2$, in a refutation, we measure the relevancy of an axiom $a1 = a2$ by computing the degree of unification between literals as follows:

- (1) Apply the mgpu (the most general partial unifier) to complementary literals to obtain $\sigma tl \neq \sigma t2$, $\sigma a1 = \sigma a2$.
- (2) Set $w=0$ (unification weight).
- (3) For $i = 1, 2$:
 if σti matches σai identically,
 then $w = w+50$,
 else if $\sigma ti, \sigma ai$ are the same function,
 say: σti is $f(b1, b2)$ and σai , $f(c1, c2)$,
 then $w = w+20$ and, furthermore,
 $w = w+15$ for each matching pair of
 corresponding arguments.

This is a simple scheme of matching which computes a weight of 100 when the mgu of complementary literals exists, and 0 when there is no degree of unification. There is also an intermediate scoring between these extremes. We now state our first principle of heuristically ordering the expansion of the refutation search tree:

- (1) Apply axioms to a negative literal in the order of higher degree of unification first and set a lower limit SDWMIN below which we suppress or postpone the application of an axiom (search directive weight minimum).
- (2) Furthermore, among axioms which satisfy SDWMIN, select the first SDLIM axioms with the highest unification scores (search directive limit).

The 18 step refutation for associativity was found using $SDWMIN=50$, i.e., we pruned the search tree of all axioms falling below this unification score. In proving the supporting lemmas for associativity, all refutations were found using $SDLIM=3$. As naturally intuitive, this

heuristic by degree of unification may be in RUE resolution, it, nonetheless, does not apply in unification resolution which requires the mgu at each deduction step.

B. Selecting the Lowest Level Disagreement Set not Containing an Irreducible Literal

Typically in adding the negated theorem to the input clause set, we introduce skolem constants and when it is evident that these constants are in effect arbitrary constants in respect to the input axioms, then we can conclude that inequalities on skolem constants like $a \neq b$ are irreducible, i.e., we cannot deduce $a=b$ from the axiom set.

For example, the negated theorem, $av(bvc) \neq (avb)vc$, introduces skolem constants a, b, c which in respect to the axioms of boolean algebra are arbitrary constants and we cannot prove two of these constants equal. Thus we should never generate an inequality like $a \neq b$ in an RUE deduction. Furthermore, inequalities like $(xv\bar{x})a \neq b$ which demodulate to irreducible literals are also irreducible and cannot appear in a refutation. This leads to the following heuristic rule:

"In an RUE deduction, choose as resolvent inequalities, the innermost inequalities not containing an irreducible literal."

Hence, in the refutation to prove $xy v x = x$, we resolved:

$$\begin{array}{l} 1 \neq b v 1 \\ \quad \vdash \quad 1 = x v \bar{x} \quad b/x \\ b v \bar{b} \neq b v 1 \end{array}$$

since resolving to $\bar{b} \neq 1$ would result in an irreducible literal. This heuristic proved successful, not only in all the boolean experiments, but also in ten experiments in group and ring theory (Digricoli, 1981). The RUE theorem prover permits the user to specify as input a list of irreducible literals, and, at the end of a run, it produces a list of ground literals present at leaf nodes of the search tree which the user may scan for irreducible literals in preparation for the next run.

C. Heuristic Substitution Selection

In the completeness analysis in (Digricoli, 1983), we specify that substitutions may be unconditionally performed in variables at the first argument level, like $P(x)$, but only in variables at lower levels, like $P(f(x))$, if certain conditions are met by the input set. This leads to longer refutations requiring extensive use of the NRF rule, when the same refutations can be stated in abbreviated form without the NRF rule, if we permit immediate substitutions at lower argument levels. In fact, it occurs in experiments performed that the following maximum unification is the refutation substitution:

"In a left-to-right scan of complementary literals, first unify at the highest argument level, and then in a second scan unify at all lower levels."

Note that we have given priority to substituting at the first argument level as specified by the completeness theory. However, the above substitution selection which enhances the efficiency of finding proofs, is not universally compatible with RUE completeness and we show this in (Digricoli, 1983).

D. Pruning by Complexity Bounds

An important method of pruning the search tree is to apply complexity bounds on resolvents which are formed. Theorem provers working in areas of the search tree having little relation to a refutation tend to produce resolvents which are too complex under a variety of attributes. The RUE theorem prover permits the user to specify complexity limits which when exceeded cause the resolvent to be discarded. These limits relate to the depth of argument nesting, the character length of a clause, the number of literals in a clause, the frequency of appearance of a constant or function symbol in a clause (or literal) and the number of distinct variables in a clause. It is true that the ideal setting of complexity bounds can only be derived by knowing a refutation. Nonetheless, it is important to know to what extent the proof search is contracted by applying complexity bounds. Experiments show that it is a good heuristic to use tight bounds to begin with (possibly derived from examining proofs in prior work with the theorem prover) and to gradually relax these bounds. The experimental results in this paper are first stated by applying complexity limits which are a profile of the refutation derived. We then suspend all use of these bounds, to determine the degradation of search which occurs. Both sets of results turn out to be favorable.

V EXPERIMENTAL RESULTS

	Steps in RUE Proof	Run 1 - with complexity bounds		Run 2 - without complexity bounds	
		CPU Time seconds	Total Unif.	CPU Time seconds	Total Unif.
B1: $x \vee 1 = 1$	5(11)	0.155	183	0.508	558
B2: $x \wedge 0 = 0$	5(11)	0.173	220	0.607	663
B3: $xy \vee x = x$	8(19)	0.654	724	0.964	1034
B4: $(xvy)x = x$	8(19)	0.647	773	0.868	1020
B1 $\wedge$ B2 $\wedge$ B3 $\wedge$ B4	16(38)	1.401	1485	3.008	2855
B5: $xv(yvz) = (xvy)vz$	18(40+)	22.431	6087	70.705	18727

(1) Under length of proof, 5(11) denotes that the 5 step RUE refutation is 11 steps when restated with the equality axioms and unification resolution.

(2) In Run 1 complexity bounds were applied which perfectly fit the refutation derived, but in Run 2 no complexity bounds were applied to limit search. Apart from this distinction, both Run 1 and 2 used the same input set and heuristics. In Run 2 of B5, complexity bounds were relaxed one level and applied.

(3) To prove lemmas B1 through B4, both individually and altogether, the following input set was used:

- | | |
|-----------------------------|------------|
| 1. $x \vee 0 = x$ | 4 variants |
| 2. $x \wedge 1 = x$ | 4 " |
| 3. $x \vee \bar{x} = 1$ | 4 " |
| 4. $x \wedge \bar{x} = 0$ | 4 " |
| 5. $x(yvz) = xy \vee xz$ | 8 " |
| 6. $x \vee yz = (xvy)(xvz)$ | 8 variants |

A complete set of variants was used for the first four axioms but only a partial set for each of the distributive axioms. In resolving with an inequality, the theorem prover first chose from each variant set, that axiom having the highest unification score with the inequality. This reduced the applicable axioms from 32 to 6, and of these only three with the highest unification scores were applied since SDLIM was set to 3. This meant that the node expansion factor was 3 instead of 32 in the search tree and the excellence of the results in proving lemmas is due primarily to this heuristic together with avoidance of irreducible literals.

(4) In proving B5 associativity, we reduced the input set to 16 clauses and introduced variants of two proven lemmas:

- | | |
|---------------------------|-------|
| 1. $x \vee 0 = x$ | |
| 2. $x = x \vee 0$ | |
| 3. $x \wedge 1 = x$ | |
| 4. $x = x \wedge 1$ | |
| 5. $x \vee \bar{x} = 1$ | |
| 6. $1 = x \vee \bar{x}$ | |
| 7. $x \wedge \bar{x} = 0$ | |
| 8. $0 = x \wedge \bar{x}$ | |
| 9. $x(yvz) = xy \vee xz$ | |
| 10. $xy \vee xz = x(yvz)$ | |
| 11. $(yvz)x = yx \vee zx$ | |
| 12. $yx \vee zx = (yvz)x$ | |
| 13. $(yvx)x = x$ | lemma |
| 14. $x = (yvx)x$ | |
| 15. $yx \vee x = x$ | lemma |
| 16. $x = yx \vee x$ | |

In this experiment, the theorem prover in resolving with an inequality chose only those input clauses whose unification score was 50 or greater (SDWMIN=50) and SDLIM was not used. Our next experiments will attempt to prove associativity without using lemmas and with the same input set of 32 clauses previously used. This will be an exceptionally long refutation.

REFERENCES

- [1] Digricoli, V.J., "Resolution by Unification and Equality", Courant Institute, New York Univ., February 1983, (available through Univ. Microfilms).
- [2] Birkhoff George and Garret, "Distributive Postulates for Systems Like Boolean Algebra", Transactions of American Mathematical Society, vol 60, July-Dec. 1946.
- [3] Digricoli, V.J., "The Efficacy of RUE Resolution: Experiments and Heuristic Theory" In Proc. IJCAI-81, Vancouver, B.C., Canada, August 1981, pp. 539-547.
- [4] Digricoli, V.J., "Automatic Deduction and Equality" In Proc. National ACM Conf., Detroit, Michigan, October 1979, pp. 240-250.
- [5] Morris, J., "E-Resolution: an Extension of Resolution to Include the Equality Relation" In Proc. IJCAI-69.
- [6] McCharen, Overbeek and Wos, "Problems and Experiments for Automated Theorem Proving", IEEE Transactions on Computers, C25-80, 1976.
- [7] Robinson, J.A., "A Machine Oriented Logic Based on the Resolution Principle", JACM 12, 1965, pp.23-41.
- [8] Brand, D., "Proving Theorems with the Modification Method", SIAM Journal of Computing, vol 4, no 4, 1975.
- [9] Harrison, M., and Rubin, N., "Another Generalization of Resolution", JACM vol 25, no 3, 1978.
- [10] Robinson, G., and Wos, L., "Paramodulation and Theorem Proving", Machine Intelligence, vol 4, 1969.
- [11] Wos, Overbeek and Henschen, "Hyperparamodulation" In Proc. 5CAD, Les Arcs, France, July 1980.
- [12] Slagle, J., "Automatic Theorem Proving with Builtin Theories Including Eq.", JACM, vol 19, no 1, January 1972.
- [13] Shostak, R., "An Algorithm for Reasoning about Equality", CACM, vol 21, No 7, July 1978.

*** Appendix I with the 18 step refutation proving associativity has been omitted due to space limitations. This refutation will be given at the conference presentation.