

The Inference Problem for Template Dependencies*

YURI GUREVICH

University of Michigan, Ann Arbor, Michigan 48109

AND

HARRY R. LEWIS[†]

*Aiken Computation Laboratory, Harvard University, 33 Oxford Street,
Cambridge, Massachusetts 02138*

A *template dependency* is a formalized integrity constraint on a relational database, stating that whenever tuples exist in the database that agree on certain attributes, an additional tuple must also be present that agrees with the others in a specified way. It is shown that the inference problem for template dependencies is undecidable, that is, there can be no algorithm for determining whether a given dependency is a logical consequence of a given finite set of dependencies. The undecidability result holds whether or not databases are considered to be necessarily finite.

INTRODUCTION

The goal of dependency theory is to formalize constraints on the data comprising a relational database. In general, a dependency is a statement to the effect that when certain tuples are present in the database, so are certain others. Such statements can be used, for example, to capture the idea that attributes are functionally related or independent in some way. Many varieties of dependencies have been proposed in the literature; see the discussions in Fagin (1980) and Yannakakis and Papadimitriou (1980), for example. The proliferation of varieties is due in part to the desire to balance two opposing forces: on the one hand, dependencies should be of a form general enough to express interesting properties, but on the other hand, the form should not be so general that natural questions about dependencies become undecidable or computationally intractable. A significant question about any class of dependencies is its *inference problem*: Given a finite set \mathbf{D} of dependencies and a single dependency D_0 , to determine whether D_0 is true in every database in which each member of \mathbf{D} is true. A solution to the inference problem carries with it the ability to determine whether two sets of

* This work was reported in preliminary form in Gurevich and Lewis (1982).

[†] Research supported by NSF Grant MCS80-05386-A01.

dependencies are equivalent, whether a set of dependencies is redundant, etc. The inference problem has been shown to be undecidable or intractable for several classes of dependencies (Chandra *et al.*, 1981, Beeri and Vardi 1981).

Template dependencies were introduced by Sadri and Ullman (1980) and have been studied in several subsequent papers. (A precise definition is given in the next section.) An important advance in comprehending their expressive power was made in Fagin *et al.* (1981), where it was shown, among other things, that there are template dependencies true in all finite databases which fail in some infinite database. The methods of Fagin *et al.* (1981) suggested a line of attack on the inference problem, on which we here report. The main result of this paper is that the inference problem for template dependencies is undecidable, whether or not we adopt the view that a database must be a finite structure.

Our results are based on the undecidability of the word problem for a class of semigroups. In brief, the result used is as follows (it is stated precisely below). Let $S = \{A_0, A_1, \dots, A_p\}$ be an alphabet, where A_p is the symbol $\mathbf{0}$, and let E be a set of equations $\{x_1 = y_1, \dots, x_n = y_n\}$, where $x_i, y_i \in S^*$. Included in E are the equations needed to make $\mathbf{0}$ a zero of the semigroup: $\mathbf{0}A_i = A_i\mathbf{0} = \mathbf{0}$ for all i . Consider semigroups generated by A_0, \dots, A_p . Then the following two sets are effectively inseparable (and hence neither is recursive):

(1) $\{E: A_0 = \mathbf{0}$ holds in all semigroups in which each member of E holds $\}$,

(2) $\{E: \text{there is a finite semigroup } G \text{ satisfying the cancellation property (explained below) in which each member of } E \text{ holds but } A_0 = \mathbf{0} \text{ fails}\}$.

This word problem is of a somewhat technical nature but is of interest outside this application. The historical context is as follows: Post (1947) showed the word problem for arbitrary semigroups to be undecidable. Turing (1950) showed the word problem to be undecidable for semigroups with cancellation, i.e., semigroups such that, for any x, y, y' , if $xy = xy'$ or $yx = y'x$, then $y = y'$. Gurevich (1966) showed that the word problem for finite semigroups is undecidable, that is, it is undecidable whether there is a finite semigroup in which a given set E of equations holds but an additional equation $s = t$ fails. For our purposes the most convenient result would have been the effective inseparability of

(3) $\{\langle E, s = t \rangle: s = t \text{ holds in all semigroups in which each member of } E \text{ holds}\}$

and

(4) $\{\langle E, s = t \rangle: \text{there is a finite semigroup with cancellation in which each member of } E \text{ holds but } s = t \text{ fails}\}$.

Unfortunately, a finite semigroup with cancellation is a group, and the word problem for finite groups is a well-known open problem.¹ Instead we are able to make do with less: We consider semigroups with zero in which cancellation is possible *except* when the terms of the equation are zero (in which case cancellation would of course be unreasonable anyway). The result is stated more precisely below but is proved elsewhere (Gurevich and Lewis, in press).

THE PROBLEM AND THE MAIN LEMMA

A *database* is for our purposes simply a relational structure. In our treatment such a structure is assumed to consist of a single relation R with a fixed number of columns or *attributes* A, B, \dots, C . A *typing* restriction is also in force: The domains of the various attributes are disjoint. For example, suppose the relation R represents the availability of garments of various styles and sizes from various suppliers. Then R has three attributes: SUPPLIER, STYLE, and SIZE, and typical members of the R relation might be $\langle \text{St. Laurent, Evening Dress, 10} \rangle$ and $\langle \text{BVD, Brief, 36} \rangle$. The disjointness of the three attribute domains is natural since no supplier is a style, etc. (Naturally, our undecidability proofs apply a fortiori if either assumption—that of the single relation, or that of the disjoint attributes—is abandoned. See Makowsky (1983) for a discussion of the semantics of the typing restriction.)

A *template dependency* is a statement, formalized in the language of predicate logic, to the effect that if certain tuples are in R , then some other single tuple is also in R . Schematically, we may write a template dependency in the form

$$\begin{aligned}
 &R(a, b, \dots, c) \\
 &\quad \& R(a', b', \dots, c') \quad (\text{the antecedents}) \\
 &\quad \vdots \\
 &\quad \& R(a'', b'', \dots, c'') \\
 &\Rightarrow R(a^*, b^*, \dots, c^*) \quad (\text{the conclusion}).
 \end{aligned}$$

meaning that if the tuples (a, b, \dots, c) , (a', b', \dots, c') , ..., and (a'', b'', \dots, c'') are in the database, then so is (a^*, b^*, \dots, c^*) . (We rule out the use of the identity sign.) Not all of a, a', \dots, a'', a^* need be distinct, and similarly for the other

¹ Recently Slobodskoi (1981) has shown the undecidability of the word problem for finite groups. By a construction like that of this paper, this result could be used to establish the undecidability of the inference problem for template dependencies over finite databases.

attributes; otherwise the dependency would be trivial. (Because of the typing restriction, however, no a can be the same as any b . That is, since variables in different columns must range over different sets of individuals, no variable can appear in two different columns.) Nor does a^* need be among a, a', \dots, a'' ; if a^*, b^*, \dots, c^* all appear among the antecedents, then the dependency is said to be *full*, otherwise *embedded*. The symbols in the antecedents are to be interpreted as universally quantified (for all a, b , etc.) and, in the case of an embedded dependency, any symbols in the conclusion not appearing in the antecedents are interpreted as existentially quantified (there exist b^* , etc.). For example, consider the dependency

$$R(a, b, c) \ \& \ R(a, b', c') \Rightarrow R(a^*, b, c')$$

for the garment database just used as an example. This dependency asserts that if a supplier supplies both garments of some style b and garments of some size c' , then there is a supplier (not necessarily the same one) of style b garments in size c' .

The question addressed here is whether a given finite set of template dependencies has another given template dependency as a logical consequence. We can distinguish two versions of the meaning of "logical consequence." In the true database interpretation, the relation R (and therefore the domain of each attribute) is of finite but arbitrary size. In the other version, R may be infinite. Fagin *et al.* (1981) prove that the two versions of "logical consequence" differ for certain systems of template dependencies. We here prove that under either view, the inference problem is undecidable. Specifically, we prove the following:

MAIN THEOREM. *Let \mathbf{D} range over finite sets of (typed) template dependencies and let D_0 range over single (typed) template dependencies. Then the following two sets are effectively inseparable:*

$\{(\mathbf{D}, D_0): D_0 \text{ holds in every (finite or infinite) database in which each member of } \mathbf{D} \text{ holds}\},$

$\{(\mathbf{D}, D_0): D_0 \text{ fails in some finite database in which each member of } \mathbf{D} \text{ holds}\}.$

It follows immediately that neither set is recursive and that there is no recursive axiomatization of the set of all (\mathbf{D}, D_0) such that D_0 holds in all finite databases satisfying \mathbf{D} . ("Effective inseparability" is a condition stronger than recursive inseparability; see Rogers, 1967 or Gurevich and Lewis, in press, for an explanation.)

The proof of the Main Theorem has two parts. The Main Lemma gives the necessary result about effective inseparability for the word problem mentioned above. The Reduction Theorem shows that this word problem can

be reduced to the given database dependency problem, so that a solution to the latter, in either the finite or the infinite case, would imply a solution to the former.

As mentioned earlier, several undecidability proofs for database dependencies have appeared elsewhere. The result of this paper is stronger than that of Chandra *et al.* (1981), that the inference problem is undecidable for type *embedded implicational dependencies* (EIDs). An EID resembles a template dependency, but the conclusion may be a conjunction of atomic formulas rather than a single atomic formula. For example, the following is an EID:

$$R(a, b, c) \ \& \ R(a, b', c') \Rightarrow R(a^*, b, c) \ \& \ R(a^*, b, c').$$

In terms of the garment supply example, this states that if one supplier supplies a garment b in a size c and also supplies some garment in size c' , then there is a supplier of garment b in both sizes c and c' . Since EIDs are more general than template dependencies, the results of this paper imply the undecidability results of Chandra (1981), but not vice versa. See also Beeri and Vardi (1981).

A result similar to the main result of this paper has been proved by Vardi (*in press*) by quite different techniques. Vardi's proof yields dependencies with an unbounded number of antecedents but a fixed number of attributes; our proof yields dependencies with a bounded number of antecedents (five at most) but an unbounded number of attributes. Since one parameter or the other must be unbounded in any undecidable class of dependencies, our results may be regarded as complementary.

Our construction was inspired by the diagrams used in Fagin *et al.* (1981) to describe dependencies succinctly. A dependency with k antecedents and one conclusion is represented by an undirected graph with $k + 1$ nodes. The nodes represent tuples in the relation, and the labels of edges are attributes on which those tuples agree (have the same value of the attribute). Thus each type of edge label represents an equivalence relation; implied edges may be omitted in diagrams to avoid clutter. A numbered node is an antecedent, and the node labelled $*$ is the conclusion. Thus the $*$ tuple has existentially quantified components on any attributes that do not label edges (or implied edges) joining that tuple to others in the diagram. For example, the diagram in Fig. 1 represents the dependency used earlier as an example: for any a, b, c, b', c' ,

$$R(a, b, c) \ \& \ R(a, b', c') \Rightarrow (\text{for some } a^*) R(a^*, b, c').$$

Node 1 represents the tuple $\langle a, b, c \rangle$, node 2 the tuple $\langle a, b', c' \rangle$, and node $*$ the tuple $\langle a^*, b, c' \rangle$. Nodes 1 and 2 have the same A attribute, nodes 1 and $*$ the same B attribute, and nodes 2 and $*$ the same C attribute, so the three edges are labelled accordingly. No attribute values need be mentioned

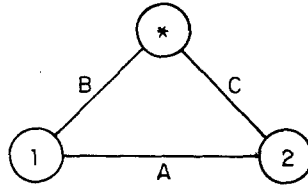


FIGURE 1

explicitly in these diagrams, since they are all quantified; only the pattern of equality among attribute values, and which is the conclusion tuple, are important.

The proof of the Main Theorem is based on a result about the word problem for semigroups. Before stating it, we need to define a few notions about semigroups. A semigroup G has zero 0 if $x0 = 0x = 0$ for each $x \in G$ (we denote the semigroup operation by juxtaposition), and has identity I if $xI = Ix = x$ for each $x \in G$. A semigroup G with zero 0 and with an identity has the *cancellation property* if it satisfies the condition,

$$(i) \quad (xy = xy' \neq 0 \text{ or } yx = y'x \neq 0) \Rightarrow y = y'.$$

Also, if G has zero 0 but has no identity, then G has the *cancellation property* if it satisfies both (i) and (ii):

$$(ii) \quad (xy = x \text{ or } yx = x) \Rightarrow x = 0.$$

(Condition (ii) seems a natural one since it describes a circumstance in which cancellation would yield the identity, if there were one. It is used in the proof to ensure that the cancellation property is preserved when an identity element is adjoined to a semigroup that does not have one already.)

MAIN LEMMA. *Let S range over alphabets containing the symbols 0 and A_0 , and let x_i and y_i range over strings in S^* . Let ϕ range over formulas of the form*

$$x_1 = y_1 \ \& \ \dots \ \& \ x_n = y_n \Rightarrow A_0 = 0$$

such that the equations $A0 = 0$ and $0A = 0$ for all $A \in S$ appear among the antecedents. Then the following two sets are effectively inseparable:

$\{\phi : \phi \text{ holds in every } S\text{-generated semigroup}\},$

$\{\phi : \phi \text{ fails in some finite } S\text{-generated semigroup without identity having the cancellation property}\}.$

This lemma is proved in Gurevich and Lewis (in press) by a combination of the methods of Turing (1950) and Gurevich (1966).

We use this lemma in a slightly stronger form below: We restrict the strings x_i and y_i appearing in the antecedents of ϕ to be of length 2 and 1, respectively. Imposing this restriction is a simple matter; if ϕ contains a conjunct $ABC = DA$, for example, we introduce new symbols E and F into S , add the equations $AB = E$ and $DA = F$, and replace the equation $ABC = DA$ by $EC = F$. Any semigroup satisfying the original formula ϕ will satisfy the new formula, with appropriate interpretations for the new symbols, and vice versa; and the cancellation property is not affected, because only the presentation of the semigroup is changed, not the semigroup itself.

PROOF OF THE MAIN THEOREM

Let us fix some particular alphabet S and some particular formula ϕ of the form specified in the Main Lemma, with $|x_i| = 2$ and $|y_i| = 1$ for each antecedent. In the ensuing construction we shall use the following equivalence relations on tuples: For each $A \in S$, the relations A' and A'' ; and additional relations E and E' . (These equivalence relations are the attributes of the dependencies, so if S contains n symbols, the relation will have $2n + 2$ attributes.) We write $a \cong_{A'} b$ to indicate that a and b are A' -equivalent, and similarly for the other equivalence relations.

The basic idea is to represent a word $A_1 A_2 \dots A_k$ over S by the structure of Fig. 2. Let us call such a structure a *bridge* for $A_1 A_2 \dots A_k$. All the elements across the bottom of a bridge are E -equivalent, all those across the top of a bridge are E' -equivalent, and each symbol A_i of the word is represented by a triangle with the apex having relations A'_i and A''_i to the two points on the base. If $x = y$ is one of the given equations and v results from w by the substitution of an occurrence of y for an occurrence of x , then the corresponding dependencies provide the means for replacing in a bridge for w a portion representing x by another portion representing y to yield a bridge for v .

For each equation $r: AB = C$ in the antecedents of ϕ , construct the four dependencies $D_i(r)$ ($i = 1, 2, 3, 4$) illustrated in Fig. 3. Let D be the set of all these dependencies. Also, let D_0 be as shown. The Main Theorem is implied

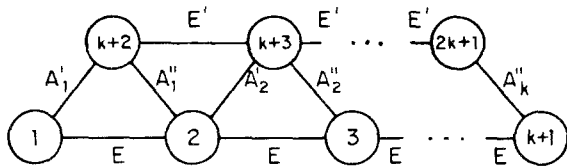


FIGURE 2

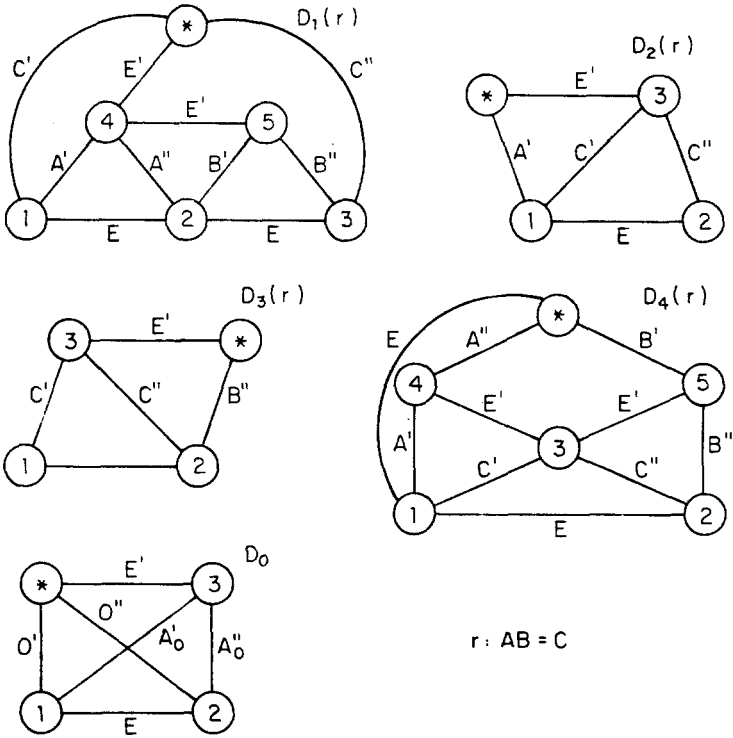


FIGURE 3

by the following theorem relating the word problem for ϕ to D_0 and the dependencies in \mathbf{D} .

REDUCTION THEOREM. (A) *If ϕ holds in every \mathbf{S} -generated semigroup, then D_0 holds in every database in which each member of \mathbf{D} holds.*

(B) *If ϕ fails in some finite \mathbf{S} -generated semigroup having the cancellation property, then there is a finite database in which each member of \mathbf{D} holds but D_0 does not.*

If the Reduction Theorem holds, then by the Main Lemma the two sets mentioned in the Main Theorem are effectively inseparable. So it remains to prove the Reduction Theorem.

Proof of (A). Suppose that ϕ holds in every \mathbf{S} -generated semigroup. Then there is a sequence of $m + 1 \geq 1$ strings $\mathbf{u}_0, \mathbf{u}_1, \dots, \mathbf{u}_m$, where \mathbf{u}_0 is \mathbf{A}_0 , \mathbf{u}_m is $\mathbf{0}$, and for $i = 0, \dots, m - 1$ \mathbf{u}_{i+1} results from \mathbf{u}_i by replacement of a single occurrence of some \mathbf{x}_i by \mathbf{y}_i or vice versa. (Otherwise, let \sim be the

equivalence relation on strings induced by such replacements; then the quotient semigroup \mathbf{S}^*/\sim would provide a counterexample to ϕ .)

Suppose that M is a finite or infinite database satisfying \mathbf{D} . We show that M satisfies D_0 . Let $a, b, d_0 \in M$ and $a \cong_E b$, $a \cong_{A_0'} d_0$, $b \cong_{A_0''} d_0$. Check by induction on $j = 0, \dots, m$ that if $\mathbf{u}_j = \mathbf{B}_0 \cdot \dots \cdot \mathbf{B}_{p-1}$ ($\mathbf{B}_1, \dots, \mathbf{B}_{p-1} \in \mathbf{S}$), then there are $c_0, \dots, c_p \in M$ and $d_1, \dots, d_p \in M$ such that $a = c_0$, $b = c_p$, and $c_k \cong_E c_{k+1}$, $d_k \cong_{E'} d_{k+1}$, $c_k \cong_{B_k'} d_{k+1}$, $d_{k+1} \cong_{B_k''} c_{k+1}$ for $k = 0, \dots, p-1$. In particular, when $j = m$ there is a d_1 such that $d_0 \cong_E d_1$, $a \cong_0 d_1$, and $d_1 \cong_0 b$, as required by D_0 .

(B) Now suppose that ϕ fails in some finite semigroup G without identity having the cancellation property. Adjoin to G an identity element I and call the resulting semigroup G' . We claim that G' also has the cancellation property. For suppose $xy = xy' \neq 0$ in G' . If $x = I$ or $y = y' = I$ or $x, y, y' \in G$, then $y = y'$. Without loss of generality we may assume that the remaining case is $x, y \in G$, $y' = I$. Then $xy = x \neq 0$ in G , which is impossible. The proof that $yx = y'x \neq 0 \Rightarrow y = y'$ is similar.

In what follows, if x is any string in \mathbf{S}^* , then x is that element of G' denoted by the term x . We construct a finite model for \mathbf{D} in which D_0 fails as follows. Let $P = \{a \in G' : \text{there is some } b \in G' \text{ such that } ab = A_0\}$. Since $I \in G'$, it follows that $I, A_0 \in P$. Also, since $A_0 \neq 0$, $0 \notin P$. For any $\mathbf{A} \in \mathbf{S}$, and $a, b \in P$, we write $a \rightarrow_A b$ if and only if $a\mathbf{A} = b$. Note that \rightarrow_A is a 1-1 partial function on P since G' has the cancellation property, and that \rightarrow_0 is empty. For every triple a, \mathbf{A}, b such that $a, b \in P$, $\mathbf{A} \in \mathbf{S}$, and $a \rightarrow_A b$, introduce a new element $\langle a, \mathbf{A}, b \rangle$, and let Q be the set of these new elements. The universe of the model for \mathbf{D} is the union of P and Q . It remains to specify the equivalence relations.

(1) For $\mathbf{A} \in \mathbf{S}$, $\cong_{A'}$ is the reflexive symmetric transitive closure of $\{\langle \langle a, \mathbf{A}, b \rangle, a \rangle : a \rightarrow_A b\}$.

FACT 1. *Each $\cong_{A'}$ equivalence class has cardinality 1 or 2. In particular, the only equivalence classes contained entirely within P or entirely within Q are trivial.*

(2) For $\mathbf{A} \in \mathbf{S}$, $\cong_{A''}$ is the reflexive symmetric transitive closure of $\{\langle \langle a, \mathbf{A}, b \rangle, b \rangle : a \rightarrow_A b\}$.

FACT 2. *Each $\cong_{A''}$ equivalence class has cardinality 1 or 2.*

(3) $a \cong_E b$ if and only if $a = b$ or both a and b are in P .

(4) $a \cong_{E'} b$ if and only if $a = b$ or both a and b are in Q .

It remains to show that this structure is a model for each dependency in \mathbf{D} but not for D_0 . In each case let us suppose that a template matches and let

us denote by t_i the element of P or Q that matches node i of the template diagram. In other words, tuples t_i and t_j agree on the components labelling the edges connecting nodes i and j of the diagram (and also agree on other components if implied by transitivity). We must show that a tuple corresponding to $*$ can be chosen in each case (D_1) – (D_4) but not in the case of (D_0) .

(D_1) If $t_1 = \dots = t_5$, then $*$ can be chosen as the same element. Suppose that t_1, t_2 , or t_3 is in Q . Then $t_1 = t_2 = t_3$, since E holds only trivially in Q . Hence $t_1 \cong_{A'} t_4$ and $t_1 \cong_{A''} t_4$. This cannot happen if $t_4 \in P$, for then $t_1 = \langle t_4, \mathbf{A}, t_4 \rangle$, which implies that $t_4 A = t_4$, and this is impossible since G has no identity and has the cancellation property. (That is, $A \neq I$, and $t_4 A = t_4$ implies $t_4 = 0$, which it is not.) So $t_4 \in Q$, and hence $t_4 = t_1$ since $\cong_{A'}$ holds only trivially in Q . By the same argument, $t_5 = t_1$ as well, and the case in which any of t_1, t_2, t_3 is in Q reduces to the case in which all five elements are identical. So assume that t_1, t_2 , and t_3 are in P ; a similar argument shows that t_4 and t_5 are in Q unless all five points are identical. Then necessarily $t_4 = \langle t_1, \mathbf{A}, t_2 \rangle$, $t_5 = \langle t_2, \mathbf{B}, t_3 \rangle$, so that $t_1 A = t_2$ and $t_1 A B = t_3$. Then $t_1 C = t_3$ and $*$ may be chosen as $\langle t_1, \mathbf{C}, t_3 \rangle$.

(D_2) If $t_1 = t_2 = t_3$, then $*$ may be chosen to be the same point. If $t_1 \in Q$, then $t_2 = t_1$ since \cong_E holds only trivially for members of Q ; also t_3 must be identical by an argument like that used above for (D_1) . So assume that t_1 and t_2 are in P ; again $t_3 \in Q$ unless all three points are identical. So $t_3 = \langle t_1, \mathbf{C}, t_2 \rangle$; and there is some t such that $t_1 C t = A_0$. Hence $t_1 A \in P$. Then let $*$ be $\langle t_1, \mathbf{A}, t_1 A \rangle$.

(D_3) Completely analogous to (D_2) .

(D_4) Again, unless the whole template collapses, t_1 and t_2 are in P and t_3, t_4 , and t_5 are in Q . Hence $t_3 = \langle t_1, \mathbf{C}, t_2 \rangle$, $t_4 = \langle t_1, \mathbf{A}, b_1 \rangle$ for some b_1 , and $t_5 = \langle b_2, \mathbf{B}, t_2 \rangle$ for some b_2 . That is, $t_1 C = t_2$, $t_1 A = b_1$, and $b_2 B = t_2$. Then $b_1 B = t_1 A B = t_1 C = t_2 = b_2 B$ and $b_1 = b_2$ by cancellation. Choose $*$ to be this element.

(NOT D_0) Let $t_1 = I$, $t_2 = A_0$, $t_3 = \langle I, \mathbf{A}_0, A_0 \rangle$; then no $*$ can exist since it is not the case that $A_0 = 0$.

This completes the proof.

REFERENCES

- BEERI, C., AND VARDI, M. (1981), The implication problem for data dependencies, in Proceedings, 8th ICALP, 1981, Lecture Notes in Computer Science, Vol. 115, pp. 73–85, Springer-Verlag, Berlin/New York.

- CHANDRA, A., LEWIS, H. R., AND MAKOWSKY, J. (1981), Embedded implicational dependencies and their inference problem, in "Proceedings, 13th ACM Symposium on Theory of Computing," pp. 342–354; revised version in *J. Comput. System Sci.*, in press.
- FAGIN, R. (1980), Horn clauses and database dependencies, in "Proceedings, 12th ACM Symposium on Theory of Computing," pp. 123–134.
- FAGIN, R., MAIER, D., ULLMAN, J., AND YANNAKAKIS, M. (1981), "Tools for Template Dependencies," IBM Technical Report, No. RJ 3033; revised version in *SIAM J. Comput.*, in press.
- GUREVICH, Y. (1966), The word problem for certain classes of semigroups, *Algebra and Logic* 5 (5), 25–35. [Russian]
- GUREVICH, Y. AND LEWIS, H. (1982), The inference problem for template dependencies, in "Proceedings of the ACM Symposium on Principles of Database Systems, Los Angeles," pp. 221–229.
- GUREVICH, Y., AND LEWIS, H. The word problem for cancellation semigroups with zero, *J. Symbolic Logic*, in press.
- MAKOWSKY, J. A. (1983), The meaning of syntactically defined dependency classes, manuscript.
- POST, E. L. (1947), Recursive unsolvability of a problem of Thue, *J. Symbolic Logic* 12, 1–11.
- ROGERS, H., JR. (1967), "Theory of Recursive Functions and Effective Computability," McGraw-Hill, New York.
- SLOBODKOI, A. M. (1982) Unsolvability of the universal theory of finite groups, *Algebra and Logic* 20 (2), 139–156; translation, January 1982.
- SADRI, F., AND ULLMAN, J. D. (1980), A complete axiomatization for a large class of dependencies in relational databases, in "Proceedings, 12th ACM Symposium on Theory of Computing," pp. 117–122; revised version, Template dependencies: A large class of dependencies in relational databases and its complete axiomatization, *J. Assoc. Comput. Mach.* 29 (2), 363–372.
- TURING, A. M. (1950), The word problem for semigroups with cancellation, *Ann. of Math.* 491–505.
- VARDI, M. The implication and finite implication problems for typed template dependencies, in "Proceedings of the ACM Symposium on Principles of Database Systems, Los Angeles," pp. 230–238; revised version in *J. Comput. Systems Sci.*, in press.
- YANNAKAKIS M., AND PAPADIMITRIOU, C. H. (1980), Algebraic dependencies, in "Proceedings, 21st Annual Symposium on Foundations of Computer Science," pp. 328–332; revised version, *J. Comput. System Sci.* 25 (1982), 2–41.