THE RELATIONAL DATA FILE AND THE DECISION PROBLEM
FOR CLASSES OF PROPER FORMULAS

Robert A. DiPaola
The Rand Corporation, Santa Monica, California

ABSTRACT

The Relational Data File (RDF) of The Rand Corporation is among the most developed of question-answering systems. The "information language" of this system is an applied predicate calculus. The atomic units of information are binary relational sentences. The system has an inference-making capacity.

As part of the actual construction and implementation of the RDF, a theory was developed by J. L. Kuhns to identify those formulas of the predicate calculus which represent the "reasonable" inquiries to put to this system. Accordingly, the classes of definite and proper formulus were defined, and their properties studied. The definite formulas share a semantic property Kuhns judged as necessarily possessed by a reasonable question to be processed by the RDF. The author has previously shown that the decision problem for the class of definite formulas is recursively unsolvable. The proper formulas are definite, and satisfy additional syntactic conditions intended to make them especially suitable for machine processing. The class of proper formulas depends on which logical primitives are employed. Different primitives give rise to different classes of formulas. A formula which can be effectively transformed into a proper equivalent is admissible. Kuhns conjectures that with respect to one particular class of proper formulas, all definite formulas are admissible. In the paper it is shown that the decision problem for several classes of proper formulas is solvable. The following results are established. Theorem 1: The class of proper formulas in prenex form on any complete set of connectives is recursive. Theorem 2: The class of proper formulas on $\neg$, $\vee$, $\exists$ is recursive. Theorem 3: The class of proper formulas on $\neg$, $\supset$, $\exists$ is recursive. Theorem 4: The class of proper formulas on $\neg$, $\supset$, $\vee$, $\exists$ is recursive. Thus, there is a mechanical decision procedure which determines whether an arbitrary formula is a member of the class. It follows that the analogues of Kuhns' conjecture for these classes are false.

KEY WORDS AND PHRASES

data retrieval system, Relational Data File, applied predicate calculus, definite formulas, proper formulas, solvable cases of the decision problem, recursively unsolvable decision problem, question-answering system

I. Introduction

The so-called "information language" of Rand's Relational Data File is an applied predicate calculus [3, 4, 5, 6, 7, 8]. As an integral part of the design and implementation of this storage and retrieval system, a theory was developed by J. L. Kuhns to identify and systematically study those formulas of this calculus that represent the "reasonable" questions to put to a computer implementation of this system, with emphasis placed upon those representations that are supposedly especially suited for machine processing [3, 4]. Accordingly, three classes of formulas were defined--definite, proper, and admissible. The definite formulas are defined semantically and are invariant under the sentential and quantificational transformations of the predicate calculus. They share a semantic condition judged as being necessarily possessed by the symbolic representations of reasonable inquiries. To elaborate, the notion of a data base as defined in [3] amounts to the common notion of a <u>structure</u> with a finite number of relations; or, from the vantage point of a formal language, it is an <u>interpretation</u> of a finite number of predicate and constant symbols. The formulas F with free variables that are definite have the property that the sets of true instances of F in an interpretation I of F and in a special extension I' of I are the same. The formulas without free variables are definite if their truth value is always preserved on passage from an interpretation I to an extension I' of I of the aforesaid special type. A precise definition is given in Sec. 2.

The proper formulas are those definite formulas that satisfy a certain syntactic condition--namely, their principal subformulas must also be proper. Thus, all subformulas must be definite. The rationale behind this definition presumably runs as follows: The definite formulas mirror the reasonable inquiries, while the proper formulas are such that all of their parts, i.e., subformulas, also have this desirable property. The admissible formulas are those that can be transformed into proper equivalents; that is, if F is admissible, there is a proper formula G, such that for each interpretation I, with domain D, F and G are satisfied in D at exactly the same instances. It is conjectured in [3] that every definite formula is admissible. Of course, even for one who accepts the theory as sound, sensible, and adequate it is <u>a priori</u> clear that the admissibility of a formula F is of little or no value unless the transformation φ of F into a proper equivalent G is effective.

The concept, then, of a proper formula involves the notion of a subformula. But determining which consecutive parts of a given formula are subformulas is, of course, a syntactic notion and depends on the identity of sentential and quantificational connectives employed. Thus, the subclasses of proper formulas, in contradistinction to the class of definite formulas, depend on which of the logical connectives are taken as primitives. This paper considers the decision problem for various classes of proper formulas. We point out that an affirmative solution to the decision problem for a particular class Π of proper formulas refutes the version of the aforementioned conjecture relative to the class of definite formulas on <u>any</u> complete set of logical connectives; that is, if Δ is a class of definite formulas on any complete set of connectives, it follows, as in [2], that Δ is not re (recursively enumerable). Hence, if Π is recursive, there is no effective transformation φ such that for each formula F, $F \in \Delta \Leftrightarrow \varphi(F) \in \Pi$.

II. Definitions

The language $\mathcal{L}$ that we use is the language of the full, pure first-order predicate calculus without equality, augmented with infinitely many individual constants. A <u>formula</u> in the predicate symbols $P_1^{n_1}, P_2^{n_2}, \ldots, P_t^{n_t}$, where the superscript denotes the rank or degree of the predicate symbol, and in the constants $c_1, c_2, \ldots, c_k$ is any formula F whose only symbols, other than sentential connectives, quantifiers, and individual variables, occur among $P_1^{n_1}, P_2^{n_2}, \ldots, P_t^{n_t}, c_1, \ldots, c_k$. An <u>interpretation</u> of F is a system $I = \langle D; R_1^{n_1}, \ldots, R_t^{n_t}, d_1, \ldots, d_k \rangle$, where D is a nonempty set, each relation $R_i^{n_i}$ is defined on D^{n_i} and assigned to $P_i^{n_i}$, and each d_j is assigned to c_j, $i = 1, \ldots, t$; $d_j \in D$, $j = 1, \ldots, k$. An interpretation I of F is said to be <u>finite</u> if the domain D of I is finite. Developments of the notion of interpretation or structure may be found in [9] or [10].

Definition

If F is a formula with m free variables and $I = \langle D; R_1^{n_1}, R_2^{n_2}, \ldots, R_t^{n_t}; d_1, \ldots, d_k \rangle$ is a finite interpretation of F, T(F, I) is the set of members of D^m that satisfy F, if $m > 0$. If $m = 0$, we call F a <u>sentence</u>, and T(F, I) = t (truth) and T(F, I) = f (falsity) according to whether F is satisfied or not satisfied in I. We say a sentence F is <u>finitely satisfiable</u> if there is a finite interpretation in which it is satisfied; F is <u>finitely valid</u> if it is satisfied in all finite interpretations.

Definition

Let F be a formula and $I = \langle D; R_1^{n_1}, \ldots, R_t^{n_t}; d_1, \ldots, d_k \rangle$ a finite interpretation of F. Let * be an individual not in D. A <u>*-extension I'</u> of I for F is the interpretation $I' = \langle D'; S_1^{n_1}, \ldots, S_t^{n_t}; d_1, \ldots, d_k \rangle$, where $D' = D \cup \{*\}$ and $S_i^{n_i}$ is the extension of $R_i^{n_i}$ from D^{n_i} to D'^{n_i}, such that $S_i^{n_i}$ is false on any member of D'^{n_i} that has * among its components.

Definition

A formula F is said to be <u>definite</u> if, for all finite interpretations I of F, T(F, I) = T(F, I'), where I' is a *-extension of I for F.

Thus, the theorems and refutables of the predicate calculus that are sentences are definite. Clearly, all atomic formulas of $\mathcal{L}$ are definite. Closure properties of the class of definite formulas are investigated at length in [3].

Remark

We recall again the following theorem [2]:

> If A_k, $k \geq 0$, is the class of definite formulas on $\mathcal{L}$ with k free variables, then A_k is not recursively enumerable.

Thus, the decision problem for each A_k is recursively unsolvable.

Examples

Consider the following formula F : $\forall x \exists y P(x, y) \supset \exists y \forall x P(x, y)$. Let $D = \{1, 2\}$ and P be interpreted in D by the binary relation R true on the pairs (1, 2) and (2, 1), and false on the pairs (1, 1) and (2, 2). Thus, in this case $I = \langle D; R \rangle$. It is easily seen that $\forall x \exists y P(x, y)$ is true in I, whereas $\exists y \forall x P(x, y)$ is false in I. By the truth-table for implication, $\forall x \exists y P(x, y) \supset \exists y \forall x P(x, y)$ is consequently false in I.

We form the $*$-extension $I' = \langle D'; S \rangle$ of I by taking $D' = \{1, 2, *\}$ and S to be the binary relation which agrees with R on $D \times D$ and is false on the pairs $(1, *)$, $(*, 1)$, $(2, *)$, $(*, 2)$, and $(*, *)$. Thus, $\forall x \exists y P(x, y)$ is false in I' since $R(*, *)$, $R(*, 1)$, and $R(*, 2)$ are all false in D'. Similarly, $\exists y \forall x P(x, y)$ is false in I'. Hence F is true in I'. Since F is false in I but true in the $*$-extension I' of I we see that F is finitely satisfiable, is not finitely valid, and is not definite.

Consider the formula G : $\exists x P(x) \& \forall x[P(x) \supset Q(x)]$. Let D be any non-empty finite set. Suppose $I = \langle D; R_1, R_2 \rangle$ is an interpretation in which R_1 and R_2 are unary relations serving as the interpretations of P and Q, respectively. If R_1 is false throughout D, then $\exists x P(x)$ and hence G are false in I. Thus, G is not finitely valid. But, if, for example, R_1 holds for but a single member of D and R_2 is universally true in D, the G holds in I. Thus, G is finitely satisfiable.

Suppose that $I' = \langle D'; S_1, S_2 \rangle$ is a $*$-extension of I. Assume that G is false in I. If $\exists x P(x)$ is false in I, then since $S_1(*)$ is false, $\exists x P(x)$ and hence G are false in I'. If $\exists x P(x)$ is true in I but $\forall x(P(x) \supset Q(x))$ is false in I, then there is an element $d \in D$ such that S_1 is true on d and S_2 is false on d. Hence $S_1(d)$ is true and $S_2(d)$ is false in D'. Thus, $\forall x(P(x) \supset Q(x))$ is false in I'. Therefore, G is false in I'. Assume, now, that G is true in I. Then $\exists x P(x)$ is true in I and hence true in I'. By assumptions, $\forall x(P(x) \supset Q(x))$ is true in I. S_1 and S_2 are both false on $*$. So, $\forall x(P(x) \supset Q(x))$ is true in I'. Therefore G is true in I'. Since I is arbitrary, it follows that G is definite.

Definition

Let $u_1, u_2, \ldots, u_k$ be the unary connectives and $b_1, b_2, \ldots, b_\ell$ the binary connectives of the language $\mathfrak{L}$. It is assumed that at least one of $\exists$ (the existential quantifier) and $\forall$ (the universal quantifier) is among $u_1, u_2, \ldots, u_k$. We inductively define the property of being a _subformula_ of a formula A of $\mathfrak{L}$.

(1) A is a _subformula_ of A.

(2) If A is $u_i(B)$ and u_i is a propositional connective, then each _subformula_ of B is a _subformula_ of A, $i = 1, 2, \ldots, k$.

(3) If A is $u_i x(B)$, u_i is either $\exists$ or $\forall$, and x is a variable that occurs free in B, then each _subformula_ of B is a _subformula_ of A, $i = 1, 2, \ldots, k$.

(4) If A is $b_j(B, C)$, then each _subformula_ of B is a _subformula_ of A and each _subformula_ of C is a _subformula_ of A, $j = 1, 2, \ldots, \ell$.

(5) A formula B is a _subformula_ of A only as prescribed by (1) through (4) above.

Definition

A formula A of $\mathfrak{L}$ is _proper_ iff each subformula of A is definite. In the following, we are mainly interested in proper formulas on the more familiar sets of connectives. If the primitive connectives are basically those of [3, 4]--namely, $\neg$, $\vee$, &, $\supset$, $\not\supset$ (but not), and $\exists$--the above definition of subformula reads as follows:

(1) A is a <u>subformula</u> of A.

(2) If A is $\neg B$, then each <u>subformula</u> of B is a subformula of A.

(3) If A is $B \vee C$, $B \& C$, $B \supset C$, $B \not\supset C$, then each <u>subformula</u> of B and each <u>subformula</u> of C is a <u>subformula</u> of A.

(4) If A is $\exists x B$, then each <u>subformula</u> of B is a <u>subformula</u> of A.

(5) A formula B is a <u>subformula</u> of A only as prescribed by (1) through (4) above.

Notation

To facilitate the frequent use of certain terms, we shall write "fv," "fs," "nfv," "nfs," "re," for "finitely valid," "finitely satisfiable," "not finitely valid," "not finitely satisfiable," and "recursively enumerable," respectively. Also, we write "$F_1 \overset{I}{\Leftrightarrow} F_2$" to mean that, for each finite interpretation I, the formulas F_1 and F_2 hold at exactly the same instances of the domain of I. If, as is usually the case, $\supset$ is among our primitive connectives, we write "$A \equiv B$" for "$(A \supset B) \& (B \supset A)$."

Definition

We define the <u>height</u> h of a formula F.

(1) If F is atomic, $h(F) = 0$.

(2) If u is a unary primitive connective of the language $\mathcal{L}$ and F is $u(A)$, then $h(F) = h(A) + 1$.

(3) If b is a binary primitive connective of $\mathcal{L}$, and F is $b(A, B)$, then $h(F) = \max(h(A), h(B)) + 1$.

III. Results

Unlike the class of definite formulas, the subclasses of proper formulas depend on which of the logical connectives are taken as primitives. Thus, classes Π_1 and Π_2 of proper formulas employing different but, in the usual sense, equivalent sets of connectives may have different properties. This dependence is, however, not absolute, as the following result demonstrates.

<u>THEOREM 1</u>. <u>The class Π_p of proper formulas in prenex form defined on any complete set of connectives is recursive</u>.

<u>Proof</u>. Consider a formula $F \in \Pi_p$. F is of the form $Qx_1Qx_2 \ldots Qx_mM(x_1, x_2, \ldots, x_m, y_1, y_2, \ldots, y_n)$, where M is quantifier-free and each Q is $\exists$ or $\forall$. Now, it is easy to see that it is decidable whether a quantifier-free formula is fv or nfs. Consequently, by using the various results governing the closure properties of the class of definite formulas given in [3] and [4], p. 11, it is decidable whether M is proper. If M is not proper, F is not proper. So, assume that M is proper. The existential quantification of a proper formula is proper [3]; therefore, if each Q is $\exists$, then F is proper. Suppose that i_o is the greatest integer not exceeding m, such that Qx_{i_o} is $\forall x_{i_o}$; then $\forall x_{i_o} \exists x_{i_o+1} \ldots \exists x_m M$ is proper iff the sentence $\exists y_1 \exists y_2 \ldots \exists y_n \exists x_1 \exists x_2 \ldots \forall x_{i_o} \exists x_{i_o+1} \ldots \exists x_m M$ is nfs [3], p. 62. Thus, $\forall x_{i_o} \exists x_{i_o+1} \ldots \exists x_m M$ is proper iff the sentence $\forall y_1 \forall y_2 \ldots \forall y_n \forall x_1 \forall x_2 \ldots \exists x_{i_o} \forall x_{i_o+1} \ldots \forall x_m \neg M$ is fv. But it is decidable

whether such sentences are fv [1], pp. 67-70, 72-74. Hence, it is decidable whether $\exists y_1 \exists y_2 \ldots \exists y_n \exists x_1 \exists x_2 \ldots \forall x_{i_o} \exists x_{i_o+1} \ldots \exists x_m M$ is fs. If the latter sentence is fs, then $\forall x_{i_o} \exists x_{i_o+1} \ldots \exists x_m M$ is not proper, and hence F is not proper. If it is nfs, then $\forall x_{i_o} \exists x_{i_o+1} \ldots \exists x_m M$ is proper. However, if $\exists y_1 \exists y_2 \ldots \exists y_n \exists x_1 \exists x_2 \ldots \forall x_{i_o} \exists x_{i_o+1} \ldots \exists x_m M$ is nfs, it follows that $Qx_j Qx_{j+1} \ldots \forall x_{i_o} \exists x_{i_o+1} \ldots \exists x_m M$ is nfs, $j = i_o-1, i_o-2, \ldots, 1$, and hence is proper. Consequently, if $\exists y_1 \exists y_2 \ldots \exists y_n \exists x_1 \exists x_2 \ldots \forall x_{i_o} \exists x_{i_o+1} \ldots \exists x_m M$ is nfs, then F is proper. Q.E.D.

For the next result, we denote by "Π" the class of proper formulas on $\neg$, $\vee$, $\exists$.

<u>LEMMA 2</u>. <u>If</u> $F \in \Pi$, <u>then it is decidable whether</u> F <u>is</u> fv <u>or</u> F <u>is</u> fs.

<u>Proof</u>. Suppose F contains no occurrence of negation signs. Then plainly F is fs. Also, the closure of a prenex form of F is of the form $\forall x_1 \forall x_2 \ldots \forall x_m \exists y_1 \exists y_2 \ldots \exists y_n M$, where $x_1, x_2, \ldots, x_m$ are the free variables of F. It is decidable whether sentences of this form are fv [1], pp. 70-71. Consequently, it is decidable whether F is fv.

We henceforth assume that F contains occurrences of negation signs. Without loss of generality, we assume that no double negations occur in F. We employ induction on the height h of F. By our assumption, since F is proper, the induction starts with h = 2. For h = 2, F is of the form $\neg \exists x P(x)$, where $P(x)$ is atomic. It is, of course, decidable whether such a formula is fv or fs. Assume that the lemma holds for all formulas F in Π of height $h \leq k$. Consider a formula F of Π of height $h = k + 1$. Suppose F contains a subformula of the form $\neg \exists u A \vee B$, where B has free variables. Since this subformula is proper, $\neg \exists u A$ must be nfs and hence $\neg \exists u A \vee B \overset{I}{\Leftrightarrow} B$. In fact, suppose F' is the formula obtained from F by replacing this particular occurrence of $\neg \exists u A \vee B$ by B. It is easy to see that F' is proper. Moreover, $F' \overset{I}{\Leftrightarrow} F$. F' is of height $h < k + 1$; hence, by the induction hypothesis, it is decidable whether F' is fv or fs. Consequently, it is decidable whether F is fv or fs.

We now assume that F contains no subformula of the form $\neg \exists u A \vee B$ (or $B \vee \neg \exists u A$), where B has free variables. Consider all occurrences of subformulas of F that have a single occurrence of a negation sign and that have negation as the senior connective. Such subformulas are sentences of the form $\neg \exists x_1 \exists x_2 \ldots \exists x_m M$, if we ignore the trivial case of the negation of proposition letters. If F is of such a form, then it is decidable whether F is fv or fs without use of the induction hypothesis [1], pp. 62-63. Suppose, therefore, that F is not of this form and such that A is a subformula of F of this type--that is, a sentence $\neg \exists x_1 \exists x_2 \ldots \exists x_m M$. We assert that A lies within the scope of no other negation sign. Suppose it did. Then there is a subformula $\neg \exists x B$ of F, such that A is a subformula of B. Since x occurs free in B, there is a subformula of B of the form $C \vee A$ or $A \vee C$, where C has free variables. This contradicts our assumption that F contains no subformulas of this form; our assertion is therefore justified.

Moreover, for the same reason, A lies within the scope of no other existential quantifier. Let $A_1, A_2, \ldots, A_k$ all be sequential occurrences of such sentences that occur as subformulas of F. Consider the left-most occurrence A_1 and next-to-left-most occurrence A_2 of such sentences. Since F is a formula on $\neg$, $\vee$, $\exists$, there are proper formulas B_1 and B_2 such that $B_1 \vee A_1 \vee B_2 \vee A_2$ is a subformula of F. The binary connective $\vee$ may occur in B_1 and B_2, but no negation signs occur in either. Hence, each occurrence of an existential quantifier in either of B_1 or B_2 is unnegated. The same considerations apply to A_2 and A_3, A_3 and A_4, $\ldots$, A_{k-1} and A_k. Thus, F is a formula of the form $B_1 \vee A_1 \vee B_2 \vee A_2 \vee \ldots \vee A_{k-1} \vee B_k \vee A_k \vee B_{k+1}$, using the associativity of $\vee$. Negation signs occur only in $A_1, A_2, \ldots, A_k$, and then only as senior connectives. (Of course, various of the formulas $B_1, B_2, \ldots, B_{k+1}$ may be empty, or F may be of the $B_1 \vee A_1$ for a nonempty B_1; however, in such cases, the situation is yet simpler and the same analysis applies.) F thus has a prenex form (on $\neg$, $\vee$, $\exists$, $\forall$) in which each universal quantifier precedes each existential quantifier. The finite validity of such formulas is decidable [1], pp. 70-71. Similarly, F has a prenex form in which each existential quantifier precedes each universal quantifier. The finite satisfiability of such formulas is decidable. Consequently, it is decidable whether F is fs or fv. Q.E.D.

We define the propositional formula P(h): If F is a formula (on $\neg$, $\vee$, $\exists$) of height h and each subformula of F of height $k < h$ is proper, then it can be effectively determined whether F is proper.

LEMMA 3. P(h) *is true for each* h.

Proof. P(0) certainly holds, since atoms are proper and h = 0 implies that F is atomic. Assume P(h) for $h \leq k$ and consider P(k + 1). Let F be a formula of height k + 1, such that each subformula A of F of height $h \leq k$ is proper.

Case 1. F is $\neg A$. Then if A is a sentence, F is proper. Otherwise, F is not proper.

Case 2. F is $\exists x A$, where x occurs free in A. Then F is proper.

Case 3. F is $A \vee B$. If the A and B have the same free variables, then F is proper [3]. By Lemma 2, it can be decided whether A is fs or B is fs. If both A and B are nfs, then F is proper. If one of A and B, say A, is nfs and each free variable of A is a free variable of B, then F is proper [3]. On the other hand, if A and B do not have the same free variables, and (1) both A and B are fs, or (2) just one, say B, is fs but A has free variables that do not occur in B, then F is not proper. Q.E.D.

We therefore have

THEOREM 4. *The class of proper formulas of* $\vee$, $\neg$, $\exists$ *is recursive*.

Proof. Consider a given formula F of height h. By 3, P(k) holds for each k. The atoms of F are proper. Thus, it can be determined whether the subformulas of height 1, 2, ..., h of F are proper. Hence, it can be determined whether F is proper. Q.E.D.

We now take up the decision problem for the class of proper formulas on $\neg$, $\supset$, $\exists$.

We define a transformation φ of the formulas on $\neg$, $\supset$, $\exists$ into the formulas on $\neg$, $\vee$, $\exists$.

$$\varphi(A) = A \text{ if } A \text{ is atomic},$$
$$\varphi(\neg A) = \neg\varphi(A),$$
$$\varphi(\exists xA) = \exists x\varphi(A),$$
$$\varphi(A\supset B) = \varphi(\neg A)\vee\varphi(B).$$

LEMMA 5. A *is proper on* $\neg$, $\supset$, $\exists$ *iff* $\varphi(A)$ *is proper on* $\neg$, $\vee$, $\exists$.

Proof. The proof proceeds by induction on the height h of A. If h = 0, then A is atomic, and $\varphi(A) = A$; hence, the assertion holds. Suppose the statement is true for all formulas B of height $h \leq k$. Consider a formula A of height $h = k + 1$.

Case 1. A is $\neg B$. Then $\varphi(A)$ is $\neg\varphi(B)$. Assume A is proper; then B is proper. Then, by the induction hypothesis, $\varphi(B)$ is proper. Since $\neg B$ is proper, A is a sentence. Hence, $\varphi(B)$ is a sentence and $\varphi(A)$ is proper. Likewise, if $\varphi(A) = \neg\varphi(B)$ is proper, $\varphi(B)$ is proper and a sentence. By definition of φ, B is a sentence; by the induction hypothesis, B is proper. Hence, A is proper.

Case 2. A is $\exists xB$. Then $\varphi(A) = \exists x\varphi(B)$. A and $\varphi(A)$ are proper iff B and $\varphi(B)$ are proper. By the induction hypothesis, B is proper iff $\varphi(B)$ is proper. Hence, A is proper iff $\varphi(A)$ is proper.

Case 3. A is $B\supset C$. Then $\varphi(A)$ is $\neg\varphi(B)\vee\varphi(C)$.

Subcase 1. B and C are both sentences. Then $\varphi(B)$ and $\varphi(C)$ are sentences. A is proper iff both B and C are proper. By the induction hypothesis, B is proper and C is proper iff $\varphi(B)$ is proper and $\varphi(C)$ is proper, respectively. Since B is a sentence, B is proper iff $\neg\varphi(B)$ is proper. But $\neg\varphi(B)\vee\varphi(C)$ is proper iff $\neg\varphi(B)$ and $\varphi(C)$ are proper [3, 4]. Therefore, A is proper iff $\neg\varphi(B)\vee\varphi(C)$ is proper.

Subcase 2. At least one of B, C is not a sentence. In this subcase, A is proper iff B is a proper fv sentence and C is proper. Thus if A is proper, C is not a sentence. Hence, $\varphi(C)$ is not a sentence and, by the induction hypothesis, $\varphi(C)$ is proper. Also, if B is a proper, fv sentence, then $\neg\varphi(B)$ is a nfs proper sentence. Hence, $\neg\varphi(B)\vee\varphi(C)$ is proper [4], p. 11. Similarly, since one of B, C is not a sentence, if $\neg\varphi(B)\vee\varphi(C)$ is proper, then $\neg\varphi(B)$ is a proper sentence [3], p. 51, $\varphi(C)$ has free variables, and hence at least $\neg\varphi(B)$ is nfs [4], p. 11. Hence, B is, by the induction hypothesis, a proper sentence and C is a proper formula with free variables. Since $\vdash F \equiv \varphi(F)$ in the predicate calculus for each formula F, B is fv. Hence, A is proper [4], p. 11. We conclude that A is proper iff $\neg\varphi(B)\vee\varphi(C)$ is proper. Q.E.D.

We thus obtain

THEOREM 6. *The class of proper formulas on* $\neg$, $\supset$, $\exists$ *is recursive*.

Proof. Indeed, an arithmetization of our formalism, together with Lemma 5, shows that the decision problem for the class of proper formulas on $\neg$, $\supset$, $\exists$ is 1-1 reducible to the decision problem for the class of proper formulas on $\neg$, $\vee$, $\exists$. By Theorem 4, the latter class is recursive. Hence, the class of proper formulas on $\neg$, $\supset$, $\exists$ is recursive. Q.E.D.

The above statement is easily extended to give

THEOREM 7. The class of proper formulas on $\neg$, $\vee$, $\supset$, $\exists$ is recursive.

Proof. We first extend the definition of φ by inclusion of the clause,

$$\varphi(A \vee B) = \varphi(A) \vee \varphi(B).$$

One more case arises in addition to the three considered in 5:

Case 4. A is $B \vee C$. Then $\varphi(A) = \varphi(B) \vee \varphi(C)$. By the induction hypothesis, B is proper iff $\varphi(B)$ is proper and C is proper iff $\varphi(C)$ is proper [4], p. 11.

Subcase 1. B and C have the same free variables. Then $\varphi(B)$ and $\varphi(C)$ have the same free variables; hence, $A = B \vee C$ is proper iff $\varphi(B) \vee \varphi(C)$ is proper.

Subcase 2. B and C do not have the same free variables. Since $\vdash C \equiv \varphi(C)$ and $\vdash B \equiv \varphi(B)$ in the predicate calculus, then (1) B and C are nfs iff $\varphi(B)$ and $\varphi(C)$ are nfs and (2) just one, say B, of B, C is nfs and each free variable of B is a free variable of C iff just one, say $\varphi(B)$, of $\varphi(B)$, $\varphi(C)$ is nfs and each free variable of $\varphi(B)$ is a free variable of $\varphi(C)$. So, again, $B \vee C$ is proper iff $\varphi(B) \vee \varphi(C)$ is proper. Q.E.D.

IV. Remarks

Reducibility

We denote by "Δ_p", "Δ_1", "Δ_2", "Δ_3" the classes of definite formulas in prenex form, on $\neg$, $\vee$, $\exists$, on $\neg$, $\supset$, $\exists$, and on $\neg$, $\vee$, $\supset$, $\exists$, respectively; by "Π_p", "Π_1", "Π_2", "Π_3", we denote the classes of definite formulas in prenex form on $\neg$, $\vee$, $\exists$, on $\neg$, $\supset$, $\exists$, and on $\neg$, $\vee$, $\supset$, $\exists$, respectively. In contrast to Theorem 1, Theorem 4, Theorem 6, and Theorem 7, the decision problem for each of Δ_p, Δ_1, Δ_2, and Δ_3 is recursively unsolvable. In fact, none of these classes of definite formulas is recursively enumerable but in fact each is a set of the highest degree of unsolvability for sets expressible by the Π_1^0 predicates of the Kleene hierarchy [2], [10]. It follows that the analogue of the conjecture of Kuhns described in the Introduction is false for all combinations of Δ's and Π's; that is, if we momentarily denote by "Δ" any of Δ_p, Δ_1, Δ_2, Δ_3 and by "Π" any of Π_p, Π_1, Π_2, Π_3, then there is no effective transformation φ such that

$$F \in \Delta \Leftrightarrow \varphi(F) \in \Pi \text{ and } F \overset{I}{\Leftrightarrow} \varphi(F).$$

In fact, none of Δ_p, Δ_1, Δ_2, Δ_3 is many-one reducible, or even Turing reducible to any of Π_p, Π_1, Π_2, Π_3. For, if one of Δ_p, Δ_1, Δ_2, Δ_3 were Turing reducible to one of Π_p, Π_1, Π_2, Π_3, then that one of Δ_p, Δ_1, Δ_2, Δ_3 would be recursive, which is not the case. The final general statement along these lines one can make is that none of Δ_p, Δ_1, Δ_2, Δ_3 is reducible to any of Π_p, Π_1, Π_2, Π_3 by a recursively enumberable function, where we understand a function f to be re if the two-place relation $f(x) = y$ is re. Thus, there is no re function f, such that $F \in \Delta \Leftrightarrow f(F) \in \Pi$.

A Concluding Comment

None of the results, Theorem 1, Theorem 4, Theorem 6, Theorem 7, is *a priori* obvious. However, if one inspects the solvable cases of the decision problem [1] on which the proofs of these results are based, one sees that the algorithms that solve the decision problems of Π_p, Π_1, Π_2, Π_3, respectively, are reasonably simple and "programmable." There is no factor intrinsic to the algorithms that would render implementation infeasible. Computational difficulties would stem rather from testing a formula for finite validity or finite satisfiability when the data base is large.

REFERENCES

(1) ACKERMANN, W. "Solvable Cases of the Decision Problem," in L. E. J. Brouwer, E. W. Beth, A. Heyting (eds.), Studies in Logic and the Foundations of Mathematics, North-Holland Publishing Co., Amsterdam, 1954.

(2) DIPAOLA, R. A. "The Recursive Unsolvability of the Decision Problem for the Class of Definite Formulas," JACM, Vol. 16, No. 2, April 1969, pp. 324-327; also, The Rand Corporation, RM-5639-PR, April 1968.

(3) KUHNS, J. L. Answering Questions by Computer: A Logical Study, The Rand Corporation, RM-5428-PR, December 1967.

(4) KUHNS, J. L. Interrogating a Relational Data File: Remarks on the Admissibility of Input Queries, The Rand Corporation, R-511-PR, November 1970.

(5) LEVIEN, R. E. Relational Data File: Experience with a System for Propositional Data Storage and Inference Execution, The Rand Corporation, RM-5947-PR, April 1969.

(6) LEVIEN, R. E. "Relational Data File II: Implementation," in G. Schector (ed.), Information Retrieval: A Critical View, Thompson Book Co., Washington, D. C., 1966.

(7) LEVIEN, R. E., and M. E. MARON "A Computer System for Inference Execution and Data Retrieval," Comm. ACM, Vol. 10, No. 11, November 1967, pp. 715-721.

(8) MARON, M. E. "Relational Data File I: Design Philosophy," in G. Schector (ed.), Information Retrieval: A Critical View, Thompson Book Co., Washington, D. C., 1966.

(9) MENDELSON, E. Introduction to Mathematical Logic, D. Van Nostrand Co., New York, 1964.

(10) SHOENFIELD, J. R. Mathematical Logic, Addison Wesley, Reading, Mass., 1967.