May 1982 Report No. STAN-CS-82-912

The Implication and Finite implication Problems for Typed Template Dependencies

by

Moshe Y. Varde

Department of Computer Science

Stanford University
Stanford, CA 94305

THE IMPLICATION AND FINITE IMPLICATION PROBLEMS FOR TYPED TEMPLATE DEPENDENCIES

Moshe Y. Vardi†

Department of Computer Science

Stanford University

Stanford, California 94305

May 1982

ABSTRACT

The class of typed template dependencies is a class of data dependencies that includes embedded multivalued and join dependencies. We show that the implication and the finite implication problems for this class are unsolvable. An immediate corollary is that this class has no formal system for finite implication. We also show how to construct a finite set of typed template dependencies whose implication and finite implication problems are unsolvable.

The class of projected join dependencies is a proper subclass of the above class, and it generalizes slightly embedded join dependencies. It is shown that the implication and the finite implication problems for this class are also unsolvable. An immediate corollary is that this class has no universe-bounded formal system for either implication or finite implication.

† Research supported by a Weizmann Post-doctoral Fellowship, Fullbright Award, and NSF grant MCS-80-12907.

1. Introduction

In the relational model one views the database as a collection of relations, each of which being a set of tuples over some domain of values [Codd1]. One notable feature of this model is its being almost devoid of semantics. A tuple in a relation represents a relationship between certain values, but from the mere syntactic definition of the relation one knows nothing about the nature of this relationship, not even if it is a one-to-one or one-to-many relationship.

Two approaches have been taken to remedy this deficiency. The first approach is to extend the relational model to capture more semantics [Codd3]. The second approach, which is the basis for this paper, is to devise means to specify the missing semantics. These semantic specifications are often called semantic or integrity constraints, since they specify which databases are meaningful for the application and which are meaningless. Thus, the database schema is conceived as a syntactic specification accompanied by a semantic specification.

Several approaches have been taken with regard to integrity constraints. Of particular interest are the constraints called *data dependencies*, or dependencies for short. Essentially, dependencies are sentences in first-order logic stating that if some tuples, fulfilling certain equalities, exist in the database then either some other hiples must also exist in the database or some values in the given tuples must be equal. The study of dependencies began with the *functional dependencies* of [Codd2]. After the introduction of *multivalued dependencies* by [Fag1, Zan] the field became chaotic for a few years in which researchers introduced many new classes of dependencies. Recently, two unifying formalisms have been suggested and turned out to be equivalent. The *class of tuple and equality generating dependencies* [BV2, Fag2†], which is equivalent to the class of *algebraic dependencies* [YP], seems to contain most cases of interest.

Most of the papers in dependency theory deal exclusively with various aspects of the *implication problem*, i.e., the problem of deciding for a given set of dependencies Σ and a

† These dependencies are called *embedded implicational dependencies* in [Fag2].

dependency σ whether Σ logically implies σ. The reason for the prominence of this problem is that an algorithm for deciding implication of dependencies enables us to decide whether two given sets of dependencies are equivalent or whether a given set of dependencies is redundant. A solution for the last two problems seems a significant step towards automated database schema design [Bern, BMSU,BR], which some researchers see as the ultimate goal for research in dependency theory [BBG]. Real life databases are inherently finite. When we restrict our attention to finite databases *we* face the *jinite implication problem,* which is independent of the implication problem.

The class of tuple and equality generating dependencies is quite expressive, in fact, expressive enough to render the implication and the finite implication problems for this class unsolvable [BV2, CLM2, Va1]. A proper subclass is the class of *template dependencies* [SU], which is general enough to contain *embedded multivalued dependencies* [Fag1], *embedded join dependencies* [MMS], and *projected join dependencies* [YP].

Usually, we require that no value appears in two different columns of a relation. Such relations are called *typed* relations, and dependencies dealing with such relations are called *typed* dependencies. If we give up this restriction then we get *untyped* relations and dependencies. Untyped template dependencies are much more expressive then typed template dependencies, and their implication and finite implication problems are unsolvable [BV1, CLM1]. However, the status of the implication and finite implication problem for typed template dependencies was left open by the above mentioned papers.

A possible way to prove solvability is to show **that** implication is equivalent to finite implication. The refutation of this possibility for typed template dependencies in [FMUY] indicated that the problems are more likely to be unsolvable.

Our ultimate result in the paper is **that** the implication and the finite implication problems for projected join dependencies are unsolvable. The proof goes in two essential steps. First, we reduce the problems for untyped template dependencies to the corresponding problems for typed template dependencies, and then we reduce them further to the corresponding problems for projected join dependencies.

The outline of the paper is as follows. In Section 2 we give the basic definitions. In Section 3 we show how to translate untyped tuples and relations to typed ones. This translation is used in Section 4 to reduce the problems for untyped td's to the corresponding problems for typed td's in a very elegant way. Since we view a template dependency as a pair consisting of a tuple and a relation, we use the translation to translate untyped dependencies to typed ones, and we also use it to translate untyped counterexample relations to typed ones. In Section 5 we show some consequences of the results in Section 4. Mainly, we show that there is a finite set of typed template dependencies whose implication and finite implication problems in the class of typed template dependencies are unsolvable. Finally, in Section 6 we use the reduction technique of [YP] to reduce the problems for typed template dependencies to the corresponding problems for projected join dependencies. We end that section with a discussion of formal systems for projected join dependencies. We distinguish between systems that are universe-bounded and those that are not, and show that the class of projected join dependencies can not have a sound and complete formal system of the first kind, but it does have such a system of the second kind. We conclude in Section 7 with some remark on the implication problem for embedded multivalued dependencies.

A preliminary version of this paper appeared in [Va3]. Unsolvability of the implication and the finite implication problem for projected join dependencies was shown independently by Gurevich and Lewis [GL1]. However, our results for template dependencies are stronger, since we show a specific set of dependencies for which the problems are unsolvable.

2. Basic Definitions

2.1. Attributes, Tuples and Relations

Attributes are symbols taken from a given finite set called the *universe.* All sets of attributes are subset of the universe. We use the letters $A, B, C, \ldots$ to denote attributes and $X, Y, \cdots$ to denote sets of attributes. We do not distinguish between the attribute A and the set $\{A\}$. The union of X and Y is denoted by XY, and the complement of X is in the universe is denoted by $\overline{X}$.

Let U be a universe. With each attribute A is associated an infinite set called its *domain,* denoted $DOM(A)$. The domain of a set of attributes X is $DOM(X)=\bigcup_{A \in X} DOM(A)$. An *X-value* is a mapping $w: X \rightarrow DOM(X)$, such that $w(A) \in DOM(A)$ for all $A \in X$. An *X-relation* is a nonempty set (not necessarily finite) of X-values. If $X = U$ then we may omit it for simplicity. A *tuple* is a U-value. We use $a, b, c, \ldots$ to denote elements of the domains, $s, t, u, \ldots$ to denote tuples, and $I, J, \ldots$ to denote relations.

For a tuple w and a set $Y \subseteq U$ we denote the restriction of w to Y by $w[Y]$. We do not distinguish between $w[A]$, which is an A-value, and w(A), which is an element of $DOM(A)$. Let I be an X-relation, and let $Y \subseteq X$. Then the *projection* of I on Y, denoted I[Y], is a Y-relation $I[Y]=\{w[Y] : w \in I\}$. The set of all attribute values in an X-relation I is $VAL(I)=\bigcup_{A \in X} I[A]$. For an X-value $w, VAL(w)$ stands for $VAL(\{w\})$.

2.2. Mappings and Valuations

We often use mappings whose domain is a subset of $DOM(U)$. Let w be an X-value, and let α be a mapping defined on VAL (w). Then we define $\alpha(w)$ as $\alpha \circ w$ (i.e., α composed with w). Thus, $\alpha(w)$ is a mapping from the domain of w to the range of α. A *valuation* is a partial mapping $\alpha: DOM(U) \rightarrow DOM(U)$ such that if $\alpha(a)$ is defined then $\alpha(a) \in DOM(A)$ for all $A \in U$ and $a \in DOM(A)$. We say that α is a valuation on a tuple w (a relation I) if a is defined exactly on $VAL(w)$ $(VAL(I))$. Let α be a valuation on a relation I, and let w be a tuple. An *extension* of α to w is a valuation **on** $I \cup \{w\}$ **that** agrees with α on $VAL(I)$.

2.3. Dependencies and Implication

For any given.application only a subset of all possible relations is of interest. This subset is defined by constraints that are to be satisfied by the relations of interest. A class of constraints that was intensively studied is the class of the so called *data dependencies.*

A *template dependency* (abbr. td) [SU] says that if some tuples, fulfilling certain equalities, exist in the relation, then necessarily another tuple (possibly with some components unspecified) exists in the relation. Formally, a td is a pair $\langle w,I\rangle$ of a tuple w and a finite relation 1. It is satisfied by a relation J, denoted $J \models \langle w,I\rangle$, if every valuation α on I such that $\alpha(I) \subseteq J$ can be extended to w so that $\alpha(w) \in J$. Let V be the maximal set such that $VAL(w[V]) \subseteq VAL(I)$. $\langle w,I\rangle$ is called *V-total.*

A *functional dependency* (abbr. fd) [Codd2] says that if two tuples agree on some of their attributes, then necessarily they agree also on other attributes. Formally, an fd is a statement $X \rightarrow$ Y for some sets of attributes X and Y. It is satisfied by a relation J, denoted $J \models X \rightarrow Y$, if for any two tuples $u, v \in J$, if $u[X] = v[X]$ then $u[Y] = v[Y]$.

From now on let Σ denote a finite set of dependencies and let σ and θ denote individual dependencies. When we want to specify explicitly the universe U we'll talk about U-dependencies. We say that Σ *implies* σ, denoted $\Sigma \models \sigma$, if $I \models \Sigma$ entails $I \models \sigma$ for every relation I. Σ *finitely implies* σ, denoted $\Sigma \models_f \sigma$, if $I \models \Sigma$ entails $I \models \sigma$ for every finite relation I.

Let Ψ be a class of dependencies. The *implication problem* for Ψ is to decide, given $\Sigma \subseteq \Psi$ and $\sigma \in \Psi$, whether $\Sigma \models \sigma$. The *finite implication problem* for Ψ is to decide, given $\Sigma \subseteq \Psi$ and $\sigma \in \Psi$, whether $\Sigma \models_f \sigma$. The two problems are independent each of the other, because one can have $\Sigma \models_f \sigma$ but $\Sigma \not\models \sigma$. In fact, if $\Sigma \models_f \sigma$ entails $\Sigma \models \sigma$ then not only are the two problems equivalent but they are also solvable.

2.4. Untyped and Typed Dependencies

Until now we have not said anything about the relationship between domains of different attributes. We now present the two extremes. If we assume that all attributes have the same domain, i.e., if the universe is $U = A_1 \cdots A_n$ and

$$DOM(U) = DOM(A_1) = \cdots = DOM(A_n),$$

then the universe, tuples, relations and dependencies are called *untyped*. If, on the other hand, we assume that different attributes have disjoint domains, i.e., $A \neq B$ entails $DOM(A) \cap DOM(B) = \emptyset$, then the universe, tuples, relations and dependencies are called *typed*.

Let us now fix a universe $U' = A'B'C'$ for the untyped case, and let

$$DOM' = DOM(U') = DOM(A') = DOM(B') = DOM(C').$$

We denote an untyped tuple w by $\langle w[A'], w[B'], w[C'] \rangle$. Beeri and Vardi [BV1] have shown that the implication and the finite implication problems for untyped td's are unsolvable. In fact their result is even stronger.

Theorem 1. [BV1] The implication and the finite implication problems for untyped td's are unsolvable even for those Σ and σ that satisfy the following conditions:

(1) σ is U'-total.

(2) All td's in Σ are $A'B'$-total.

(3) If $\Sigma \not\models_{(f)} \sigma$ then $\Sigma \cup \{A'B' \rightarrow C'\} \not\models_{(f)} \sigma$.

Furthermore, there is even a fixed σ that satisfies the above conditions, for which the problems are still unsolvable. ❑

3. Translating Untyped Tuples and Relations to Typed Ones

We use a typed universe $U = ABCDEF$. To every element $a \in DOM'$ there correspond three distinct elements $a^1 \in DOM(A)$, $a^2 \in DOM(B)$ and $a^3 \in DOM(C)$. $DOM(A)$, $DOM(B)$

and DOM (C) have also special elements a 0, $b0$ and c 0, correspondingly. Thus $DOM(A)=\{a0\}\cup\{a^1 : a \in DOM'\}$, $DOM(B)=\{b0\}\cup\{b^2 : b\in DOM'\}$, and $DOM(C)=\{c0\}\cup\{c^3 : c\in DOM'\}$. The other domains are: $DOM(D)=\{d0\}\cup\{w$: w is an untyped tuple), $DOM(E)=\{e0\}\cup DOM'$ and $DOM(F)=\{f0,f1, \ldots \}$.

We denote a typed tuple w by $\langle w[A], \ldots, w[F]\rangle$.

We use mappings between DOM' and $DOM = DOM(A)\cup\cdots\cup DOM(F)$. Three such mappings are the one-to-one mappings ^{1}, ^{2} and ^{3} defined earlier. The inverse mapping is $\varphi : \varphi(a^1)=\varphi(a^2)=\varphi(a^3)=a$.

The basic idea is to represent an untyped tuple w = $\langle a ,b,c \rangle$ by a typed tuple $T(w)=\langle a^1,b^2,c^3,w,e0,f1\rangle$. Note that $\varphi(T(w)[ABC])=w$. To represent an untyped relation by a typed one we have to convey the message that a^1, a^2, and a^3 are just three names for the same element. For this wc use the typed tuple $N(a)=\langle a^1,a^2,a^3,d0,a,f1\rangle$. We also use a typed tuple $s=\langle a0,b0,c0,d0,e0,f0\rangle$. *Now* we represent an untyped relation I by replacing every tuple $w\in I$ by $T(w)$, by adding $N(a)$ for every $a \in VAL(I)$ and by adding s, that is,

$$T(I)=\left[\bigcup_{w\in I} T(w)\right]\cup\left[\bigcup_{a\in VAL(I)} N(a)\right]\cup\{s\}$$

Example 1.

Let I be the untyped relation:

	A'	B'	C'
w_1:	a	b	c
w_2:	b	a	c

$T(I)$ is the typed relation:

	A	B	C	D	E	F
s:	$a0$	$b0$	$c0$	$d0$	$e0$	$f0$
$T(w_1)$:	a^1	b^2	c^3	w_1	$e0$	$f1$
$T(w_2)$:	b^1	a^2	c^3	w_2	$e0$	$f1$
$N(a)$:	a^1	a^2	a^3	$d0$	a	$f1$
$N(b)$:	b^1	b^2	b^3	$d0$	b	$f1$
$N(c)$:	c^1	c^2	c^3	$d0$	c	$f1$

We now make a few observations on T. First, T is a monotone operator on relations, i.e., $I \subseteq J$ entails $T(I) \subseteq T(J)$. Secondly, T preserve finiteness, i.e., if I is finite then $T(I)$ is also finite. Furthermore, if we restrict our attention to finite relations, then T can be viewed as an effective translation. Finally, $T(Z)$ has a very specific structure. In particular, it satisfies certain functional dependencies.

Lemma 1. Let I be an untyped relation. Then

$$T(I) \models \{AD \rightarrow U, BD \rightarrow U, CD \rightarrow U, ABCE \rightarrow U\}.$$

Proof. Let us show that $T(I) \models AD \rightarrow U$ (the proof for $BD \rightarrow U$ and $CD \rightarrow U$ is analogous.) Let $u,v \in T(I)$ and $u[AD]=v[AD]$. If $u \neq v$ then $u[D]=v[D]=d0$. If $u=s$ then $v=s$ because $a0 \neq a^1$ for all $a \in DOM'$, and if u = N(a) for some $a \in VAL$ (I) then v = $N(a)$ because ^{1} is one-to-one. So $u[AD]=v[AD]$ implies u = v.

Let *us* now *show* that $T(I) \models ABCE \rightarrow U$. Let $u,v \in T(I)$ and $u[ABCE]=v[ABCE]$. If $u \neq v$ then $u[E]=v[E]=e0$. If $u=s$ then v $=s$ and vice versa, because $a0 \neq a^1$ for all

$a \in DOM'$. It follows that u $= T(p)$ and v $= T(q)$ for some $p,q \in I$. But $u[ABC] = v[ABC]$ entails $p = q$, because ^{1}, ^{2}, and ^{3} are one-to-one. Necessarily, u = v. □

4. The Reduction

Our goal is to reduce the (finite) implication problem for untyped td's to the (finite) implication problem for typed td's via a many-to-one reduction. So far we have shown how to translate untyped tuples and relations to typed ones. To translate an untyped td $\sigma = \langle w, J \rangle$ to a typed td, we translate both the antecedent I and the consequent w, i.e., $T(\sigma) = \langle T(w), T(I) \rangle$.

Example 2.

Let σ be the untyped td $\langle w, I \rangle$, Z=(u):

	A	B	C
w:	b	a	d
u:	a	b	c

$T(a)$ is the typed td $\langle T(w), T(I) \rangle$:

	A	B	C	D	E	F
T(w):	b^1	a^2	d^3	w	e0	f1
T(I):	a0	b0	c0	d0	e0	f0
	a^1	b^2	c^3	u	e0	f1
	a^1	a^2	a^3	d0	a	f1
	b^1	b^2	b^3	d0	b	f1
	c^1	c^2	c^3	d0	c	f1

We'll also define later the translation function T on sets of untyped td's so that given untyped Σ and σ, $\Sigma \models \sigma$ iff $T(\Sigma) \models T(\sigma)$ and $\Sigma \models_f \sigma$ iff $T(\Sigma) \models_f T(\sigma)$. Thus, given an untyped relation I such that $I \models \Sigma$ but $I \not\models \sigma$ [†], we'll show that $T(I) \models T(C)$ but $T(I) \not\models T(\sigma)$. We'll also define T^{-1} the "inverse" of T that translates typed relations into untyped ones, so that given a typed relation I such that $I \models T(C)$ but $I \not\models T(\sigma)$, we'll show that $T^{-1}(I) \models \Sigma$ but

---

† Such a relation is called a *counterexample* relation for the implication $\Sigma \models_{(f)} \sigma$.

$T^{-1}(I) \not\models \sigma$. Both T and T^{-1} preserve finiteness, which makes the reduction *conservative.* That means that both the finite implication problem and the implication problem arc reduced simultaneously.

Our first candidate for $T(C)$ is $\{T(\theta) : \theta \in \Sigma\}$. Indeed, as the next lemma shows, that works fine in one direction, from $I \models \Sigma$ and $I \not\models \sigma$ to $T(I) \models T(\Sigma)$ and $T(I) \not\models T(\sigma)$. Because of Theorem 1, we don't have to deal with arbitrary untyped td's but only with $A'B'$-total untyped td's, i.e., untyped td's $\langle w,I\rangle$ where $VAL(w[A'B'])\subseteq VAL(I)$.

Lemma 2. Let I be an untyped relation and let θ be an $A'B'$-total untyped td. Then $I \models \theta$ if and only if $T(I) \models T(\theta)$.

Proof. Let θ be $\langle w,J\rangle$, $w = \langle a,b,c\rangle$.

If: Suppose that $T(I) \models T(\theta)$. Let α be a valuation on J such that $\alpha(J)\subseteq I$. Define a valuation β on $T(J)$ as follows: β is the identity on $\{a0,b0,c0,d0,e0,f0,f1\}$, $\beta(d^i)=\alpha(d)^i$ and $\beta(d)=\alpha(d)$ for all $d\in VAL(I)$, and $\beta(t)=\alpha(t)$ for all $t\in J$. Let $t=\langle d,e,f\rangle\in J$. Then $T(t)=\langle d^1,e^2,f^3,t,e0,f1\rangle$ and

$$\beta(T(t))=\langle\alpha(d)^1,\alpha(e)^2,\alpha(f)^3,\alpha(t),e0,f1\rangle=T(\alpha(t)).$$

Let $d \in VAL(J)$. Then $N(d)=\langle d^1,d^2,d^3,d0,d,f1\rangle$ and

$$\beta(N(d))=\langle\alpha(d)^1,\alpha(d)^2,\alpha(d)^3,d0,\alpha(d),f1\rangle=N(\alpha(d)).$$

Also, $\beta(s)=s$, so we get $\beta(T(J))=T(\alpha(J))\subseteq T(I)$. By assumption, β can be extended to $T(w)$ so that $\beta(T(w))\in T(I)$. But $\beta(a^1)=\alpha(a)^1\neq a0$, so $\beta(T(w))\neq s$. That is, there is a tuple $u\in I$ such that $\beta(T(w))=T(u)$, because $\beta(e0)=e0$. If $c \in VAL(J)$ then

$$\alpha(w)=\langle\alpha(a),\alpha(b),\alpha(c)\rangle=\varphi(\langle\beta(a^1),\beta(b^2),\beta(c^3)\rangle)=$$
$$=\varphi(\beta(T(w))[ABC])=\varphi(T(u)[ABC])=u\in I.$$

Otherwise, we define $\alpha(c)=\varphi(\beta(c^3))$ and get $\alpha(w)=u$.

Only if: Suppose that $I \models \theta$. Let α be a valuation on $T(J)$ such that $\alpha(T(J))\subseteq T(I)$. If $|\alpha(T(J))| = 1$ then $\alpha(T(J))=\{u\}$ for some $u \in T(I)$. It is easy to see that α can be extended to $T(w)$ so that $\alpha(T(w))=u \in T(I)$, so we can assume that $|\alpha(T(J))|>1$. What we'll now show is that a maps $T(J)$ to $T(I)$ in a very specific way.

Claim 1. $\alpha(T(J)-\{s\}) \subseteq T(I)-\{s\}$.

Assume to the contrary that there is a tuple $u \in T(J)-\{s\}$ such that $\alpha(u)=s$. Then $\alpha(f1)=f0$. But $f0$ has a unique occurrence in $T(Z)$, so it follows that $\alpha(T(J)-\{s\})=\{s\}$. Thus, $\alpha(d0)=d0$ and $\alpha(e0)=e0$. But for every $u \in T(I)-\{s\}$ either $u[D]\neq d0$ or $u[E]\neq e0$, so necessarily $a(s)$ = s and $|(T(J))| = 1$ - contradiction.

Claim 2. $\alpha(s)=s$.

Assume to the contrary that there is a tuple $u \in I$ such that $\alpha(s)=T(u)$. Then $\alpha(d0)=u$. But u has a unique occurrence in T(I), so it follows that for all $d \in VAL(J), \alpha(N(d))=T(u)$. Let $v=\langle e,f,g\rangle \in J$. Then $\alpha(N(e))=\alpha(N(f))=\alpha(N(g))=T(u)$. I.e., $\alpha(e^1)=T(u)[A]$, $\alpha(f^2)=T(u)[B]$ and $\alpha(g^3)=T(u)[C]$. Also, $\alpha(e0)=e0=T(u)[E]$, and consequently, $\alpha(T(v)[ABCE])=T(u)[ABCE]$. By Lemma 1, $T(I)\models ABCE\rightarrow U$, so $\alpha(T(v))=T(u)$. It follows that $|\alpha(T(J))| = 1$ - contradiction.

If $\alpha(s)\neq$ s, then the only other possibility is that there is a value $d \in VAL(I)$ such that a(s) = N(d). Then $\alpha(e0)=d$. But d has a unique occurrence in $T(I)$, so it follows that for all $u \in J, \alpha(T(u))=N(d)$. If $e \in VAL(J)$, then there is a tuple $v \in J$ such that either $v[A']=e$, or $v[B']=e$, or $v[C']=e$; so either $T(v)[A]=e^1$, or $T(v)[B]=e^2$, or $T(v)[C]=e^3$. But $\alpha(T(v))=N(d)$, so either $\alpha(e^1)=N(d)[A]$, or $\alpha(e^2)=N(d)[B]$, or $\alpha(e^3)=N(d)[C]$. Also, $\alpha(d0)=d0=N(d)[D]$, so either $\alpha(N(e))[AD]=N(d)[AD]$, or $\alpha(N(e))[BD]=N(d)[BD]$, or $\alpha(N(e))[CD]=N(d)[CD]$. By Lemma 1, $T(I)\models\{AD\rightarrow U, BD\rightarrow U, CD\rightarrow U\}$, so in either case $\alpha(N(e))=N(d)$. It follows that $|\alpha(T(J))| = 1$ - contradiction.

Claim 3. For every tuple $u \in J$ there is a tuple $v \in I$ such that $\alpha(T(u))=T(v)$.

Assume to the contrary that $\alpha(T(u))=N(d)$ for some $d \in VAL(I)$. Then, $\alpha(e0)=d$. But d has a unique occurrence in $T(I)$, so a(s)= N(d) - contradicting Claim 2.

Claim 4. For each value $d \in VAL(J)$ there is a value $e \in VAL(I)$ such that $\alpha(N(d))=$ N(e). Assume to the contrary that $\alpha(N(d))=$ T(u) for some $u \in I$. Then $\alpha(d0)=u$. But u has a unique occurrence in T(I), so $\alpha(s)=T(u)$ - contradicting Claim 2.

Claim 5. α can be extended to $T(w)$ so that $\alpha(T(w)) \in T(I)$.
Define a valuation β on J by $\beta(d)=\varphi(\alpha(d^i))$. β is well-defined, because, by Claim 4, $\alpha(d^i)=e^i$ for some $e \in VAL(I)$. Let $u=\langle d,e,f\rangle \in J$. Then, by Claim 3, $\alpha(T(u))=T(v)$ for some v $\in I$. But now

$$\beta(u)=\varphi(\langle\alpha(d^1),\alpha(e^2),\alpha(f^3)\rangle)=\varphi(\alpha(T(u)[ABC]))=$$

$$=\varphi(T(v)[ABC])=v \in I.$$

That is, $\beta(J) \subseteq I$. It follows that β can be extended to w so that $\beta(w) \in I$. Either c $\in VAL$ (I) and $\alpha(c^3)=\beta(c)^3$, or we can define $\alpha(c^3)$ to be $\beta(c)^3$. Also, we can define at(w) to be $\beta(w)$, and get $\alpha(T(w))=T(\beta(w)) \in T(I)$. • I

Things are more complicated when, given a counterexample relation to the implication $T(C) \models_{(f)} T(\sigma)$, we try to find a counterexample relation to the implication $\Sigma \models_{(f)} \sigma$. The reason for that is that the counterexample relation I', $I' \models T(\Sigma)$ and I' $\not\models T(\sigma)$, is not necessarily a translation $T(I)$ of some untyped relation I. Thus, it is not sufficient to define T^{-1} in the obvious way on the collection $(T(J)$: J is an untyped relation}. On the other hand, it is not clear how to define T^{-1} on the collection (I' : I' is a typed relation}.

The solution is to ensure that the typed counterexample relations have some structure to them. For example, we require that they satisfy the fd's that are satisfied by $T(I)$ as in Lemma 1. But that is not enough. $T(I)$ also has the property that if $T(\langle a,b,c\rangle) \in T(I)$ then also $N(a),N(b),N(c) \in T(I)$. Unfortunately, we can not express this property by a td, so we'll have to do with a weaker statement, saying that if $T(\langle a,b,c\rangle) \in T(I)$ and also $N(a),N(b) \in T(I)$, then also $N(c) \in T(I)$. The reason that this weaker statement suffices is that we are dealing with $A'B'$-total dependencies. The weaker statement can be expressed by a typed td $\sigma_0=\langle w_0,I_0\rangle$, $I_0=\{s,w_1,w_2,w_3\}$:

	A	B	C	D	E	F
s:	a0	bo	co	d0	e0	f0
w_1:	a1	b2	c3	d1	e0	f1
w_2:	a1	a2	a3	d0	e1	f1
w_3:	b1	b2	b3	d0	e2	f1
w_0:	c1	c2	c3	d0	e3	j2

Let Σ_0 be the set $\{\sigma_0, AD \rightarrow U, BD \rightarrow U, CD \rightarrow U, ABCE \rightarrow U\}$. We are now in position to define our inverse mapping T^{-1}.

Lemma 3. Let σ be a U'-total untyped td, and let I' be a typed relation such that $I' \not\models T(\sigma)$ and $I' \models \Sigma_0$ Then we can construct an untyped relation $T^{-1}(I') = I$ such that $I \not\models \sigma$, and for every $A'B'$-total untyped td θ such that I' $\models$ T(8) we have $I \models \theta$.

Proof.

Let σ be $\langle w, J\rangle, w = \langle a,b,c\rangle, \{a,b,c\} \subseteq VAL(J)$. $I' \not\models \langle T(w), T(J)\rangle$, i.e., there is a valuation' α such that $\alpha(T(J)) \subseteq I'$ but α can not be extended to $T(w)$ so that $\alpha(T(w)) \in I'$. Assume, without loss of generality, that $\alpha(s) = s$ (we can always rename values to assure that), in particular $\alpha(d0) = d0$, and $\alpha(e0) = e0$. We define an equivalence relation $\equiv$ on $VAL(I')$ as follows: $d \equiv e$ if $d = e$ or if there is a tuple $u \in I'$ such that $u[D] = d0$ and $\{d,e\} \subseteq VAL(u[ABC])$. Clearly, $\equiv$ is reflexive and symmetric. To show that it is transitive, suppose that $d \equiv e, e \equiv f$, $d \neq e$, and $e \neq f$. I.e., there are tuples $u, v \in I'$ such that $u[D] = v[D] = d0, \{d,e\} \subseteq VAL(u[ABC])$ and $\{e,f\} \subseteq VAL(v[ABC])$. Since I' is typed, either $u[A] = v[A] = e$, $u[B] = v[B] = e$, or $u[C] = v[C] = e$; that is, either $u[AD] = v[AD]$, $u[BD] = v[BD]$ or $u[CD] = v[CD]$. But $I' \models \{AD \rightarrow U, BD \rightarrow U, CD \rightarrow$ U$\}$, so in either case u = v and $d \equiv f$. Note that, since I' is typed, for all $u, v \in I', u[A] \equiv v$[A] iff $u[A] = v[A]$, $u[B] \equiv v[B]$ iff $u[B] = v[B]$, and $u[C] \equiv v[C]$ iff $u[C] = v[C]$.

Let $\rho: VAL(I') \rightarrow DOM'$ be a mapping such that $\rho(d) = \rho(e)$ iff $d \equiv e$. We define I by:

$I = \{\rho(u[ABC]) : u \in I', u[E] = e0, u[F] = \alpha(f1)$

and there are tuples $u_1, u_2, u_3 \in I'$ such that

$u_1[D] = u_2[D] = u_3[D] = d0,\ u_1[F] = u_2[F] = u_3[F] = \alpha(f1),$

$u_1[A] = u[A], u_2[B] = u[B],$ and $u_3[C] = u[C]\}$

(The intuition is that u looks like $T(\langle e, f, g\rangle)$ and u_1, u_2, and u_3 look like $N(a)$, $N(b)$, and N(c), respectively.) Observe that if I' is finite then so is I.

Claim 1. $I \not\models \sigma$

We want to define a valuation β such that $\beta(J) \subseteq I$ but $\beta(w) \notin I$. If $d \in VAL(J)$, then $\alpha(N(d)) \in I'$ and $\alpha(N(d)[D]) = \alpha(d0) = d0$, It follows that $\alpha(d^1) \equiv \alpha(d^2) \equiv \alpha(d^3)$. We define a valuation β on J by: $j?(d) = \rho(\alpha(d^1)) = \rho(\alpha(d^2)) = \rho(\alpha(d^3))$. Let v $= \langle d, e, f\rangle \in J$. Then it is easy to verify that $\alpha(T(v)), \alpha(N(d)), \alpha(N(e))$, and $\alpha(N(f))$ satisfy the conditions for u, u_1, u_2, and u_3 in the definition of I'. It follows that

$$\beta(v) = \langle \rho(\alpha(d^1)), \rho(\alpha(e^2)), \rho(\alpha(f^3))\rangle = \rho(\alpha(T(v))[ABC]) \in I.$$

Consequently, $\beta(J) \subseteq I$.

Suppose now that $\beta(w) = \langle \beta(a), \beta(b). \beta(c)\rangle \in I$. I.e., there is a tuple $u \in I'$ such that $u[E] = e0, u[F] = \alpha(f1)$ and $\beta(w) = \rho(u[ABC])$. Now $a \in VAL(J)$, so $\alpha(N(a)) \in I'$. Consequently, $\beta(a) = \rho(\alpha(a^1)) = \rho(\alpha(N(a))[A]) = \rho(u[A])$, so $\alpha(N(a))[A] \equiv u[A]$, and consequently $\alpha(a^1) = u[A]$. Similarly, $\alpha(b^2) = u[B]$ and $\alpha(c^3) = u[C]$; that is $\alpha(T(w)[ABCEF]) = u[ABCEF]$. Defining $\alpha(w) = u[D]$ wc get $\alpha(T(w)) = u \in I'$ - contradiction.

Claim 2. $I' \models T(\theta)$ entails $I \models \theta$.

Let θ be $\langle u, K\rangle$, and let β bc a valuation on K such that $\beta(K) \subseteq I$. We want to define a valuation y such. that $\gamma(T(K)) \subseteq I'$. Then y can bc cxtcndcd to $T(u)$ so that $\gamma(T(u)) \in I'$, and from this we'll bc able to cxtcndcd β to u so that $\beta(u) \in I$. Let v $= \langle d, e, f\rangle \in K$, thcn $\beta(v) \in I$. That is, there arc tuples $t, t_1, t_2, t_3 \in I'$ such that $t[F] = t_1[F] = t_2[F] = t_3[F] = \alpha(f1)$, $t[E] = e0$,

$t_1[D]=t_2[D]=t_3[D]=d0$, $t[A]=t_1[A]$, $t[B]=t_2[B]$, $t[C]=t_3[C]$ and $\beta(v)=\rho(t[ABC])$. Furthermore, we claim that I, t_1, t_2, and t_3 arc unique.

Suppose that x satisfies the same condition as t. In particular, $\beta(v)=\rho(t[ABC])=\rho(x[ABC])$, that is, $t[A]\equiv x[A]$, $t[A]\equiv x[B]$, and $t[C]\equiv x[C]$, and therefore $t[ABC]=x[ABC]$. But also $t[E]=x[E]=e0$ and $I'\models ABCE\rightarrow U$, so $x=t$.

Suppose that x_1 satisfies the same conditions as t_1. In particular, $x_1[A]=$,[A]= $t_1[A]$ and $x_1[D]=t_1[D]=d0$. But $I'\models AD\rightarrow U$, so $x_1=t_1$. Similarly, because I' $\models \{BD\rightarrow U, CD\rightarrow U\}$, t_2 and t_3 are unique.

We define now a valuation y on $T(v)$, $N(d)$, N(e) and $N(f)$ by: $\gamma(T(v))=t$, $\gamma(N(d))=t_1$, $\gamma(N(e))=t_2$, and $\gamma(N(f))=t_3$. Obscrvc that $y(dO)=dO$, $\gamma(e0)=e0$, and $\gamma(f1)=\alpha(f1)$. We have to show that in a similar manner we can define y on all tuples in K. Thus, suppose for example that x = $\langle d,g,h\rangle\in K$, then there exist tuplcs $y,y_1,y_2,y_3\in I'$ satisfying conditions analogous to the conditions above for I, t_1, t_2 and t_3. But then, $\beta(d)=\rho(t[A])=\rho(y[A])$ so $y_1[A]=y[A]=t[A]=t_1[A]$. Also, $y_1[D]=t_1[D]=d0$, so $t_1[AD]=y_1[AD]$ and, since $I'\models AD\rightarrow U$, $y_1=t_1$. It follows that defining $\gamma(T(x))=y$ and $\gamma(N(d))=y_1$ is consistent with the definition $\gamma(T(v))=t$ and $\gamma(N(d))=t_1$. Defining $\gamma(s)=s$ wc get that $\gamma(T(K))\subseteq I'$. Let $u=\langle d,e,f\rangle$. Since $I'\models\langle T(u),T(K)\rangle$, wc can cxtcnd y to $T(u)$ so that $\gamma(T(u))=z\in I'$.

Our aim is now to show that $\rho(z[ABC])\in I$. Recall that $\{d,e\}\subseteq VAL(K)$, so let $z_1=\gamma(N(d))\in I'$ and $z_2=\gamma(N(e))\in I'$. We want to have some z_3 that looks like $\gamma(N(f))$, but if $f\notin VAL(K)$ then wc don't know whcthcr $\gamma(N(f))\in I'$. Now we havc to usc thc fact that $I'\models\sigma_0$. Define a valuation δ on I_0 so that $\delta(s)=s$, $\delta(w_1)=z$, $\delta(w_2)=z_1$ and $\delta(w_3)=z_2$. δ is wcll-dcfincd bccausc $\delta(a1)=z[A]=z_1[A]$, $\delta(b2)=z[B]=z_2[B]$, $\delta(d0)=d0$, $\delta(e0)=e0$ a n d $\delta(f1)=\alpha(f1)$. Sincc $I'\models\sigma_0$, we can cxtcnd 8 to w_0 so that $z_3=\delta(w_0)\in$ I'. (Clearly, if $f\in VAL(K)$ thcn z_3 is just $\gamma(N(f))$. In particular, $z_3[C]=z[C]=\delta(c3)$, so $\rho(z[ABC])\in I$.

To complete the proof of the claim we show how to get that $\beta(u)$ is $\rho(z[ABC])$. *Now* $d \in VAL(K)$, so v $= \gamma(N(d)) \in I'$. But

$$\rho(v[A]) = \rho(v[B]) = \rho(v[C]) = \beta(d),$$

and $v[A] = \gamma(d^1) = z[A]$, so $\rho(z[A]) = \beta(d)$. Similarly, $\rho(z[B]) = \beta(e)$. If $f \in VAL(K)$ then $\rho(z[C]) = \beta(f)$. Otherwise, wc can define $\beta(f) = \rho(z[C])$. In either case, $\beta(u) = \rho(z[ABC])$. ❑

Following Lemma 3, we are inclined to define *T(C)* as $\{T(B)\colon \theta \in \Sigma\} \cup \Sigma_0$. But now we see that Lemma 2 does not yet prove the correctness of the first direction of the reduction. That is, given an untyped relation I such that $I \models \Sigma$ and $I \not\models \sigma$, Lemma 2 ensures that *T(I)* $\not\models T(\sigma)$ and *T(I)* $\models \{T(\theta) : \theta \in \Sigma\}$. Also, Lemma 1 ensures that *T(I)* satisfies the fd's in Σ_0. But does *T(I)* satisfy σ_0? Let a be a valuation such that $\alpha(I_0) \subseteq$ *T(I)* and $|\alpha(I_0)| > 0$. If $\alpha(s)$ = s, then, as in the proof of Lemma 2, we can show that for some $\langle d,e,f\rangle \in I$ *we* have that $\alpha(w_1) = T(\langle d,e,f\rangle)$, $\alpha(w_2) =$ *N(d),* and $\alpha(w_3) = N(e)$. So we can extend a to w_0 to get $\alpha(w_0) = N(f) \in$ *T(I).* But, unlike in the proof of Lemma 2, -we can not show that necessarily a(s)= s, so we can not prove that *T(1)* $\models \sigma_0$. However, given an additional constraint on I, specifically, $I \models A$ B '→ C', wc can prove that *T(I)* $\models \sigma_0$.

Lemma 4. Let I be an untyped relation. If $I \models A$ B $' \to C'$ then *T(I)* $\models \sigma_0$.

Proof. Preliminary to showing that $T(I) \models \sigma_0$, let us show that *T(I)* $\models$ *ABE-W.* Let $u,v \in T(I)$ and $u[ABE] = v[ABE]$. If ufv then $u[E] = v[E] = e0$. If $u = s$ then v $= s$ and vice versa, because $a0 \neq a^1$ for all $a \in DOM'$. It follows that $u =$ *T(p)* and v = *T(q)* for some $p,q \in I$. But $u[AB] = v[AB]$ entails $p[A'B'] = q[A'B']$, becausc [1] and [2] arc one-to-one, and $p[A'B'] = q[A'B']$ entails $p = q$ because $I \models A'B' \to C'$. Ncccssarily, u = v.

Let us show that *T(Z)* $\models \sigma_0$. Suppose that a is a valuation on I_0 such that $\alpha(I_0) \subseteq T(I)$. If α maps either w_1, w_2, or w_3 to s then $\alpha(f1) = f0$ so $\alpha(I_0) =$ {s), and α can be extended to w_0 so that $\alpha(w_0)$ = s. Consequently, we can assume that $\alpha(I_0 - \{s\}) \subseteq$ *T(I)* $-$ {s). Supposc that $\alpha(s) = s$. Then $\alpha(e0) = e0$, so $\alpha(w_1) = T(t)$ for some $t = \langle d,e,f\rangle \in I$. Also $\alpha(d0) = d0$, *so*

$\alpha(w_2)=N(d)$ and $\alpha(w_3)=N(e)$. Wc can extend α to w_0 so that $\alpha(w_0)=N(f)\in T(I)$.

Suppose that $a(s)=T(t)$ for some $t\in I$. Then $\alpha(d0)=t$, so $\alpha(w_2)=\alpha(w_3)=T(t)$. Thus, $\alpha(w_1[A])=\alpha(w_2[A])=T(t)[A]$, $\alpha(w_1[B])=\alpha(w_3[B])=T(t)[B]$ and $\alpha(w_1[E])=\alpha(e0)=e0=T(t)[E]$; that is, $\alpha(w_1)[ABE]=T(t)[ABE]$. But $T(I)\models ABE\rightarrow U$, so $\alpha(w_1)=T(t)$. We have shown that $\alpha(I_0)=\{T(t)\}$, consequently, a can be extended to w_0 so that $\alpha(w_0)=T(r)$.

Finally, suppose that a(s)= $N(a)$ for some $a\in VAL(I)$. Then $\alpha(e0)=a$, so $\alpha(w_1)=N(a)$. Now $\alpha(w_2[D])=\alpha(w_3[D])=d0=N(a)[D]$, $\alpha(w_2[A])=\alpha(w_1[A])=N(a)[A]$ and $\alpha(w_3[B])=\alpha(w_1[B])=N(a)[B]$; that is $\alpha(w_2)[AD]=N(a)[AD]$ and $\alpha(w_3)[BD]=N(a)[BD]$. But $T(I)\models\{AD\rightarrow U, BD\rightarrow U\}$, so $\alpha(w_2)=\alpha(w_3)=N(a)$. Wc have shown that $\alpha(I_0)=\{N(a)\}$, consequently, α can be extended to w_0 so that $\alpha(w_0)=N(a)$. • I

There is another problem with our proposed $T(\Sigma)$. It is not a set of td's! Fortunately, we know how to replace fd's by td's. First, observe that an fd $X\rightarrow Y$ is equivalent to the set of fd's $\{X\rightarrow A : A\in Y-X\}$. Thus, wc can assume that all fd's in Σ arc of the form $X\rightarrow A$ with $A\notin X$. We now define $\theta_{X\rightarrow A}$ as a U-total td $\langle u,\{u_1,u_2,u_3\}\rangle$, where

(1) $u_1[X]=u_2[X]$ and $u_1[B]\neq u_2[B]$ for $B\in\overline{X}$,

(2) $u_3[A]=u_2[A]$ and $u_1[A]\neq u_3[B]\neq u_2[A]$ for $B\in\overline{A}$, and

(3) $u[A]=u_1[A]$ and $u[\overline{A}]=u_3[\overline{A}]$.

Example 3.

$\theta_{AD\rightarrow B}$ is $\langle u,\{u_1,u_2,u_3\}$:

	A	B	C	D	E	F
u:	a3	b1	c3	d3	e3	f3
u_1:	a1	b1	c1	d1	e1	f1
u_2:	a1	b2	c2	d1	e2	j2
u_3:	a3	b2	c3	d3	e3	f3

Lemma 5. [BV3][†] Let Σ be a set of typed td's and fd's. Let Σ' be the set obtained by replacing each fd $X \rightarrow A$ in Σ by $\theta_{X \rightarrow A}$. Then $\Sigma \models$ C', and for all typed td's $\sigma, \Sigma \models \sigma$ if and only if $\Sigma' \models \sigma$ and $\Sigma \models_f \sigma$ if and only if $\Sigma' \models_f \sigma$. •I

Thus, wc define $T(\Sigma)$ as $(\{T(B)\colon \theta \in \Sigma\} \cup \Sigma_0)'$, with $'$ defined as in the lemma. We are now in position to prove the main result.

Theorem 2. The implication and the finite implication problem for typed td's are unsolvable.

Proof. Let Σ and σ be as in Theorem 1. We claim that $\Sigma \models_{(f)} \sigma$ iff $T(\Sigma) \models_{(f)} T(\sigma)$. Since T is an effective translation, the claim follows.

Suppose first that $\Sigma \not\models_{(f)} \sigma$, then by condition (3) of Theorem 1, $\Sigma \cup \{A'B' \rightarrow C'\} \not\models_{(f)} \sigma$. Thus, there is an untyped (finite) relation I such that $I \models \Sigma, I \models A'B' \rightarrow C'$ and $I \not\models \sigma$. By Lemmas 1 and 4, $T(I) \models \Sigma_0$, and by Lemma 2, $T(I) \models \{T(\theta) : \theta \in \Sigma\}$ and $T(I) \not\models T(\sigma)$. It follows by Lemma 5 that $T(I) \models T(E)$, so $T(C) \not\models_{(f)} T(\sigma)$.

Suppose now that $T(\Sigma) \not\models_{(f)} T(\sigma)$. By Lemma 5, we have that $\{T(\theta) : \theta \in \Sigma\} \cup \Sigma_0 \not\models_{(f)} T(\sigma)$. Thus, there is a typed (finite) relation I', such that $I' \models \{T(\theta) : \theta \in \Sigma\}$, $I' \models \Sigma_0$, and $I' \not\models T(\sigma)$. Note that by condition (1) in Theorem 1 we can assume that σ is U'-total. Let $I = T^{-1}(I')$ as in Lemma 3. By that lemma we know that $I \models \Sigma$ and $I \not\models \sigma$, so $\Sigma \not\models_{(f)} \sigma$. □

Let us make two observations. First, by Theorem 1, there is a fixed untyped σ such that deciding whether $\Sigma \models_{(f)} T(\sigma)$ is unsolvable. Secondly, it is easy to see that the set $\{(\Sigma, \sigma) : \Sigma \not\models_f \sigma\}$ is recursively enumcrablc. It follows that the finite implication problem for typed td's is not even partially solvablc. Thus, there is no sound and complete formal system for finite implication of typed td's. In contrast, see [BV4, SU] for sound and complete systems for implication of typed td's.

[†] The same result was also shown in [SU] for unrestricted implication.

5. Some Consequences

Let Ψ be a class of dependencies and $\Sigma \subseteq \Psi$. The (finite) implication problem for Σ *in* Ψ is to decide, given $\sigma \in \Psi$, whether $\Sigma \models_{(f)} \sigma$ Note that the unsolvability results of Theorems 1 and 2 does not say anything about the solvability of the (finite) implication problem for specific C's. For example, it is known that the (finite) implication problem for $\varnothing$ in the class of (typed) td's is solvable [BV1, SU]. Also, in [FMUY] it is shown there is a typed td σ that implies all typed td's. Thus, the (finite) implication problem for $\{\sigma\}$ in the class of typed td's is trivially solvable. It is conceivable that for every fixed Σ its (finite) implication problem in the class of (typed) td's is solvable, yet there is no effective way to find, when given a specific Σ, the decision procedure for that Σ. In [BV1] a fixed set Σ_1 of untyped td's is presented, whose implication problem in the class of untyped td's is unsolvable. Using a result from [GL2] we can get a much stronger result involving recursive inseparability. Recall ([Ro]) that two sets X and Y are recursively inseparable if there is no recursive set containing X and disjoint from Y.

Theorem 3. There is a set Σ_2 of untyped A$'B$'-total td's such that the set

$$(a : \sigma \text{ is a } U' - \text{ total untyped td and } \Sigma_2 \models a)$$

and the set

$$\{\sigma : \sigma \text{ is a } U' - \text{ total untyped td and } \Sigma_2 \cup \{A'B' \rightarrow C'\} \not\models_f \sigma\}$$

are recursively inseparable.

Proof. An *equational implication* for semigroups (abbr. ei) is a sentence of the form

$$\forall y_1 \cdots \forall y_n (s_1 = t_1 \wedge \cdots \wedge s_k = t_k \rightarrow s_{k+1} = t_{k+1}),$$

where $k,n > 0$ and the s_i's and t_i's are terms built from the y_i's by means of the semigroup multiplication symbol. In [GL2] it is shown that the set

$$\{\varphi : \varphi \text{ is an ei that holds in all semigroups})$$

and the set

$$\{\varphi : \varphi \text{ is an ei that fails in some finite semigroup})$$

are recursively inseparable. Using the technique of [BV1] to reduce questions about ei's in groupoids to implication of untyped td's, we can prove the claim, where Σ_2 expresses the axioms for semigroups. □

Corollary 1. The implication and the finite implication problem for Σ_2 in the class of untyped td's are unsolvable.

Proof. Observe first that the theorem entails that the set

$$\{\sigma : \sigma \text{ is an untyped td and } \Sigma_2 \models a)$$

and the set

$$(a : \sigma \text{ is an untyped td and } \Sigma_2 \not\models_f \sigma\}$$

are also recursively inseparable. The claim then follows because by definition a set that is recursively inseparable from some other set can not be recursive. •I

We now note that the td's in the statement of Theorem 3 satisfy the conditions of Theorem 1, so by applying the reduction of the previous section we get inseparability results for typed td's.

Theorem 4. There is a set Σ_3 of typed td's such that the set

$$\{a : \sigma \text{ is a typed td and } \Sigma_3 \models \sigma\}$$

and the set

$$\{\sigma : \sigma \text{ is a typed td and } \Sigma_3 \not\models_f \sigma\}$$

are recursively inseparable. □

Corollary 2. The implication and the finite implication problem for Σ_3 in the class of typed td's are unsolvable. □

An interesting question is whether we can decide, given a set Σ of (typed) td's, if its (finite) implication problem in the class of (typed) td's is solvable or not. In [Va2] it is shown that for set Σ of untyped td's and equality generating dependencies this problem is unsolvable, By techniques similar to those employed in proving Lemma 5, it can be shown that the problcm is unsolvable also for sets Σ of untyped td's. However, the proof method does not extend to the typed case.

Corollary 2 has an interesting consequence. Let Ψ be a class of dependencies and $\Sigma \subseteq \Psi$. A finite relation I such that for all $\sigma \in \Psi$, we have that $I \models \sigma$ if and only if $\Sigma \models_f \sigma$ is called a *finite Armstrong relation for* Σ *in* Ψ [Fag2].

Theorem 5. Σ_3 does not have a finite Armstrong relation in the class of typed td's.

Proof. Suppose to the contrary that I is a finite Armstrong relation for Σ_3 in the class of typed td's. Let σ be a typed td. Now $\Sigma_3 \models_f \sigma$ iff $I \models \sigma$. But the set $\{\sigma : I \models$ a) is recursive, which means that the finite implication problem for Σ_3 in the class typed td's is solvable - contradiction. Cl

We mention that in [FMUY] a set of two typed td's is defined, which does not have a finite Armstrong relation in the class of typed td's.

6. Projected Join Dependencies

In this section we are dealing exclusively with the typed case. Let U be a universe, and let $\mathbf{R}=(R_1, \ldots, R_k)$ be a sequence without repetition of subsets of U, with $\bigcup_{i=1}^{k} R_i = R \subseteq U$. The *project-join* mapping $m_{\mathbf{R}}$ maps U-relations to R -relations as follows:

$$m_{\mathbf{R}}(I)=\{t : t \text{ is an } R-\text{value s.t. } t[R_i] \in I[R_i] \text{ for } i=1, \ldots, k).$$

Let $X \subseteq R$. A *projected join dependency* (abbr. pjd) [YP] is a statement $*[\mathbf{R}]_X$. It is satisfied by a relation I if $(m_{\mathbf{R}}(I))[X]=I[X]$. The interest in pjd's comes from the question whether we can compute $I[X]$ when given the projections $I[R_1], \ldots, I[R_k]$.

Several special cases of pjd's have been investigated in the literature. If $X = R$, then we drop the subscript X and call $*[\mathbf{R}]$ a *join dependency* [ABU, Ri]. If R = U, then $*[\mathbf{R}]$ is called *total* otherwise it is called *embedded* [MMS]. If we have above $\mathbf{R}=(R_1,R_2)$ then the join dependency is also called a *multivalued dependency* (abbr. mvd) [Fag1]. A total mvd $*[R_1,R_2]$

is also denoted by $R_1 \cap R_2 \twoheadrightarrow R_1 - R_2$. According the definition of satisfaction for pjd's, $I \models X \twoheadrightarrow Y$ exactly when, 4.r all $u,v \in I$, if $u[X]=v[X]$, then there is a $w \in I$ with $w[XY]=u[XY]$ and $w[\overline{XY}]=v[\overline{XY}]$. Clearly, if $I \models X \rightarrow Y$, then also $I \models X \twoheadrightarrow Y$.

Even though pjd's and td's look on the surface completely different, we can in fact view pjd's as special td's. A td $\langle w,I \rangle$ is called *shallow* [YP], if whenever u and v are two distinct tuples in I and $u[A]=v[A]$, then

(1) if s and t are two distinct tuples in I and $s[A]=t[A]$ then $s[A]=t[A]=u[A]=v[A]$, and

(2) either $w[A]=u[A]=v[A]$ or $w[A] \notin VAL(I)$.

Lemma 6. For every shallow td σ there exist a pjd 8, and for every pjd θ there exists a shallow td σ, such that for all relations I, $I \models \sigma$ if and only if $I \models \theta$.

Proof. The claim follows from the connection between relational expressions and tableaux as described in [ASU]. ❑

Thus, instead of talking about pjd's we can talk about shallow td's. Our aim in this section is to show that the implication and the finite implication problem for td's are reducible to the corresponding problems for shallow td's. The reduction is essentially due to Yannakakis and Papadimitriou [YP]. However, they have dealt only with the implication problem, and their proof-theoretic technique does not extend to finite implication. In contrast, our proof, which is model-theoretic, shows that the reduction is conservative (i.e., preserve finiteness of relations), and therefore proves simultaneously the correctness of the reduction for both implication and finite implication.

We note that for a fixed universe U there are only finitely many U-pjd's, so the (finite) implication problem is solvable. Thus, unlike the case with arbitrary td's, we have to deal here with arbitrary universes. In fact, the basic idea of the reduction is that given Σ, σ over a universe U, we translate them to shallow $\tilde{\Sigma}, \tilde{\sigma}$ over a bigger universe $\tilde{U}$, whose size depends on the size of the td's in $\Sigma \cup \{\dot{\sigma}\}$.

More specifically, let

$$m = \max\{k : \langle w, I\rangle \in \Sigma \cup \{\sigma\} \text{ and } |I| = k\},$$

and let $n = m(m-1)/2$. Then we take

$$\tilde{U} = \{A_i : A \in U \text{ and } 0 \leq i \leq n\}.$$

The intended interpretation is that the $A_0 \cdots A_n$-values in the new universe encode the A -values in the old universe. For domain we take $DOM(A_i) = \{A_i\} \times N$ (N is the set of natural numbers). However, when describing A_i-values we'll usually omit the first component of the pair; i.e., we write $w[A_i] = 1$ instead of the more precise $w[A_i] = \langle A_i, 1\rangle$. We assume without loss of generality that $DOM(U) \subseteq N$.

A U-td θ is translated to a shallow $\tilde{U}$-td $\tilde{\theta}$ as follows. Let θ be $\langle w, I\rangle$. We can assume without loss of generality that I = $\{w_1, \ldots, w_m\}$. Let us fix some enumeration of the set $\{\{i,j\} : 1 \leq i,j \leq m$ and i Sj). By $A_{i,j}$ we mean A_k, where k is the ordinal number of $\{i,j\}$ in that enumeration. $\tilde{\theta}$ is $\langle u, \tilde{I}\rangle$, $\tilde{I} = \{u_1, \ldots, u_m\}$. $\tilde{I}$ is constructed so that $u_i[A_{i,j}] = u_j[A_{i,j}]$ iff $w_i[A] = w_j[A]$, so that the equalities between A -values in I are spread over $A_1, \ldots, A_n$ in $\tilde{I}$, which makes $\tilde{\theta}$ shallow.

More precisely, $\tilde{I}$ is defined as follows.

(1) For $A \in U, 1 \leq k \leq m : u_k[A_0] = k$.

(2) For $A \in U, 1 \leq i,j,k \leq m, i \neq j$: For k different from i and j, let $u_k[A_{i,j}] = k$. If $u_i[A] \neq u_j[A]$ then $u_i[A_{i,j}] = i$ and $u_j[A_{i,j}] = j$. Otherwise, $u_i[A_{i,j}] = u_j[A_{i,j}] = min\{i,j\}$.

u is defined as follows.

(1) For A $\in U$: If $w[A] \in VAL(I)$ then $w[A] = w_k[A]$ for some $1 \leq k \leq m$, so $u[A_0] = k = u_k[A_0]$. Otherwise, $u[A_0] = m$ -l-l.

(2) For $A \in I$, $1 \leq i \leq n$: Let $u[A_i] = m+1$.

We leave it to the reader to show that $\tilde{\theta}$ is indeed shallow.

Example 4. Let U = ABC, and let θ be a td over U, $\theta = \langle w, I \rangle$, $I = \{w_1, w_2, w_3\}$:

	A	B	C
w:	a	b	$c3$
w_1:	a	$b1$	$c1$
w_2:	$a1$	b	$c1$
w_3:	$a1$	$b1$	$c2$

Now $\tilde{U} = A_0 \cdots A_3 B_0 \cdots B_3 C_0 \cdots C_3$. Let $A_{1,2} = A_1$, $A_{1,3} = A_2$, and $A_{2,3} = A_3$. $\tilde{\theta}$ is $\langle u, I \rangle$, $I = \{u_1, u_2, u_3\}$:

	A_0	A_1	A_2	A_3	B_0	B_1	B_2	B_3	C_0	C_1	C_2	C_3
u:	1	4	4	4	2	4	4	4	4	4	4	4
u_1:	1	1	1	1	1	1	1	1	1	1	1	1
u_2:	2	2	2	2	2	2	2	2	2	1	2	2
u_3:	3	3	3	2	3	3	1	3	3	3	3	3

The following lemma describes the relationship between U-relations and $\tilde{U}$ relations on one hand and θ and $\tilde{\theta}$ on the other hand. We use U_0 to denote the set $\{A_0 : A \in U\}$.

Lemma 7. Let I be a U-relation, and let $\tilde{I}$ be a c-relation such that

(1) There is a one-to-one mapping $\gamma : DOM(U) \rightarrow DOM(U_0)$ such that $\gamma(I) = \tilde{I}[U_0]$.

(2) $\tilde{I} \models A_i \rightarrow A_j$ for all A $\in U$ and $0 \leq i,j \leq n$.

Then for all td's θ over U. $I \models \theta$ if and only if $\tilde{I} \models \tilde{\theta}$.

Proof. We first show that for every $s \in I$ there is a unique $t \in \tilde{I}$ such that y(s)= $t[U_0]$. Clearly, there is at least one such t because $\gamma(s) \in \tilde{I}[U_0]$. Suppose that y(s)= $t[U_0] = v[U_0]$. Now for all $A \in U$ and $1 \leq i \leq n$, we have $t[A_0] = v[A_o]$ and $I' \models A_0 \rightarrow A_i$, so $t[A_i] = v[A_i]$. It follows that t = v. We say that *t comes from* s. Observe that if t_1, t_2 come from s_1, s_2, respectively, then for all $A \in U$ and $0 \leq i \leq n$, *we* have $s_1[A] = s_2[A]$ iff $t_1[A_i] = t_2[A_i]$.

Let $\theta = \langle w, J \rangle$, $J = \{w_1, \ldots, w_m\}$, and $\tilde{\theta} = \langle u, \tilde{J} \rangle$, $\tilde{J} = \{u_1, \ldots, u_m\}$.

If: Suppose that $\tilde{I} \models \tilde{\theta}$. Let β be a valuation on J such that $\beta(J) \subseteq I$. Let $t_1, \ldots, t_m \in \tilde{I}$ come from $\beta(w_1), \ldots, \beta(w_m)$, respectively. Now if $u_i[A_{i,j}] = u_j[A_{i,j}]$, then $w_i[A] = w_j[A]$, and $\beta(w_i)[A] = \beta(w_j)[A]$. Consequently $t_i[A_{i,j}] = t_j[A_{i,j}]$. Thus, we can define a valuation α on $\tilde{J}$ so that $\alpha(u_k) = t_k$. Since we assumed that $\tilde{I} \models \tilde{\theta}$, α can be extended to u so that $\alpha(u) \in \tilde{I}$. Let $\alpha(u)$ come from $s \in I$. We extend β to w so that $\beta(w) = s$. If $w[A] \notin VAL(I)$, then we define $\beta(w[A]) = s[A]$. Otherwise, $w[A] = w_k[A]$ for some $l < k \leq m$. But in that case, $u[A_0] = u_k[A_0]$, so $\alpha(u)[A_0] = t_k[A_0]$ and $\beta(w[A]) = \beta(w_k)[A] = s[A]$. *So we* have that $\beta(w) = s$ as desired.

Only if: Suppose that $I \models$ 8. Let **a** be a valuation on J' such that $\alpha(J') \subseteq I'$. The tuples $\alpha(u_1), \ldots, \alpha(u_m)$ come from some tuples $s_1, \ldots, s_m \in I$, respectively. We claim that if $w_i[A] = w_j[A]$, then $s_i[A] = s_j[A]$. Indeed, if $w_i[A] = w_j[A]$ then $u_i[A_{i,j}] = u_j[A_{i,j}]$, so necessarily $\alpha(u_i)[A_{i,j}] = \alpha(u_j)[A_{i,j}]$, and consequently $s_i[A] = s_j[A]$. 'Thus, we can define a valuation β on J so that $\beta(w_k) = s_k$ for $1 \leq k \leq m$. Since wc assume that $I \models \theta$, β can be extended to w so that $\beta(w) \in I$. Let $t \in I'$ come from $\beta(w)$. We extend α to u so that $\alpha(u) = t$. If $u[A_i] \notin VAL(I')$, then we define $\alpha(u[A_i]) = t[A_i]$. Otherwise, $u[A_0] = u_k[A_0]$ for some $1 \leq k \leq m$. But in that case $w[A] = w_k[A]$, so $\beta(w)[A] = s_k[A]$ and $\alpha(u[A_0]) = \alpha(u_k)[A_0] = t[A_0]$. So wc have $\alpha(u) = t$ as desired. Cl

By means of Lemma 7 wc can show that the (finite) implication problem for td's is reducible to the (finite) implication problem for fd's and pjd's. Let $\tilde{\Sigma}$ be

$$\{\tilde{\theta} : \theta \in \Sigma\} \cup \{A_i \rightarrow A_j : A \in U \text{ and } 0 \leq i,j \leq n\}.$$

Lemma 8. $\Sigma \models \sigma$ if and only if $\tilde{\Sigma} \models \tilde{\sigma}$ and $\Sigma \models_f \sigma$ if and only if $\tilde{\Sigma} \models_f \tilde{\sigma}$.

Proof. As is Section 4, wc show that $\Sigma \not\models_{(f)} \sigma$ iff $\tilde{\Sigma} \not\models_{(f)} \tilde{\sigma}$ by constructing counterexample relations.

Suppose that $\Sigma \not\models_{(f)} \sigma$. Then there is a (finite) U-relation I such that $I \models \Sigma$ and $I \not\models \sigma$. We construct a e-relation $\tilde{I}$ by duplicating I $n+1$ times. That is,

$$\tilde{I} = \{s : s \text{ is a } \tilde{U}-\text{value and there is } t \in I \text{ s.t. for all } A \in U \text{ and } 0 \leq i \leq n, s[A_i] = \langle A_i, t[A] \rangle\}$$

Observe that of I is finite then so is $\tilde{I}$. Also, it is easy to verify that for all $A \in U$ and $0 \leq i,j \leq n$, wc have $\tilde{I} \models A_i \rightarrow A_j$. By Lemma 7, $\tilde{I} \models \tilde{\Sigma}$ and $\tilde{I} \not\models \tilde{\sigma}$. It follows that $\tilde{\Sigma} \not\models_{(f)} \tilde{\sigma}$.

Suppose that $\tilde{\Sigma} \not\models_{(f)} \tilde{\sigma}$. Then there is a (finite) $\tilde{U}$-rclation $\tilde{I}$ such that $\tilde{I} \models \tilde{\Sigma}$ and $\tilde{I} \not\models \tilde{\sigma}$. Let I be a U-relation that is isomorphic to $\tilde{I}[U_0]$. That is, there is a one-to-one mapping $\gamma : DOM(U) \rightarrow DOM(U_0)$ such that $\gamma(I) = \tilde{I}[U_0]$. Clearly, if $\tilde{I}$ is finite then so is I. By Lemma 7, $I \models \Sigma$ and $I \not\models \sigma$. It follows that $\Sigma \not\models_{(f)} \sigma$. □

It seems now that wc only need to apply Lemma 5 to get rid of the fd's in $\tilde{\Sigma}$. Alas! A brief inspection reveals that $\theta_{A_i \rightarrow A_j}$ is not shallow. Fortunately, in our case it suffices to replace $A_i \rightarrow A_j$ by $A_i \rightarrow\rightarrow A_j$.

Lemma 9. Assume $3 \leq n, 0 \leq i,j,k \leq n, i \neq j, j \neq k$, and $i \neq k$. Then

$$\{A_p \rightarrow\rightarrow A_q : p,q \in \{i,j,k\}\} \models \theta_{A_i \rightarrow A_j}$$

Proof. Let us describe a o-value w as $(w[A_i], w[A_j], w[A_k], w[\overline{A_i A_j A_k}])$. Then $\theta_{A_i \rightarrow A_j}$ is $\langle t, \{u, v, w\} \rangle$:

	A_i	A_j	A_k	$\overline{A_i A_j A_k}$
t:	a2	b1	c3	x3
u:	a1	b1	c1	x1
v:	a1	b2	c2	x2
w:	a2	b2	c3	x3

Suppose that

$$\tilde{I} \models \{A_p \rightarrow\rightarrow A_q : p,q \in \{i,j,k\}\}.$$

Let α bc a valuation such that $\alpha(u), \alpha(v), \alpha(w) \in \tilde{I}$. $\alpha(u)$, $\alpha(v)$, and $\alpha(w)$ look like u, v and w, except that we have additional equalities like $\alpha(a1)=\alpha(a2)$. Since additional equalities do not bother us we can assume that $u, v, w \in \tilde{I}$. We now use the fact that $\tilde{I}$ satisfies the mvd's above to infer that $\tilde{I}$ must contain certain tuples. E.g., from v and w we can infer by $A_j \twoheadrightarrow A_k$ that $(a1,b2,c3,x2)\in\tilde{I}$. The following figure shows a chain of such inferences.

	A_i	A_j	A_k	$\overline{A_i A_j A_k}$	
u:	a1	$b1$	c1	x1	
v:	$a1$	$b2$	$c2$	$x2$	
w:	$a2$	$b2$	$c3$	$x3$	
s_1	$a2$	$b2$	c2	x3	(From w and v by $A_j \twoheadrightarrow A_k$)
s_2	$a1$	$b2$	$c2$	$x3$	(From s_1 and v by $A_k \twoheadrightarrow A_i$)
s_3	$a1$	$b1$	$c2$	$x3$	(From s_2 and u by $A_i \twoheadrightarrow A_k$)
s_4	$a2$	$b1$	c2	x3	(From s_3 and s_1 by $A_k \twoheadrightarrow A_i$)
t	$a2$	b1	$c3$	x3	(From s_4 and w by $A_i \twoheadrightarrow A_k$)

Thus, $t\in\tilde{I}$ and $\tilde{I}\models\theta_{A_i \to A_j}$. (Essentially, what we have done here is proving the implication by the *chase* proof procedure of [ABU, BV3, MMS, SU].) ∎

Corollary. Assuming $3 \le n$,

$$\{\theta_{A_i \to A_j} : 0 \le i,j \le n\} \models\!\dashv \{A_i \twoheadrightarrow A_j : 0 \le i,j \le n\}.$$

Proof. The lemma gives us one direction of the implication. The second direction follows from Lemma 5 together with the fact that $X \to A \models X \twoheadrightarrow A$. □

Since there is no loss of generality in assuming **that** $3 \le n$, we get the desired reduction.

Theorem 6. The implication and finite implication problems for pjd's are unsolvable.

Proof. Let Σ,σ over U be given. By Lemma 8, $\Sigma \models_{(f)} \sigma$ iff

$$\{\tilde{\theta} : \theta \in \Sigma\} \cup \{A_i \to A_j : 0 \le i,j \le n\} \models_{(f)} \tilde{\sigma}.$$

By Lemma 5, the last implication holds iff

$$\{\tilde{\theta} : \theta \in \Sigma\} \cup \{\theta_{A_i \to A_j} : 0 \leq i,j \leq n\} \models_{(f)} \tilde{\sigma}.$$

By Lemma 9, this implication holds iff

$$\{\tilde{\theta} : \theta \in \Sigma\} \cup \{A_i \to\to A_j : 0 \leq i,j \leq n\} \models_{(f)} \tilde{\sigma}.$$

Since $\{\tilde{\theta} : \theta \in \Sigma\} \cup \{A_i \to\to A_j : 0 \leq i,j \leq n\}$ is a set of shallow td's and pjd's, and it can be constructed effectively, the claim follows. □

Analogously to the observation in Section 4, unsolvability of the finite implication problem for pjd's entails that the problem is not even partially solvable, and consequently there is no sound and complete formal system for finite implication of pjd's. In this observation, the only thing we assume about formal systems is that having a formal system for a problem renders it partially solvable.

We now make our notion of a formal system more precise. Most generally, what we mean by having a formal system for implication is that having an effective way of checking proofs. There is however a subtle point here. Unlike the case with td's where the universe is clear from the syntax, this is not the case with pjd's. In fact, pjd's are oblivious to the universe in a much stronger way. Let θ be the pjd $*[R_1, \ldots, R_k]_X$. We define $attr(\theta) = \bigcup_{i=1}^{k} R_i$, and for a set Σ of pjd's *we* define $attr(\Sigma) = \bigcup_{\theta \in \Sigma} attr(\theta)$. Now given a set Σ of pjd's and a pjd σ, the only thing we know about the universe is that it contains $attr(\Sigma \cup \{\sigma\})$. It follows that we can have different notions of implication, depending on the universe. That is, Σ (finitely) U-implies σ, denoted $\Sigma(U) \models_{(f)} \sigma$, if for all (finite) U-relations I we have that $I \models \Sigma$ entails $I \models \sigma$. Fortunately, all these "different" notions of implication turn out to be the same. We leave the easy proof of the following lemma to the reader.

Lemma 10. Let $\Sigma \cup \{\sigma\}$ be a set of pjd's. Then for all U such that $attr(\Sigma \cup \{\sigma\}) \subseteq U$ we have that $\Sigma(U) \models_{(f)} \sigma$ iff $\Sigma(attr(\Sigma \cup \{\sigma\})) \models_{(f)} \sigma$. □

Thus, we can go on using the notation $\Sigma \models \sigma$ without specifying the universe. However, when it comes to formal system the question pops up again. Do we want our formal system to handle proofs within fixed universes or not? We call a formal system of the first kind *universe- bounded.*

More precisely, a formal system for implication of pjd's is a recursive set Π whose elements are pairs $(\Sigma,\langle\sigma_1, \ldots, \sigma_k\rangle)$, where Σ is a set of pjd's, and $\sigma_1, \ldots, \sigma_k$ is a sequence without repetition of pjd's. The intended interpretation for Π is that $(\Sigma,\langle\sigma_1, \ldots, \sigma_k\rangle)\in\Pi$ when $\sigma_1, \ldots, \sigma_k$ is a proof that $\Sigma \models \sigma_k$. Thus, we say that Π is *sound* if $(\Sigma,\langle\sigma_1, \ldots, \sigma_k\rangle)\in\Pi$ entails that $\Sigma \models \sigma_k$, and we say that Π is *complete* if whenever Σ is a set of pjd's and σ is a pjd such that $\Sigma \models \sigma$ then there is a pair $(\Sigma,\langle\sigma_1, \ldots, \sigma_k\rangle)\in\Pi$ with $\sigma_k = \sigma$. If the formal system Π is universe-bounded then instead of pairs it consists of triples $(U,\Sigma,\langle\sigma_1, \ldots, \sigma_k\rangle)$, where U is a universe, Σ is a set of U-pjd's, and $\sigma_1, \ldots, \sigma_k$ is a sequence without repetition of U-pjd's. We say that Π is *sound* if $(U,\Sigma,\langle\sigma_1, \ldots, \sigma_k\rangle)\in\Pi$ entails that $\Sigma \models \sigma_k$, and we say that Π is *complete* if whenever Σ is a set of U-pjd's and σ is a U-pjd such that $\Sigma \models \sigma$ then there is a triple $(U,\Sigma,\langle\sigma_1, \ldots, \sigma_k\rangle)\in\Pi$ with $\sigma_k = \sigma$.

Theorem 7. There is no sound and complete universe-bounded formal system for pjd's.

Proof. The argument is essentially that of [BV3]. Suppose that Π is a sound and complete formal system for implication of pjd's. Let Σ be a set of pjd's, and let σ be a pjd. Take $U = attr(\Sigma \cup \{\sigma\})$. There are only finitely many U-pjd's, and therefore there are only finitely many triples $(U,\Sigma,\langle\sigma_1, \ldots, \sigma_k\rangle)$, where $\sigma_1, \ldots, \sigma_k$ is a sequence without repetition of U-pjd's with $\sigma_k = \sigma$. We can enumerate all these triples, and $\Sigma \models \sigma$ iff one of them is in Π. It follows that the implication problem for pjd's is solvable - contradiction. □

The crucial point in the proof, and the only property of pjd's used, is that there are only finitely many U-pjd's for any fixed U. Thus, the argument applies as well to any class of dependencies with that property.

Let $\theta = \langle w, I \rangle$ be a U-td. For any $A \in U$, we define $REP(\theta, A)$ is the set of repeating A -values in 8. That is,

$$REP(A) = \{u[A] : u \in I \text{ and either } u[A] = w[A] \text{ or } u[A] = v[A] \text{ for some } v \in I, v \neq u\}.$$

We say that θ is *k-simple* if for all A $\in U$ we have that $|REP(\theta, A)| \leq k$. Thus, the class of shallow td's is exactly the class of 1-simple td's. The generalized join dependencies of [Sc] are equivalent to 2-simple td's.

Sciore [Sc] has argued heuristically that one can not prove implication of k-simple td's without using $k+1$-simple td's, and conjectured that this is really the case. Since for every fixed U and k there are only finitely many k-simple U-td's, the argument in the proof of Theorem 7 shows that Sciore is right and there can be no sound and complete universe-bounded formal system for k-simple td's.

Two qualifications should be made. First, the proof of Theorem 7 relies on the unsolvability of the implication problem, and therefore does not apply to classes of dependencies for which the implication problem is solvable. Indeed, Sciore's conjecture that no class of td's that contain the class of total join dependencies but is properly contained in the class of td's has a sound and complete formal system is false. In [BV5] a universe-bounded formal system for total join dependencies is shown to be sound and complete. Secondly, the proof of Theorem 7 applies only to universe-bounded formal systems. Furthermore, since the reduction in this section shows us how to transform arbitrary td's to pjd's, it is not difficult to take a formal system for td's (see [BV4, SU]) and to transform it to a formal system for pjd's. The resulting system is of course not universe-bounded.

Theorem 8. There is a sound and complete formal system for pjd's. ❑

Question. Is there a sound and complete formal system for embedded join dependencies? For embedded multivalued dependencies?

7. Concluding Remarks

The solvability of the (finite) implication problem for embedded multivalued dependencies is one of the outstanding open question in dependency theory. One of the motivation to studying larger and larger classes of dependencies was the hope the the regularity of the more general classes, which in some senses are more natural then the narrower classes, would enable us to discover the elusive algorithm for deciding implication.

Unfortunately, a series of negative results shattered, more or less, that hope. First, in [BV1,CLM1] it was shown that the (finite) implication problem for untyped td's is unsolvable. Then in [BV2, CLM2, Val] unsolvability was shown also for typed tuple generating dependencies. Finally, here and in [GL2] unsolvability was extended to projected join dependencies. Projected join dependencies seem to be a very slight generalization of embedded join dependencies, and we believe that the unsolvability screw can be tightened that further. What about embedded multivalued dependencies? That question still haunts and baffles us.

REFERENCES

[ABU] Aho, A.V., Beeri, C., Ullman, J.D.: The theory of joins in relational databases. ACM Trans. on Database Systems 4(1979), pp. 297-314.

[ASU] Aho, A.V., Sagiv, Y., Ullman, J.D.: Equivalence among relational expressions. SIAM J. on Computing 8(1979), pp. 2 18-246.

[Bern] Bernstein, P.A.: Synthesizing third normal form relations from functional dependencies. ACM Trans. on Database Systems 1(1976), pp. 277-298.

[BMSU] Beeri, C., Mendelzon, A.O., Sagiv, Y, Ullman, J.D.: Equivalence of relational database schemes. SIAM J. on Comput. 10(1981), pp. 647-656.

[BR] Beeri, C., Rissanen, J.: Faithful representation of relational database sheemes. IBM Research Report, San Jose, 1980.

[BBG] Beeri, C., Bernstein, P.A., Goodman, N.: A sophisticates' introduction to database normalization theory. Proc. Int'l Conf. on VLDB, Berlin, 1978, pp.113-124.

[BV1] Beeri, C., Vardi, M.Y.: The implication problem for data dependencies. Proc. XP1 Workshop on Relational Database Theory, Stony Brook, June 1980. Also, Technical Report, Department of Computer Science, The Hebrew University of Jerusalem, May 1980.

[BV2] Beeri, C., Vardi, M.Y.: The implication problem for data dependencies. Proc. 8th ICALP, Acre, Israel, 1981, in Lecture Notes in Computer Science 115, Springer-Verlag, 1981, pp. 73-85.

[BV3] Beeri, C., Vardi, M.Y.: A proof procedure for data dependencies. Technical Report, Department of Computer Science, The Hebrew University of Jerusalem, Dec. 1980.

[BV4] Beeri, C., Vardi, M.Y.: Formal systems for tuple and equality generating dependencies. Technical Report, Department of Computer Science, The Hebrew University of Jerusalem, April 1981.

[BV5] Beeri, C., Vardi, M.Y.: Formal systems for join dependencies. Technical Report, Department of Computer Science, The Hebrew University of Jerusalem, April 1981.

[CLM1] Chandra, A.K., Lewis, H.R., Makowsky, J.A.: Embedded implicational dependencies and their inference problem. Proc. XP1 Workshop on Relational Database Theory, Stony Brook, June 1980.

[CLM2] Chandra, A.K., Lewis, H.R., Makowsky, J.A.: Embedded implicational dependencies and their inference problem. Proc. 13th ACM Ann. Symp. on Theory of Computing, 1981, pp. 342-354.

[Codd1] Codd, E.F.: A relational model for large shared data bases. Comm. of ACM 13(1970), pp. 377-387.

[Codd2] Codd, E.F.: Further normalization of the database relational model. In *Data Base Systems* (R. Rustin, ed.), Prentice-Hall, 1972, pp. 33-64.

[Codd3] Codd, E.F.: Extending the database relational model to capture more meaning. ACM Trans. on Database Systems 48(1980), pp. 397-434.

[Fag1] Fagin, R.: Multivalued dependencies and a new normal form for relational databases. ACM Trans. on Database Systems 2(1977), pp. 262-278.

[Fag2] Fagin, R.: Horn clauses and database dependencies. Proc. 12th ACM Ann. Symp. on Theory of Computing, 1980, pp. 123-134. To appear in JACM.

[FMUY] Fagin, R., Maier, D., Ullman, J.D., Yannakakis, M.: Tools for template dependencies. IBM Research Report RJ3033, May 1981. To appear in SIAM J. on Computing.

[GL1] Gurevich, Y., Lewis, H.R.: The inference problem for template dependencies. Proc. ACM Symp. on Principles of Database Systems, Los Angeles, 1982, pp. 221-229.

[GL2] Gurevich, Y., Lewis, H.R.: The word problem for cancellation semigroups with zero. Technical Report TR-08-82, Aiken Computation Lab, Harvard University, 1982.

[MMS] Maier, D., Mendelzon, A.O., Sagiv, Y.: Testing implications of data dependencies. ACM Trans. on Database Systems 2(1977), pp. 201-222.

[Ri] Rissanen, J.: Theory of relations for databases - a tutorial survey. Proc. 7th Symp. on Math. Found. of Computer Science, 1978, pp. 537-551.

[Ro] Rogers, H.: Theory of recursive functions and effective computability. McGraw-Hill, 1967.

[Sc] Sciore, E.: A complete axiomatization for full join dependencies. J. of ACM 29(1982), pp. 373-393.

[SU] Sadri, F., Ullman, J.D.: Template Dependencies - A large class of dependencies in relational databases and its complete axiomatization. J. of ACM 29(1982), pp. 363-372.

[Va1] Vardi, M.Y.: The implication problem for data dependencies in relational databases. Ph.D. Thesis (in Hebrew), The Hebrew University of Jerusalem, Sep. 1981.

[Va2] Vardi, M.Y.: Global decision problems for relational databases. Proc. 22nd IEEE Symp. on Foundation of Computer Science, Nashville, 1981, pp. 198-202.

[Va3] Vardi, M.Y.: The implication and the finite implication problems for typed template dependencies. Proc. ACM Symp. on Principles of Database Systems, Los Angeles, 1982, pp. 230-238.

[YP] Yannakakis, M., Papadimitriou, C.: algebraic dependencies. 21st IEEE Ann. Symp. on Found. of Computer Science, 1980, pp. 328-332. To appear in JCSS.

[Zan] Zaniolo, C.: Analysis and design of relational schemata for database systems. Technical Report UCLA-ENG-7769, UCLA, 1976.