

Extracting Information from Resolution Proof Trees¹

David Luckham

*Stanford University, Artificial Intelligence Project,
Stanford, California*

Nils J. Nilsson

*Stanford Research Institute, Artificial Intelligence Group,
Menlo Park, California*

Recommended by R. Burstall

ABSTRACT

Procedures for generating proofs within a logical inference system can be applied to many information-retrieval and automatic problem-solving tasks. These applications require additional procedures for extracting information from the proofs when they are found. We present an extraction procedure for proofs generated by the resolution principle. The procedure uses a given proof to find solutions for existential quantifiers in the statement proved in terms of known quantities in the initial data. This procedure relies heavily on basic subfunctions in the resolution program, so that it requires relatively little additional programming. The correctness of the procedure is proved, and examples are given to illustrate how it operates and to show that it cannot be simplified at certain points without loss of generality.

1. Introduction

Recently there have been described attempts to apply automatic-deduction programs to information-retrieval and problem-solving tasks [1-3]. In systems designed for these applications (often called *question-answering systems*), it is necessary to extend the basic proof procedure by adding some sort of mechanism for extracting information from the proofs that the procedure generates.

¹ The research reported here was supported in part by the Advanced Research Projects Agency of the Office of the Secretary of Defense (ARPA Order No. SD-183 at Stanford University); and by the Advanced Research Projects Agency of the Department of Defense and the National Aeronautics and Space Administration (Contract NAS12-2221 at Stanford Research Institute).

The simplest illustration of this sort of application can be roughly described as follows. Suppose a certain body of information, I , is described in first-order logic by a set of axioms or hypotheses, S , say. Further suppose that a question, Q , about I can be formulated as "Does there exist an x such that $P(x)$, given the hypotheses, S ?" An equivalent formulation is "Does S logically imply $(\exists x)P(x)$?" Then if a proof of $S \Rightarrow (\exists x)P(x)$ can be found, the answer to Q is "yes". Furthermore, in the light of a "yes" answer, it is reasonable and useful to ask for an example of the sort of object in I that has just been proved to exist.

Generally, an example of the object proved to exist might be given by the interpretation in I of an expression involving function symbols and constant symbols occurring in S . A naive and highly impractical procedure for constructing such an example would be to enumerate the terms t_1, t_2, t_3, \dots in the Herbrand universe of S and to ask simultaneously if $S \Rightarrow P(t_1), S \Rightarrow P(t_2), \dots, S \Rightarrow (P(t_1) \vee P(t_2)), \dots$ etc.

Green [4] has proposed that a special "answer predicate" be added to the theorem being proved. Its role is to collect the substitutions made by a resolution-based proof procedure during the search for a refutation of the negation of the theorem. Finally (after a refutation is found), the arguments of the answer predicate are taken to be the desired instances of the existentially quantified variables of the theorem. In this paper we shall study an alternative procedure that *extracts* information from a proof *after* the proof has already been found. The instances obtained using this procedure are sometimes more (and never less) general than those found by Green's answer-predicate method. We prove that our method yields "correct" answers; the proof also justifies Green's method.

We must emphasize that the major problem in these applications of proof procedures is still the problem of finding a proof. The answer-extraction procedure should not affect in any adverse way the efficiency of the proof procedure, either by increasing the length of the clauses or by adding to the computation time required for some of the basic operations (e.g. by requiring extra substitutions to be performed when resolvents are computed). Having obtained the proof, one can *then* afford to spend a little extra time extracting good answers.

Essentially, this problem of obtaining instances for the existentially quantified variables in a theorem is well known in classical proof theory as the "realization of existential quantifiers." Indeed, for the usual first-order predicate calculus, if S is an open formula (e.g. a conjunction of axioms in Skolem free-variable form) and there is a proof p of $S \Rightarrow (\exists x)P(x)$, then there are terms t_1, t_2, \dots, t_n composed from the function symbols of S (and a bound on their complexity is computable from p) such that the disjunction, $P(t_1) \vee P(t_2) \vee \dots \vee P(t_n)$ is a logical consequence of a finite conjunction of

Artificial Intelligence 2 (1971), 27-54

substitution instances of S . Also, there is a recursive operation which yields both the terms $\{t_i\}$ and a propositional calculus proof of the disjunction from the conjunction, given the original proof p . In the case where the consequence of S has a more complicated quantifier structure (and is not purely existential) the terms will generally be functions of some of the universal variables—e.g., a proof of $S \Rightarrow (\forall x)(\exists y)P(x, y)$ will yield a finite disjunction of the form, $\bigvee_{i \leq n} P(x, t_i(x))$. As Mostowski [5] puts it, “If all existential assumptions in the axioms are made explicit, then all existential theorems will also be explicit.” The realization procedure will of course depend on the inference rules used in the proof p , and its practicability may depend on p itself.

What we shall do in the following results in a new proof of this classical meta-theorem for the case where the original first-order proof is obtained by refutation using the Resolution Principle as the (single) rule of inference. The procedure for extracting information results as a by-product of the meta-proof and involves transforming the given resolution refutation in a very simple way. It requires little extra programming since its basic subfunctions are those of the resolution procedure.

We shall motivate the discussion by considering some examples of how the procedure works, which will also serve to illustrate why it cannot be simplified at certain points without loss of generality. It is assumed that the reader is familiar with the methods and vocabulary of resolution-based proof procedures.

2. Examples

The answer-extraction procedure modifies a resolution refutation tree to yield a direct proof of an *answer statement*. Although a complete formulation of the most general version of the procedure must provide for several fine points, the general idea is quite simple and can be illustrated by some specific examples.

Example 1. Consider the following set of statements:

- (1) $(\forall x \forall y)\{P(x, y) \wedge P(y, z) \Rightarrow G(x, z)\}$
- (2) $(\forall y \exists x)\{P(x, y)\}$.

(We might interpret these as claiming

“For all x and y if x is the parent of y and y is the parent of z , then x is the grandparent of z ,”

and

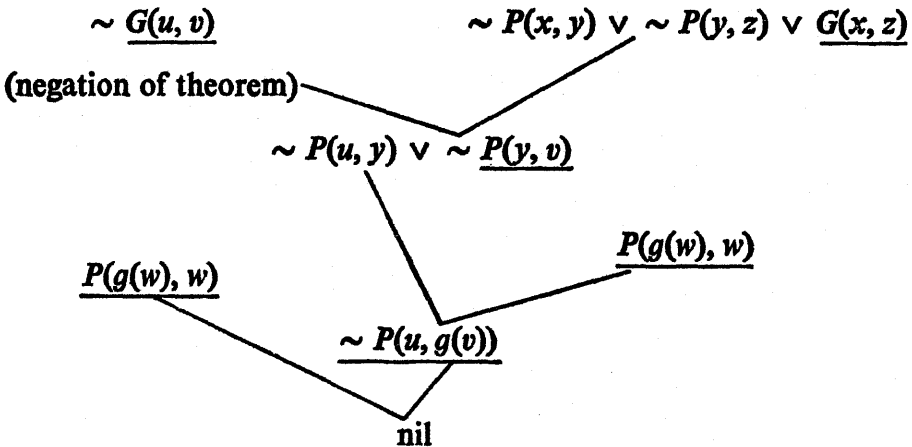
“Everyone has a parent.”)

Given these statements as hypotheses, suppose we wanted to ask the question: “Do there exist individuals x and y such that $G(x, y)$?” (That is, are there x and y such that x is the grandparent of y ?)

Artificial Intelligence 2 (1971), 27–54

We pose the question as a theorem to be proved: $(\exists x)(\exists y)G(x, y)$.

The theorem is easily proved by a resolution refutation showing the inconsistency of the set of clauses obtained from the free-variable Skolem form of the hypotheses and the negation of the theorem. The refutation tree is shown below:



(In this and the following examples we underline those literals unified in a resolution.)

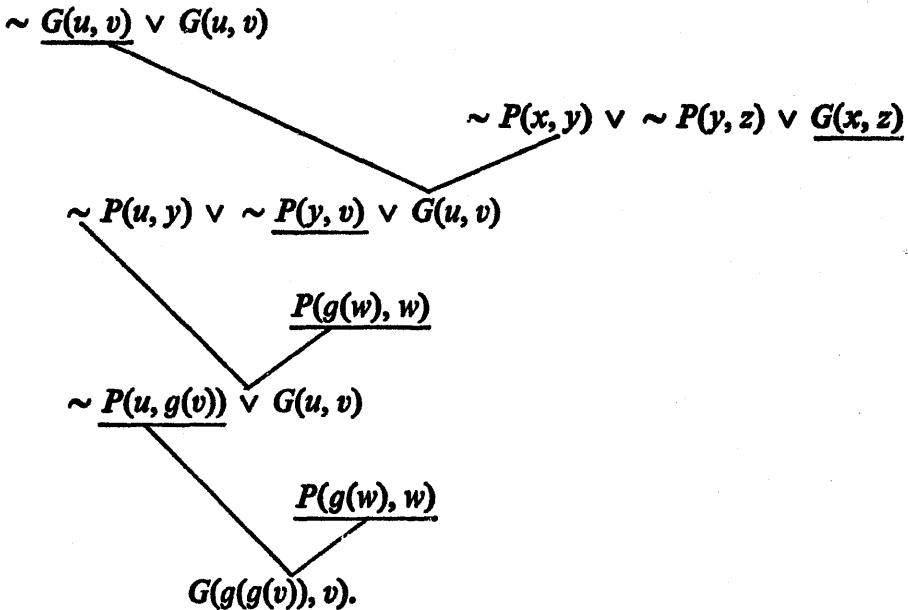
The clause $P(g(w), w)$ contains a Skolem function, g , introduced to eliminate the existential quantifier in the hypothesis $(\forall y \exists x)\{P(x, y)\}$. This function is defined so that $(\forall y)P(g(y), y)$. (The function g can be interpreted as a function that is defined to name a parent of any individual y .)

To extract information from this refutation telling us about what values of x and y satisfy $G(x, y)$, we modify this refutation tree as follows. Transform the clause $\sim G(u, v)$ obtained from the negation of the theorem into a tautology by adding the literal $G(u, v)$. Performing the same² resolutions as in the original refutation tree, we now obtain the proof tree: (see top of p. 31).

This proof tree shows that $(\forall v)G(g(g(v)), v)$ follows from the original set of hypotheses plus a tautology; thus it follows from the original set of hypotheses alone. Moreover, the statement $(\forall v)G(g(g(v)), v)$ provides a complete answer to the question: "Are there x and y such that x is the grandparent of y ?" The answer in this case involves the definitional function g : Any v and the parent of the parent of v are examples of individuals satisfying the conditions of the question.

This simple example illustrates the general idea of the tree-modifying process: Transform those clauses arising from the negation of the theorem to be proved into tautologies and construct a new tree using the resolutions of the old tree, thus obtaining a proof of a formula providing additional

² To be made precise in the next section (Lemma 1).



information about the existentially quantified variables in the theorem. This formula we shall call the *answer statement*.

Some important complications of this process will now be illustrated by additional examples.

Example 2. In this example we shall illustrate the way in which more complex clauses arising from the negation of the theorem are transformed into tautologies.

Consider the following set of clauses:

- (1) $\sim A(x) \vee F(x) \vee G(f(x))$
- (2) $\sim F(x) \vee B(x)$
- (3) $\sim F(x) \vee C(x)$
- (4) $\sim G(x) \vee B(x)$
- (5) $\sim G(x) \vee D(x)$
- (6) $A(g(x)) \vee F(h(x))$.

We desire to prove from these hypotheses the theorem:

$$(\exists x)(\exists y)\{\{B(x) \wedge C(x)\} \vee \{D(y) \wedge B(y)\}\}.$$

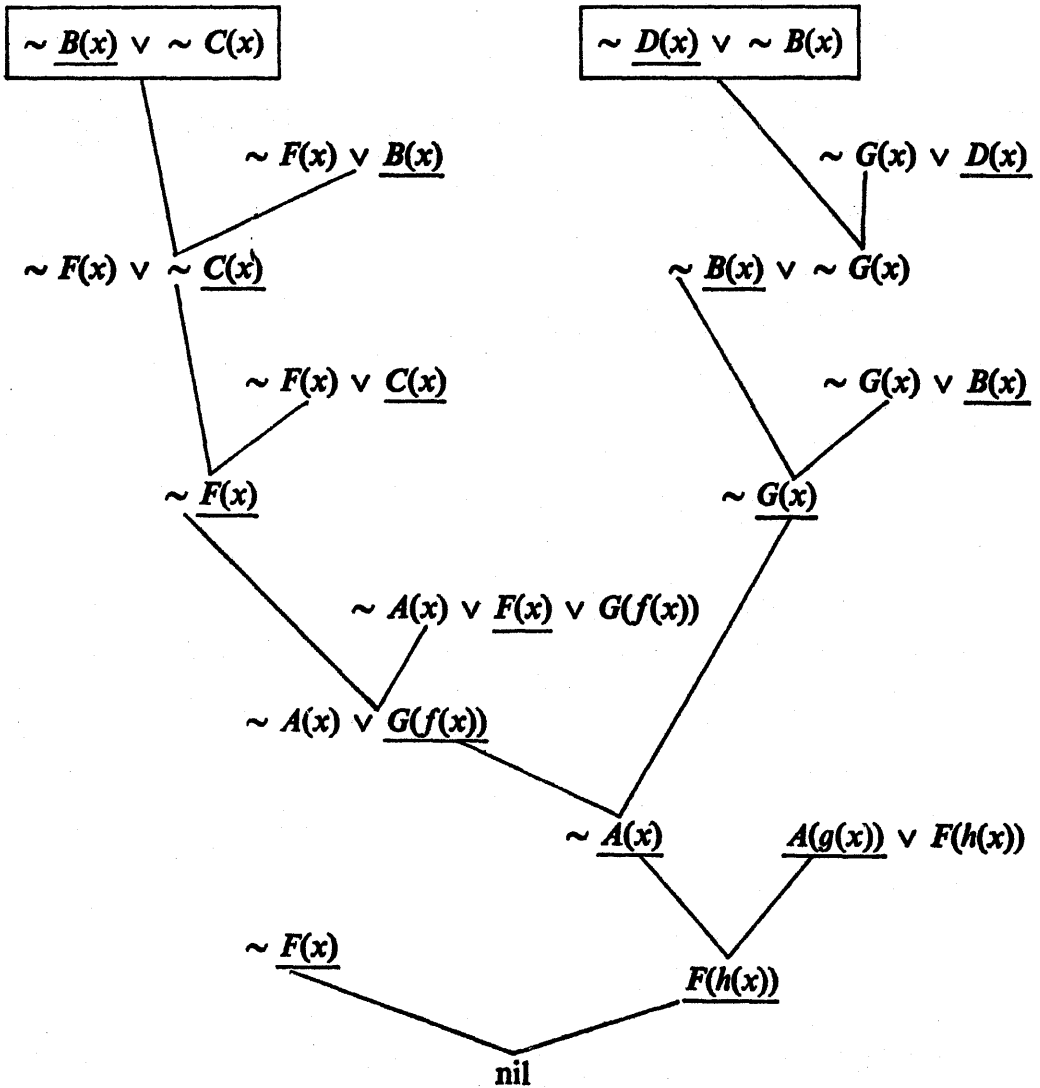
The negation of this theorem produces two clauses each with two literals

$$\sim B(x) \vee \sim C(x)$$

and

$$\sim B(x) \vee \sim D(x).$$

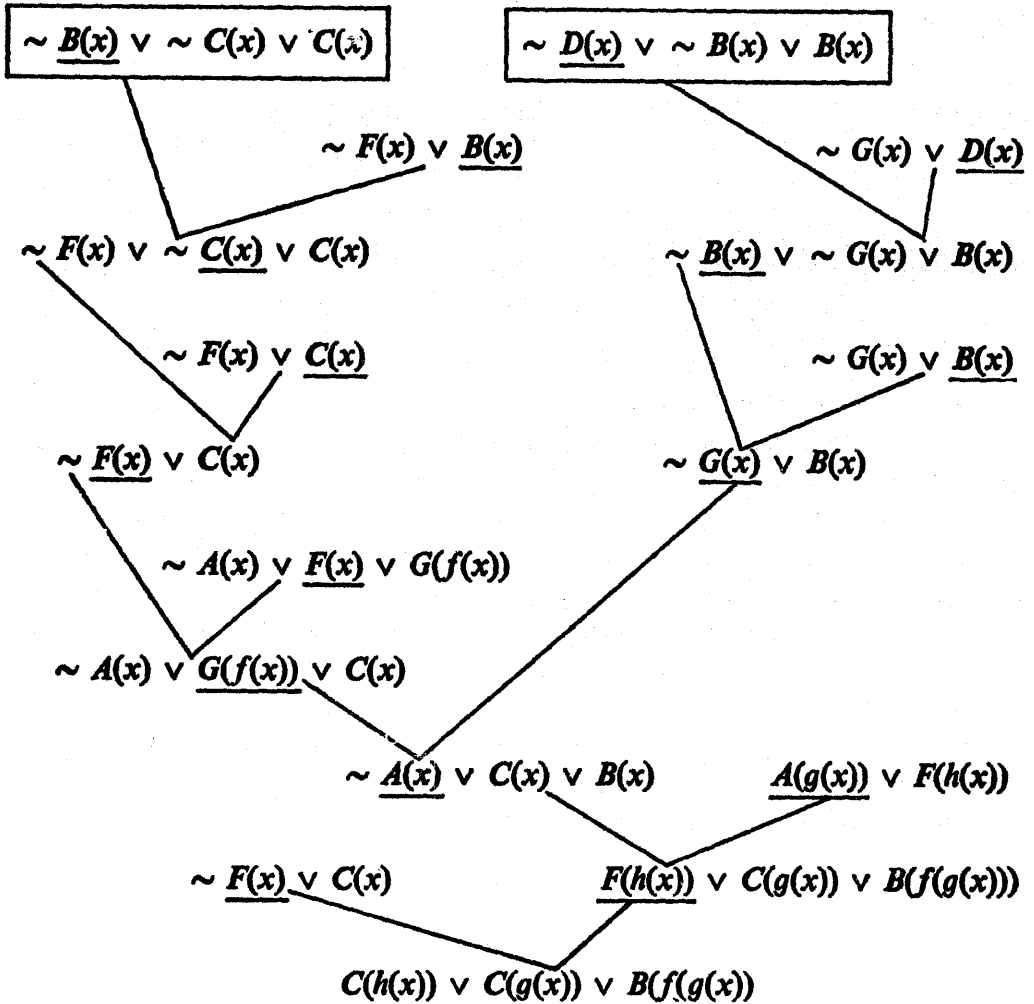
A refutation tree proving the theorem is shown overleaf:



Now to extract information by transforming this tree, we must convert the two clauses coming from the negation of the theorem (shown in boxes) into tautologies. Note that there are four ways in which this can be done:

- (1) Add $B(x)$ to the first clause and $B(x)$ to the second
- (2) Add $C(x)$ to the first clause and $D(x)$ to the second
- (3) Add $B(x)$ to the first clause and $D(x)$ to the second
- (4) Add $C(x)$ to the first clause and $B(x)$ to the second.

Each of these corresponds to a different tree transformation; we shall perform all four and conjoin the results. The proof tree on p. 33 is the result of one of the transformations:



The other three produce proofs for

$$B(h(w)) \vee B(g(w)) \vee B(f(g(w)))$$

$$B(h(y)) \vee B(g(y)) \vee D(f(g(y)))$$

and

$$C(h(z)) \vee C(g(z)) \vee D(f(g(z))).$$

Combining all of these expressions into a formula of first-order logic yields:

$$\begin{aligned}
 & (\forall w)[B(h(w)) \vee B(g(w)) \vee B(f(g(w)))] \\
 & \wedge (\forall x)[C(h(x)) \vee C(g(x)) \vee B(f(g(x)))] \\
 & \wedge (\forall y)[B(h(y)) \vee B(g(y)) \vee D(f(g(y)))] \\
 & \wedge (\forall z)[C(h(z)) \vee C(g(z)) \vee D(f(g(z)))]
 \end{aligned}$$

which is equivalent to

$$\begin{aligned} & \{(\forall x)[B(h(x)) \vee B(g(x)) \vee B(f(g(x)))] \\ & \wedge [C(h(x)) \vee C(g(x)) \vee B(f(g(x)))] \\ & \wedge [B(h(x)) \vee B(g(x)) \vee D(f(g(x)))] \\ & \wedge [C(h(x)) \vee C(g(x)) \vee D(f(g(x)))]\}. \end{aligned}$$

Now by distribution, this formula is equivalent to

$$\begin{aligned} & (\forall x)\{ \underline{[B(g(x)) \wedge C(g(x))] \vee [D(f(g(x))) \wedge B(f(g(x)))]} \\ & \vee [B(h(x)) \wedge C(h(x))] \}. \end{aligned}$$

We note that, here, the answer statement has a form somewhat different from the form of the theorem. The underlined part of the answer statement is readily seen to be similar to the entire theorem with $g(x)$ taking the place of the existentially quantified variable x in the theorem, and $f(g(x))$ taking the place of the existentially quantified variable y in the theorem. But, in this example, there is the extra disjunct $[B(h(x)) \wedge C(h(x))]$ in the answer statement. This disjunct is seen to be similar to one of the disjuncts of the theorem with $h(x)$ now taking the place of the existentially quantified variable x of the theorem.

In general, if the theorem itself is in disjunctive normal form, then our answer-extraction process will produce a statement that is a disjunction of expressions, each of which is similar in form either to the entire theorem or to one or more disjuncts of the entire theorem. For this reason we claim that this statement can be meaningfully interpreted as an "answer" to the question represented by the theorem.

Example 3. Consider the axioms describing an associative system with right inverses and a right identity element:

- | | |
|--|-----------------|
| (1) $P(x_1, g(x_1), e)$ | right inverses |
| (2) $P(x_1, e, x_1)$ | right identity |
| (3) $\sim P(x_1, x_2, x_3) \vee \sim P(x_2, x_4, x_5)$ | } associativity |
| $\vee \sim P(x_1, x_5, x_6) \vee P(x_3, x_4, x_6)$ | |
| (4) $\sim P(x_1, x_2, x_3) \vee \sim P(x_2, x_4, x_5)$ | |
| $\vee \sim P(x_3, x_4, x_6) \vee P(x_1, x_5, x_6)$ | |

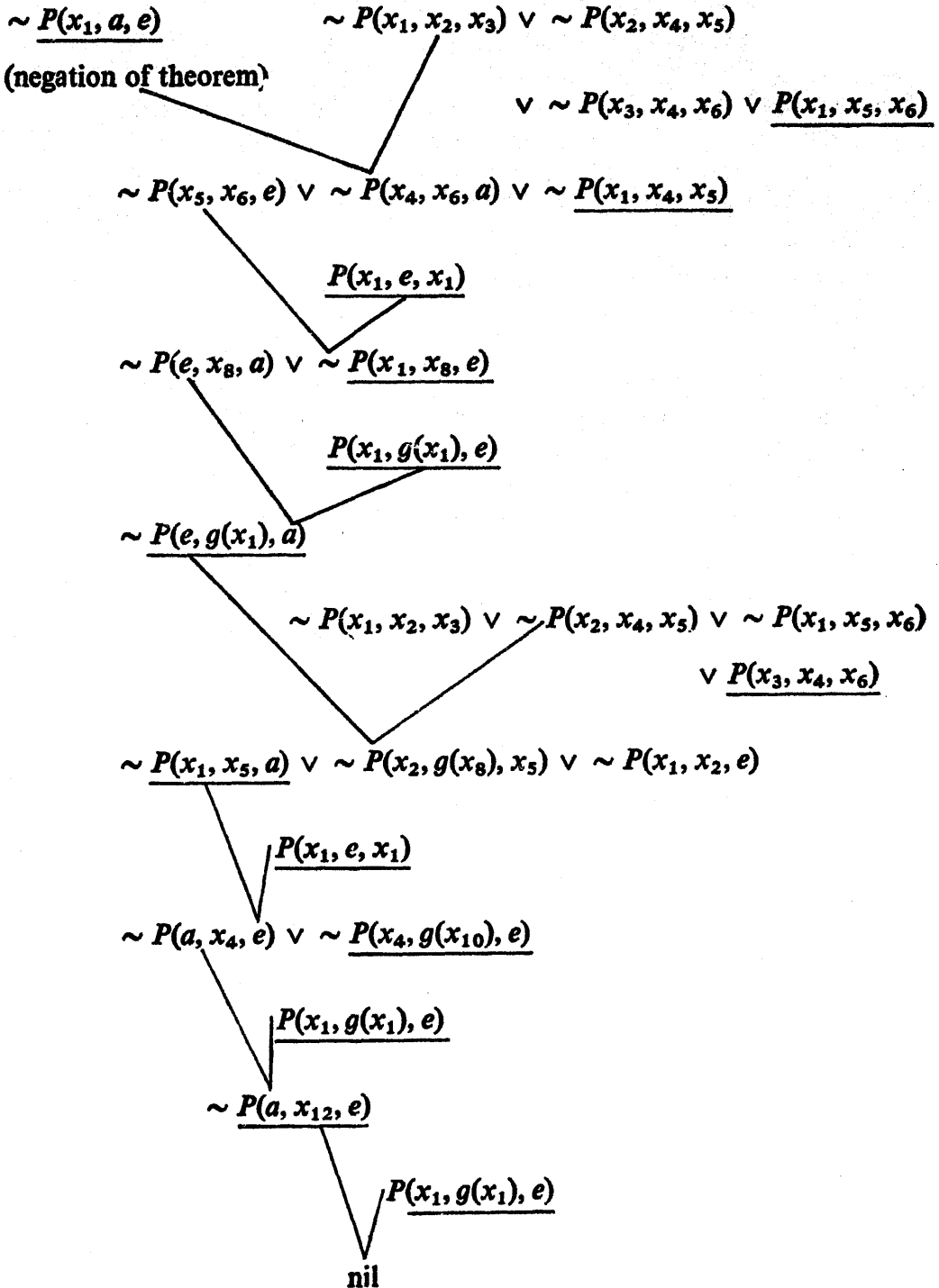
Suppose we ask of this system, "Does every element have a left inverse?" To answer, we attempt to prove the theorem:

$$(\forall x \exists y)P(y, x, e).$$

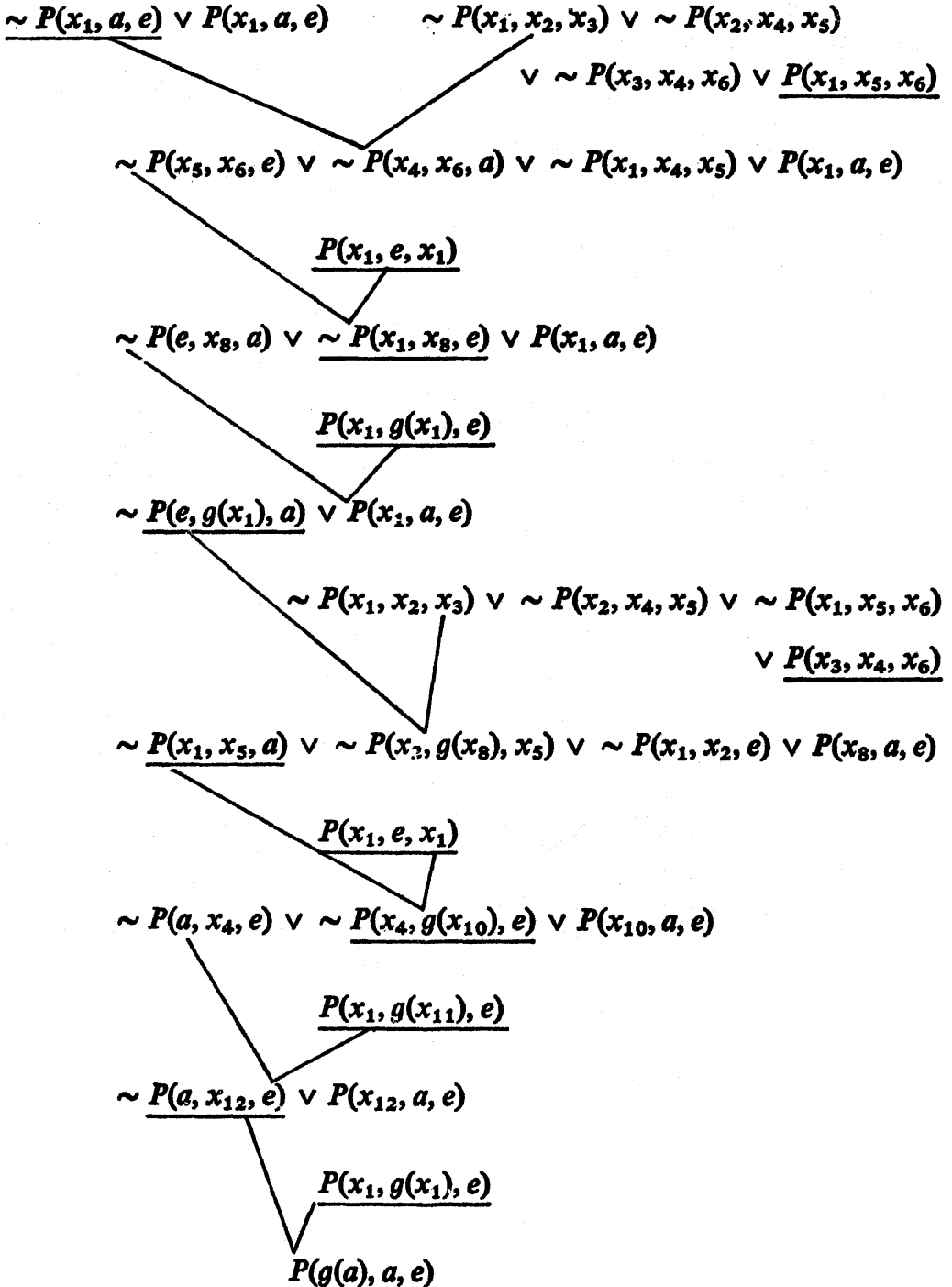
In this case we seek to extract an expression for the left inverse from the proof of its existence.

The negation of the theorem produces the clause $\sim P(y, a, e)$ where the constant " a " is a Skolem function (of no arguments) introduced to eliminate

an existentially quantified variable. A refutation tree proving the theorem is as follows:



In order to extract an answer statement from this refutation, we transform the clause, $\sim P(x_1, a, e)$, arising from the negation of the theorem into the tautology $\sim P(x_1, a, e) \vee P(x_1, a, e)$. Our refutation tree then becomes:



Now recall that our theorem was $(\forall x \exists y)P(y, x, e)$, but our transformed tree is a proof of $P(g(a), a, e)$. Our problem is how to interpret the constant a appearing in this "answer." The constant a , alleged to have no left inverse, was originally introduced in an attempt to spoil the conjectured theorem. Our proof of $P(g(a), a, e)$ shows that any such constant does in fact have a left inverse given by $g(a)$. Since a was arbitrary we might suspect that we could actually have proved the stronger result $(\forall x)P(g(x), x, e)$ which would serve as a satisfactory answer to the original question. More generally, suppose from some set \mathcal{A} of axioms we can prove a formula $P(f(x))$ using the resolution principle. Then, by the soundness of the resolution principle, $\mathcal{A} \Rightarrow (\forall x)P(f(x))$ is a theorem of predicate logic. Now, if f does not occur in \mathcal{A} , the formula $\mathcal{A} \Rightarrow (\forall f)(\forall x)P(f(x))$ is also provable, and hence $\mathcal{A} \Rightarrow (\forall y)P(y)$ is a theorem. Well known inference principles justify the replacement of all terms beginning with Skolem functions occurring in the answer statement but not in the axioms by *new* universally quantified variables. Such a replacement procedure will always yield a valid answer statement, but it might not yield the most general one as we shall show later in the paper.

3. Stage 1: Extracting Solutions to Existential Quantifiers

We now consider the procedure and its correctness in detail. Suppose we have a resolution proof for a theorem, T (or rather a refutation of $\sim T$) from a set of axioms, S . The procedure for extracting an answer statement breaks naturally into two stages. In the first stage we transform the refutation tree into a proof tree of an answer statement, ANS, by converting those clauses arising from $\sim T$ into tautologies. It is convenient to regard ANS as a preliminary answer to the question. We shall show that S implies ANS and also that ANS implies T . This is used to prove similar results for the output from the next stage.

In the second stage, terms in ANS beginning with those Skolem functions that appear in $\sim T$ but not in S , are replaced by variables. There are two alternative methods. The first is simply to carry out the necessary replacement of terms by variables on ANS itself, converting it to ANS'. The formula ANS' is an answer to the question T in the following sense.

Suppose T is in prenex normal form, and the quantifier-free matrix of T is in disjunctive normal form. ANS' is a disjunction of substitution instances of conjunctions from the matrix of T (as illustrated in the examples of Section 2), in which a variable is replaced by a functional term only if it is an existentially quantified variable in T . Hopefully, the terms in ANS' are correct solutions to the existential quantifiers in T . That this is in fact true, is established by showing: (i) S implies ANS', so the terms are provably solutions to some question having the same matrix as T , and (ii) ANS' implies T , so the terms are solutions to a question that is no weaker than T .

(For example, (ii) eliminates the possibility that ANS' is a solution to $(\forall x)(\exists y)P(x, y)$ whereas T is $(\exists y)(\forall x)P(x, y)$.) One may get an answer to a stronger question than was asked.

The alternative method at Stage 2 involves eliminating the extra Skolem functions from the entire proof tree of ANS . This sometimes produces more general answers than the first (simple) method. It is described in Section 4.

3.1. Correctness of the First Stage

We assume the usual definitions and notation that have been adopted for the theory of resolution proof trees (see [6-9]). The reader will be familiar with "leaf node," "branch," "ordinal level," "ancestor," and such like terms. A resolution proof tree of a clause A is denoted by $Tr(A)$. The *base set* of a proof tree is the set of clauses occurring at the leaf nodes of the tree. The *base sequence* of Tr is the sequence of clauses occurring at the leaf nodes of Tr enumerated from left to right, and includes repetitions of clauses that occur at more than one leaf.

First we need the following technical lemma about resolution proof trees. It is closely related to Lemma 4 of Andrews [8] but is stated for the general case instead of the ground case.

LEMMA 1. *Let $Tr(A)$ be a resolution proof tree with base set S . Let the clause C at leaf node α be replaced by $C \cup B$ and each clause D at a node β below α be replaced by the appropriate $D \cup B'\tau$ as follows:*

if

$$D = (E - \mathcal{L})\sigma \cup (F - \mathcal{M})\sigma$$

and E is replaced by $E \cup B'\rho$, then $B'\rho = B'\rho - \mathcal{L}$, and $\tau = \rho\sigma$.

Then we obtain a resolution proof tree $Tr'(A')$ isomorphic to $Tr(A)$, having base set $S \cup \{C \cup B\}$, and such that $A' = A \cup \tilde{B}\theta$ where $\tilde{B} \subseteq B$ and θ is the composition of the substitutions used to compute resolvents on the branch from α to $root(Tr)$.

REMARKS 1. Notice that Lemma 1 in fact contains an algorithm for constructing Tr' given Tr and B .

2. Notice that θ is independent of B .

3. Clearly the lemma can be extended to allow the addition of different B 's to more than one leaf node.

Let us represent the theorem T by the form, $\bar{Q}(\bar{X}, \bar{Y})M(\bar{X}, \bar{Y})$, where \bar{Q} is the quantifier prefix, \bar{X} is the set of universally quantified variables, \bar{Y} is the set of existentially quantified variables, and $M(\bar{X}, \bar{Y})$ is the quantifier-free matrix. The free variable Skolem form of the negation of T may be written in different ways, each representation reflecting greater detail, as follows:

$$\begin{aligned}
 \sim T(\bar{F}, \bar{Y}) &= \sim M(\bar{F}, \bar{Y}), \\
 &= C_1(\bar{F}(1), \bar{Y}(1)) \wedge \dots \wedge C_m(\bar{F}(m), \bar{Y}(m)), \\
 &= \bigwedge_{i=1}^m (l_{i1} \vee \dots \vee l_{iq_i})(\bar{F}(i), \bar{Y}(i)),
 \end{aligned}$$

where \bar{F} is the set of new Skolem functions replacing the existential quantifiers in $\sim T$ in the standard way; C_1, \dots, C_m are the clauses in a c.n.f. expression for $\sim M(\bar{F}, \bar{Y})$; $\bar{F}(i)$ and $\bar{Y}(i)$ are those function symbols (from \bar{F}) and variables (from \bar{Y}) that occur in C_i , and l_{i1}, \dots, l_{iq_i} are the literals of C_i . Correspondingly, we can represent the Skolem free variable form of the unnegated statement T by,

$$\begin{aligned}
 T(\bar{X}, \bar{G}) &= M(\bar{X}, \bar{G}) \\
 &= \bigvee_{i=1}^m \sim C_i(\bar{X}(i), \bar{G}(i)) \\
 &= \bigvee_{i=1}^m (\sim l_{i1} \wedge \dots \wedge \sim l_{iq_i})(\bar{X}(i), \bar{G}(i)).
 \end{aligned}$$

Let us denote provability by the usual rules of first order logic by \vdash_L , and provability by the resolution principle by \vdash_R .

Suppose now, that T is a logical consequence of a set of axioms and hypotheses, $\{A_1, \dots, A_p\}$.³

$$A_1 \wedge \dots \wedge A_p \vdash_L \bar{Q}(\bar{X}, \bar{Y})M(\bar{X}, \bar{Y}). \quad (1)$$

By the completeness of the resolution principle, we then have,

$$A_1 \wedge \dots \wedge A_p \wedge \sim T(\bar{F}, \bar{Y}) \vdash_R \text{nil}. \quad (2)$$

Let $\text{Tr}_1(\text{nil})$ be a proof tree with base set $\{A_1, \dots, A_p, \sim T(\bar{F}, \bar{Y})\}$. Now let \mathcal{C} be a *choice sequence* containing the negation, $\sim l_{ij}(\bar{F}(i), \bar{Y}(i))$, of exactly one literal from each occurrence of clause C_i in the base sequence. \mathcal{C} may contain different literals for different occurrences of C_i . Using the procedure of Lemma 1 we add the member $\sim l_{ij}$ of \mathcal{C} to the corresponding occurrence of C_i at a leaf of Tr_1 . This transforms $\text{Tr}_1(\text{nil})$ into an isomorphic tree, $\text{Tr}_2(B)$, which contains a tautology C'_i , at each leaf node corresponding to a leaf of Tr_1 which contains C_i .

We will assume for the moment that every $\sim l_{ij}$ that was added to a leaf of Tr_1 to form Tr_2 has a successor in B . This means that in applying Lemma 1, no literal having a $\sim l_{ij}$ as an ancestor was eliminated by a resolution in Tr_2 . Let us call this assumption A .

The formula B is of the form $\bigvee_i \sim l_{ij}\theta'_i$, where θ'_i is the substitution θ defined in Lemma 1 on the branch from the r th leaf (which contains $C_i \vee \sim l_{ij}$) of

³ In what follows, we use the same notation to denote both open and closed forms of the axioms.

Tr_2 to its root. There may be more than one occurrence of an instance of $\sim l_{ij}$ for some i , or there may be no occurrences, depending on the number of leaves containing C_i . However, we assume B is not empty because this would imply that the axioms are inconsistent. We may write B as

$$\bigvee_i \sim l_{ij} (\bar{F}(i)\theta_i^r, \bar{Y}(i)\theta_i^r). \\ A_1 \wedge \dots \wedge A_p \wedge \bigwedge_k C_k'(\bar{F}(k), \bar{Y}(k)) \vdash_R \bigvee_i \sim l_{ij}(\bar{F}(i)\theta_i^r, \bar{Y}(i)\theta_i^r). \quad (3)$$

Now 3 holds for any choice sequence, \mathcal{C} , and the θ_i^r is independent of the $\sim l_{ij}$ added at the r th leaf node. Hence, for all choice sequences we have,

$$A_1 \wedge \dots \wedge A_p \wedge \bigwedge_k C_k' \wedge (\bigwedge_i l_{ij}(\bar{F}(i)\theta_i^r, \bar{Y}(i)\theta_i^r)) \vdash_R \text{nil}. \quad (4)$$

By the soundness of the resolution principle and the fact that tautologies are theorems of logic, we have for any choice sequence,

$$A_1 \wedge \dots \wedge A_p \vdash_L \bar{\forall}(\bigvee_i \sim l_{ij}(\bar{F}(i)\theta_i^r, \bar{Y}(i)\theta_i^r)), \quad (5)$$

where $\bar{\forall}$ denotes the universal closure of the formula within its scope. Thus, the conjunction of all possible right-hand sides of 5 (for each choice sequence) is also a logical consequence of the axioms. By applying the quantifier rules of logic and the distribution laws (governing the \wedge and \vee connectives) to this conjunction, we can collect together all those literals having an ancestor in the occurrence of C_i at the r th leaf node, for each i and r . Since, by assumption A , no such literals are eliminated in Tr_2 , this yields,

$$A_1 \wedge \dots \wedge A_p \vdash_L \bar{\forall}(\bigvee_i \sim C_i(\bar{F}(i)\theta_i^r, \bar{Y}(i)\theta_i^r)), \quad (6)$$

where $\sim C_i$ is the i th conjunction in the disjunctive normal form of M (see above). It is the right-hand side of 6 that we take to be the *preliminary answer statement*, ANS. By 6, ANS logically follows from the axioms.

If assumption A was false, this would simply mean that one or more of the disjuncts in the right-hand side of 6 could be deleted. For example, suppose a successor of $\sim l_{11}$ when added to C_1 at leaf α_1 , is eliminated by a resolution in Tr_2 . Then $\sim l_{11}\theta_1^1$ does not occur in B (at step 3). So, restricting the above argument (steps 4 to 6) to those choice sequences which are fixed to choose $\sim l_{11}$ for C_1 at α_1 , results in deleting $\sim C_1(\bar{F}(1)\theta_1^1, \bar{Y}(1)\theta_1^1)$ from 6. A more precise (preliminary) answer would then be obtained. This sort of inefficiency in the proof is easily tested for.

We shall use the substitutions, θ_i^r to realize the existentially quantified variables in the theorem.

STAGE 1. The first stage of our procedure merely involves computing the terms $\bar{F}(i)\theta_i^r$ and $\bar{Y}(i)\theta_i^r$ and constructing the right-hand side of 6. One way of doing this is to extend Green's procedure [4] as follows: First the clauses at *Artificial Intelligence 2* (1971), 27-54

the leaves of Tr_1 are checked to see if they belong to the negation of the theorem; if C_i in the negation of the theorem occurs at the r th leaf node, a "new" predicate atom, $ANS_i(\bar{Y}(i))$ is appended to that occurrence. The resolvents of the tree are then recomputed according to Lemma 1. Notice that $ANS_i(\bar{Y}(i))$ contains only those existential variables that occur in C_i . This yields a proof tree of a disjunction of the form, $\bigvee_i ANS_i(\bar{Y}(i)\theta_i)$, which contains all the information necessary to compute a preliminary answer statement.

This method is inadequate if one wants to check if the answer statement can be improved because assumption A fails for some literal. This must be tested separately, for example, by computing the tree Tr_2 for those choice sequences which choose the first literal from each C_i , the second literal from each C_i , and so on.

It remains to show $\vdash_L ANS \Rightarrow T$. This is the case if and only if the conjunction of the set of clauses $\{ANS, \sim T\}$ is unsatisfiable. Recall that $\sim T = C_1(\bar{F}(1), \bar{Y}(1)) \wedge \dots \wedge C_m(\bar{F}(m), \bar{Y}(m))$. Let $\{\alpha_1, \alpha_n, \dots, \alpha_p\}$ be the sequence (from left to right) of those leaves of Tr_1 at which clauses in $\sim T$ occur, and suppose that the corresponding sequence of clauses at these leaves is $\{C_u, C_v, \dots, C_w\}$. Let $\{\theta_u^l, \theta_v^n, \dots, \theta_w^p\}$ be the sequence of substitutions from Lemma 1 for these leaves; each θ_j^i is the composition of substitutions on the branch of Tr_1 from leaf α_i to root Tr_1 . Then ANS , in the form of the righthand side of 5, is the conjunction,

$$ANS = \wedge [\sim l_{ui}(\bar{F}(u), \bar{Y}(u))\theta_u^l \vee \sim l_{vj}(\bar{F}(v), \bar{Y}(v))\theta_v^n \vee \dots \\ \sim l_{wk}(\bar{F}(w), \bar{Y}(w))\theta_w^p],$$

where the conjunction is taken over all possible ways of choosing a literal l_{ij} from each occurrence of C_i in the base sequence of Tr_1 . None of these disjunctions is empty if the axioms are consistent.

Now consider the following conjunction of instances of clauses from $\sim T$ containing $C_j\theta_j^i$ for each C_j and θ_j^i in the above sequences:

$$C_u(\bar{F}(u), \bar{Y}(u))\theta_u^l \wedge C_v(\bar{F}(v), \bar{Y}(v))\theta_v^n \wedge \dots \wedge C_w(\bar{F}(w), \bar{Y}(w))\theta_w^p.$$

If this is consistent, it is possible to choose one literal from each clause in the conjunction without choosing a complementary pair. But any such choice must contain the negation of every literal in one of the clauses of ANS . Thus the conjunction, $ANS \wedge (\sim T)\theta_u^l \wedge \dots \wedge (\sim T)\theta_w^p$ is inconsistent. Since the terms of the θ_j^i substitutions belong to the Herbrand domain of $\{ANS, \sim T\}$, it follows from Herbrand's theorem that the conjunction of these clauses is unsatisfiable.

We have thus proved the following theorem:

THEOREM 1. *If T is a logical consequence of consistent axioms \mathcal{A} , and ANS*
Artificial Intelligence 2 (1971), 27-54

is the preliminary answer statement obtained from a resolution refutation with base set $\mathcal{A} \cup \{\sim T\}$, then $\mathcal{A} \vdash_L \text{ANS}$ and $\text{ANS} \vdash_L T$.

4. Stage 2: Eliminating Skolem Functions

Stage 1 of the extraction process produces a preliminary answer statement given by the right-hand side of 6:

$$\text{ANS} = \bar{\forall}(\bigvee_i \sim C_i(\bar{F}(i)\theta_i, \bar{Y}(i)\theta_i)).$$

Now, if T is not purely existential, ANS will contain terms beginning with Skolem function symbols introduced in $\sim T$ and not occurring in the axioms or initial hypotheses, namely, the terms $\bar{F}(i)\theta_i$. Such terms may also occur in $\bar{Y}(i)\theta_i$. These terms are not interpretable from the initial data. Their replacement by "new" variables (not already occurring in ANS) will yield an answer, ANS', which is also a consequence of the axioms. These replacements are easily justified on the basis of standard inference rules of logic (see the reasoning following Example 3, Section 2).

STAGE 2(a). Let $\{z_1, z_2, z_3, \dots\}$ be a sequence of variables not occurring in ANS. Set $\text{ANS}_0 = \text{ANS}$. For $i = 0, 1, 2, \dots$, let ANS_{i+1} be obtained from ANS_i by replacing every occurrence of the first (leftmost) term beginning with a symbol from \bar{F} by z_{i+1} ; if there are no such occurrences, let $\text{ANS}' = \text{ANS}_i$.

Clearly there is a substitution σ which reverses this replacement operation so that $\text{ANS} = (\text{ANS}')\sigma$. Therefore $\vdash_L \text{ANS}' \Rightarrow \text{ANS}$. Thus we have,

$$A_1 \wedge \dots \wedge A_p \vdash_L \text{ANS}' \text{ and } \text{ANS}' \vdash_L T. \quad (7)$$

To see in what sense we mean that ANS' can be considered as an answer to the original question, suppose we replace by x_i each occurrence of a z -variable that replaced a term beginning with f_i in going from ANS to ANS'. We will then have an instance of ANS':

$$\bar{\forall}(\bigvee_i \sim C_i(\bar{X}(i), \bar{Y}(i)\theta_i^{r*})). \quad (8)$$

Finally, note that 8 implies a generally weaker statement of the form,

$$\bar{\forall}(M(\bar{X}, \bar{Y}\theta_1) \vee \dots \vee M(\bar{X}, \bar{Y}\theta_q)). \quad (9)$$

The description of both stages of one version of the procedure is now complete. We shall show that a more complex operation in Stage 2 can yield more general answer statements. We first present an example to illustrate that, without this modification, the procedure as it stands might not produce sufficiently general answer statements.

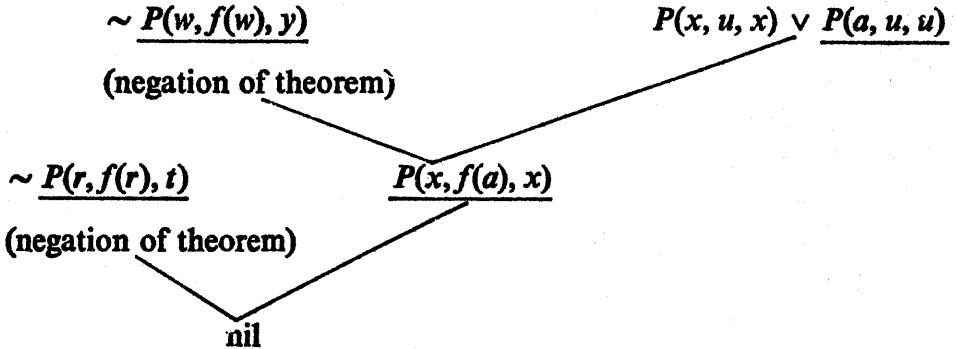
Example 4. Suppose from the hypothesis

$$(\forall x \forall u)[P(x, u, x) \vee P(a, u, u)]$$

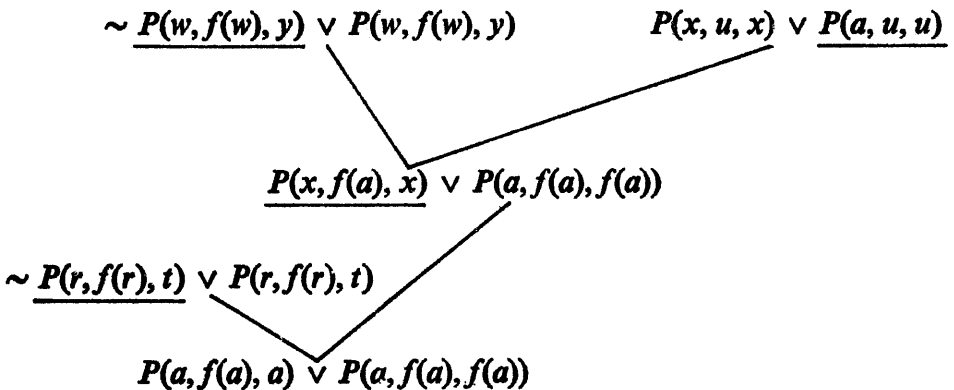
we want to prove

$$(\exists w \forall v \exists y) P(w, v, y).$$

A refutation tree for the set of clauses obtained from the free-variable Skolem form of the hypothesis and the negation of the theorem is shown below:



Stage 1 of the procedure yields the following resolution tree for ANS:



In Stage 2(a), we replace the Skolem function $f(a)$ in the answer statement by a new universally quantified variable, say z , to obtain the final answer statement:

$$(\forall z)[P(a, z, a) \vee P(a, z, z)].$$

It is easily seen that a more general answer statement, namely the hypothesis itself in this example, should be given. In the above resolution tree, the unification of terms within the Skolem function f in the final resolution, reduced the generality of the answer statement. This sort of loss of generality can be avoided by reconstructing the proof tree for ANS using new variables in place of terms beginning with f . That is, instead of merely replacing f -terms in ANS by new variables, we do so throughout the ANS tree. This replacement operation in the ANS tree constitutes an alternative Stage 2(b) procedure.

The main idea of the transformation algorithm is easily stated: In those clauses converted into tautologies we replace those Skolem functions not appearing in the axioms by new variables, called z -variables. With the clauses resulting from this replacement operation we will then construct a tree Tr_3 isomorphic to the ANS tree, Tr_2 . The resolutions used in Tr_3 will be computed from those used in Tr_2 according to a certain rule described in the next section.

We are particularly concerned with a special property of this rule, namely that in Tr_3 no terms are ever substituted for z -variables. Since the z -variables correspond to the universally quantified variables in the original theorem, we want them to appear in the answer statement at the root of Tr_3 . Thus we need to be certain that substitutions in Tr_3 do not eliminate any of them. One of the major tasks will be to show that the standard unification process employed in Tr_3 does not substitute terms for z -variables. Of necessity this argument must be rather detailed and requires some special definitions and lemmas concerning the process of replacing Skolem functions by z -variables.

4.1. Notation and Definitions

In addition to the usual definitions and notation of resolution proof theory (see [6–9]), we will need some special definitions to explain the procedure.

If \mathcal{A} is a set of atoms, we denote the result of applying the standard unification algorithm (defined in [6 or 7]) by $\mathcal{U}(\mathcal{A})$. If \mathcal{A} is unifiable, $\mathcal{U}(\mathcal{A}) = \sigma$ (a simplest or most-general unifier), else $\mathcal{U}(\mathcal{A}) = F$ (false). Substitutions are sets of replacements; here we use the notation, $\{(t \rightarrow x)\}$ to denote that a substitution replaces x by t . We note in passing that if $\mathcal{U}(\mathcal{A}) = \sigma$, then σ has the property that if $(t' \rightarrow x) \in \sigma$ then x does not belong to any of the terms t' such that $(t' \rightarrow x) \in \sigma$. Finite or infinite sequences of variables are denoted by \bar{X} , \bar{Y} , \bar{Z} , In the usual formulation of resolution and unification, lexical ordering is used to make certain decisions (e.g. what to substitute for what). We shall use the lexical ordering of all well-formed terms (variables and functional terms) in the theory below; in any implementation some more practical ordering would be used. We assume that all the variables used in any resolution proof belong to a sequence \bar{X} . We shall need an infinite sequence of “new” variables, \bar{Z} , disjoint from \bar{X} , all the members of which succeed all members of \bar{X} in the lexical ordering; we denote this by $\bar{X} < \bar{Z}$. The bar notation, $\bar{\mathcal{A}}$, is also used to denote the set of negations of literals in \mathcal{A} ; this will not be confused with the vector notation for sequences of variables.

Let \mathcal{A} be a set of well-formed expressions (e.g. atoms or terms) all of whose variables belong to \bar{X} . Let $\bar{F} = \{f_1, f_2, \dots, f_k\}$ be a finite sequence of (some of the) function symbols in \mathcal{A} , and let $\bar{Z} = \{\bar{Z}_1, \bar{Z}_2, \dots, \bar{Z}_k\}$ be a finite sequence of disjoint infinite sequences of variables, one for each member of \bar{F} .

We assume further that all the proof variables precede the variables in \bar{Z} , i.e., $x_i < z_j$ for all $x_i \in \bar{X}$ and $z_j \in \bar{Z}$, and in addition, that if $i < j$, all members of \bar{Z}_i precede all members of \bar{Z}_j .

DEFINITION. \mathcal{A}^* denotes the result of performing the following *replacement* operation on the expressions of \mathcal{A} : For each i in the order $i = 1, 2, 3, \dots$ all occurrences of the j th term beginning with f_i in the lexical ordering of all terms (i.e., the term beginning with f_i which is preceded in the lexical order by $j - 1$ terms beginning with f_i) are replaced in \mathcal{A} by z_i^j (the j th variable of \bar{Z}_i).

In forming \mathcal{A}^* , all occurrences in \mathcal{A} of the *same* term beginning with a function symbol in \bar{F} , are replaced by the *same* z -variable. All terms beginning with f_i are replaced before any term beginning with f_{i+1} . We shall refer to this as a **-operation*. In what follows, \bar{F} and \bar{Z} are fixed so that the **-operation* is uniquely determined. \mathcal{A}^* is said to be obtained by a **-operation* on \mathcal{A} w.r.t. \bar{F} .

REMARKS. 1. The **-operation* is defined for any set of pairs of expressions (e.g. substitutions). Thus, σ^* denotes the result of performing the **-operation* on the expressions of σ .

2. If all variables of \mathcal{A} belong to \bar{X} , then $|\mathcal{A}| = |\mathcal{A}^*|$. That is, the **-operation* does not unify distinct expressions; this is because distinct *new* variables are substituted for distinct terms.

3. $(\mathcal{A} \cup \mathcal{B})^* = \mathcal{A}^* \cup \mathcal{B}^*$; $(\mathcal{A} - \mathcal{B})^* = (\mathcal{A}^* - \mathcal{B}^*)$.

Certain substitutions have the effect of leaving unaltered in any expression they operate on, the set of \bar{Z}_i 's having members in the expression. The number of distinct members of any \bar{Z}_i may be changed. Such substitutions play a special role in what follows so we give them a name (a temporary name which the reader may want to forget at the end): *z-conserving substitutions*.

DEFINITION. A substitution σ is called *z-conserving* (notation: σ is z.c.) if for each \bar{Z}_i in \bar{Z} , the only terms it substitutes for variables in \bar{Z}_i are other variables in \bar{Z}_i . That is, in any member of σ of the form $(t \rightarrow z_i^j)$, the term t must also be a member of \bar{Z}_i .

DEFINITION. The *depth of nesting* function, d , is defined on terms, t , and finite sets of terms, \mathcal{A} , as follows:

$$d(t) = 0 \text{ if } t \text{ is a variable or constant,}$$

$$d(f(t_1 \dots t_n)) = \max(d(t_1), \dots, d(t_n)) + 1,$$

$$d(\mathcal{A}) = \max \{d(t) : t \in \mathcal{A}\}.$$

4.2. Preliminary Lemmas

LEMMA 2. Let \mathcal{A} be any set of terms all of whose variables belong to \bar{X} and θ any substitution that replaces only variables belonging to \bar{X} . Then there exists a z.c. substitution, μ , such that

$$(\mathcal{A}\theta)^* = \mathcal{A}^*\theta^*\mu.$$

PROOF. By induction on $d(\mathcal{A})$. In addition, we shall also prove that the z-variables replaced by μ result from applying the $*$ -operation to functional terms of depth at most $d(\mathcal{A})$.

First let us consider the simplified situation where \bar{F} contains a single function letter, f , and the corresponding \bar{Z} contains one set of variables, $\bar{Z} = \{z^1, z^2, z^3, \dots\}$.

Basis step, $d(\mathcal{A}) = 0$. Consider any term t such that $d(t) = 0$.

If t is a constant, $t\theta = t$.

Therefore, $(t\theta)^* = t^*$

$= t^*\theta^*$ because all variables replaced by θ^* belong to

\bar{X} and t^* is either a constant or a z^i .

If t is a variable, x , say, then

$$(x\theta)^* = \begin{cases} t_1^* & \text{if } (t_1 \rightarrow x) \in \theta \\ x & \text{otherwise} \end{cases} = x^*\theta^*.$$

If $d(\mathcal{A}) = 0$, \mathcal{A} is a set of constants and variables so, by remark (3) above, we have proved that

$$(\mathcal{A}\theta)^* = \mathcal{A}^*\theta^*.$$

Induction step, $d(\mathcal{A}) = n + 1$. Induction Hypothesis: For any set of terms \mathcal{B} , all the variables of which belong to \bar{X} , such that $d(\mathcal{B}) \leq n$, and any substitution θ that replaces only variables in \bar{X} , there is a z.c. substitution μ such that

(i) $(\mathcal{B}\theta)^* = \mathcal{B}^*\theta^*\mu,$

(ii) All the z-variables replaced by μ belong to \mathcal{B}^* .

We consider first a single term t such that $d(t) = n + 1$.

CASE 1.

$$t = f(t_1, t_2, \dots, t_m).$$

Thus

$$t\theta = f(t_1\theta, \dots, t_m\theta),$$

and

$$(t\theta)^* = z^p$$

where the term $f(t_1\theta, \dots, t_m\theta)$ is replaced by z^p under the $*$ -operation.

But,

$$t^*\theta^* = z^q\theta^* = z^q$$

where t is replaced by z^a under the $*$ -operation, and no z -variable is replaced by θ (nor by θ^*).

Hence

$$\begin{aligned}(t\theta)^* &= t^*\theta^*\{(z^p \rightarrow z^q)\} \\ &= t^*\theta^*\mu\end{aligned}$$

where μ is z.c. and satisfies the hypothesis since z^a belongs to $\{t\}^*$.

CASE 2.

$$t = g(t_1, t_2, \dots, t_m).$$

Let

$$\mathcal{B} = \{t_1, t_2, \dots, t_m\}.$$

We note that $d(\mathcal{B}) = n$ so that by the induction hypothesis,

$$(\mathcal{B}\theta)^* = \mathcal{B}^*\theta^*\mu \quad (1)$$

where μ is z.c. Now,

$$\begin{aligned}t\theta &= g(t_1, \dots, t_m)\theta = g(t_1\theta, \dots, t_m\theta). \\ (t\theta)^* &= g((t_1\theta)^*, \dots, (t_m\theta)^*)\end{aligned}$$

since the $*$ -operation does not replace g ,

$$\begin{aligned}&= g(t_1^*\theta^*\mu, \dots, t_m^*\theta^*\mu) \text{ by (1),} \\ &= g(t_1^*, \dots, t_m^*)\theta^*\mu \\ &= t^*\theta^*\mu.\end{aligned}$$

Thus the induction hypothesis is extended to single terms of depth $n + 1$.

Let $d(\mathcal{A}) = n + 1$. Let \mathcal{B} be the set of all proper subterms of terms of \mathcal{A} , together with all terms \mathcal{A} of depth less than $n + 1$. Note that $d(\mathcal{B}) = n$. So, by hypothesis, $(\mathcal{B}\theta)^* = \mathcal{B}^*\theta^*\mu$ where μ is z.c. and replaces only z -variables belonging to \mathcal{B}^* .

Then

$$(\mathcal{A}\theta)^* = \{(t_1\theta)^*, (t_2\theta)^*, \dots, (t_m\theta)^*\}$$

where $d(t_i) \leq n + 1$,

$$= \{t_1^*\theta^*\mu_1, t_2^*\theta^*\mu_2, \dots, t_m^*\theta^*\mu_m\}$$

by the above extension to single terms of depth $n + 1$; each μ_i is either μ (if t_i falls under Case 2 or $d(t_i) \leq n$), or is a unit substitution of the form $(z^p \rightarrow z^q)$ where $z^q = t_i^*$ and $d(t_i) = n + 1$, so that z^q is not replaced by μ . Such unit substitutions occur under Case 1 above. Any two such z^q 's must be distinct because $t_i^* = t_j^*$ implies $t_i = t_j$. We may therefore conclude that $\mu' = \mu_1 \cup \mu_2 \cup \dots \cup \mu_m$ is an unambiguous substitution. Clearly μ' is z.c. and only replaces variables belonging to \mathcal{A}^* .

From the above equation, $(\mathcal{A}\theta)^* = \mathcal{A}^*\theta^*\mu'$. This completes the induction step.

It is clear that the lemma is now established for unit \bar{F} , whenever \mathcal{A} contains variables (and θ replaces variables) that do *not* occur in the range of the $*$ -operation. The restriction to \bar{X} is not essential. The extension to $*$ -operations on more than one function symbol can be done by iterating the one-function operations. For example, let $\bar{F} = \{f_1, f_2\}$, $\bar{Z} = \{Z_1, Z_2\}$, and let $*i$ denote the operation of replacing terms beginning with f_i by members of Z_i , $i = 1$ or 2 . Then

$$\begin{aligned} (\mathcal{A}\theta)^* &= ((\mathcal{A}\theta)^{*1})^{*2} \\ &= (\mathcal{A}^{*1}\theta^{*1}\mu_1)^{*2} \quad \text{by the lemma for one function symbol,} \\ &= (\mathcal{A}^{*1}\theta^{*1})^{*2}(\mu_1)^{*2}\mu_2 \quad \text{by the lemma for one function symbol,} \\ &= (\mathcal{A}^{*1}\theta^{*1})^{*2}\mu_1\mu_2 \quad \text{since } \mu_1 \text{ only contains members of } Z_1, \\ &= \mathcal{A}^{*1*2}\theta^{*1*2}\mu_2'\mu_1\mu_2 \quad \text{by the lemma for one function symbol,} \\ &= \mathcal{A}^*\theta^*\mu \quad \text{where } \mu \text{ is z.c.} \end{aligned}$$

This completes the proof of the lemma.

COROLLARY 1. *If \mathcal{A} is a set of atoms and θ is any substitution, and all variables of \mathcal{A} and θ belong to \bar{X} , then $(\mathcal{A}\theta)^* = \mathcal{A}^*\theta^*\mu$ where μ is z.c.*

COROLLARY 2. *Let \mathcal{A} be a unifiable set of atoms all of whose variables belong to \bar{X} , and let $\mathcal{U}(\mathcal{A}) = \theta$. Then there is a μ such that $\theta\mu$ is a unifier of \mathcal{A}^* and $\theta^*\mu$ is z.c.*

PROOF. Since $\mathcal{U}(\mathcal{A}) = \theta$, all the variables of θ belong to \bar{X} . Hence, by Corollary 1, there is a z.c. substitution μ such that $\mathcal{A}^*\theta^*\mu = (\mathcal{A}\theta)^*$. $(\mathcal{A}\theta)^*$ is a unit set. Therefore $\theta^*\mu$ unifies \mathcal{A}^* and is clearly z.c.

LEMMA 3. *Let \mathcal{A} be a unifiable set of atoms, let θ be a z.c. unifier of \mathcal{A} , and let $\mathcal{U}(\mathcal{A}) = \rho$. Then*

1. ρ is z.c.
2. there is a τ such that $\rho\tau = \theta$ and τ is z.c.

PROOF. Since θ unifies \mathcal{A} , there is a τ such that $\rho\tau = \theta$. Thus, the composition $\rho\tau$ is z.c. Now, ρ is an output of \mathcal{U} and therefore has the property: for any variable v , if $(t \rightarrow v) \in \rho$ then v does not occur in any term t' such that $(t' \rightarrow v') \in \rho$. That is, if v is replaced by ρ then it is *not* substituted by ρ . It follows from this that if $(t \rightarrow z) \in \rho$ and $(t' \rightarrow z) \in \tau$ then $(t' \rightarrow z)$ is *redundant* in the sense that it may be omitted from τ without affecting $\rho\tau$. Hence we may assume that τ is chosen without redundant replacements, in which case all replacements of τ appear in $\rho\tau$. Thus τ is z.c.

Consider ρ , and suppose $(t \rightarrow z_i)$ belongs to ρ . If t is a functional term, *Artificial Intelligence* 2 (1971), 27-54

then $(\tau \rightarrow z_i)$ belongs to $\rho\tau$, contradicting z.c. If $(z_j \rightarrow z_i)$ belongs to ρ , this can be prevented from appearing in $\rho\tau$ only if τ contains a replacement of the form $(z_i^* \rightarrow z_j)$; but then this latter replacement is nonredundant and will appear in $\rho\tau$, again contradicting z.c. Finally, $(x_j \rightarrow z_i)$ cannot belong to ρ because $\bar{X} < \bar{Z}$. Thus the only replacements of z -variables in ρ are of the form $(z_i^* \rightarrow z_i)$, so that ρ is z.c.

LEMMA 4. *Let E be a resolvent of clauses A and B all of whose variables belong to \bar{X} . Let C and D be clauses and τ_1 and τ_2 z.c. substitutions with the property that $C\tau_1 = A^*$ and $D\tau_2 = B^*$. Then there is a resolvent H of C and D obtained by a z.c. unifier, and a z.c. substitution τ_3 such that $H\tau_3 = E^*$.*

PROOF. We may assume that A and B have no common variables, and C and D have no common variables. Let

$$E = (A - \mathcal{L}_1)\theta \cup (B - \mathcal{M}_1)\theta,$$

where

$$\mathcal{U}(\mathcal{L}_1 \cup \bar{\mathcal{M}}_1) = \theta.$$

Define subsets of literals, $\mathcal{L}_2 \subseteq C$ and $\mathcal{M}_2 \subseteq D$ by,

$$\begin{aligned} l \in \mathcal{L}_2 &\Leftrightarrow l \in C \ \& \ l\tau_1 \in \mathcal{L}_1^*, \\ m \in \mathcal{M}_2 &\Leftrightarrow m \in D \ \& \ m\tau_2 \in \mathcal{M}_1^*. \end{aligned}$$

Note that

$$\text{and } \left. \begin{aligned} \mathcal{L}_2\tau_1 &= \mathcal{L}_1^* \\ \mathcal{M}_2\tau_2 &= \mathcal{M}_1^* \end{aligned} \right\}. \quad (2)$$

First of all, we show that $\mathcal{L}_2 \cup \bar{\mathcal{M}}_2$ is unifiable. Since C and D have no common variables, $\tau = \tau_1 \cup \tau_2$ is an unambiguous substitution (and is z.c.). Now,

$$((\mathcal{L}_1 \cup \bar{\mathcal{M}}_1)\theta)^* = (\mathcal{L}_1^* \cup \bar{\mathcal{M}}_1^*)\theta^*\mu,$$

by Lemma 2, Corollary 1, since all the variables of \mathcal{L}_1 , $\bar{\mathcal{M}}_1$ and θ belong to \bar{X} ,

$$= (\mathcal{L}_2 \cup \bar{\mathcal{M}}_2)\tau\theta^*\mu \quad \text{by (2).}$$

But $((\mathcal{L}_1 \cup \bar{\mathcal{M}}_1)\theta)^*$ is a unit, so $\tau\theta^*\mu$ unifies $\mathcal{L}_2 \cup \bar{\mathcal{M}}_2$. In addition, $\tau\theta^*\mu$ is a z.c. substitution. Thus we may apply Lemma 3: Let

$$\mathcal{U}(\mathcal{L}_2 \cup \bar{\mathcal{M}}_2) = \rho$$

where ρ is z.c. Then

$$H = (C - \mathcal{L}_2)\rho \cup (D - \mathcal{M}_2)\rho$$

is a resolvent of C and D , and is obtained by a z.c. unifier.

It remains to show that there is a z.c. τ_3 such that $H\tau_3 = E^*$. First,

$$\begin{aligned} E^* &= ((A - \mathcal{L}_1)\theta \cup (B - \mathcal{M}_1)\theta)^* \\ &= (A - \mathcal{L}_1)^*\theta^*\mu \cup (B - \mathcal{M}_1)^*\theta^*\mu \end{aligned}$$

by Lemma 2, Corollary 1,

$$= (A^* - \mathcal{L}_1^*)\theta^*\mu \cup (B^* - \mathcal{M}_1^*)\theta^*\mu$$

by Remark 3 above.

Now by Lemma 3 there is a z.c. substitution, τ_3 , such that $\rho\tau_3 = \tau\theta^*\mu$. Then,

$$\begin{aligned} H\tau_3 &= (C - \mathcal{L}_2)\rho\tau_3 \cup (D - \mathcal{M}_2)\rho\tau_3 \\ &= (C - \mathcal{L}_2)\tau\theta^*\mu \cup (D - \mathcal{M}_2)\tau\theta^*\mu \\ &= (C - \mathcal{L}_2)\tau_1\theta^*\mu \cup (D - \mathcal{M}_2)\tau_2\theta^*\mu \end{aligned}$$

because C and D have no common variables,

$$= (C\tau_1 - \mathcal{L}_2\tau_1)\theta^*\mu \cup (D\tau_2 - \mathcal{M}_2\tau_2)\theta^*\mu$$

from the definition of \mathcal{L}_2 and \mathcal{M}_2 ,

$$\begin{aligned} &= (A^* - \mathcal{L}_1^*)\theta^*\mu \cup (B^* - \mathcal{M}_1^*)\theta^*\mu \\ &= E^*. \end{aligned}$$

REMARK. Notice from the proof of Lemma 4 that given A, B, E, C, D, τ_1 and τ_2 it is possible to compute H and τ_3 . This involves the following sequence of steps, each of which is clearly computable:

- (1) Find \mathcal{L}_1 and \mathcal{M}_1 by recomputing E .
- (2) Compute \mathcal{L}_1^* and \mathcal{M}_1^* , and then, using C, D, τ_1, τ_2 , compute \mathcal{L}_2 and \mathcal{M}_2 .
- (3) Compute H .
- (4) Find a z.c. τ_3 such that $H\tau_3 = E^*$.

4.3. Alternative Procedure for Eliminating Skolem Functions

The following theorem justifies the alternative procedure:

THEOREM 2. *Let $\text{Tr}(A)$ be a proof tree of A with base sequence S . Let S^* be the result of performing a $*$ -operation on the members of S with respect to some set of Skolem function symbols occurring in S . There exists a proof tree, $\text{Tr}'(B)$, satisfying:*

- (1) S^* is the base sequence of Tr' .
- (2) Each resolvent in Tr' is obtained by a z.c. unifier.
- (3) There is an isomorphism, m , mapping Tr' onto Tr .

- (4) For every node α in Tr' , the clause $C(\alpha)$ at α subsumes the clause $C(m(\alpha))$, at $m(\alpha)$ in Tr . In particular, B subsumes A . Furthermore, for each α there is a z.c. substitution τ_α such that $C(\alpha)\tau_\alpha = C(m(\alpha))^*$.

PROOF. By induction on the level of $\text{Tr}(A)$, using Lemma 4. We denote the level of Tr by $l(\text{Tr})$.

BASIS STEP. If $l(\text{Tr}(A)) = 0$, the theorem is clearly satisfied merely by applying a $*$ -operation to S ; m is the 1-1 correspondence $\{\text{root}(\text{Tr}') \rightarrow \text{root}(\text{Tr})\}$ and τ_α is the empty substitution.

INDUCTION STEP. Suppose the theorem is true for all trees of level n or less, and let $l(\text{Tr}(A)) = n + 1$. Let the parent clauses of A be C and D , so that $\text{Tr}(A)$ is obtained by composing $\text{Tr}(C)$, $\text{Tr}(D)$ and $\text{root}(\text{Tr}(A))$ in the obvious way.

First, apply the $*$ -operation to the base sequence, S , of $\text{Tr}(A)$. If the base sequences of $\text{Tr}(C)$, $\text{Tr}(D)$ are respectively S_1 , S_2 , then $\{S_1, S_2\} = S$, so that applying a $*$ -operation to S is equivalent to applying the $*$ -operation to S_1 and S_2 . Since $l(\text{Tr}(C)) \leq n$, and $l(\text{Tr}(D)) \leq n$, the theorem is true for these trees by hypothesis. So let $\text{Tr}'_1(E)$, m_1 , τ_1 , and $\text{Tr}'_2(F)$, m_2 , τ_2 be trees, isomorphisms and z.c. substitutions satisfying the theorem for $\text{Tr}(C)$ and $\text{Tr}(D)$ respectively.

Let θ be the necessary *change of variables* so that E and $F\theta$ have no common variables. Clearly θ can be chosen to be z.c. There is an inverse substitution (denoted by θ^{-1}) which is also z.c., such that $F\theta\theta^{-1} = F$. Then $E\tau_1 = C^*$ and $F\theta\tau_2 = D^*$, where $\tau_2 = \theta^{-1}\tau_2$ and is obviously z.c. Applying Lemma 4, there is a resolvent B of E and $F\theta$ obtained by a z.c. unifier, and a z.c. τ_3 such that $B\tau_3 = A^*$.

Form $\text{Tr}'(B)$ by composing Tr'_1 and Tr'_2 with a new node α which is the immediate successor of $\text{root}(\text{Tr}'_1)$ and $\text{root}(\text{Tr}'_2)$; let $C(\alpha) = B$ and $m = m_1 \cup m_2 \cup \{\alpha \rightarrow \text{root}(\text{Tr}(A))\}$. Then $\text{Tr}'(B)$, m , τ_3 satisfy the theorem for $\text{Tr}(A)$. This completes the induction step.

STEP 2(b). The proof of Theorem 2 clearly indicates an algorithm for constructing Tr_3 given Tr_2 and a set \bar{F} of Skolem functions to be eliminated. We sketch the main steps of such an algorithm as follows. Assume a standard enumeration of nodes of a tree such that no node is enumerated unless both of its parents have already been enumerated, and also that the leaves are enumerated first.

Let node α' in Tr_3 correspond to α in Tr_2

- (0) If $\bar{F} = \phi$, set $\text{Tr}_3 = \text{Tr}_2$.
 (1) Compute the sequence S^* w.r.t. \bar{F} . Enumerate the leaf nodes of Tr_2 , and for each such α , construct a node α' and label it with *both* the clause $C(\alpha)^*$ from S^* and the substitution $\tau_\alpha = \text{nil}$.

- (2) Enumerate the next node of Tr_2 , say α , with immediate predecessors β and γ . Select $C(\alpha)$, $C(\beta)$, $C(\gamma)$, $C(\beta')$, $C(\gamma')$, $\tau_{\beta'}$, $\tau_{\gamma'}$. Find a z.c. change of variables, θ , such that $C(\beta')$ and $C(\gamma')\theta$ have no common variables, and use the algorithm of Lemma 4 (remark) to compute a resolvent H of $C(\beta')$, $C(\gamma')\theta$ and a z.c. $\tau_{\alpha'}$ such that $H\tau_{\alpha'} = C(\alpha)^*$.
- (3) Construct a new node α' immediately below β' and γ' , and label it with H and $\tau_{\alpha'}$.
- (4) If α is not $\text{root}(\text{Tr}_2)$ go to (2). Otherwise, eliminate all substitutions $\tau_{\alpha'}$ from the constructed nodes $\{\alpha'\}$; the resulting proof-tree is Tr_3 .

It should be noted that the standard unification and resolution algorithms are used in computing Tr_3 . This yields correct answers providing the proof variables, \bar{X} , (those normally used by the procedure) precede in lexical order the "new" variables, \bar{Z} , introduced to replace the Skolem functions. In fact (Lemma 3) any condition will do that forces the unification algorithm to unify an \bar{X} -variable and a \bar{Z} -variable by substituting for the \bar{X} -variable.

4.4. The Correctness of the Procedure

We must now show that the answer statement ANS' produced by the Stage 2(b) logically follows from the axioms and implies the theorem.

First we note directly from condition 4 of Theorem 2, that ANS' subsumes ANS and thus $\text{ANS}' \vdash_L \text{ANS}$ and $\text{ANS}' \vdash_L T$. Furthermore, Tr_3 is a resolution proof tree for ANS' with a base set consisting of the axioms and tautologies (since the $*$ -operation preserves the tautologies in Tr_2). By the same reasoning justifying 1–6 in Section 3.2 we conclude that ANS' logically follows from the axioms. Thus we have a theorem for ANS' (obtained using either of the second stages) analogous to Theorem 1 for ANS :

THEOREM 3. *If T is a logical consequence of consistent axioms \mathcal{A} , and ANS' is the answer statement obtained by applying Stages 1 and 2(a or b) to a resolution refutation with base set $\mathcal{A} \cup \{\sim T\}$, then $\mathcal{A} \vdash_L \text{ANS}'$ and $\text{ANS}' \vdash_L T$.*

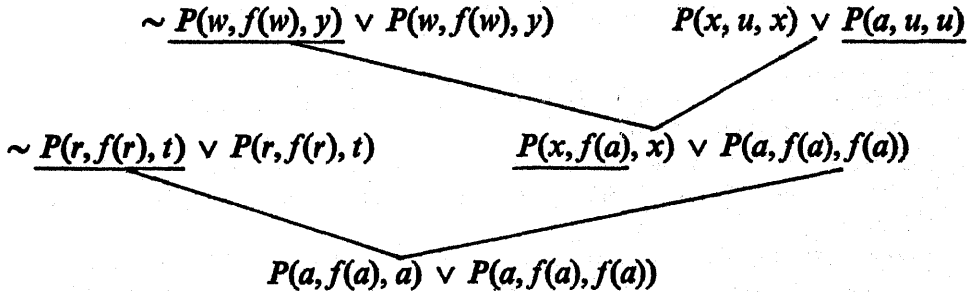
Let us illustrate the complete procedure using Stage 2(b) by considering Example 4 again. Recall that from the single hypothesis

$$(\forall x \forall u)[P(x, u, x) \vee P(a, u, u)]$$

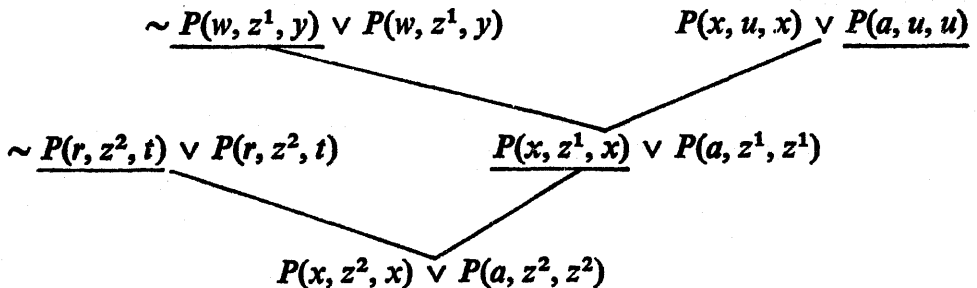
we wanted to prove

$$(\exists x \forall v \exists y)P(w, v, y).$$

We repeat below the tree Tr_2 for ANS:



Now, in Stage 2(b) we use the algorithm of Theorem 2 to create a new tree Tr_3 . This step produces the following proof tree:



We note in particular that the unifying substitution used in the final resolution of this tree is more general than the corresponding one of Tr_2 . Converting the clause at the root of this proof tree to a closed formula in classical logic finally produces the answer statement:

$$(\forall x \forall z)[P(x, z, x) \vee P(a, z, z)].$$

In this case the answer statement is equivalent to the hypothesis.

In conclusion we note that the answer statement extracted by this process depends on the refutation obtained by the proof procedure. When alternative refutations exist, there may thus be alternative answer statements some stronger than others. In the general situation, of course, we have no way of knowing whether the answer statement corresponding to a particular refutation is the strongest possible answer statement.

REFERENCES

1. Green, C. and Raphael, B. The use of theorem-proving techniques in question-answering systems. *Proc. 23rd Nat'l. Conf. ACM*, Brandon Systems Press, Princeton, New Jersey (1968), 169-181.
2. Slagle, J. Experiments with a deductive question-answering program. *Comm. ACM*, 8 (December 1965), 792-798.
3. Waldinger, R. and Lee, R. PROW: a step toward automatic program-writing. *Proc. Int'l. Joint Conf. on Artificial Intelligence*, Washington, D.C. (May 1969).

4. Green, C. Theorem-proving by resolution as a basis for question-answering systems. *Machine Intelligence 4*, Meltzer, B., and Michie, D. (Eds.), American Elsevier Publishing Co., Inc., New York (1969), 183–205.
5. Mostowski, A. Thirty years of foundational studies. *Acta Philosophica Fennica*, **17** (1964), 47–48.
6. Robinson, J. A. A machine-oriented logic based on the resolution principle. *J. ACM*, (1) **12** (January 1965), 23–41.
7. Luckham, D. The resolution principle in theorem proving. *Machine Intelligence 1*, Collins, N. and Michie, D. (Eds.), American Elsevier Publishing Co., Inc., New York (1967), 47–61.
8. Andrews, P. Resolution with merging. *J. ACM*, (3) **15** (July 1968), 367–381.
9. Luckham, D. Refinement theorems in resolution theory. *Proc. IRIA 1968 Symp. on Automatic Demonstration*, Springer Verlag Lecture Notes in Mathematics No. 125, pp. 163–190 (1970).

Accepted February 7, 1971