# THE DECISION PROBLEM FOR DATABASE DEPENDENCIES

M.Y. VARDI *

*Department of Applied Mathematics, The Weizmann Institute of Science, Rehovot, Israel*

Received 13 August 1980; revised version received 31 March 1981

Database, dependency, decision problem, recursive, rec' sively enumerable

## 1. Introduction

In the last decade a great deal of interest has been aroused in sentences of first order logic which are adequate as database constraints – the so called *data dependencies*. Since the introduction of *functional dependencies* by Codd [5], several additional classes of dependencies have been investigated, e.g., *multivalued dependencies* [8,17], *join dependencies* [1,13], etc. These classes are all syntactically specified. Recently, there has been an effort to characterize these sentences model-theoretically [4,9]. Chandra et al. [4] define a dependency as a sentence that holds in the trivial structure and is domain independent, i.e., the truth or falsity of such a sentence in some structure is independent of the domain, and depends only on the relations of the structure. The decision problem for dependencies is to decide whether a given sentence is a dependency. We show that this problem is recursively unsolvable. In fact, the set of dependencies is not recursively enumerable. This is closely related to the recursive unsolvability of the decision problem for *definite* formulas of Di Paola [7].

## 2. Dependencies

The language **L** that we use is the language of first order logic with equality with predicate symbols $\mathbf{R}_1$, $\mathbf{R}_2, \ldots$ – the arity of $\mathbf{R}_i$ is i, however, each sentence $s$ contains at most one predicate symbol [1]. Thus, $s(\mathbf{R}_i)$ denotes the sentence $s$ with the predicate symbols $\mathbf{R}_i$.

A structure for L, $M = \langle D, R_2, R_3, \cdots \rangle$, is *finite* if D is finite. Unless explicitly stated otherwise, the structures dealt with are assumed to be finite. When concerned with a sentence $s(\mathbf{R}_i)$ we denote a structure by $\langle D, R_i \rangle$. A structure $\langle D, R_i \rangle$ is *empty* if $R_i = \phi$, is *trivial* if $|D| = |R_i| = 1$, and is *simple* if for some $a \in D$, $R_i = \{\langle a, \ldots, a \rangle\}$. Note that a trivial structure is simple.

A sentence $s(\mathbf{R}_i)$ is *domain independent* if for every two structures $M_1 = \langle D_1, R_i \rangle$ and $M_2 = \langle D_2, R_i \rangle$ (i.e., different domains but the same relation), we have $M_1 \models s$ iff $M_2 \models s$, i.e., s holds in $M_1$ iff it holds in $M_2$. (A domain independent sentence is called *definite* in [7], *permissible* in [6], *range restricted* in [12], and *safe* in [11]). A sentence $s$ is a *dependency* if it is domain independent and holds in the trivial structure.

## 3. Some lemmas

Let $R_i$ be an i-ary relation, $i > 1$. We define the *projection* of $R_i$, denoted $p(R_i)$, as the i-1-ary relation:

$$p(R_i) = \{\langle a_1, \ldots, a_{i-1} \rangle \quad \text{for some } a_i, \langle a_1, \ldots, a_i \rangle \in R_i\}.$$

* Current address: Department of Computer Science, The Hebrew University of Jerusalem, Jerusalem, Israel.

[1] The assumption that the database consists of a single relation is called the 'universal relation assumption'. Our results clearly hold for sentences with several predicate symbols.

0020-0190/81/0000–0000/$02.50 © 1981 North-Holland

Let $s(R_i)$ be a sentence. By replacing each occurrence of an atomic formula $R_i(v_1, ..., v_i)$ in s by $\exists x R_{i+1}(v_1, ..., v_i, x)$, where $x$ is a new variable, we get a new sentence $s'(R_{i+1})$.

**Lemma 1.** (a) If $\langle D, R_{i+1}\rangle \models s'$ then $\langle D, p(R_{i+1})\rangle \models s$. (b) If $\langle D, R_i\rangle \models s$ then $\langle D, R_i \times D\rangle \models s'$ [2].

**Proof.** For a formula $f(R_i)$, let $f'(R_{i+1})$ be the result of applying the above transformation to f. For a formula $f(R_i)$ with free variables $x_1, ..., x_k$ and a structure $M = \langle D, R_i\rangle$, we define $T(f, M)$ as the set of members of $D^k$ for which $f$ holds in M, i.e.,

$$T(f, M) = \{\langle a_1, ..., a_k\rangle \mid \langle M, a_1, ..., a_k\rangle \models f\}.$$

If $k = 0$ then $T(f, M) = \mathit{true}$ if $M \models f$, and $T(f, m) = \mathit{false}$ otherwise. It can easily be shown, by a routine induction on the structure of $f$, that:

(a) $T(f', \langle D, R_{i+1}\rangle) = T(f, \langle D, p(R_{i+1})\rangle)$.
(b) $T(f, \langle D, R_i\rangle) = T(f', \langle D, R_i \times D\rangle)$.

The claim follows as a specific case.

Let $s(R_i)$ be a sentence, and let $x$ be a variable not occurring in $s$. We replace each atomic formula $R_i(v_1, ..., v_i)$ by the conjunction $v_1 = x \wedge \cdots \wedge v_i = x$, and prefix the resulting formula with $\exists x$ to get a new sentence $s^+$.

**Lemma 2.** (a) Let $M = \langle D, R_i\rangle$ be a simple structure. $M \models s$ implies $\langle D\rangle \models s^+$. (b) If $\langle D\rangle \models s^+$ and $a \in D$ then $\langle D, \{\langle a, ..., a\rangle\}\rangle \models s$.

**Proof.** Analogous to the proof of Lemma 1.

Let $s(R_i)$ be a sentence, and let $x$ be a variable not occurring in $s$. We replace each atomic formula $R_i(v_1, ..., v_i)$ by $x \neq x$ and prefix the resulting formula with $\exists x$ to get a new sentence $s^*$.

**Lemma 3.** $\langle D, \phi\rangle \models s$ if and only if $\langle D\rangle \models s^*$.

**Proof.** Analogous to the proof of Lemma 1.

**Lemma 4.** The set of sentences having a non-empty model is not recursive.

**Proof.** By Lemma 3, $s$ has an empty model iff $s^*$ is satisfiable. Since $s^*$ is an equality sentence we can effectively test whether it is satisfiable [13]. Thus, the set of sentences having an empty model is recursive. If the set of sentences having a non-empty model were recursive, then the set of satisfiable sentences would also be recursive. But the last set is not recursive [15] – a contradiction. The claim follows.

## 4. The main result

**Theorem 1.** The set of domain independent sentences is not recursive.

**Proof.** We show that the decision problem for sentences having a non-empty model is reducible to the decision problem for domain independent sentences. Let $s(R_i)$ be a sentence. We construct another sentence $t(R_{i+1})$: $s' \wedge \exists x_1 \cdots \exists x_i \forall x_{i+1} R_{i+1}(x_1, ..., x_{i+1})$. $s$ has a non-empty model iff $t$ is not domain independent: suppose first that $s$ has no non-empty model, then $t$ has no model, hence, it is domain independent. Suppose now that $s$ has a non-empty model $M = \langle D, R_i\rangle$, then by Lemma 1, $s'$ has a non-empty model $M_1 = \langle D, R_{i+1}\rangle$, where $R_{i+1} = R_i \times D$. $M_1$ is also a model of $t$. Let $a$ be any element not in D, and let $M_2 = \langle D \cup \{a\}, R_{i+1}\rangle$. Since $M_2 \not\models t$, t is not domain independent.

A sentence $s(R_i)$ is *weakly domain independent* if for every two non-simple structures $M_1 = \langle D_1, R_i\rangle$ and $M_2 = \langle D_2, R_i\rangle$, we have $M_1 \models s$ iff $M_2 \models s$.

**Lemma 5.** The set of weakly domain independent sentences is not recursive.

**Proof.** We show that the decision problem for domain independent sentences is reducible to the decision problem for weakly domain independent sentences. Let $s(R_i)$ be a sentence. If there are two simple structures $M_1 = \langle D_1, R_i\rangle$ and $M_2 = \langle D_2, R_i\rangle$, such that $M_1 \models s$ but $M_2 \not\models s$, then $s$ is not domain independent. If, on the other hand, we have that either for all simple structures M, $M \models s$, or for all simple structures M, $M \not\models s$, then $s$ is domain independent iff $s$ is weakly domain independent. Suffice it to show that we can effectively decide whether there are two simple struc-

[2] $\times$ denotes the Cartesian product.

tures $M_1 = \langle D_1, R_i \rangle$ and $M_2 = \langle D_2, R_i \rangle$, such that $M_1 \models s$ but $M_2 \not\models s$. By Lemma 2 and the known results concerning equality formulas [2,13], this is the case iff there are two domains D and D', such that $\langle D \rangle \models s^+$ and $\langle D' \rangle \not\models s^+$ where $|D| = p$, $|D'| = q$, $p, q \leq k + 1$, and k is the number of quantifiers in $s$. The claim follows.

**Theorem 2.** The set of dependencies is not recursive.

**Proof.** We show that the decision problem for weakly domain independent sentences is reducible to the decision problem for dependencies. Let $s(\mathbf{R}_i)$ be a sentence. We construct another sentence $t$: $s \wedge \exists x(R_i(x, ..., x) \wedge \forall y_1 \cdots \forall y_i(R_i(y_1, ..., y_i) \to x = y_1 = \cdots = y_i))$. Obviously, $t$ holds in all simple structures and especially in the trivial structure. Also, $s$ is weakly domain independent iff $t$ is domain independent. Thus, $s$ is weakly domain independent iff $t$ is a dependency.

**Corollary 1.** The set of dependencies is not recursively enumerable.

**Proof.** Obviously, the set of sentences which does not hold in the trivial structure is recursive. Also, the set of sentences which are not domain independent is recursively enumerable. Hence, the set of all sentences which are not dependencies is recursively enumerable. Since the set of dependencies is not recursive, it follows that it is not recursively enumerable.

**Corollary 2.** The set of dependencies $s(\mathbf{R}_3)$ is not recursively enumerable.

**Proof.** Since the set of satisfiable sentences $s(\mathbf{R}_2)$ is not recursive [15], the result of Lemma 4 holds for the sentences $s(\mathbf{R}_2)$, and the results of Theorem 1, Lemma 5 and Theorem 2 hold for the sentences $s(\mathbf{R}_3)$.

Let us now admit also infinite structures. A sentence $s(\mathbf{R}_i)$ is *strongly domain independent* if for every two structures (finite or infinite) $M_1 = \langle D_1, R_i \rangle$, and $M_2 = \langle D_2, R_i \rangle$, we have $M_1 \models s$ iff $M_2 \models s$. A sentence $s$ is a *strong dependency* if it is strongly domain independent and holds in the trivial structure.

**Theorem 3.** The set of strong dependencies $s(\mathbf{R}_3)$ is not recursive.

**Proof.** Since the set of satisfiable sentences $s(\mathbf{R}_2)$ is not recursive [3], we can prove the claim by repeating the chain of arguments leading to Theorem 2.

**Theorem 4.** The set of strong dependencies is recursively enumerable.

**Proof.** Suffice it to show that the set of strongly domain independent sentences $s(\mathbf{R}_i)$ is recursively enumerable, for all $i \geq 1$. We show that the decision problem for strongly domain independent sentences $s(\mathbf{R}_i)$ is reducible to the validity problem for first order logic with equality which is recursively enumerable [10]. Let $f$ be a formula, and let P be a unary predicate symbol. If we replace every subformula $\forall x(g)$ or $\exists x(g)$ in $f$, by the formula $\forall x(P(x) \to g)$ or $\exists x(P(x) \wedge g)$ respectively, then the resulting formula $f(P)$ is said to be obtained by *relativizing* $f$ to P [14]. Let $s(\mathbf{R}_i)$ be a sentence, let P and Q be two unary predicate symbols, and let $s(\mathbf{R}_i, P)$ and $s(\mathbf{R}_i, Q)$ be the result of relativizing $s$ to P and Q respectively. We construct three sentences:

$$t_1\colon \forall x_1 \cdots \forall x_i(R_i(x_1, ..., x_i) \to P(x_1) \wedge \cdots \wedge P(x_i)),$$

$$t_2\colon \forall x_1 \cdots \forall x_i(R_i(x_1, ..., x_i) \to Q(x_1) \wedge \cdots \wedge Q(x_i)),$$

$$t\colon t_1 \wedge t_2 \to (s(\mathbf{R}_i, P) \leftrightarrow s(\mathbf{R}_i, Q)).$$

The reader can verify that $s$ is strongly domain independent iff $t$ is valid.

*Remark.* The solvability of the decision problem for the monadic predicate calculus [2,11] entails the solvability of the decision problem for (strong) dependencies $s(R_1)$.

**Acknowledgement**

I wish to thank Ron Fagin for his helpful comments.

**References**

[1] A.V. Aho, C. Beeri and J.D. Ullman, The theory of joins in relational databases, ACM Trans. Database Systems 4 (3) (1979) 297–314.

[2] W. Ackermann, Solvable cases of the decision problem (North-Holland, Amsterdam, 1954).

[3] A. Church, A note on Eintscheidungsproblem, J. Symbolic Logic 1 (1936) 40–41.

[4] A.K. Chandra, H.R. Lewis and J.A. Makowsky, Embedded implicational dependencies and their inference problem, Proc. XP1 Workshop on Relational Database Theory, Stony Brook (1980).

[5] E.F. Codd, Further normalization of the data base relational model, in: R. Rustin, Ed., Data Base Systems (Prentice-Hall, Englewood Cliffs, NJ, 1972) 33–64.

[6] E.C. Cooper, On the expressive power of query languages for relational databases, Techn. Rep. TR-14-80, Center for Research in Computing Technology, Harvard University (1980).

[7] R. Di Paola, The recursive unsolvability of the decision problem for the class of definite formulas, J. ACM 16 (2) (1969) 324–327.

[8] R. Fagin, Multivalued dependencies and a new normal form for relational databases, ACM Trans. Database Systems 2 (3) (1977) 262–278.

[9] R. Fagin, Horn clauses and database dependencies, Proc. 12th Annual ACM Symp. Theory Comput. (1980) 123–134.

[10] K. Godel, Die Vollstandigkeit der Axiome des Logischen Funktionenkalkuls, Monatshefte Math. Phys. 37 (1930) 349–360.

[11] L. Lowenheim, Uber Moglichkeiten im Relativekalkul, Math. Ann. 76 (1915) 447–470.

[12] J.M. Nicolas, A property of logical formulas corresponding to integrity constraints on database relations, Proc. Workshop on Formal Bases for Data Bases, Toulouse (1979).

[13] J. Rissanen, Theory of relations for databases – a tutorial survey, Proc. 7th Symp. on Mathematical Foundations of Computer Science, Poland, 1978, Lecture Notes in Computer Science 64 (Springer-Verlag) 537–551.

[14] A. Tarski, A. Mostowski and R.M. Robinson, Undecidable theories (North-Holland, Amsterdam, 1953).

[15] B.A. Trachtenbrot, Impossibility of an algorithm for the decision problem in finite classes, Dokl. Akad. Nauk SSSR 70 (1950) 569–572; translated in: Amer. Math. Soc. Trans. Series 2, 23 (1963) 1–5.

[16] C. Zaniolo, Analysis and design of relational schemata for database systems, Techn. Rep. UCLA-ENG-7769, Department of Computer Science, UCLA (1976).