Theorem Proving via General Matings

PETER B. ANDREWS

Carnegie-Mellon University, Pittsburgh, Pennsylvania

ABSTRACT An approach to automatic theorem proving using matings of arbitrary sentences is discussed No use is made of conjunctive normal form (clauses) or prenex normal form, since these forms tend to introduce superfluous redundancy, complicate the search for a proof, and impede analysis of the essential logical structure of the proposed theorem. A complete exposition of the logical foundations of theorem proving via general matings is given, starting with proofs of appropriate versions of Herbrand's Theorem. It is shown that one may restrict quantifier duplication to outermost quantifiers without loss of completeness, though with possible loss of efficiency.

General matings could be used as the basis for a variety of theorem-proving procedures, and there are many opportunities for research in this area. A procedure using the criterion of path acceptability for matings is discussed. This criterion is easily visualized in terms of a two-dimensional format for formulas. An implementation by Eve Cohen has yielded encouraging preliminary results. Some implementation issues are discussed.

KEY WORDS AND PHRASES. automatic theorem proving, matings

CR CATEGORIES 3.69, 5 21

1. *Introduction*

Much of the research in automatic theorem proving has been focused on developing efficient methods for deriving contradictions from sets of clauses, which represent the conjuncts (disjunctions of literals) of a wff (well-formed formula) whose matrix is in conjunctive normal form. The advantages of conjunctive normal form were pointed out in [13] and incorporated into the widely studied resolution method [26]. Many theorems of mathematics and other disciplines lend themselves naturally to representation as sets of clauses. However, experience with a wide variety of theorems [2] has shown that in many other cases, the use of clausal form has serious disadvantages, since the repeated use of the distributive law $[P \vee (Q \wedge R)] \equiv [(P \vee Q) \wedge (P \vee R)]$ involved in the reduction to conjunctive normal form often causes wild proliferation of literals.

These difficulties arise both with respect to theorems of first-order logic and theorems of higher order logic. Indeed, many theorems of higher order logic require only the methods of first-order logic for their proofs and can easily be translated into theorems of first-order logic. Theorem A (see (1) of Figure 1) is such a theorem, and we use it to illustrate our point. The theorem is expressed in the notation of Church's

This material is based on work supported by the National Science Foundation under Grant MCS 78-01462

A preliminary version of this paper was presented to the Fourth Workshop on Automated Deduction in Austin, Texas, on February 1, 1979

Author's address Department of Mathematics, Carnegie-Mellon University, Pittsburgh, PA 15213

THEOREM A:

(1) $[\#F_{\alpha\beta} \cdot R_{o\beta} \cup S_{o\beta}] = \cdot[\#F_{\alpha\beta}R_{o\beta}] \cup \cdot\#F_{\alpha\beta}S_{o\beta}$

Negate, eliminate definitions, skolemize, and delete quantifiers:

(2) $[[[\sim R_{o\beta}x^6_\beta \wedge \sim S_{o\beta}x^6_\beta] \vee \cdot P^6_{o\alpha\beta}x^6_\beta\, x^1_\alpha \wedge \sim P^6_{o\alpha\beta}x^6_\beta \cdot F_{\alpha\beta}x^6_\beta]$
$\vee \cdot[\sim R_{o\beta}x^7_\beta \vee \cdot P^7_{o\alpha\beta}x^7_\beta x^1_\alpha \wedge \sim P^7_{o\alpha\beta}x^7_\beta \cdot F_{\alpha\beta}x^7_\beta]$
$\wedge \cdot \sim S_{o\beta}x^8_\beta \vee \cdot P^8_{o\alpha\beta}x^8_\beta x^1_\alpha \wedge \sim P^8_{o\alpha\beta}x^8_\beta \cdot F_{\alpha\beta}x^8_\beta]$
$\wedge \cdot[[R_{o\beta}x^9_\beta \vee S_{o\beta}x^9_\beta] \wedge \cdot \sim P^9_{o\alpha}x^1_\alpha \vee P^9_{o\alpha} \cdot F_{\alpha\beta}x^9_\beta]$
$\vee \cdot[R_{o\beta}x^{10}_\beta \wedge \cdot \sim P^{10}_{o\alpha}x^1_\alpha \vee P^{10}_{o\alpha} \cdot F_{\alpha\beta}x^{10}_\beta]$
$\vee \cdot S_{o\beta}x^{11}_\beta \wedge \cdot \sim P^{11}_{o\alpha}x^1_\alpha \vee P^{11}_{o\alpha} \cdot F_{\alpha\beta}x^{11}_\beta$

Eliminate conjunctions and form clauses (with type symbols omitted).

(c1) $\sim Rx^6 \vee P^6x^6x^1 \vee \sim Rx^7 \vee P^7x^7x^1$
(c2) $\sim Rx^6 \vee P^6x^6x^1 \vee \sim Rx^7 \vee \sim P^7x^7Fx^7$
(c3) $\sim Rx^6 \vee P^6x^6x^1 \vee \sim Sx^8 \vee P^8x^8x^1$
(c4) $\sim Rx^6 \vee P^6x^6x^1 \vee \sim Sx^8 \vee \sim P^8x^8Fx^8$
(c5) $\sim Rx^6 \vee \sim P^6x^6Fx^6 \vee \sim Rx^7 \vee P^7x^7x^1$
(c6) $\sim Rx^6 \vee \sim P^6x^6Fx^6 \vee \sim Rx^7 \vee \sim P^7x^7Fx^7$
(c7) $\sim Rx^6 \vee \sim P^6x^6Fx^6 \vee \sim Sx^8 \vee P^8x^8x^1$
(c8) $\sim Rx^6 \vee \sim P^6x^6Fx^6 \vee \sim Sx^8 \vee \sim P^8x^8Fx^8$
(c9) $\sim Sx^6 \vee P^6x^6x^1 \vee \sim Rx^7 \vee P^7x^7x^1$
(c10) $\sim Sx^6 \vee P^6x^6x^1 \vee \sim Rx^7 \vee \sim P^7x^7Fx^7$
(c11) $\sim Sx^6 \vee P^6x^6x^1 \vee \sim Sx^8 \vee P^8x^8x^1$
(c12) $\sim Sx^6 \vee P^6x^6x^1 \vee \sim Sx^8 \vee \sim P^8x^8Fx^8$
(c13) $\sim Sx^6 \vee \sim P^6x^6Fx^6 \vee \sim Rx^7 \vee P^7x^7x^1$
(c14) $\sim Sx^6 \vee \sim P^6x^6Fx^6 \vee \sim Rx^7 \vee \sim P^7x^7Fx^7$
(c15) $\sim Sx^6 \vee \sim P^6x^6Fx^6 \vee \sim Sx^8 \vee P^8x^8x^1$
(c16) $\sim Sx^6 \vee \sim P^6x^6Fx^6 \vee \sim Sx^8 \vee \sim P^8x^8Fx^8$
(c17) $Rx^9 \vee Sx^9 \vee Rx^{10} \vee Sx^{11}$
(c18) $Rx^9 \vee Sx^9 \vee Rx^{10} \vee \sim P^{11}x^1 \vee P^{11}Fx^{11}$
(c19) $Rx^9 \vee Sx^9 \vee \sim P^{10}x^1 \vee P^{10}Fx^{10} \vee Sx^{11}$
(c20) $Rx^9 \vee Sx^9 \vee \sim P^{10}x^1 \vee P^{10}Fx^{10} \vee \sim P^{11}x^1 \vee P^{11}Fx^{11}$
(c21) $\sim P^9x^1 \vee P^9Fx^9 \vee Rx^{10} \vee Sx^{11}$
(c22) $\sim P^9x^1 \vee P^9Fx^9 \vee Rx^{10} \vee \sim P^{11}x^1 \vee P^{11}Fx^{11}$
(c23) $\sim P^9x^1 \vee P^9Fx^9 \vee \sim P^{10}x^1 \vee P^{10}Fx^{10} \vee Sx^{11}$
(c24) $\sim P^9x^1 \vee P^9Fx^9 \vee \sim P^{10}x^1 \vee P^{10}Fx^{10} \vee \sim P^{11}x^1 \vee P^{11}Fx^{11}$

FIGURE 1

formulation of type theory [10] and says that if F is a function and R and S are subsets of the domain of F, then the image under F of the union of R and S is the union of the images of R and S under F. (We use $\#F$ to denote the function which maps sets to their images under F.) After negating Theorem A, eliminating definitions, Skolemizing, and deleting universal quantifiers, one obtains (2). (The details of this reduction may be found in Appendix A.) Roman letters are constants, and italicized letters are variables. Lines (c1)–(c24) are the clauses of the conjunctive normal form of (2). Line (2) contains 20 literals, but there are 104 literal-occurrences in clauses (c1)–(c24). Thus this simple mathematical statement has been transformed into a highly redundant form in which the basic logical structure of the theorem is no longer apparent.

One approach to automatic theorem proving which does not involve clauses is to seek directly to construct a proof in "natural deduction" format. Well-structured logical arguments are commonly presented in such a format, and natural deduction is certainly one of the important frontiers of research in automatic theorem proving. Of course, the fact that natural deduction provides a congenial format for communicating proofs does not necessarily mean that it provides the best context for discovering them. One can envision theorem-proving systems which use a variety of

Theorem B·

(B) $\exists x \forall y [Px \equiv Py] \supset [\exists x Px \equiv \forall y Py]$

Negate, and eliminate $\equiv$ and $\supset$.

(C) $\exists x \forall y [(\sim Px \vee Py) \wedge (\sim Py \vee Px)]$
$\wedge ([\exists x Px \wedge \exists y \sim Py] \vee [\forall y Py \wedge \forall x \sim Px])$

Skolemize:

(D) $\forall y [(\sim Pc \vee Py) \wedge (\sim Py \vee Pc)]$
$\wedge ([Pd \wedge \sim Pe] \vee [\forall z Pz \wedge \forall x \sim Px])$

Display in two-dimensional format:

(D′)
$$\left[\begin{array}{l} \forall y \left[\begin{array}{l} \sim Pc \vee Py \\ \sim Py \vee Pc \end{array}\right] \\ \left[\begin{array}{l} Pd \\ \sim Pe \end{array}\right] \vee \left[\begin{array}{l} \forall z Pz \\ \forall x \sim Px \end{array}\right] \end{array}\right]$$

Instantiate quantifiers

(H)
$$\left[\begin{array}{lll} \sim Pc & \vee & Pd \\ \sim Pd & \vee & Pc \\ \sim Pc & \vee & Pe \\ \sim Pe & \vee & Pc \\ \left[\begin{array}{l} Pd \\ \sim Pe \end{array}\right] & \vee & \left[\begin{array}{l} Pc \\ \sim Pd \end{array}\right] \end{array}\right]$$

FIGURE 2

methods to discover the essential ingredients of a proof and then construct proofs in whatever style is most congenial to the reader.

While people generally find it easier to grasp the ideas of a proof when it is presented in natural deduction style rather than resolution style, both of these proof methods involve breaking the wff into parts. For purposes of analyzing the logical structure of a theorem, such methods have the disadvantage that they tend to focus one's attention, and methods, on isolated parts of a wff and may encourage one to overlook certain aspects of its global structure.

Thus we are led to consider how one might prove a theorem without breaking it into parts. One such method is illustrated in Figure 2 with a proof of Theorem B. (The converse of Theorem B is also valid, and together these theorems justify distributing quantifiers over $\equiv$ in certain circumstances.) We negate Theorem B, eliminate $\equiv$ and $\supset$, and Skolemize to obtain (D). We often find it convenient to display wffs (especially if they are complex) in a two-dimensional format, with disjunctions being displayed horizontally but with conjunctions being displayed vertically. Line (D′) of Figure 2 displays (D) in such a format. Next we instantiate the quantifiers of (D′) to obtain (H). (The reason for this lettering of the wffs will become clear.) Note that $\forall y$ is instantiated with two terms, so that the subformula of (D′) having the form $\forall y R(y)$ is replaced by $R(\mathrm{d}) \wedge R(\mathrm{e})$. (H) is a contradiction (as we shall see), so Theorem B is established.

Naturally, we could dualize this approach to get a proof rather than a refutation. This would involve starting with a tautology and using existential generalization to derive the theorem.

Obviously, the basic problem is to proceed automatically from (D′) to (H) in Figure 2; one needs methods to decide which instantiations of the quantifiers to make so that the instantiated wff will be contradictory. This is the same problem that had to be faced when resolution was invented, and we use the same solution. The purpose

of the instantiations is to make certain literal-occurrences complementary, so we use a unification algorithm to find the substitution terms which will do this. The problem of deciding which literals should be made complementary remains. As when dealing with clauses, we can analyze this problem in terms of matings [1] of literal-occurrences. However, we now completely avoid the use of conjunctive normal form.

It was shown in [1] that matings are naturally induced by resolution-style refutations. Actually, it appears that all refutation and proof procedures for first-order logic tacitly involve the construction of matings, which embody much of the essential logical structure of the final refutations or proofs. It is our hope that procedures which focus directly on constructing matings for wffs in their natural form can avoid some of the redundancies and irrelevancies involved in constructing proofs and refutations, and that investigation of such procedures will yield new ideas and avenues for progress.

So much of the literature on automatic theorem proving has concentrated on wffs in prenex normal form and conjunctive normal form, in spite of the substantial practical disadvantages of these forms for many nontrivial problems, that a complete exposition of the logical foundations for dealing with more general classes of wffs of first-order logic seems to be needed. We here present such an exposition, assuming only that the reader has had a basic introduction to first-order logic. We also present an example of a refutation procedure based on matings of arbitrary wffs.

The basic logical ideas underlying this approach to theorem proving go back to Herbrand [20] (translated in [21]), whose proof of the fundamental theorem contained errors [15]. The ideas we use are closely related to those of Quine [25] and Prawitz [24], except that Quine restricts his attention to conjunctions of wffs in prenex normal form, and in [24] matrices are kept in conjunctive normal form and the scopes of quantifiers are not minimized.

A concise outline of the procedure we discuss below can be found in [9, Sec. 4], and related issues are discussed in other papers of Bibel. Another approach to theorem proving without clauses has been developed by Wilkins [30].

2. *Logical Preliminaries*

2.1 TERMINOLOGY. In our formal development we shall be concerned with wffs of a system of first-order logic whose primitive connectives and quantifiers are $\sim$ (not), $\wedge$ (and), $\vee$ (or), $\forall$ (for all), and $\exists$ (there exists . . . such that); $\supset$ (implication) and $\equiv$ (equivalence) are to be regarded as abbreviations. We could easily extend our treatment to a system in which $\supset$ is also primitive, but the details are straightforward and would merely clutter our exposition. Of course, it may be useful to include $\supset$ among the primitive connectives of computerized systems, and this should cause no difficulty. $\vDash A$ means that A is valid. When A is quantifier-free, $\vDash A$ means that A is tautological.

We write θA for the result of applying a substitution θ to an expression A. As an alternative substitution notation, we may write a wff A as $A(x_1, \ldots, x_n)$ to indicate that $x_1, \ldots, x_n$ occur as free variables in A and then use $A(t_1, \ldots, t_n)$ as a notation for the result of simultaneously substituting t_i for the free occurrences of x_i in A for $i = 1, \ldots, n$.

A wff C is in *negation normal form* (*nnf*) and is a *negation normal formula* (*nnf*) iff the scope of each occurrence of $\sim$ in C is atomic. A *negation normal sentence* (*nns*) is a sentence (wff without free variables) in nnf. A wff can easily be transformed into an equivalent nnf by using the laws

$$\sim\sim M \equiv M, \quad \sim[M \wedge N] \equiv [\sim M \vee \sim N], \quad \sim[M \vee N] \equiv [\sim M \wedge \sim N],$$
$$\sim\forall x\, M \equiv \exists x \sim M, \quad \text{and} \quad \sim\exists x\, M \equiv \forall x \sim M.$$

In general, the use of normal forms (such as negation, conjunctive, disjunctive, and prenex normal forms) provides conceptual simplicity which facilitates theoretical discussions, but it may make particular examples more cumbersome. (Figure 1 illustrated this phenomenon for the case of conjunctive normal form. Another example is provided by increases in the degrees of Skolem functions which may be caused by putting a wff into prenex normal form before Skolemizing.) However, when one puts a wff into nnf, one can obtain a wff with no more literal-occurrences than the original one, and having the same essential logical structure. Therefore, without any real loss of generality we may often confine our attention to wffs in nnf.

An occurrence of a quantifier or a well-formed (wf) part of a wff C is *positive* (*negative*) in C iff it is in the scope of an even (odd) number of occurrences of $\sim$. A wff is *universal* iff all its universal quantifiers occur positively and all its existential quantifiers occur negatively in it. It is well known how to introduce Skolem functions into a wff C so as to obtain a universal wff D (called the *Skolemized form* of C) such that C has a model if and only if D has a model. (We say that a wff *has a model* iff its universal closure is satisfiable; for wffs with free variables, this is not quite the same as satisfiability, though the two phrases are sometimes confused in the literature of theorem proving.)

Given a wff B which we wish to show is valid, we let C be a negation normal form of $\sim\bar{B}$ (where $\bar{B}$ is the universal closure of B), and let D be the Skolemized form of C (as in Figure 2). Then D is a universal nns which has no model if and only if B is valid. Thus we shall concentrate on the problem of refuting universal negation normal sentences.

2.2 Compound Instances. We now give a complete proof of a fundamental theorem (Theorem 1A below) which may be regarded as a form of Herbrand's Theorem.

We first define a *compound instance* (*c-instance*) of a universal nnf D to be the result of replacing each wf part of D of the form $\forall x\, B(x)$ by $B(t_1) \wedge \cdots \wedge B(t_n)$, where $n \geq 1$, and for each $i \leq n$, t_i is a closed (variable-free) term; different terms $t_1, \ldots, t_n$ may be chosen for different occurrences of quantifiers in D.

To avoid any ambiguities, we define the c-instances of D inductively as follows:

(a) If D is a literal, D is the only c-instance of D.
(b) If D is $[D_1 * D_2]$, where $*$ is $\vee$ or $\wedge$, and H_i is a c-instance of D_i for $i = 1, 2$, then $[H_1 * H_2]$ is a c-instance of D.
(c) If D is $\forall x\, C(x)$, $n \geq 1$, and for each $i (1 \leq i \leq n)$, t_i is a closed term and H_i is a c-instance of $C(t_i)$, then $[H_1 \wedge \cdots \wedge H_n]$ is a c-instance of D.

Lemma 1. *Let G and H be c-instances of a universal nnf D. Then there is a c-instance K of D such that $\models K \supset [G \wedge H]$.*

Proof. The proof is by induction on the number of occurrences of $\wedge$, $\vee$, and $\forall$ in D.

Case 1. D is a literal. Then $G = D = H$, and we let $K = D$ also.

Case 2. D has the form $D_1 * D_2$, where $*$ is $\wedge$ or $\vee$. Then G has the form $G_1 * G_2$ and H has the form $H_1 * H_2$, where G_i and H_i are c-instances of D_i for $i = 1, 2$. By the inductive hypothesis, for each i there is a c-instance K_i of D_i such that $\models K_i \supset [G_i \wedge H_i]$. Let $K = K_1 * K_2$. K is a c-instance of D, and $\models K \supset [G \wedge H]$.

Case 3. D has the form $\forall x\, C(x)$. Then G has the form $E_1 \wedge \cdots \wedge E_n$ and H has the form $E_{n+1} \wedge \cdots \wedge E_m$, where for each i there is a closed term t_i such that E_i is a c-instance of $C(t_i)$. Clearly we can let K be $G \wedge H$. □

A *truth assignment* is an assignment of truth values (t or f) to atomic wffs. If G is a quantifier-free wff and Φ is a truth assignment which assigns truth values to all atomic wf parts of G, we let $\mathscr{V}_\Phi G$ denote the truth value of G with respect to Φ; this is computed in the usual way using the truth tables for the propositional connectives. We call G a *truth-functional contradiction* (*t-f contradiction*) iff $\mathscr{V}_\Phi G = \mathrm{f}$ for all such truth assignments Φ. We say that Φ *verifies* G iff $\mathscr{V}_\Phi G = \mathrm{t}$. If $\mathscr{S}$ is a set of quantifier-free wffs and Φ is a truth assignment which verifies each member of $\mathscr{S}$, we say that Φ *truth-functionally satisfies* (*t-f satisfies*) $\mathscr{S}$.

LEMMA 2. *Let D be a universal nnf. If no c-instance of D is a t-f contradiction, then there is a truth assignment which verifies every c-instance of D.*

PROOF. This is just an application of the Compactness Theorem for propositional calculus [16, p. 59], which says that if $\mathscr{S}$ is a set of wffs of propositional calculus and every finite subset of $\mathscr{S}$ is satisfiable, then $\mathscr{S}$ is satisfiable too. Let $\mathscr{S}$ be the set of all c-instances of D, and let $\{H_1, \ldots, H_n\}$ be a finite subset of $\mathscr{S}$. Using Lemma 1 we obtain a c-instance K of D such that $\vDash K \supset [H_1 \wedge \cdots \wedge H_n]$. Since K is not a t-f contradiction, there is a truth assignment Ψ which verifies K, so Ψ t-f satisfies $\{H_1, \ldots, H_n\}$. Thus every finite subset of $\mathscr{S}$ is t-f satisfiable, and so is $\mathscr{S}$ also. □

THEOREM 1A. *Let D be a universal nns of first-order logic. D has no model iff D has a c-instance which is t-f contradictory.*

PROOF. It is easy to see by induction on the number of occurrences of $\wedge$, $\vee$, and $\forall$ in D that if D is a universal nns and G is any c-instance of D, then $\vDash D \supset G$. Hence if G is t-f contradictory, it is false in any model, so D is also, so D has no model.

For the proof in the other direction, suppose D is a universal nns which has no t-f contradictory c-instance. Then by Lemma 2 there is an assignment Φ of truth values to the atoms that occur in c-instances of D which verifies every c-instance of D.

We next construct a model for D. We let the domain of individuals of our model be the set of closed terms of our language. (We assume that the language contains at least one individual constant; apart from this requirement, it need contain no constants other than the individual, function, and predicate constants which occur in D.) As in Henkin's completeness proof, we interpret the individual constants and function symbols so that each term (regarded as an expression of the language) denotes itself (regarded as an element of the domain of individuals). Also, we interpret the predicate symbols so that if P is any n-ary predicate and $t_1, \ldots, t_n$ are individuals, then $Pt_1, \ldots, t_n$ is true in the interpretation iff $\Phi(Pt_1, \ldots, t_n) = \mathrm{t}$. This specifies an interpretation $\mathscr{M}$ for our system of first-order logic. For any sentence B we let $\mathscr{V}^{\mathscr{M}}B$ be the truth value of B in this interpretation, and we let $\mathscr{M} \vDash B$ mean that B is true in $\mathscr{M}$.

Next we show, by induction on the number of occurrences of $\wedge$, $\vee$, and $\forall$ in B, that if B is any universal nns such that Φ verifies every c-instance of B, then $\mathscr{M} \vDash B$.

Case 1. B is a literal. Then B is the only c-instance of B, and $\mathscr{V}^{\mathscr{M}}B = \mathscr{V}_\Phi B$ by the interpretation of predicates.

Case 2. B has the form $[B_1 \wedge B_2]$. For $i = 1, 2$, let G_i be any c-instance of B_i. Then $[G_1 \wedge G_2]$ is a c-instance of B, so $\mathscr{V}_\Phi[G_1 \wedge G_2] = \mathrm{t}$, so $\mathscr{V}_\Phi G_i = \mathrm{t}$. Therefore $\mathscr{M} \vDash B_i$ by inductive hypothesis; so $\mathscr{M} \vDash B$.

Case 3. B has the form $[B_1 \vee B_2]$. Suppose there are c-instances G_1 of B_1 and G_2 of B_2 such that $\mathscr{V}_\Phi G_1 = \mathrm{f}$ and $\mathscr{V}_\Phi G_2 = \mathrm{f}$. Then $\mathscr{V}_\Phi[G_1 \vee G_2] = \mathrm{f}$, which contradicts our assumption about B, since $[G_1 \vee G_2]$ is a c-instance of B. Therefore for $i = 1$ or $i = 2$, Φ verifies every c-instance of B_i, so $\mathscr{M} \vDash B_i$ by inductive hypothesis, so $\mathscr{M} \vDash B$.

Case 4. B has the form $\forall x\, C(x)$. Let t be any closed term. Every c-instance of $C(t)$ is a c-instance of B and so is verified by Φ, so $\mathscr{M} \models C(t)$ by inductive hypothesis. Therefore $\mathscr{M} \models \forall x\, C(x)$, since every individual of $\mathscr{M}$ is the denotation of some closed term.

This completes the inductive argument and establishes that $\mathscr{M} \models D$, so D has a model. □

2.3 AMPLIFICATIONS, MATINGS, AND ACCEPTABILITY. Since we will find terms with which to instantiate quantifiers by applying a unification algorithm, we now wish to shift our attention to those aspects of the problem concerning which actual decisions must be made during the search for a refutation. These decisions concern how many instantiations of each quantifier should be performed, and which literal-occurrences should be made complementary.

We say that a wff F is *normal* iff no variable occurs both free and bound in F, and distinct quantifier-occurrences of F have distinct variables. By appropriate alphabetic changes of bound variables, any wff E can be transformed (or *normalized*) into a normal wff F (called a *normal form of* E) such that $\models [E \equiv F]$.

Let D be a universal nns. We next define the set of *amplifications* of D. We say that a wff R' is obtained from a wff R by *quantifier duplication* iff R' is the result of replacing some wf part of R of the form $\forall x\, M$ by $\forall x\, M \wedge \forall x\, M$. If there is a sequence $D_1, \ldots, D_n$ of wffs (where $n \geq 1$) such that D_{i+1} is obtained from D_i by quantifier duplication for each $i < n$, we say that D_n is obtained from D_1 by a *sequence of quantifier duplications.* Now suppose that E is obtained from D by some sequence of quantifier duplications, F is a normal form of E, and G is the result of deleting all quantifiers of F. Then G is called an *amplification* of D (see Figure 3).

Let G be a quantifier-free wff. We let $\mathscr{L}(G)$ be the set of occurrences of literals in G and $\mathscr{L}(G)^2$ be the set of ordered pairs of elements of $\mathscr{L}(G)$. A *mating* $\mathscr{M}$ of G is a binary relation on $\mathscr{L}(G)$ such that there is a substitution θ such that $\theta K = \sim\theta L$ whenever $L \mathscr{M} K$ (i.e., whenever L and K are mated literal-occurrences). In first-order logic, whenever such a substitution θ exists, there is an essentially unique most general such substitution $\theta_{\mathscr{M}}$, which we call the substitution *associated* with $\mathscr{M}$. We say that $\mathscr{M}$ is a *refutation mating* of G iff G is false with respect to every assignment of truth values to atoms that gives opposite truth values to literals which have mated literal-occurrences.

In Figure 4 we present a mating of line G of Figure 3, which is displayed in two-dimensional format, by drawing lines between mated literal-occurrences. That this is a mating can be verified by comparison with line (H) of Figure 2, where the substitution associated with this mating has been applied. We shall see that it is a refutation mating.

Constructing a mating involves two processes: (a) the *pairing* process, which decides which pairs of literal-occurrences to mate, and (b) the *unification* process, which determines whether there is a substitution which makes mated pairs complementary. The pairing process will need criteria, which we shall call *acceptability criteria*, to decide whether a given mating is a refutation mating. We discuss one acceptability criterion below, but it is clear that the problem of devising new and better acceptability criteria provides a rich field for future research. It is useful for acceptability criteria to have the following properties:

(1) When a mating fails the criteria, they should suggest ways in which the mating might profitably be altered.
(2) The criteria should be compatible with a step-by-step construction of a mating, so that information acquired in checking the criteria at one step can be used at the next step.

(D) $\forall y[(\sim Pc \vee Py) \wedge (\sim Py \vee Pc)]$
$\wedge ([Pd \wedge \sim Pe] \vee [\forall z\, Pz \wedge \forall x \sim Px])$

Duplicate quantifier:

(E) $\forall y[(\sim Pc \vee Py) \wedge (\sim Py \vee Pc)]$
$\wedge \forall y[(\sim Pc \vee Py) \wedge (\sim Py \vee Pc)]$
$\wedge ([Pd \wedge \sim Pe) \vee [\forall z\, Pz \wedge \forall x \sim Px])$

Normalize.

(F) $\forall y[(\sim Pc \vee Py) \wedge (\sim Py \vee Pc)]$
$\wedge \forall w[(\sim Pc \vee Pw) \wedge (\sim Pw \vee Pc)]$
$\wedge ([Pd \wedge \sim Pe] \vee [\forall z\, Pz \wedge \forall x \sim Px])$

Delete quantifiers.

(G) $[(\sim Pc \vee Py) \wedge (\sim Py \vee Pc)]$
$\wedge [(\sim Pc \vee Pw) \wedge (\sim Pw \vee Pc)]$
$\wedge ([Pd \wedge \sim Pe) \vee [Pz \wedge \sim Px])$

FIGURE 3

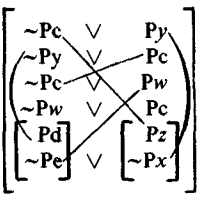

FIGURE 4

(3) The criteria should be compatible with a process in which the construction of a refutation mating is combined with choosing an appropriate amplification of the sentence to be refuted.

We next describe an acceptability criterion which is related to disjunctive normal form, and which is discussed, in slightly different terminology, by Bibel [5] and Prawitz [24].

Motivated by the two-dimensional representation of nnfs discussed earlier, we define a *vertical path* through a quantifier-free nnf G to be a sequence of members of $\mathscr{L}(G)$ which corresponds to one of the disjuncts (conjunctions of literals) in the disjunctive normal form of G. Intuitively, one chooses a vertical path through G by choosing one disjunct (M or N) from each disjunction $[M \vee N]$ of G and deleting all parts of G which are not chosen. One vertical path through Figure 4 contains literals $\sim Pc$, $\sim Py$, $\sim Pc$, $\sim Pw$, Pd, and $\sim Pe$, and another contains literals Py, $\sim Py$, Pw, $\sim Pw$, Pz, and $\sim Px$.

More formally, we can define the vertical paths through G inductively as follows:

(1) If G is a literal L, then the one-term sequence $\langle L \rangle$, whose only term is the given occurrence of L in G, is the only vertical path through G.
(2) If G has the form $[G_1 \vee G_2]$, then every vertical path through G_1, as well as every vertical path through G_2, is a vertical path through G.
(3) If G has the form $[G_1 \wedge G_2]$ and P_i is any vertical path through G_i for $i = 1, 2$, then the concatenation P_1P_2 of these sequences is a vertical path through G.

LEMMA 3. *Let G be a quantifier-free nnf, and let Φ be a truth assignment to the atoms of G. Then Φ verifies G iff there is a vertical path through G, all of whose literals are verified by Φ.*

PROOF. This is easily established by induction, considering the cases where G is a literal or is of the form $[G_1 \vee G_2]$ or $[G_1 \wedge G_2]$. □

Let $\mathscr{M}$ be a mating of a quantifier-free nnf G. We say that $\mathscr{M}$ is *path-acceptable* (*p-acceptable*) iff every vertical path through G contains a mated pair of literal-occurrences.

LEMMA 4. *Let G be a quantifier-free nnf and let $\mathscr{M}$ be a mating of G.*

(*a*) *If there is a substitution θ such that θG is t-f contradictory, then G has a p-acceptable mating.*

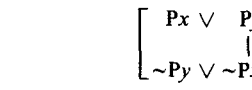

FIGURE 5

FIGURE 6

(*b*) *If $\mathcal{M}$ is p-acceptable, then $\mathcal{M}$ is a refutation mating.*
(*c*) *If $\mathcal{M}$ is a refutation mating, then $\theta_{\mathcal{M}}G$ is t-f contradictory.*

COROLLARY. *Let G be a quantifier-free nnf. The following are equivalent:*

(1) *There is a substitution θ such that θG is t-f contradictory.*
(2) *G has a p-acceptable mating.*
(3) *G has a refutation mating.*

Notice that Figure 5 shows that the converse of (b) is false, and Figure 6 shows that the converse of (c) is false.

PROOF OF LEMMA 4

(a) Suppose θG is t-f contradictory. Let $\mathcal{M} = \{(L, K) \in \mathscr{L}(G)^2 \mid \theta K = \sim\theta L\}$. Clearly $\mathcal{M}$ is a mating of G. To show that $\mathcal{M}$ is p-acceptable, let P be any vertical path through G. P corresponds to a vertical path, which we may call θP, through θG. θP must contain a pair $(\theta L, \theta K)$ of complementary literals (which correspond to literals L and K of G which are on P); otherwise we could assign truth values to atoms of θG so that all literals on θP would be verified, thus verifying θG (by Lemma 3), which is contradictory. Since L and K are mated by $\mathcal{M}$, we see that $\mathcal{M}$ is p-acceptable.

(b) Let $\mathcal{M}$ be p-acceptable. Suppose there is a truth assignment Φ to the atoms in G which verifies G and gives opposite values to mated literals. By Lemma 3 there is a vertical path through G, all of whose literals are verified by Φ. But since every vertical path through G contains a mated pair, this is a contradiction. Therefore no such assignment Φ exists, so $\mathcal{M}$ is a refutation mating.

(c) Follows from the definition of a refutation mating. □

We can now prove two alternative forms of Theorem 1A which establish the completeness and soundness of refutation procedures based on matings.

THEOREM 1B. *Let D be a universal nns of first-order logic. D has no model iff some amplification of D has a refutation mating.*

PROOF. Suppose some amplification of D has a refutation mating. Indeed, let E, F, and G be as in the definition of amplification, and let $\mathcal{M}$ be a refutation mating for G. Let $\forall x_1, \ldots, \forall x_n$ be the quantifiers of F, in the left-to-right order in which they occur in F. It is easy to see that $\models D \equiv E$, $\models E \equiv F$, $\models F \equiv \forall x_1 \cdots \forall x_n G$, and $\models \forall x_1 \cdots \forall x_n G \supset \theta_{\mathcal{M}}G$, so $\models D \supset \theta_{\mathcal{M}}G$. Since $\theta_{\mathcal{M}}G$ is contradictory, $\models \sim D$, so D has no model.

Suppose D has no model. Then by Theorem 1A, D has a c-instance H which is t-f contradictory. It can be seen that there is an amplification G of D and a substitution θ such that $H = \theta G$. By Lemma 4, G has a refutation mating. □

THEOREM 1C. *Let D be a universal nns of first-order logic. D has no model iff some amplification of D has a p-acceptable mating.*

PROOF. By Theorem 1B and the corollary to Lemma 4. □

Figure 4 exhibits a p-acceptable mating of (G) of Figure 3. Thus (D) of Figure 3 has no model, and we see again that Theorem B is valid.

3. *A Refutation Procedure*

We now provide a general outline of a refutation procedure based on p-acceptability. For the sake of generality and expository simplicity, the basic procedure described here is rather naive, although we offer a few suggestions about ways in which it could be made more sophisticated in an actual implementation. Our main purpose here is to provide a framework for discussion of ideas and a starting point for future research.

In particular, we speak as though unifying substitutions are always to be directly computed and applied to the appropriate wffs. In an actual implementation one should consider more sophisticated ways of handling substitutions, as discussed in [11], [27], and [28]. In constructing a mating, both the pairing process and the unification process involve computational effort, and each generates information which can be useful to the other. If the unification process determines that there is no substitution which unifies the atoms of a pair of literals and is compatible with the mating as constructed so far, then the pairing process need not consider adding that pair of literals to the mating. On the other hand, if the pairing process decides directly that it would not be useful to add a given pair of literals to the mating, then the unification algorithm need not consider whether an appropriate substitution exists. Thus it may often be desirable for these processes to work in parallel and to interact with each other, with the balance of effort being determined by the relative efficiencies of the algorithms and the complexities of their tasks. This seems essential when one deals with wffs of higher order logic, where unifying substitutions may not be unique and the unification algorithm may not terminate. However, we regard these matters as implementation details beyond the scope of the present discussion.

When pairing and unification are done in parallel, the pairing process may work with a set of pairs of literal-occurrences which is not known to be a mating, since it is not yet known whether an associated substitution exists. Such a set of pairs is called a *potential mating*. Sometimes, speaking loosely, we refer to a potential mating as a mating.

Choices must be made at various points in the procedure described below, and appropriate heuristics must be used to make these choices. Although the success of a working program will depend crucially on these heuristics, we say little about them here, since more experience with various heuristics is needed.

The fundamental data structures involved in the refutation procedure are the following:

(1) The wff B to be proved.
(2) A universal nns D which is produced from B by step 1 below and thereafter remains fixed. D is called the *initial wff* of the refutation process.
(3) A universal nns F obtained from D by quantifier duplications and alphabetic changes of bound variables. F changes as the procedure progresses. The quantifiers of F simply serve as markers to facilitate additional quantifier duplications and will be ignored much of the time. Thus we regard F as an amplification of D. In an obvious sense, each literal-occurrence of F corresponds to a literal-occurrence of D.
(4) A mating $\mathscr{M}$ of F, and the associated substitution $\theta_{\mathscr{M}}$.
(5) A connection graph $\mathscr{C}(D)$ of D.
(6) A connection graph $\mathscr{C}_{\mathscr{M}}(F)$ of F relative to $\mathscr{M}$.

The connection graphs are defined as follows. Let F be an nnf, and let M and N be in $\mathscr{L}(F)$. M and N are *potential mates* with respect to a mating $\mathscr{M}$ of F iff

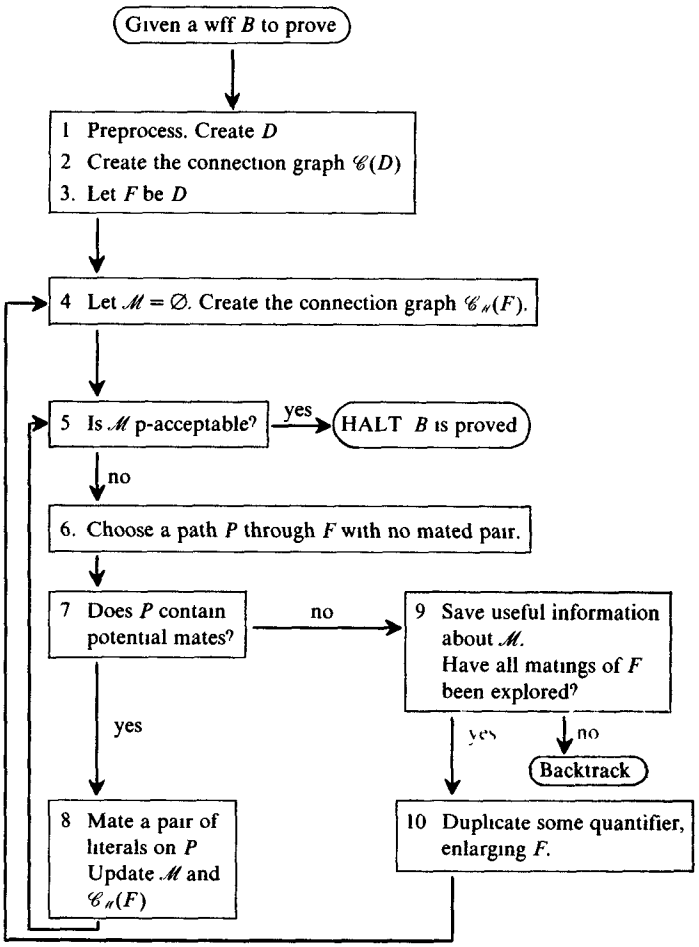

FIG 7. The refutation procedure

some vertical path contains both M and N and there is a substitution σ such that $\sigma(\theta_{\mathscr{M}}N) = \sim\sigma(\theta_{\mathscr{M}}M)$. We define

$$\mathscr{C}_{\mathscr{M}}(F) = \{(M, N) \in \mathscr{L}(F)^2 \mid M \text{ and } N \text{ are potential mates with respect to } \mathscr{M}\}.$$

This is a binary relation, which can be represented as a graph, as in [23]. We define $\mathscr{C}(D)$ to be the connection graph of D with respect to the empty mating.

THE REFUTATION PROCEDURE. The refutation procedure is summarized in Figure 7. We are given a wff B which we wish to show is valid. Let C be a negation normal form of $\sim\bar{B}$, where $\bar{B}$ is the universal closure of B.

Step 1. Preprocessing

(1a) Simplify and normalize C, so as to obtain a nns C_1 which is provably equivalent to C, while minimizing the number of literal-occurrences in C_1 and also minimizing the degrees of the Skolem functions to be introduced in step 1b. Choose between alternative simplified forms.

Thus a wf part of C of the form

$$\forall w \forall x \exists y \exists z [\mathrm{P}wy \wedge \mathrm{Q}xz]$$

should be replaced by

$$\forall w \forall x [\exists y Pwy \wedge \exists z Qxz].$$

However, it serves no purpose to replace $\forall w \exists y \exists z [Pwy \wedge Qyz]$ by $\forall w \exists y [Pwy \wedge \exists z Qyz]$, since in each case the Skolemized form is $\forall w [Pw(fw) \wedge Q(fw)(gw)]$.

(1b) Skolemize to eliminate existential quantifiers. To do this, proceeding sequentially from left to right, replace each wf part $\exists y M(x_1, \ldots, x_n, y)$ of C_1 by $M(x_1, \ldots, x_n, fx_1 \cdots x_n)$, where $x_1, \ldots, x_n$ are the free variables of $\exists y M$, and f is a new n-ary function constant. (If $n = 0$, f is a new individual constant.)

(1c) Push in universal quantifiers so as to obtain a provably equivalent nns in which the scopes of universal quantifiers are as small as possible. For example, replace $\forall x \forall y [Pxy \vee Qy]$ by $\forall y [\forall x Pxy \vee Qy]$. Thus, if C contains a wf part of the form $\forall x \exists y [Pxy \wedge Qxy]$, this is replaced by $\forall x [Px(fx) \wedge Qx(fx)]$ in step 1b, and by $\forall x Px(fx) \wedge \forall z Qz(fz)$ in step 1c.

(1d) Let the sentence obtained by step 1 be called D.

Further discussion of preprocessing can be found in [14].

Step 2. Create the connection graph $\mathscr{C}(D)$.

In an actual implementation of this procedure, it might be useful not to create $\mathscr{C}(D)$ all at once, but to compute and store parts of it as the information is needed. The important point is to avoid the necessity of computing this information more than once. A similar comment applies to $\mathscr{C}_{\mathscr{M}}(F)$ below.

Step 3. Let F be D.

Step 4. Let $\mathscr{M}$ be the empty mating of F. Create $\mathscr{C}_{\mathscr{M}}(F)$, using $\mathscr{C}(D)$.

Step 5. Test $\mathscr{M}$. Is $\mathscr{M}$ p-acceptable? If so, the refutation is complete. Otherwise, continue to step 6.

We remark that since the number of vertical paths in a wff can be quite large, considerable attention should be paid to the efficiency with which this test is carried out. Of course, one need not examine the vertical paths separately and completely. As soon as one has found a pair of mated literals on a subpath of a vertical path, one can exclude from further consideration all extensions of that subpath. Also, since $\mathscr{M}$ is tested and constructed in stages, information about which vertical paths contain mated pairs can be saved from one stage to the next. At each stage, all one really does is search for one new vertical path which does not contain any mated pair, starting where one left off at the last stage.

Various methods from propositional calculus can be used to simplify F temporarily as the construction of $\mathscr{M}$ progresses, and we digress briefly to discuss two of these. Let M be the wff

$$([L_1 \wedge P_1] \vee \cdots \vee [L_n \wedge P_n]) \wedge ([L'_1 \wedge \cdots \wedge L'_n \wedge Q] \vee R),$$

where $n \geq 1$, the L_i and L'_i are literals, and the P_i, Q, and R are arbitrary wffs; in particular, they may be the empty conjunction $\triangle$ (which is true) or the empty disjunction $\square$ (which is false). Let N be the wff $([L_1 \wedge P_1] \vee \cdots \vee [L_n \wedge P_n]) \wedge R$. Suppose F contains an occurrence of M, and F^* is obtained from F by replacing that occurrence of M by an occurrence of N. (See Figure 8. Note that if $L'_i = \sim L_i$ for each i, then $\vDash F \equiv F^*$. Also note what the reduction from F to F^* looks like when $n = 1$ or R is $\square$.) Let $\mathscr{M}$ be a mating of F such that for each $i \leq n$, $L_i \mathscr{M} L'_i$ or $L'_i \mathscr{M} L_i$. There

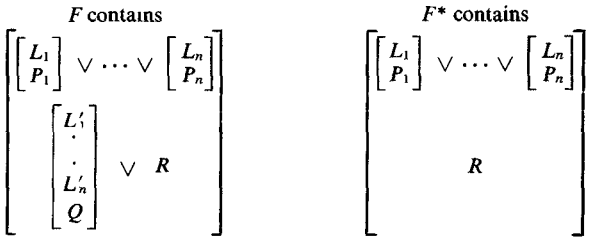

FIGURE 8

is a natural correspondence between $\mathscr{L}(\theta_{\mathscr{M}}F^*)$ and a subset of $\mathscr{L}(F)$. Let $\mathscr{M}^*$ be the mating of $\theta_{\mathscr{M}}F^*$ induced by $\mathscr{M}$ under this correspondence. It is easy to see that there is a p-acceptable mating of F which is an extension (superset) of $\mathscr{M}$ if and only if there is a p-acceptable mating of $\theta_{\mathscr{M}}F^*$ which is an extension of $\mathscr{M}^*$. Thus we may reduce F and $\mathscr{M}$ to $\theta_{\mathscr{M}}F^*$ and $\mathscr{M}^*$ in our search for a p-acceptable mating (though F and $\mathscr{M}$ may have to be restored later if no p-acceptable extension of $\mathscr{M}^*$ is found).

Of course, we shall feel free to use elementary laws of conjunction and disjunction, such as commutativity, associativity, the idempotent laws, $M \vee \square \equiv M$, $M \wedge \square \equiv \square$, and $M \wedge \triangle \equiv M$. Using these laws and reductions of the sort just discussed, we can often drastically simplify F, or even reduce it to $\square$, which has a p-acceptable mating since it has no vertical paths. This is illustrated in Figures 9 and 10, where we assume complementary literals are mated. In both figures, (a′) is obtained from (a) by elementary laws, (b) is obtained from (a′) by a reduction, etc. The dotted lines surround the parts of the wff involved in the reduction steps. Figure 9 represents the mating for Theorem B in Figure 4.

Another reduction of the same general type, which was introduced by Prawitz [24] for reducing matrices in conjunctive normal form, is to replace $(L \vee Q) \wedge (L' \vee R)$ by $[L \wedge R] \vee [L' \wedge Q]$, where L and L' are mated literals. See Figure 11.

Both of these reductions reduce the number of vertical paths in the nnf, and together they can be used to establish that many nnfs of propositional calculus are contradictory. Note, however, that neither reduction is applicable to the contradictory nnf in Figure 12.

Step 6. Choose a vertical path P through F with no mated pair.

Presumably at least one such path can be found in step 5. However, it may be worthwhile to choose this path carefully. For example, with the aid of the connection graph $\mathscr{C}_{\mathscr{M}}(F)$ one can seek such a path with a minimum number of pairs of potential mates.

Step 7. Is there at least one pair of potential mates on P? If so, go to step 8. If not, go to step 9. It may be a useful heuristic to give P high priority for consideration by step 6 at later stages of the procedure.

Step 8. Choose a pair (M, N) of potential mates on P, and replace $\mathscr{M}$ by $\mathscr{M} \cup \{(M, N)\}$. (We call this process *mating* M and N.) Adjust $\theta_{\mathscr{M}}$ and $\mathscr{C}_{\mathscr{M}}(F)$ appropriately. Return to step 5.

We remark that in steps 5–8 we simply check whether $\mathscr{M}$ is acceptable and change it if it is not. The particular way we change it is guided by our criterion of acceptability. We add a pair of literals to $\mathscr{M}$ because it makes $\mathscr{M}$ closer to being acceptable.

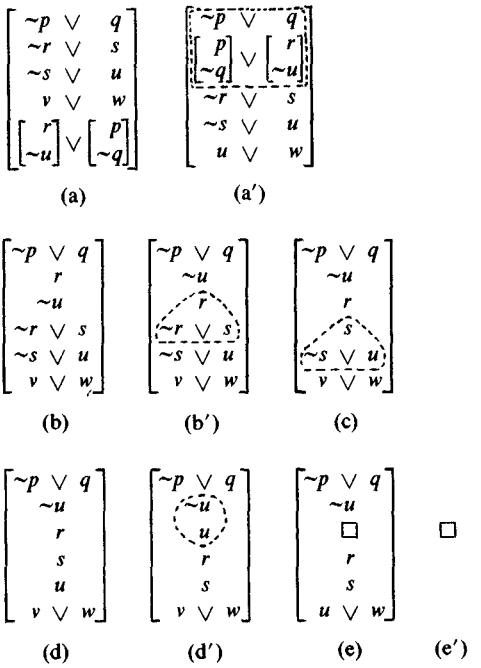

FIGURE 9

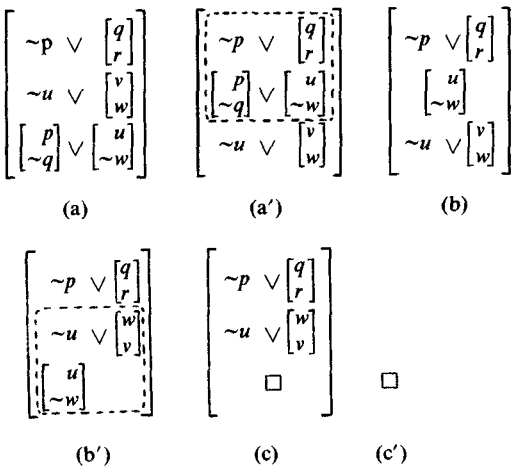

FIGURE 10

Of course, if an inappropriate choice is made at step 8, the system will eventually have to backtrack. To alleviate this problem, more sophisticated methods of deciding how to alter $\mathscr{M}$ than those implicit in steps 6–8 may be needed. We briefly mention one such method, which was suggested by Eve Cohen.

Let us say that we *fix* a vertical path P by mating a pair of literal-occurrences on P. While there may be many ways of fixing a given path, some of these may not be compatible with any way of fixing another path (because the associated substitutions are not compatible) and so should not be used. By considering all possible ways of simultaneously fixing all paths in a set of vertical paths, starting with a unit set and progressively enlarging it, one can eliminate many inappropriate ways of fixing paths.

F contains $\begin{bmatrix} L \vee Q \\ L' \vee R \end{bmatrix}$ F^* contains $\begin{bmatrix} L \\ R \end{bmatrix} \vee \begin{bmatrix} L' \\ Q \end{bmatrix}$

FIGURE 11

$$\left[\begin{array}{l} \begin{bmatrix} p \\ w \end{bmatrix} \vee \begin{bmatrix} q \\ x \end{bmatrix} \\ \sim p \vee \begin{bmatrix} q \vee \sim w \\ x \end{bmatrix} \\ \begin{bmatrix} r \\ y \end{bmatrix} \vee \sim q \\ \sim r \vee \begin{bmatrix} \sim x \vee \sim y \\ w \vee q \end{bmatrix} \end{array}\right]$$

FIGURE 12

If, within the computational resources allocated, one can eliminate all but one way of simultaneously fixing all paths in the set, one can fix all those paths simultaneously without fear of error.

We can also apply to steps 8 and 6 a heuristic which is analogous to the set-of-support strategy [31] for resolution. Not all literal-occurrences need have mates in a p-acceptable mating (particularly if inappropriate quantifier duplications have occurred), but it is sometimes clear from the statement of a theorem that certain literal-occurrences must have mates, or that most literal-occurrences in a certain set must have mates. In such cases, high priority should be given to mating such literal-occurrences.

Step 9. Have all matings of F been explored? If so, go to step 10. If not, backtrack and try another mating. In either case, record information about the current mating which may eventually be useful in choosing a quantifier to duplicate or in reconstructing this mating after a quantifier has been duplicated.

Step 10. Choose a quantifier of F, duplicate it, normalize the resulting wff, and call it F. Go to step 4.

The choice of a quantifier to duplicate should be made as intelligently as possible, since every quantifier duplication enlarges the wff with which the mating program must deal. Nevertheless, an inappropriate quantifier duplication does not prevent an acceptable mating from eventually being found, so we never backtrack past this point. Of course, the heuristic used to choose quantifiers to duplicate should be designed so that the procedure will be complete.

Sometimes it may be desirable to go from step 9 to step 10 even if further matings of F remain to be explored, or to go from step 10 directly to step 7 and continue work on the current mating in the enlarged wff. Of course, such acts may complicate backtracking and necessitate special measures to ensure the completeness of the procedure.

4. *Quantifier Duplication*

Since the complexity of the search for an acceptable mating grows drastically as the number of literal-occurrences in the wff F increases, it is worthwhile to devote considerable effort to keeping this number small. Thus we make the scopes of universal quantifiers as small as possible so that duplications of these quantifiers will not create more new literal-occurrences than necessary.

If the refutation procedure is to be complete, one must duplicate quantifiers in a systematic way as the search progresses. One would really like a way to know just how many times each quantifier needs to be duplicated, but such knowledge could be used to construct a decision procedure for first-order logic and is therefore not

attainable in general. Nevertheless, some procedures for duplicating quantifiers are better than others.

To ensure completeness of the refutation procedure, one could proceed by stages and at each stage duplicate each quantifier in the current wff (working from right to left through the wff). However, if a literal-occurrence is in the scope of n universal quantifiers, this would produce 2^n copies of it at the next stage, which might be excessive. A scheme which would produce just one additional copy of each literal-occurrence (which is in the scope of some quantifier) is to duplicate only those quantifiers (called *outermost* quantifiers) which are not in the scopes of other quantifiers. The question naturally arises whether such a procedure is complete. We shall show that it is, but first we need some more information about instances of a wff.

We define a *simple instance* of a universal nnf D to be the result of replacing each wf part of D of the form $\forall x\, B(x)$ by $B(t)$ for some closed term t. (Thus simple instances are c-instances where n (in the definition of c-instances) is always 1.)

LEMMA 5. *Let D be a universal nns, and let H be any c-instance of D. Then H is equivalent to some conjunction of simple instances of D.*

PROOF. By induction on the number of occurrences of $\wedge$, $\vee$, and $\forall$ in D.

Case 1. D is a literal. Then H is D, and a simple instance of D.

Case 2. D is $[D_1 * D_2]$, where $*$ is $\wedge$ or $\vee$, and H is $[H_1 * H_2]$, where H_i is a c-instance of D_i for $i = 1, 2$. By inductive hypothesis, $\vdash H_1 \equiv \bigwedge_{j=1}^{p} J_j$ and $\vdash H_2 \equiv \bigwedge_{k=1}^{q} K_k$, where each J_j is a simple instance of D_1 and each K_k is a simple instance of D_2. It can be seen that $\bigwedge_{j=1}^{p} \bigwedge_{k=1}^{q} [J_j * K_k]$ is a conjunction of simple instances of D which is equivalent to H.

Case 3. D is $\forall x\, C(x)$. Then H is $\bigwedge_{i=1}^{n} H_i$, where H_i is a c-instance of $C(t_i)$ and t_i is a closed term for $1 \leq i \leq n$. By the inductive hypothesis, $\vdash H_i \equiv \bigwedge_{j=1}^{p_i} J_{ij}$, where J_{ij} is some simple instance of $C(t_i)$ for each i and j. Thus $\bigwedge_{i=1}^{n} \bigwedge_{j=1}^{p_i} J_{ij}$ is a conjunction of simple instances of D which is equivalent to H. □

THEOREM 2. *Let D be a universal nns of first-order logic. D has no model iff there is a refutation mating for some amplification G of D such that in forming G from D only outermost quantifiers are duplicated.*

PROOF. Suppose D has no model. Then by Theorem 1A, D has a c-instance H which is t-f contradictory. Let us designate as *molecules* of D those wf parts of D which are in the scope of no quantifier or negation of D and properly contain no other such wf parts. (For example, if D is $\forall x\, Px \vee [\forall y[Qy \vee Pa] \wedge \sim Pb]$, the molecules of D are $\forall x\, Px$, $\forall y[Qy \vee Pa]$, and $\sim Pb$.) Let $M_1, \ldots, M_k$, where $k \geq 1$, be the occurrences of molecules of D. D can be built up from the M_i using $\wedge$ and $\vee$, so there is a wff $B(p_1, \ldots, p_k)$ of propositional calculus, containing only the connectives $\wedge$ and $\vee$ and only the propositional variables $p_1, \ldots, p_k$, such that D is $B(M_1, \ldots, M_k)$. Thus H is $B(H_1, \ldots, H_k)$, where H_i is a c-instance of M_i for each i. By Lemma 5 each H_i is equivalent to $\bigwedge_{j=1}^{n_i} K_{ij}$, where K_{ij} is some simple instance of M_i for each $j \leq n_i$ and each $i \leq k$. Hence $B(\bigwedge_{j=1}^{n_1} K_{1j}, \ldots, \bigwedge_{j=1}^{n_k} K_{kj})$, which we call J, is contradictory. Let E be obtained from D by duplicating the outermost quantifier of M_i enough times to create n_i copies of M_i for each $i \leq k$. Let G be obtained from E by normalizing and deleting quantifiers. It can be seen that there is a substitution θ such that $J = \theta G$. By Lemma 4, G has a refutation mating.

The proof in the other direction follows by Theorem 1B. □

While the procedure outlined above is complete, it is rather primitive, since no information obtained in the search for an acceptable mating is used to choose the quantifier to duplicate. The following very simple example illustrates a possible disadvantage of duplicating only outermost quantifiers. Suppose the wff contains the molecule $\forall x (Px \vee \forall y\, Qxy)$ and to obtain an acceptable mating this molecule must be expanded to $\forall x (Px \vee [\forall y\, Qxy \wedge \forall z\, Qxz])$. Duplicating the outermost quantifier will instead yield

$$\forall x (Px \vee \forall y\, Qxy) \wedge \forall w (Pw \vee \forall z\, Qwz).$$

This will also permit an acceptable mating to be found (since Px and Pw can be given the same mates), but the search for the mating will be more complicated in this case.

One would like to develop a set of heuristics for duplicating quantifiers. An example of such a heuristic for first-order logic is the following: Suppose L is a literal-occurrence and $P_1, \ldots, P_n$ are vertical paths such that for each $i \leq n$, L occurs in every pair of potential mates which fix P_i, but no mating simultaneously fixes $P_1, \ldots,$ and P_n. Then duplicate some quantifier with L in its scope.

5. *Implementation*

A computer program based on these ideas has been constructed by Eve Cohen, with assistance from Harry Porta and Dale Miller. The program handles sentences of type theory as well as those of first-order logic, and uses Huet's unification algorithm [22] for type theory. Of course, it is not logically complete for type theory.

Experimentation with and development of the program are continuing, but the program has already been able to prove a number of theorems, such as Theorem A, which seemed quite intractable (within reasonable restrictions on time and memory) for an earlier program [2] (based on [1]) which reduced wffs to clauses. Appendix B gives a sampling of theorems proved by the program.

It has been found that if the theorem to be proved contains $\equiv$ as an abbreviation, the way this abbreviation is eliminated can significantly affect the time required to prove the theorem. Every wf part of the form $[M \equiv N]$ may be replaced by $[(\sim M \vee N) \wedge (M \vee \sim N)]$ or by $([M \wedge N] \vee [\sim M \wedge \sim N])$. As can be seen from Figure 13, which choice is made can affect the number of vertical paths in the wff. (Note that if M is a complex formula which initially contains quantifiers, a naive system may spend considerable time fixing all vertical paths which go through both M and $\sim M$.) One can minimize the number of vertical paths by treating each occurrence of $\equiv$ appropriately. (Positive and negative occurrences should be handled differently.) We illustrate this (in a rather naive way, since the example is really too simple) in Figure 14, where we find an amplification for Theorem B slightly different from that obtained in Figures 2–4. Figure 4 has 32 vertical paths, but G in Figure 14 has only 8.

Let Theorem C be $\exists x\, \forall y\, [Px \equiv Py] \equiv [\exists x\, Px \equiv \forall y\, Py]$. Treating $\equiv$ naively (uniformly), the computer produced a complete automatic proof of Theorem C in 37 seconds. With the same heuristics but with $\equiv$ handled so as to minimize the number of vertical paths, the computer required only 21 seconds to prove the theorem.

Similar considerations govern the production of clauses. Let Theorem D be

$$\begin{aligned}&(\exists x\, \forall y\, [Px \equiv Py] \equiv [\exists x\, Qx \equiv \forall y\, Py])\\ &\quad \equiv (\exists x\, \forall y\, [Qx \equiv Qy] \equiv [\exists x\, Px \equiv \forall y\, Qy])\end{aligned}$$

(which is indeed valid). If one eliminates $\equiv$ from the negation of Theorem D naively,

$$\begin{bmatrix} \sim M \vee N \\ M \vee \sim N \end{bmatrix} \qquad \begin{bmatrix} M \\ N \end{bmatrix} \vee \begin{bmatrix} \sim M \\ \sim N \end{bmatrix}$$

FIG. 13. Two ways of treating $[M \equiv N]$

Theorem B.

(B) $\exists x\, \forall y\, [Px \equiv Py] \supset [\exists x\, Px \equiv \forall y\, Py]$

Negate, and eliminate $\equiv$ and $\supset$.

(C) $\exists x\, \forall y\, ([Px \wedge Py] \vee [\sim Px \wedge \sim Py])$
$\wedge\, [\exists x\, Px \wedge \exists y\, \sim Py] \vee [\forall y\, Py \wedge \forall x\, \sim Px])$

Skolemize.

(D) $\forall y\, ([Pc \wedge Py] \vee [\sim Pc \wedge \sim Py])$
$\wedge\, ([Pd \wedge \sim Pe] \vee [\forall z\, Pz \wedge \forall x\, \sim Px])$

Duplicate quantifier and normalize.

(F) $\forall y\, ([Pc \wedge Py] \vee [\sim Pc \wedge \sim Py])$
$\wedge\, \forall w\, ([Pc \wedge Pw] \vee [\sim Pc \wedge \sim Pw])$
$\wedge\, ([Pd \wedge \sim Pe] \vee [\forall z\, Pz \wedge \forall x\, \sim Px])$

Delete quantifiers:

(G)
$$\left[\begin{bmatrix} Pc \\ Py \end{bmatrix} \vee \begin{bmatrix} \sim Pc \\ \sim Py \end{bmatrix}\right] \wedge \left[\begin{bmatrix} Pc \\ Pw \end{bmatrix} \vee \begin{bmatrix} \sim Pc \\ \sim Pw \end{bmatrix}\right] \wedge \left[\begin{bmatrix} Pd \\ \sim Pe \end{bmatrix} \vee \begin{bmatrix} Pz \\ \sim Px \end{bmatrix}\right]$$

FIGURE 14

one gets 2704 clauses; if one does it so as to minimize the number of clauses, one gets 128 clauses; if one does it so as to minimize the number of vertical paths (and thereby maximize the number of clauses), one gets 16,384 clauses! Of course, in propositional calculus the extra clauses would all be tautologies, but this is not the case for first-order logic.

Experience with the program has revealed certain problems which must be dealt with to obtain an efficient implementation, which we mention briefly.

A mating can be built up in a multitude of ways, differing in the orders in which pairs of literal-occurrences are put in. It is important to have a redundancy check, so that a mating is not constructed if it, or a subset of it, has previously been rejected.

Symmetries in the wff under consideration also produce redundant matings. Symmetries may occur in the original statement of the theorem, and they are created whenever quantifiers are duplicated. Symmetries [29] can be regarded as automorphisms of $\mathscr{L}(F)$ under which the wff F is carried into an alphabetic variant of itself. For example, suppose a wf part $\forall x A(x)$ has been duplicated to form $\forall x A(x) \wedge \forall y A(y)$. Let $\tau : \mathscr{L}(F) \to \mathscr{L}(F)$ be defined so that τL = the copy of L in $A(y)$ (respectively, $A(x)$) if L is in $A(x)$ (respectively, $A(y)$), and $\tau L = L$ if L is not in $A(x)$ or $A(y)$. Given any such symmetry τ and a mating $\mathscr{M}$, there is a mating

$$\tau\mathscr{M} = \{(\tau L, \tau K) \mid (L, K) \in \mathscr{M}\}$$

which is symmetrical to $\mathscr{M}$. Clearly $\tau\mathscr{M}$ is acceptable iff $\mathscr{M}$ is. Thus a mating should be rejected if a mating symmetrical to it, or a subset of it, has already been rejected.

Updating the connection graph as the mating is constructed is very beneficial in limiting the search. However, to facilitate backtracking, one must keep track of the connection graph for each step of the mating process to which one might backtrack, and this might cause storage problems. One way to keep track of the information can be roughly described as follows. The original connection graph can be represented as a list of pairs, which can be rearranged at will. Suppose the current mating has been

constructed in n stages, with a pair having been added at each stage, and let θ_i be the substitution associated with the mating at the ith stage, for $i = 1, \ldots, n$. Let S_i be a list of the pairs which have been found to be incompatible with θ_i (i.e., pairs (M, N) such that $\theta_i N$ and $\sim\theta_i M$ are not unifiable), but not with θ_j for any $j < i$, and let T be a list of the pairs in the connection graph which are not in $\bigcup_{i=1}^{n} S_i$. Store the connection graph in the order T, S_n, S_{n-1}, ..., S_2, S_1, with markers to separate the sublists. Then when one backtracks from stage n to stage $n-1$, one simply removes the marker between T and S_n. Of course, one need not update the entire connection graph at each stage; one can simply keep pairs in T until one has occasion to examine them.

Good ways of handling wffs containing = must still be devised. If the system is not restricted to first-order logic, one can define $[A = B]$ as $\forall p[\sim pA \vee pB]$, where p is a predicate variable. Literals whose atoms start with predicate variables can be mated with arbitrary literals, so the large number of potential mates for literals such as $\sim pA$ and pB produces an enormous number of matings which must be explored when seeking a proof of a theorem in which = occurs frequently. However, once one chooses a substitution for p which mates $\sim pA$ with some literal, the possible mates for pB become quite limited. Therefore, mates for $\sim pA$ and pB should be chosen simultaneously. This heuristic for handling = has proved quite helpful, though additional heuristics are needed.

Appendix A

Here we show how Theorem A is processed. Theorem A is

$$[\#F_{\alpha\beta} \cdot R_{o\beta} \cup S_{o\beta}] \equiv \cdot[\#F_{\alpha\beta}R_{o\beta}] \cup \cdot\#F_{\alpha\beta}S_{o\beta}$$

NEGATE:

(1) $[\sim\cdot[[\#F_{\alpha\beta}] \cdot R_{o\beta} \cup S_{o\beta}] \equiv \cdot[[\#F_{\alpha\beta}]R_{o\beta}] \cup \cdot[[\#F_{\alpha\beta}]S_{o\beta}]$

$\equiv$, or equality between sets, is defined as $[\lambda P_{o\alpha} Q_{o\alpha} \forall x_\alpha [P_{o\alpha}x_\alpha \equiv Q_{o\alpha}x_\alpha]]$.
INSTANTIATE $\equiv$, and eliminate the propositional connective $\equiv$ in a way which will minimize the number of clauses to be produced:

(2) $\exists x_\alpha^1 \cdot [\sim[[\#F_{\alpha\beta}] \cdot R_{o\beta} \cup S_{o\beta}]x_\alpha^1 \vee \sim[[[\#F_{\alpha\beta}]R_{o\beta}] \cup \cdot[\#F_{\alpha\beta}]S_{o\beta}]x_\alpha^1]$
$\wedge [[[\#F_{\alpha\beta}] \cdot R_{o\beta} \cup S_{o\beta}]x_\alpha^1 \vee [[[\#F_{\alpha\beta}]R_{o\beta}] \cup \cdot[\#F_{\alpha\beta}]S_{o\beta}]x_\alpha^1]$

$\cup$ is defined as $[\lambda P_{o\alpha} \lambda Q_{o\alpha} \lambda x_\alpha [P_{o\alpha}x_\alpha \vee Q_{o\alpha}x_\alpha]]$.
INSTANTIATE $\cup$:

(3) $\exists x_\alpha^1 \cdot [\sim[[\#F_{\alpha\beta}] \cdot \lambda x_\beta^2 \cdot R_{o\beta}x_\beta^2 \vee \cdot S_{o\beta}x_\beta^2]x_\alpha^1$
$\vee \cdot\sim[[\#F_{\alpha\beta}]R_{o\beta}]x_\alpha^1 \wedge \sim[[\#F_{\alpha\beta}]S_{o\beta}]x_\alpha^1]$
$\wedge \cdot[\#F_{\alpha\beta}][\lambda x_\beta^4 \cdot R_{o\beta}x_\beta^4 \vee \cdot S_{o\beta}x_\beta^4]x_\alpha^1$
$\vee \cdot[[\#F_{\alpha\beta}]R_{o\beta}]x_\alpha^1 \vee [[\#F_{\alpha\beta}]S_{o\beta}]x_\alpha^1$

$\#$ is defined as $[\lambda F_{\alpha\beta} \lambda D_{o\beta} \lambda y_\alpha \cdot \exists x_\beta [D_{o\beta}x_\beta \wedge \cdot y_\alpha = F_{\alpha\beta}x_\beta]]$.
INSTANTIATE $\#$:

(4) $\exists x_\alpha^1 \cdot [[\forall x_\beta^6 \cdot [\sim R_{o\beta}x_\beta^6 \wedge \sim S_{o\beta}x_\beta^6] \vee \cdot\sim\cdot x_\alpha^1 = \cdot F_{\alpha\beta}x_\beta^6]$
$\vee \cdot[\forall x_\beta^7 \cdot \sim R_{o\beta}x_\beta^7 \vee \sim\cdot x_\alpha^1 = \cdot F_{\alpha\beta}x_\beta^7]$
$\wedge \cdot\forall x_\beta^8 \cdot \sim S_{o\beta}x_\beta^8 \vee \sim\cdot x_\alpha^1 = \cdot F_{\alpha\beta}x_\beta^8]$
$\wedge \cdot[\exists x_\beta^9 \cdot [R_{o\beta}x_\beta^9 \vee S_{o\beta}x_\beta^9] \wedge \cdot x_\alpha^1 = \cdot F_{\alpha\beta}x_\beta^9]$
$\vee \cdot[\exists x_\beta^{10} \cdot R_{o\beta}x_\beta^{10} \wedge \cdot x_\alpha^1 = \cdot F_{\alpha\beta}x_\beta^{10}]$
$\vee \cdot\exists x_\beta^{11} \cdot S_{o\beta}x_\beta^{11} \wedge \cdot x_\alpha^1 = \cdot F_{\alpha\beta}x_\beta^{11}$

$=$ is defined as $[\lambda x_\alpha \lambda y_\alpha \cdot \forall P_{o\alpha}[\sim P_{o\alpha}x_\alpha \vee P_{o\alpha}y_\alpha]]$.

INSTANTIATE =:

$$(5)\quad \exists x^1_\alpha \cdot [[\forall x^6_\beta \cdot [\sim \mathrm{R}_{o\beta} x^6_\beta \wedge \cdot \sim \cdot \mathrm{S}_{o\beta} x^6_\beta]$$
$$\vee \cdot \exists P^6_{o\beta} \cdot P^6_{o\alpha} x^1_\alpha \wedge \sim P^6_{o\alpha} \cdot \mathrm{F}_{\alpha\beta} x^6_\beta]$$
$$\vee \cdot [\forall x^7_\beta \cdot \sim \mathrm{R}_{o\beta} x^7_\beta \vee \cdot \exists P^7_{o\alpha} \cdot P^7_{o\alpha} x^1_\alpha \wedge \sim P^7_{o\alpha} \cdot \mathrm{F}_{\alpha\beta} x^7_\beta]$$
$$\wedge \cdot \forall x^8_\beta \cdot \sim \mathrm{S}_{o\beta} x^8_\beta \vee \cdot \exists P^8_{o\alpha} \cdot P^8_{o\alpha} x^1_\alpha \wedge \sim P^8_{o\alpha} \cdot \mathrm{F}_{\alpha\beta} x^8_\beta]$$
$$\wedge \cdot [\exists x^9_\beta \cdot [\mathrm{R}_{o\beta} x^9_\beta \vee \mathrm{S}_{o\beta} x^9_\beta] \wedge \cdot \forall P^9_{o\alpha} \cdot \sim P^9_{o\alpha} x^1_\alpha \vee \cdot P^9_{o\alpha} \cdot \mathrm{F}_{\alpha\beta} x^9_\beta]$$
$$\vee \cdot [\exists x^{10}_\beta \cdot \mathrm{R}_{o\beta} x^{10}_\beta \wedge \cdot \forall P^{10}_{o\alpha} \cdot \sim P^{10}_{o\alpha} x^1_\alpha \vee P^{10}_{o\alpha} \cdot \mathrm{F}_{\alpha\beta} x^{10}_\beta]$$
$$\vee \cdot \exists x^{11}_\beta \cdot \mathrm{S}_{o\beta} x^{11}_\beta \wedge \cdot \forall P^{11}_{o\alpha} \cdot \sim P^{11}_{o\alpha} x^1_\alpha \vee P^{11}_{o\alpha} \cdot \mathrm{F}_{\alpha\beta} x^{11}_\beta$$

SKOLEMIZE:

$$(6)\quad [[\forall x^6_\beta \cdot [\sim \mathrm{R}_{o\beta} x^6_\beta \wedge \sim \mathrm{S}_{o\beta} x^6_\beta]$$
$$\vee \cdot \mathrm{P}^6_{o\alpha\beta} x^6_\beta \mathrm{x}^1_\alpha \wedge \sim \cdot \mathrm{P}^6_{o\alpha\beta} x^6_\beta \cdot \mathrm{F}_{\alpha\beta} x^6_\beta]$$
$$\vee \cdot [\forall x^7_\beta \cdot \sim \mathrm{R}_{o\beta} x^7_\beta \vee \cdot \mathrm{P}^7_{o\alpha\beta} x^7_\beta \mathrm{x}^1_\alpha \wedge \sim \mathrm{P}^7_{o\alpha\beta} x^7_\beta \cdot \mathrm{F}_{\alpha\beta} x^7_\beta]$$
$$\wedge \cdot \forall x^8_\beta \cdot \sim \mathrm{S}_{o\beta} x^8_\beta \vee \cdot \mathrm{P}^8_{o\alpha\beta} x^8_\beta \mathrm{x}^1_\alpha \wedge \sim \mathrm{P}^8_{o\alpha\beta} x^8_\beta \cdot \mathrm{F}_{\alpha\beta} x^8_\beta]$$
$$\wedge \cdot [[\mathrm{R}_{o\beta} \mathrm{x}^9_\beta \vee \mathrm{S}_{o\beta} \mathrm{x}^9_\beta] \wedge \cdot \forall P^9_{o\alpha} \cdot \sim P^9_{o\alpha} \mathrm{X}^1_\alpha \vee P^9_{o\alpha} \cdot \mathrm{F}_{\alpha\beta} \mathrm{x}^9_\beta]$$
$$\vee \cdot [\mathrm{R}_{o\beta} \mathrm{x}^{10}_\beta \wedge \cdot \forall P^{10}_{o\alpha} \cdot \sim P^{10}_{o\alpha} \mathrm{x}^1_\alpha \vee P^{10}_{o\alpha} \cdot \mathrm{F}_{\alpha\beta} \mathrm{x}^{10}_\beta]$$
$$\vee \cdot \mathrm{S}_{o\beta} \mathrm{x}^{11}_\beta \wedge \cdot \forall P^{11}_{o\alpha} \cdot \sim P^{11}_{o\alpha} \mathrm{x}^1_\alpha$$

Deleting the quantifiers yields (2) of Figure 1.

Appendix B

Here we list some theorems of first-order and higher order logic which have been proven in automatic mode by the program mentioned in Section 5 using the naive scheme of duplicating outermost quantifiers which was discussed in Section 4. No heuristic for dealing with symmetries was used. The times given for the proofs are not of great significance, since the theorems were proven with various heuristics at various stages in the refinement of the program.

Some of the definitions which appear in these theorems were explained in Appendix A. Explanations of the others are as follows:

COMMUTATIVE is defined as $[\lambda H_{\alpha\beta\beta} \cdot \forall X_\beta \forall Y_\beta \cdot [[H_{\alpha\beta\beta} X_\beta] Y_\beta] = \cdot [H_{\alpha\beta\beta} Y_\beta] X_\beta]$.

$\circ$ (composition of functions) is defined as $[\lambda F_{\alpha\beta} \lambda G_{\beta\gamma} \lambda X_\gamma \cdot F_{\alpha\beta} \cdot G_{\beta\gamma} X_\gamma]$.

$\mathscr{P}$ (the powerset operation) is defined as $[\lambda P_{o\alpha} \lambda Q_{o\alpha} \cdot Q_{o\alpha} \subseteq P_{o\alpha}]$.

$\subseteq$ is defined as $[\lambda P_{o\alpha} \lambda Q_{o\alpha} \cdot \forall X_\alpha \cdot [\sim \cdot P_{o\alpha} X_\alpha] \vee \cdot Q_{o\alpha} X_\alpha]$.

$\cap$ is defined as $[\lambda P_{o\alpha} \lambda Q_{o\alpha} \lambda X_\alpha \cdot [P_{o\alpha} X_\alpha] \wedge \cdot Q_{o\alpha} X_\alpha]$.

Theorems Proved by the System

THM5 (Cantor's Theorem for Sets. The power set of a set has more members than the set.)

$$\sim \exists G_{o\iota} \forall F_{o\iota} \exists J_\iota \cdot [G_{o\iota} J_\iota] = F_{o\iota} \quad (11 \text{ seconds})$$

THM24 (The computer must find a noncommutative operator.)

$$\sim [U_\iota = V_\iota] \supset \exists G_{\iota(\iota)(\iota)} \cdot \sim \text{COMMUTATIVE } G_{\iota(\iota)(\iota)} \quad (11 \text{ seconds})$$

THM25 (This is Quine's modification of Russell's paradox; let Rxy mean $x \in y$.)

$$\sim \exists Y_\iota \forall X_\iota \cdot [R_{o\iota} X_\iota Y_\iota] \equiv \sim \exists Z_\iota \cdot [R_{o\iota} X_\iota Z_\iota] \wedge [R_{o\iota} Z_\iota X_\iota] \quad (10 \text{ seconds})$$

THM29

$$[\# F_{\alpha\beta} \cdot \# G_{\beta\gamma} S_{o\gamma}] \equiv \cdot \# [F_{\alpha\beta} \circ G_{\beta\gamma}] S_{o\gamma} \quad (26 \text{ seconds})$$

THM34 (Theorem A)

$$[\#F_{\alpha\beta} \cdot R_{o\beta} \cup S_{o\beta}] \equiv \cdot[\#F_{\alpha\beta}R_{o\beta}] \cup \cdot\#F_{\alpha\beta}S_{o\beta} \quad \text{(243 seconds)}$$

THM36

$$[R_{o\iota} = S_{o\iota}] \supset \cdot R_{o\iota} \subseteq S_{o\iota} \quad \text{(5 seconds)}$$

THM44 (Useful in eliminating a quantifier on a second-order variable)

$$\exists S_{o\iota} \forall X_\iota [[[S_{o\iota}X_\iota] \vee \cdot P_{o\iota}X_\iota] \wedge \cdot[\sim S_{o\iota}X_\iota] \vee \cdot Q_{o\iota}X_\iota] \equiv \forall Y_\iota \cdot [P_{o\iota}Y_\iota] \vee \cdot Q_{o\iota}Y_\iota \quad \text{(12 seconds)}$$

THM45

$$[[P_{o\iota\iota}D_\iota E_\iota] \wedge \forall X_\iota \forall Y_\iota [[P_{o\iota\iota}X_\iota Y_\iota] \supset \cdot[P_{o\iota\iota}Y_\iota X_\iota] \wedge \cdot Q_{o\iota\iota}X_\iota Y_\iota] \wedge \forall U_\iota \forall V_\iota [[Q_{o\iota\iota}U_\iota V_\iota] \supset \cdot Q_{o\iota\iota}U_\iota U_\iota]] \supset \cdot[Q_{o\iota\iota}D_\iota D_\iota] \wedge \cdot Q_{o\iota\iota}E_\iota E_\iota \quad \text{(21 seconds)}$$

THM46A

$$[\mathscr{P}[D_{o\iota} \cap E_{o\iota}]] \equiv \cdot[\mathscr{P}D_{o\iota}] \cap \cdot\mathscr{P}E_{o\iota} \quad \text{(23 seconds)}$$

THM53 (Distributes a quantifier over an equivalence)

$$\forall X_\iota [[P_{o\iota}X_\iota] \equiv \exists Y_\iota \cdot P_{o\iota}Y_\iota] \equiv \cdot \forall X_\iota [P_{o\iota}X_\iota] \equiv \exists Y_\iota \cdot P_{o\iota}Y_\iota \quad \text{(23 seconds)}$$

THM55 (Theorem C) (Distributes quantifiers over an equivalence)

$$\exists X_\iota \forall Y_\iota [[P_{o\iota}X_\iota] \equiv \cdot P_{o\iota}Y_\iota] \equiv \cdot \exists X_\iota [P_{o\iota}X_\iota] \equiv \cdot \forall Y_\iota \cdot P_{o\iota}Y_\iota \quad \text{(21 seconds)}$$

THM56

$$\forall X_\iota [[P_{o\iota}X_\iota] \equiv \forall Y_\iota \cdot P_{o\iota}Y_\iota] \equiv \cdot \exists X_\iota [P_{o\iota}X_\iota] \equiv \forall Y_\iota \cdot P_{o\iota}Y_\iota \quad \text{(16 seconds)}$$

THM58

$$[[X_{o\alpha} \cap Y_{o\alpha}] \cup Z_{o\alpha}] \equiv \cdot[X_{o\alpha} \cup Z_{o\alpha}] \cap \cdot Y_{o\alpha} \cup Z_{o\alpha} \quad \text{(9 seconds)}$$

ACKNOWLEDGMENTS. I would like to thank Eve Cohen for her many vital contributions to this research project. I would also like to thank Wolfgang Bibel and Jörg Siekmann for helpful comments.

REFERENCES

(Note. References [3, 4, 6–8, 12, 17–19] are not cited in the text.)

1 ANDREWS, P B. Refutations by matings *IEEE Trans. Comput. C-25* (1976), 801–807.

2 ANDREWS, P.B., AND COHEN, E L Theorem proving in type theory. 5th Int. Joint Conf on Artificial Intelligence, Cambridge, Mass., Aug 1977, p 566

3. BIBEL, W An approach to a systematic theorem proving procedure in first-order logic. *Comput 12* (1974), 43–55.

4. BIBEL, W. A syntactic connection between proof procedures and refutation procedures. In *Lecture Notes in Computer Science 48*, H Tzschach, H Waldschmidt, and H K-G. Walter, Eds., Springer-Verlag, New York, Berlin, Heidelberg, 1978, pp. 215–224

5 BIBEL, W. Tautology testing with a generalized matrix reduction method. *Theor. Comput. Sci 8* (1979), 31–44

6 BIBEL, W On matrices with connections *J ACM* (to appear)

7. BIBEL, W A comparative study of several proof procedures. In *Proc. AISB-80 Conference*, S Hardy, Ed, Univ. of Sussex, Sussex, England, 1980, pp 11–18.

8 BIBEL, W A theoretical basis for the systematic proof method *Proc 9th Symposium on Mathematical Foundations of Computer Science*, Lecture Notes in Computer Science 88, P Dembinski, Ed, Springer-Verlag, Berlin, Heidelberg, New York, 1980, pp 154–167

9. BIBEL, W., AND SCHREIBER, J Proof search in a Gentzen-like system of first order logic. Proc. Int.

Computing Symp. 1975, E. Gelenbe and D. Potier, Eds., North-Holland, Amsterdam, 1975, pp 205–212.
10. CHURCH, A. A formulation of the simple theory of types. *J. Symbolic Logic 5* (1940), 56–68
11. COX, P.T Locating the source of unification failure. Proc. 2nd Nat. Conf. of the Canadian Society for Computational Studies of Intelligence, Toronto, Ontario, Canada, July 1978, pp. 20–29.
12. DAVIS, M. Eliminating the irrelevant from mechanical proofs. In *Experimental Arithmetic, High Speed Computing and Mathematics*, Proc. Symp. in Applied Mathematics XV, American Mathematical Society, Providence, R.I., 1963, pp. 15–30.
13. DAVIS, M., AND PUTNAM, H. A computing procedure for quantification theory *J ACM 7*, 3 (July 1960), 201–215.
14. DE CHAMPEAUX, D. Sub-problem finder and instance checker—two cooperating processors for theorem provers. Proc. 4th Workshop on Automated Deduction, Austin, Texas, Feb. 1979, pp. 110–114
15. DREBEN, B , ANDREWS, P., AND AANDERAA, S. False lemmas in Herbrand. *Bull. Amer. Math. Soc. 69* (1963), 699–706.
16. ENDERTON, H.B. *A Mathematical Introduction to Logic.* Academic Press, New York, 1972, 295 pp
17. GILMORE, P.C. A proof method for quantification theory: Its justification and realization *IBM J. Res. Dev. 4* (1960), 28–35
18. HAILPERIN, T. A form of Herbrand's theorem. *Z. Math Logik Grundl. Math. 15* (1969), 107–120.
19. HENSCHEN, L J Theorem proving by covering expressions. *J. ACM 26*, 3 (July 1979), 385–400.
20. HERBRAND, J. Recherches sur la théorie de la démonstration. *Travaux de la Société des Sciences et des Lettres de Varsovie, Classe III Sciences Mathematiques et Physiques* 33 (1930)
21. HERBRAND, J *Logical Writings* Harvard University Press, Cambridge, Mass., 1972.
22. HUET, G.P. A unification algorithm for typed λ-calculus. *Theor. Comput. Sci 1* (1975), 27–57
23. KOWALSKI, R A proof procedure using connection graphs. *J. ACM 22*, 4 (Oct. 1975), 572–595.
24 PRAWITZ, D A proof procedure with matrix reduction In *Lecture Notes in Mathematics 125*, M Laudet, D. Lacombe, L Nolin, and M Schutzenberger, Eds., Springer-Verlag, New York, Berlin, Heidelberg, 1970, pp. 207–213.
25 QUINE, W.V. A proof procedure for quantification theory. *J Symbolic Logic 20* (1955), 141–149.
26. ROBINSON, J.A. A machine-oriented logic based on the resolution principle. *J. ACM 12*, 1 (Jan 1965), 23–41.
27. SICKEL, S A search technique for clause interconnectivity graphs *IEEE Trans Comput C-25* (1976), 823–835
28. VAN VAALEN, J An extension of unification to substitutions with an application to automatic theorem proving 4th Int. Joint Conf on Artificial Intelligence, Tbilisi, USSR, 1975, 77–81.
29. WEYL, H. *Symmetry.* Princeton University Press, Princeton, N.J , 1952, 168 pp.
30. WILKINS, D.E QUEST: A non-clausal theorem proving system Master's Th., Univ of Essex, Colchester, Essex, England, 1973.
31 WOS, L , ROBINSON, G.A , AND CARSON, D F Efficiency and completeness of the set of support strategy in theorem proving *J. ACM 12*, 4 (Oct. 1965), 536–541

RECEIVED JUNE 1979, REVISED FEBRUARY 1980, ACCEPTED FEBRUARY 1980