provide a heuristically useful gauge of the degree of “belongingless” of each individual to a given high-level type.

There are in addition other commands, such as FORCE, which permits one to ignore a particular relationship in a type if one suspects that it is spurious. It is easy to think of other commands which would be useful in investigating the structure of the data and uncovering high-level types. One particular set of commands, which will be implemented soon, will permit the user to give a name to a subset of individuals or variables and make it current at a later time by merely saying GET,xxx, where xxx is the name. Also planned are optimizing routines which will heuristically attempt to build the most cohesive possible high-level types around a few individuals selected by the researcher to represent a hypothetical type, within constraints imposed by the researcher. (For example, the correlation between variables 2 and 7 must be positive.) This latter command is likely to cause problems because of excessive use of CPU time.

This program, which is called INTERFORM, for INTERactive pattern FORMation, has as yet been tested only on made-up data, not real-world social science data. It is planned to eventually embed it in a general-purpose program for the analysis of social science data, along the lines of SDC’s TRACE.

REFERENCES

1. L McQuitty
   The mutual development of some theories of types and some pattern analytic methods
   Presidential Address Division V Evaluation and Measurement The American Psychological Association
   September 1965
2. L McQuitty
   Dr McQuitty’s papers have appeared frequently in Education and Psychological Measurement for the last several years and the interested reader is referred to this source
3. L Press M Rogers
   IDEA—A conversational heuristic program for interactive data exploration and analysis
   Proc of the 22nd National ACM Conference 1967
4. G Shure R Meecher W Moore
   TRACE: Time-shared routines for analysis, classification and evaluation
   Spring Joint Computer Conference Proceedings 1967
5. I E Robinson D Foster
   The simulation of human systems
   Digest of the Second Conference on Applications of Simulation 1968
6. P F Winch
   The idea of a social science 1958
7. The concept of anomie was developed by French sociologist Emile Durkheim. An anomie individual is conceptualised as being alienated, unable to find meaning or structure in the organization of the world or society. It is reasonable to postulate that, within the theoretical framework developed in this paper, anomie individuals would lack clear membership in any high-order type—i.e., the variations in their beliefs would be unstructured and random.
Completeness results for E-resolution*

by ROBERT ANDERSON

University of Texas
Austin, Texas

INTRODUCTION

Since their introduction in 1965,7 resolution based deductive systems for the first-order predicate calculus have been extensively investigated and utilized by researchers in the field of automatic theorem-proving by computer. Part of this research has been directed at finding techniques for treating the equality relation within the framework of resolution based deductive systems.2,3,4,5,9,10 Perhaps the most natural treatment of equality, introduced so far, is by means of the paramodulation principle which when used in conjunction with resolution forms a complete deductive system for the first-order predicate calculus with equality.5,6,11 A very similar technique for treating equality was introduced4 and called E-resolution. In fact E-resolution can be viewed as a restricted form of paramodulation and resolution. The purpose of this paper is to define E-resolution in terms of paramodulation and resolution and to prove the completeness of E-resolution and several modifications of E-resolution.

PRELIMINARIES AND TERMINOLOGY

The reader is assumed to be familiar with the notation and terminology of resolution and paramodulation.5,6,7,11 In addition the reader is assumed to be familiar with the technique introduced in Reference 1, for establishing the completeness of resolution based deductive systems. In that regard, recall that the technique is based on mathematical induction on the parameter k (the excess literal parameter) defined as follows: For any set S of clauses, k(S) is defined to be (the total number of appearances of literals in S) minus (the number of clauses in S). To define E-resolution in terms of paramodulation and resolution we need the following definitions:

Definition 1. If S is a set of clauses and C is a clause in S and l is a literal in C then define

\[ P^{(0)}(S, C, l) = \{C\} \]

\[ P^{(i)}(S, C, l) = \text{the set of all clauses which can be obtained from } C \text{ by paramodulating from some clause in } S \text{ into the literal } l \text{ in } C. \]

and by induction,

\[ P^{(n)}(S, C, l) = \text{the set of all clauses which can be obtained from clauses } C' \in P^{(n-1)}(S, C, l) \text{ by paramodulating from some clause in } S \text{ into the literal } l' \text{ (in } C') \text{, which is descended from the literal } l \text{ in } C. \]

Definition 2.

\[ P^{(\omega)}(S, C, l) = \bigcup_{n \in \mathbb{Z}} P^{(n)}(S, C, l). \]

Thus \( P^{(\omega)}(S, C, l) \) consists of all the clauses which can be obtained from C by paramodulating any finite number of times from clauses in S into the literal l (or its descendents) in C. There is no paramodulation into any other literal in C.

Definition 3. If S is a set of clauses then \( C_2 \) is an E-resolvent of S if and only if there exist clauses \( C_1 \) and \( C_2 \) in S and literals \( l_1 \) in \( C_1 \) and \( l_2 \) in \( C_2 \) and there exist clauses \( C'_1 \in P^{(\omega)}(S, C_1, l_1) \) and \( C'_2 \in P^{(\omega)}(S, C_2, l_2) \) such that \( C_1 \) is a resolvent of \( C'_1 \) and \( C'_2 \) and the literals resolved upon in \( C'_1 \) and \( C'_2 \) are those descended from \( l_1 \) and \( l_2 \) respectively.

Thus E-resolution consists of paramodulation and resolution used in the following manner: Two clauses are selected and a literal is selected from each clause as possible literals to be resolved upon. One then searches (in a depth first manner) for all possible ways of uni-
From the collection of the Computer History Museum (www.computerhistory.org)

COMPLETENESS RESULTS FOR

E-RESOLUTION

Theorem 1. (Ground Completeness of E-resolution)
If \( S \) is an \( R \)-unsatisfiable set of ground clauses, and \( S' \) is the set of ground clauses obtained by adding to \( S \) all ground instances of the clause \( (x = x) \) and \( T \subset S \) is such that \( S' - T \) is \( R \)-satisfiable, then \( T \) can be deduced from \( S' \) by E-resolution with \( T \) as set of support.

Proof. The proof follows the general outline given in theorem 1 of Reference 1 and is by induction on \( k(S) \).

(i) Suppose \( k(S) = 0 \). Then \( S \) must consist entirely of unit clauses. Since paramodulation and resolution is known to be complete, we know that \( T \) can be deduced from \( S' \) by paramodulation and resolution. But since \( S' \) consists entirely of unit clauses there can be only one resolution involved, since any resolvent of unit clauses is \( T \). Thus this deduction of \( T \) must consist of a series of (0 or more) paramodulations followed by a single resolution. It is clear that all relevant paramodulations can be made into the two literals which are finally resolved upon. Thus \( T \) is in fact an E-resolvent of \( S' \).

(ii) The induction step \( (k(S) = N) \) follows identically the outline given in theorem 1 of Reference 1 with the word "resolution" replaced by the word "E-resolution."

Wos and Robinson extended the concept of set of support to cover paramodulation as well as resolution and showed the completeness of paramodulation and resolution with set of support. Since E-resolution is a restricted form of paramodulation and resolution, their definition of set of support can be immediately applied to E-resolution. Using this definition we obtain

Theorem 2. (Ground completeness for E-resolution with set of support) If \( S \) is an \( R \)-unsatisfiable set of ground clauses and \( S' \) is the set of ground clauses obtained by adding to \( S \) all ground instances of the clause \( (x = x) \) and \( T \subset S \) is such that \( S' - T \) is \( R \)-satisfiable, then \( T \) can be deduced from \( S' \) by E-resolution with \( T \) as set of support.

Proof. The proof is again by induction on \( k(S) \).

(i) For \( k(S) = 0 \) we know that \( S' \) consists entirely of unit clauses. Since paramodulation and resolution with set of support is complete, we know that there must exist a deduction of \( T \) from \( S' \) by paramodulation and resolution with \( T \) as set of support. As in theorem 1 the fact that \( S' \) consists entirely of unit clauses assures us that any such proof can be given the particular form defined as E-resolution and thus we obtain a deduction of \( T \) by E-resolution with \( T \) as set of support.

(ii) The proof of the induction step \( (k(S) = N) \) follows exactly the outline given in theorem 3 of Reference 1 where the word "resolution" is replaced by the word "E-resolution" and the word "satisfiable" is replaced by the word "\( R \)-satisfiable."

In order to extend the concept of hyper-resolution to hyper-E-resolution we need the following technical definitions:

Definition 4. If \( S \) is a set of clauses and \( C \in S \) and literals \( l_1, \ldots, l_n \) occur in \( C \) then define

\[ P^{(0)}(S, C, \{l_1, \ldots, l_n\}) = \{C\} \]

\[ P^{(1)}(S, C, \{l_1, \ldots, l_n\}) = \{\text{the set of all clauses which can be obtained from } C \text{ by paramodulating from some clause in } S \text{ into one of the literals } l_1, \ldots, l_n \text{ in } C \text{ and where the clauses from which paramodulation occur consist entirely of positive equality literals.} \]

and by induction,

\[ P^{(N)}(S, C, \{l_1, \ldots, l_n\}) = \{\text{the set of all clauses which can be obtained from clauses } C' \in P^{(N-1)}(S, C, \{l_1, \ldots, l_n\}) \text{ by paramodulating from some clause in } S \text{ into one of the literals } l'_1, \ldots, l'_n \text{ (in } C') \text{ descended from } l_1, \ldots, l_n \text{, and where the clauses from which paramodulation occur consist entirely of positive equality literals.} \]