

# Reasoning About Embedded Dependencies Using Inclusion Dependencies

Miika Hannula<sup>(✉)</sup>

Department of Mathematics and Statistics, University of Helsinki,  
P.O. Box 68, 00014 Helsinki, Finland  
`miika.hannula@helsinki.fi`

**Abstract.** The implication problem for the class of embedded dependencies is undecidable. However, this does not imply lackness of a proof procedure as exemplified by the chase algorithm. In this paper we present a complete axiomatization of embedded dependencies that is based on the chase and uses inclusion dependencies and implicit existential quantification in the intermediate steps of deductions.

**Keywords:** Axiomatization · Chase · Implication problem · Dependence logic · Embedded dependency · Tuple generating dependency · Equality generating dependency · Inclusion dependency

## 1 Introduction

Embedded dependencies generalize the concept database dependencies within the framework of first-order logic. Their implication is undecidable but however recursively enumerable, thus enabling complete axiomatizations. A standard example of such a proof procedure is the chase that was invented in the late 1970s [1, 2], and then soon extended to equality and tuple generating dependencies [3]. In this paper we present an axiomatization for the class of embedded dependencies that simulates the chase at the logical level using inclusion dependencies. In particular, completeness of the rules is obtained by constructing deductions in which all the intermediate steps are inclusion dependencies, except for the first and the last step. These inclusion dependencies consist of attributes of which some are new, i.e., such that they are not allowed to appear at any earlier stage of the deduction.

As a background example, consider the combined class of functional and inclusion dependencies. It is well known that the corresponding implication problem is undecidable, lacking hence finite axiomatization [4, 5]. One strategy in such situations has been to search for axiomatizations within a more general class of dependencies, and partly for this reason many different dependency notions were introduced in the 1980s. For instance, a textbook on dependency theory from 1991 considers more than 80 different dependency classes [6]. In [7] Mitchell proposed another strategy by presenting an axiomatization of functional and inclusion dependencies using a notion of new attributes which should

be thought of as implicitly existentially quantified. In this paper we take an analogous approach, and present an axiomatization for embedded dependencies where new attributes correspond to new values obtained from an associated chasing sequence. These attributes can be thought of as implicitly existentially quantified in the sense of *team semantics*, that is, a semantic framework that has *teams*, i.e., sets of assignments, as its underlying concept [8]. Team semantics is compositionally applicable to logics that extend first-order logic with various database dependencies [9, 10]. In this setting, *inclusion logic*, i.e., first-order logic with additional inclusion dependencies, captures the positive fragment of greatest fixed-point logic and hence all PTIME recognizable classes of finite, ordered models [11–13]. Therefore, inclusion dependencies with new attributes can be thought of as greatest fixed-point logic expressions. This may in part enable succinct intermediate steps in deductions in contrast to axiomatic systems that simulate the chase by composing first-order definable dependencies.

The methods described in this paper generalize the axiomatization of conditional independence and inclusion dependencies presented in [14]. It is also worth noting that extending relations with new attributes reminds of algebraic dependencies, that are, typed embedded dependencies defined in algebraic terms. The complete axiomatization of algebraic dependencies presented in [15] involves also an extension schema that introduces new copies of attributes.

## 2 Preliminaries

For two sets  $A$  and  $B$ , we write  $AB$  to denote their union, and for two sequences  $\mathbf{ab}$ , we write  $\mathbf{ab}$  to denote their concatenation. For a sequence  $\mathbf{a} = (a_1, \dots, a_n)$  and a mapping  $f$ , we write  $f(\mathbf{a})$  for  $(f(a_1), \dots, f(a_n))$ . We denote by  $\text{id}$  the identity function and by  $\text{pr}_i$  the function that maps a sequence to its  $i$ th projection. For a function  $f$  and  $A \subseteq \text{Dom}(f)$ , we write  $f|_A$  for the restriction of  $f$  to  $A$ , and for a set of mappings  $F$ , we write  $F|_A$  for  $\{f|_A : f \in F\}$ .

We start by fixing two countably infinite sets  $\text{Val}$  and  $\text{Att}$ , the first denoting possible values of relations and the second attributes. For notational convenience, we will assume that  $\text{Val} = \text{Att}$ . For  $R \subseteq \text{Att}$ , a *tuple* over  $R$  is a mapping  $R \rightarrow \text{Val}$ , and a *relation* over  $R$  is a set of tuples over  $R$ . We may sometimes write  $r[R]$  to denote that  $r$  is a relation over  $R$ . Values of a relation  $r$  over  $R$  are denoted by  $\text{Val}(r)$ , i.e.,  $\text{Val}(r) := \{t(A) : t \in r, A \in R\}$ . Let  $f$  be a *valuation*, i.e., a mapping  $\text{Val} \rightarrow \text{Val}$ . Then for a tuple  $t$ , we write  $f(t) := f \circ t$ , and for a relation  $r$ ,  $f(r) := \{f(t) : t \in r\}$ . A valuation  $f$  *embeds* a relation  $r$  (a tuple  $t$ ) to  $r'$  if  $f(r) \subseteq r'$  ( $f(t) \in r'$ ). Since we are usually interested only valuations of a relation, we say that  $f : \text{Val}(r) \rightarrow \text{Val}$  is a *valuation on  $r$* . For a valuation  $f$  on  $r$ , we say that  $g$  is an *extension* of  $f$  to another relation  $r'$  if  $g$  is a valuation on  $r'$  such that it agrees with  $f$  on values of  $\text{Val}(r) \cap \text{Val}(r')$ .

*Embedded dependencies* (ed's) can be written using first-order logic in the following way.

**Definition 1 (Embedded Dependency).** *Embedded dependency is a first-order sentence of the form*

$$\forall x_1, \dots, x_n (\phi(x_1, \dots, x_n) \rightarrow \exists z_1 \dots \exists z_k \psi(y_1, \dots, y_m))$$

where  $\{z_1, \dots, z_k\} = \{y_1, \dots, y_m\} \setminus \{x_1, \dots, x_n\}$  and

- $\phi$  is a (possibly empty) conjunction of relational atoms using all of the variables  $x_1, \dots, x_n$ ;
- $\psi$  is a conjunction of relational and equality atoms using all of the variables  $z_1, \dots, z_k$ ;
- there are no equality atoms in  $\psi$  involving existentially quantified variables.

If at most one relation symbol occurs in an ed, then we say that the ed is *unirelational*, and otherwise it is *multirelational*. An ed is called *typed* if there is an assignment of variables to column positions such that variables in relation atoms occur only in their assigned position, and each equality atom involves a pair of variables assigned to the same position. Otherwise we say that an ed is *untyped*. If  $\psi$  contains only one atom, then we say that the ed is *single-head*, and otherwise it is *multi-head*. A single-head ed where  $\psi$  is an equality is called an *equality generating dependency* (egd). If  $\psi$  is a conjunction of relational atoms, then the ed is called a *tuple generating dependency* (tgd). For notational simplicity, we restrict attention to unirelational ed's. It is easy to see that any ed is equivalent to a set of tgd's and egd's, and hence we restrict attention to ed's that belong to either of these subclasses.

The following alternative tableau presentation for egd's and tgd's are used in this paper.

**Definition 2.** *Let  $T$  and  $T'$  be finite relations over  $R$ , and  $x, y \in \text{Val}(T)$ . Then  $(T, x = y)$  and  $(T, T')$  are an egd and a tgd over  $R$ , respectively, with the below satisfaction relation for a relation  $r$  over  $S \supseteq R$ :*

- $r \models (T, x = y) \Leftrightarrow$  for all valuations  $f$  such that  $f(T) \subseteq r|_R$ , it holds that  $f(x) = f(y)$ .
- $r \models (T, T') \Leftrightarrow$  for all valuations  $f$  on  $T$  such that  $f(T) \subseteq r|_R$ , there is an extension  $g$  of  $f$  to  $T'$  such that  $g(T') \subseteq r|_R$ .

Sometimes we write  $\sigma[R]$  to denote that  $\sigma$  is a dependency over  $R$ . If  $T$  or  $T'$  is a singleton, then we may omit the set braces in the notation, e.g., write  $(T, t)$  instead of  $(T, \{t\})$ .

We also extend valuations to dependencies. For an egd  $\sigma = (T, x = y)$  we write  $\text{Val}(\sigma) = \text{Val}(T)$ , and for a tgd  $\tau = (T, T')$  we write  $\text{Val}(\sigma) = \text{Val}(T) \cup \text{Val}(T')$ . Moreover, if  $f$  is a valuation, then  $f(\sigma) = (f(T), f(x) = f(y))$  and  $f(\tau) = (f(T), f(T'))$ .

**Example 1.** Consider the relation  $r$  and the tgd's  $\sigma_1 := (\{t, t'\}, \{u\})$  and  $\sigma_2 := (\{t, t'\}, \{v, v'\})$  obtained from Fig. 1.<sup>1</sup> We notice that there are two valuations

<sup>1</sup> In a tableau presentation of a dependency  $\sigma$ , the distinct values of  $\sigma$  are sometimes denoted by blank cells.

	A	B	C
s <sub>0</sub>	0	1	2
s <sub>1</sub>	3	0	1
s <sub>2</sub>	2	3	0
s <sub>3</sub>	1	4	3

	A	B	C
t	x	y	z
t'		x	y
u	z		x

	A	B	C
t	x	y	z
t'		x	y
v	z	a	x
v'			a

**Fig. 1.** Relation  $r$  and tgd's  $\sigma_1, \sigma_2$

on  $\{t, t'\}$  that embed  $\{t, t'\}$  to  $r$ , namely  $f := \{(x, 0), (y, 1), (z, 2)\}$  and  $g := \{(x, 3), (y, 0), (z, 1)\}$ . Then  $r \models \sigma_1$  since  $f$  and  $g$  embed  $u$  into  $r$ , witnessed by tuples  $s_2$  and  $s_3$ , respectively. We also notice that  $r \not\models \sigma_2$  since, although  $f \cup \{(a, 3)\}$  embeds  $\{v, v'\}$  into  $r$ , no extension of  $g$  does the same.

Next we define inclusion dependencies which are examples of possibly untyped tgd's.

**Definition 3 (Inclusion Dependency).** *Let  $A_1, \dots, A_n$  and  $B_1, \dots, B_n$  be (not necessarily distinct) tuples of attributes. Then  $A_1 \dots A_n \subseteq B_1 \dots B_n$  is an inclusion dependency (ind) over  $R = \{A_i, B_i : i = 1, \dots, n\}$  with the following semantic rule for a relation  $r$  over  $S \supseteq R$ :*

$$r \models A_1 \dots A_n \subseteq B_1 \dots B_n \Leftrightarrow \forall s \in r \exists s' \in r \forall i = 1, \dots, n : s(A_i) = s'(B_i).$$

The axiomatization presented in the next section involves inclusion dependencies that introduce new attributes. These attributes are here interpreted as existentially quantified in *lax team semantics* sense [9]:

$$r \models \exists A \phi \Leftrightarrow r[f/A] \models \phi \text{ for some } f : r \rightarrow \mathcal{P}(\text{Val}) \setminus \{\emptyset\}, \quad (1)$$

where  $r[f/A] := \{t(x/A) : x \in f(A)\}$  and  $t(x/A)$  is the mapping that agrees with  $t$  everywhere except that it maps  $A$  to  $x$ . Interestingly, inclusion logic formulae with this concept of existential quantification can be characterized with positive greatest fixed-point logic formulae (see Theorem 15 in [11]).

### 3 Axiomatization

In this section we present an axiomatization for the class of all embedded dependencies. The axiomatization contains an identity rule and three rules for the chase. We also involve conjunction in the language and therefore incorporate its usual introduction and elimination rules in the definition. Regarding the equalities that appear in the rules, note that both  $AB \subseteq AA$  and  $AB \subseteq BB$  indicate that the values of  $A$  and  $B$  coincide in each row. Therefore, we use  $A = B$  to denote ind's of either form. For a tgd (an egd)  $\sigma$ , we say that  $x \in \text{Val}(\sigma)$  is *distinct* if it appears at most once as a value in  $\sigma$ . Namely,

- for a tgd  $\sigma = (T, T')[R]$ ,  $x$  is distinct if for all  $t, t' \in T \cup T'$  and  $A, B \in R$ , if  $t(A) = x = t'(B)$ , then  $t = t'$  and  $B = B'$ ;

- for an egd  $\sigma = (T, y = z)[R]$ ,  $x$  is distinct if  $x \notin \{y, z\}$  and for all  $t, t' \in T$  and  $A, B \in R$ , if  $t(A) = x = t'(B)$ , then  $t = t'$  and  $B = B'$ .

Lastly, note that in the following rules we assume that values can appear as attributes and vice versa.

**Definition 4.** *In addition to the below rules we adopt the usual introduction and elimination rules for conjunction. In the last three rules, we assume that  $\mathbf{A}$  is a sequence listing the attributes of  $R$ .*

*EE Equality Exchange:*

$$\text{if } A = B \wedge \sigma, \text{ then } \tau.$$

where  $\sigma$  is an ind and  $\tau$  is obtained from  $\sigma$  by replacing any number of occurrences of  $A$  by  $B$  and any number of occurrences of  $B$  by  $A$ .

*CS Chase Start:*

$$(T^*, \text{id})[RS] \wedge \bigwedge_{t \in T} t(\mathbf{A}) \subseteq \mathbf{A}$$

where  $T = T^*|_R$ ,  $S = \text{Val}(T)$  consists of new attributes, and  $R$  consists of distinct values.

*CR Chase Rule:*

$$\text{tgd: if } (T, T')[R] \wedge \bigwedge_{t \in T} f \circ t(\mathbf{A}) \subseteq \mathbf{A}, \text{ then } \bigwedge_{t' \in T'} f \circ t'(\mathbf{A}) \subseteq \mathbf{A},$$

$$\text{egd: if } (T, x = y)[R] \wedge \bigwedge_{t \in T} f \circ t(\mathbf{A}) \subseteq \mathbf{A}, \text{ then } f(x) = f(y),$$

where tgd:  $f$  is a valuation that it is 1-1 on  $\text{Val}(T') \setminus \text{Val}(T)$ , and  $f(x)$  is a new attribute for  $x \in \text{Val}(T') \setminus \text{Val}(T)$ .

*CT Chase Termination:*

$$\text{tgd: if } (T^*, \text{id})[RS] \wedge \bigwedge_{t' \in T'} u \circ t'(\mathbf{A}) \subseteq \mathbf{A}, \text{ then } (T, T')[R],$$

$$\text{egd: if } (T^*, \text{id})[RS] \wedge x = y, \text{ then } (T, x = y)[R],$$

where  $T = T^*|_R$ ,  $S = \text{Val}(T)$ , and  $\text{Val}(T^*|_S)$  consists of distinct values. Moreover, tgd:  $u$  is a mapping  $\text{Val}(T') \rightarrow \text{Att}$  that is the identity on  $\text{Val}(T) \cap \text{Val}(T')$ , and egd:  $x, y \in \text{Val}(T)$ .

For a dependency  $\sigma$  over  $R$ , we let  $\text{Att}(\sigma) := R$ , and for a set of dependencies  $\Sigma$ , we let  $\text{Att}(\Sigma) := \bigcup_{\sigma \in \Sigma} \text{Att}(\sigma)$ .

**Definition 5.** *A deduction from  $\Sigma$  is a sequence  $(\sigma_1, \dots, \sigma_n)$  such that:*

1. Each  $\sigma_i$  is either an element of  $\Sigma$ , an instance of [CS], or follows from one or more formulae of  $\{\sigma_1, \dots, \sigma_{i-1}\}$  by one of the rules presented above.
2. For each  $A \in \text{Att}(\sigma_i)$ , if  $A$  is new in  $\sigma_i$ , then  $A \notin \text{Att}(\Sigma \cup \{\sigma_1, \dots, \sigma_{i-1}\})$ , and otherwise  $A \in \text{Att}(\Sigma \cup \{\sigma_1, \dots, \sigma_{i-1}\})$ .

We say that  $\sigma$  is provable from  $\Sigma$ , written  $\Sigma \vdash \sigma$ , if there is a deduction  $(\sigma_1, \dots, \sigma_n)$  from  $\Sigma$  with  $\sigma = \sigma_n$  and such that no attributes in  $\sigma$  are new in  $\sigma_1, \dots, \sigma_n$ .

We will also use the following rules that are derivable from [EE]:

ES Equality Symmetry:

$$\text{if } A = B, \text{ then } B = A.$$

ET Equality Transitivity:

$$\text{if } A = B \wedge B = C, \text{ then } A = C.$$

One may find the chase rules slightly convoluted at first sight. However, the ideas behind the rules are relatively simple as illustrated in the following examples.

**Example 2 (Chase Start).** Let  $\sigma_0 := (\{t_0, t_1\}, \{u_0\})[RS]$  be as in Fig. 2, for  $R := \{A, B, C\}$  and  $S := \{x, y, z\}$ . Then

$\sigma_0 =$	$t_0$	$A$	$B$	$C$	$x$	$y$	$z$
	$t_1$	$x$	$y$	$z$			
	$u_0$				$x$	$y$	$z$

$\sigma_1 =$	$t_0$	$A$	$B$	$C$
	$t_1$	$x$	$y$	$z$
	$u_1$	$z$	$x$	

$\sigma_2 =$	$t_0$	$A$	$B$	$C$
	$t_1$	$x$	$y$	$z$
	$u_2$	$z$	$v$	
	$u_3$	$v$	$z$	

**Fig. 2.** Dependencies  $\sigma_0, \sigma_1, \sigma_2$

$$\tau := \sigma_0 \wedge xyz \subseteq ABC \wedge xy \subseteq BC$$

is an instance of [CS]. Here  $x, y, z$  are interpreted either as values or as new attributes. By the latter we intuitively mean that any relation  $r[ABC]$  can be extended to some  $r'[ABCxyz]$  such that  $r' \models \tau$ . For instance, one can define  $r' := q(r)$  where  $q$  is the following SPJR query

$$ABC \bowtie (\pi_{xyz}(\sigma_{xy=BC}(\rho_{xyz/ABC}(ABC) \bowtie ABC)))$$

where  $\sigma$  refers to (S)election,  $\pi$  to (P)rojection,  $\bowtie$  to (J)oin, and  $\rho$  to (R)ename operator. Then  $q(r)$  is a relation over  $RS$  such that its restriction to  $xyz$  lists all  $abc$  for which there exist  $s, s' \in r$  such that  $s(ABC) = abc$  and  $s'(BC) = ab$ . Let  $\sigma_1 = (\{t_0, t_1\}, \{u_1\})[R]$  be as in Fig. 2. Now,

$$r \models \sigma_1 \Leftrightarrow q(r) \models zx \subseteq AC.$$

Hence proving  $\Sigma \models \sigma_1$  reduces to showing that  $\Sigma \cup \{\tau\} \models zx \subseteq AC$ .

**Example 3 (Chase Rule).** Assume

$$\sigma_2 \wedge xyz \subseteq ABC \wedge xy \subseteq BC \quad (2)$$

where  $\sigma_2 = (\{t_0, t_1\}, \{u_2, u_3\})[R]$  is as in Fig. 2, for  $R := \{A, B, C\}$ . Then, interpreting  $f$  as id, one can derive with one application of [CR]

$$zv \subseteq AC \wedge vz \subseteq AC \quad (3)$$

from (2). Note that in (3)  $v$  is interpreted as a *new* attribute, and the idea is that any relation  $r[R]$  satisfying (2) and with  $v \notin R$  can be extended to a relation  $r'[R \cup \{v\}]$  satisfying (3) by introducing suitable values for  $v$ .

**Example 4 (Chase Termination).** Assume

$$\sigma_0 \wedge zx \subseteq AC \quad (4)$$

where  $\sigma_0 = (\{t_0, t_1\}, \{u_0\})[RS]$  is as in Fig. 2, for  $R := \{A, B, C\}$  and  $S := \{x, y, z\}$ . Then, letting  $u = \text{id}$ , one can derive  $\sigma_1$  as in Fig. 2 from (4) with one application of [CT].

## 4 Soundness Theorem

In this section we show that the axiomatization presented in the previous section is sound. First note that the next lemma follows from the definitions of egd's, tgd's and ind's.

**Lemma 1.** *Let  $\sigma$  be a dependency over  $R$ , and let  $r$  and  $r'$  be relations over supersets of  $R$  and with  $r|_R = r'|_R$ . Then  $r \models \sigma \Leftrightarrow r' \models \sigma$ .*

Then we prove the following lemma which implies soundness of the axioms. For attribute sets  $R, R'$  with  $R \subseteq R'$  and a relation  $r$  over  $R$ , we say that a relation  $r'$  over  $R'$  is an extension of  $r$  to  $R'$  if  $r'|_R = r$ . Recall from Eq. 1 that exactly such extensions are used in the existential quantification of lax team semantics.

**Lemma 2.** *Let  $r$  be a relation over  $\text{Att}(\Sigma)$  such that  $r \models \Sigma$ , and let  $(\sigma_1, \dots, \sigma_n)$  be a deduction from  $\Sigma$ . Then there exists an extension  $r'$  of  $r$  to  $\text{Att}(\Sigma \cup \{\sigma_1, \dots, \sigma_n\})$  such that  $r' \models \Sigma \cup \{\sigma_1, \dots, \sigma_n\}$ .*

*Proof.* We prove the claim by induction on  $n$ . We denote by  $R_n$  the set  $\text{Att}(\Sigma \cup \{\sigma_1, \dots, \sigma_n\})$ . Assuming the claim for  $n-1$ , we first find an extension  $r_{n-1}$  of  $r$  to  $R_{n-1}$  such that  $r_{n-1} \models \Sigma \cup \{\sigma_1, \dots, \sigma_{n-1}\}$ . If  $\sigma_n$  is obtained by an application of a conjunction or some ind rule, then it is easy to see that we may choose  $r_n := r_{n-1}$ . Hence, it suffices to consider the cases where  $\sigma_n$  is obtained by using one of the chase rules. Due to Lemma 1, it suffices to find an extension  $r_n$  of  $r_{n-1}$  to  $R_n$  such that  $r_n \models \sigma_n$ . In the following cases,  $\mathbf{A}$  denotes a sequence listing the attributes of  $R \subseteq R_{n-1}$ .

**Case [CS].** Assume that  $\sigma_n$  is obtained by [CS] and is of the form

$$(T^*, \text{id})[RS] \wedge \bigwedge_{t \in T} t(\mathbf{A}) \subseteq \mathbf{A}$$

where  $T = T^*|_R$ ,  $S = \text{Val}(T)$  consists of *new attributes* and  $R$  of *distinct values*. Let  $r_n := r_{n-1} \bowtie r$  be an extension of  $r_{n-1}$  to  $R_n = R_{n-1}S$ , where

$$r := \{h : h \text{ is a valuation on } T \text{ such that } h(T) \subseteq r_{n-1}|_R\}.$$

We claim that  $r_n \models \sigma_n$ . Consider the first conjunct of  $\sigma_n$ , and let  $h$  be a valuation on  $T^*$  such that  $h(T^*) \subseteq r_n|_{RS}$ . Then  $h|_S$  is a valuation on  $T$  such that  $h(T) \subseteq r_n|_R = r_{n-1}|_R$ , i.e.,  $h|_S = t_0|_S$  for some  $t_0 \in r_n$ . Since  $R$  consists of *distinct values* and thus  $R \cap \text{Dom}(h) = \emptyset$ , we may define  $h'$  as an extension of  $h$  with  $A \mapsto t_0(A)$ , for  $A \in R$ . Then  $h'|_{RS} = t_0|_{RS} \in r_n|_{RS}$ , and therefore  $r_n \models (T^*, \text{id})[RS]$ .

Consider then  $t(\mathbf{A}) \subseteq \mathbf{A}$ , for  $t \in T$ , and let  $t_0 \in r_n$ . By the definition,  $t_0|_S = h$  for some valuation  $h$  on  $T$  such that  $h(T) \subseteq r_n|_R$ , and hence we obtain that  $t_0 \circ t(\mathbf{A}) = h \circ t(\mathbf{A}) = t_1(\mathbf{A})$  for some  $t_1 \in r_n$ . Therefore,  $r_n \models t(\mathbf{A}) \subseteq \mathbf{A}$ .

**Case [CR].** Assume that  $\sigma_n$  is of the form (i)  $\bigwedge_{t' \in T'} f \circ t'(\mathbf{A}) \subseteq \mathbf{A}$  or (ii)  $f(x) = f(y)$ , and is obtained by [CR] from

- (i)  $(T, T')[R] \wedge \bigwedge_{t \in T} f \circ t(\mathbf{A}) \subseteq \mathbf{A}$ ,
- (ii)  $(T, x = y)[R] \wedge \bigwedge_{t \in T} f \circ t(\mathbf{A}) \subseteq \mathbf{A}$ ,

where in case (ii)  $f$  is a valuation on  $T \cup T'$  such that it is 1-1 on  $S := \text{Val}(T') \setminus \text{Val}(T)$  and  $f(x)$  is a *new attribute* for  $x \in S$ . Let  $s \in r_{n-1}$ . Since  $r_{n-1} \models \bigwedge_{t \in T} f \circ t(\mathbf{A}) \subseteq \mathbf{A}$ , we first obtain that  $s \circ f(T) \subseteq r_{n-1}|_R$ .

- (i) Since  $r_{n-1} \models (T, T')[R]$  we find a mapping  $g : S \rightarrow \text{Val}$  such that  $h(T') \subseteq r_{n-1}|_R$ , for  $h = g \cup (s \circ f)$ . Since  $f$  is 1-1 on  $S$ , we can now define  $r_n$  as the relation obtained from  $r_{n-1}$  by extending each  $s \in r_{n-1}$  with  $f(x) \mapsto g(x)$  for  $x \in S$ . Then for each  $s \in r_n$ ,  $s \circ f(T') \subseteq r_n|_R$ , and hence we obtain that  $r_n \models \bigwedge_{t' \in T'} f \circ t'(\mathbf{A}) \subseteq \mathbf{A}$ .
- (ii) It suffices to show that  $r_{n-1} \models f(x) = f(y)$ . Since  $s \circ f(x) = s \circ f(y)$  by  $r_{n-1} \models (T, x = y)[R]$ , this follows immediately.

**Case [CT].** Assume that  $\sigma_n$  is of the form (i)  $(T, T')[R]$  or (ii)  $(T, x = y)[R]$  and is obtained by [CT] from

- (i)  $(T^*, \text{id})[RS] \wedge \bigwedge_{t' \in T'} u \circ t'(\mathbf{A}) \subseteq \mathbf{A}$ , where  $u$  is a mapping  $\text{Val}(T') \rightarrow \text{Att}$  that is the identity on  $\text{Val}(T) \cap \text{Val}(T')$ ,
- (ii)  $(T^*, \text{id})[RS] \wedge x = y$ , where  $x, y \in \text{Val}(T)$ .

Moreover, in both cases  $T = T^*|_R$ ,  $S = \text{Val}(T)$ , and  $\text{Val}(T^*|_S)$  consists of *distinct values*. It suffices to show that  $r_{n-1} \models \sigma_n$ , so let  $h$  be a valuation on  $T$  such that  $h(T) \subseteq r_{n-1}|_R$ . Since  $\text{Val}(T^*|_S)$  consists of distinct values,  $h$  can be extended to a valuation  $h'$  on  $T^*$  such that  $h'(T^*) \subseteq r_{n-1}|_{RS}$ . Since  $r_{n-1} \models (T^*, \text{id})[RS]$ , there is an extension  $h''$  of  $h'$  to attributes in  $R$  such that  $h''|_{RS} \in r_{n-1}|_{RS}$ . Hence, we obtain that  $h|_S \in r_{n-1}|_S$ . Let then  $s \in r_{n-1}$  be such that it agrees with  $h$  on  $S$ .



- (i) Since  $r_{n-1} \models \bigwedge_{t' \in T'} u \circ t'(\mathbf{A}) \subseteq \mathbf{A}$ , we obtain that  $s \circ u(T') \subseteq r_{n-1}|_R$ . Moreover, we notice that  $s \circ u = h$  on  $\text{Val}(T) \cap \text{Val}(T')$ .
- (ii) Since  $r_{n-1} \models x = y$ , we obtain that  $s(x) = s(y)$ . Then  $h(x) = h(y)$  since  $x, y \in S$ .

Hence, in both cases we obtain that  $r_{n-1} \models \sigma_n$ . This concludes the [CT] case and the proof.  $\square$

Using the previous lemma, soundness of the rules follows.

**Theorem 1.** *Let  $\Sigma \cup \{\sigma\}$  be a finite set of egd's and tgd's over  $R$ . Then  $\Sigma \models \sigma$  if  $\Sigma \vdash \sigma$ .*

*Proof.* Let  $r$  be a relation such that  $r \models \Sigma$ , and assume that  $(\sigma_1, \dots, \sigma_n)$  is a deduction from  $\Sigma$  where  $\sigma = \sigma_n$  contains no attributes that appear as new in  $\sigma_1, \dots, \sigma_n$ . If  $R' := \text{Att}(\Sigma \cup \{\sigma_1, \dots, \sigma_n\})$ , then by Lemma 2 we find an extension  $r'$  of  $r|_R$  to  $R'$  such that  $r' \models \sigma$ . Then using Lemma 1 we obtain that  $r \models \sigma$ .  $\square$

## 5 Chase Revisited

In this section we define the chase for the class of egd's and tgd's. The chase algorithm was generalized to typed egd's and tgd's in [3], and here we present the chase using notation similar to that in [16]. First let us assume, for notational convenience, that there is a total, well-founded order  $<$  on the set  $\text{Val}$ , e.g.,  $x_1 < x_2 < x_3 < \dots$  for  $\text{Val} = \{x_1, x_2, x_3, \dots\}$ . Let  $\Sigma \cup \{\sigma\}$  be a set of egd's and tgd's over  $R$ . A *chasing sequence* of  $\sigma$  over  $\Sigma$  is a (possibly infinite) sequence  $\sigma_0, \sigma_1, \dots, \sigma_n, \dots$  where  $\sigma_0 = \sigma$ , and  $\sigma_{n+1}$  is obtained from  $\sigma_n$ , with  $T := \text{pr}_1(\sigma_n)$ , according to either of the following rules.

Let  $\tau \in \Sigma$  be of the form  $(S, x = y)$ , and suppose that there is a valuation  $f$  on  $S$  such that  $f(S) \subseteq T$  but  $f(x) \neq f(y)$ . Then  $\tau$  (and  $f$ ) can be applied to  $\sigma_n$  as follows:

- **egd rule:** Let  $\sigma_{n+1} := g(\sigma_n)$  where  $g : \text{Val} \rightarrow \text{Val}$  is the identity everywhere except that it maps  $f(y)$  to  $f(x)$  if  $f(x) < f(y)$ , and  $f(x)$  to  $f(y)$  if  $f(y) < f(x)$ .

Let  $\tau \in \Sigma$  be of the form  $(S, S')$ , and suppose that there is a valuation  $f$  on  $S$  such that  $f(S) \subseteq T$ , but there exists no extension  $f'$  of  $f$  to  $S'$  such that  $f(S') \subseteq T$ . Then  $\tau$  can be applied to  $\sigma_n$  as follows:

- **tgd rule:** List all  $f_1, \dots, f_n$  that have the above property, and for each  $f_i$  choose a *distinct extension* to  $S'$ , i.e., an extension  $f'_i$  to  $S'$  such that each variable in  $\text{Val}(S') \setminus \text{Val}(S)$  is assigned a distinct new value greater than any value in  $\text{Val}(\sigma_0) \cup \dots \cup \text{Val}(\sigma_n)$ . Moreover, no new value is assigned by two  $f'_i, f'_j$  where  $i \neq j$ . Then we let  $\sigma_{n+1} : (T \cup f'_1(S') \cup \dots \cup f'_m(S'), \text{pr}_2(\sigma_n))$ .

Construction of a chasing sequence is restricted with the following two conditions:

- (i) Whenever an egd is applied, it is applied repeatedly until it is no longer applicable.
- (ii) No dependency is starved, i.e., each dependency that is applicable infinitely many times is applied infinitely many times.

Let  $\overline{(\Sigma, \sigma)} = \sigma_0, \sigma_1, \dots$  be a chasing sequence of  $\sigma$  over  $\Sigma$ . Due to the possibility of applying egd's, a chasing sequence may not be monotone with respect to  $\subseteq$ . Hence, depending on whether  $\sigma$  is a tgd or an egd, we define

- egd:  $\text{chase}(\overline{(\Sigma, \sigma)}) := (T^1, x = y)$ ,
- tgd:  $\text{chase}(\overline{(\Sigma, \sigma)}) := (T^1, T^2)$ ,

where  $T^i := \{u : \exists m \forall n \geq m (u \in \text{pr}_i(\sigma_n))\}$  and  $x = y$  is  $\text{pr}_2(\sigma_n)$  for  $n \in \mathbb{N}$  such that  $\text{pr}_2(\sigma_n) = \text{pr}_2(\sigma_m)$  for all  $m \geq n$ . Note that “newer” values introduced by the tgd rule are always greater than the “older” ones, and values may only be replaced with smaller ones. Hence, no value can change infinitely often, and therefore  $\text{chase}(\overline{(\Sigma, \sigma)})$  is always well defined and non-empty.

We also associate each chasing sequence with the following descending valuations  $\rho_n$ , for  $n \geq 0$ . We let  $\rho_0 = \text{id}$ ,  $\rho_{n+1} = g \circ \rho_n$  if  $\sigma_{n+1}$  is obtained by an application of the egd rule where  $\sigma_{n+1} = g(\sigma_n)$ , and  $\rho_{n+1} = \text{id} \circ \rho_n$  otherwise. We then define  $\rho(x) = \lim_{n \rightarrow \infty} \rho_n(x)$ , i.e.,  $\rho(x) = \rho_n(x)$  if  $n \in \mathbb{N}$  such that  $\rho_m(x) = \rho_n(x)$  for all  $m \geq n$ . Then we obtain that

$$\text{chase}(\overline{(\Sigma, \sigma)}) = \bigcup_{n=0}^{\infty} \rho(\sigma_n).$$

A dependency  $\tau$  is *trivial* if

- $\tau$  is of the form  $(T, x = x)$ , or
- $\tau$  is of the form  $(T, T')$  and there is a valuation  $f$  on  $T'$  such that  $f$  is the identity on  $\text{Val}(T) \cap \text{Val}(T')$  and  $f(T') \subseteq T$ .

It is well-known that the chase algorithm captures unrestricted implication of dependencies. For the proof of the following proposition, see Appendix of the arXiv version of the paper [17].

**Proposition 1.** *Let  $\Sigma \cup \{\sigma\}$  be a set of egd's and tgd's over  $R$ . Then the following are equivalent:*

- (i)  $\Sigma \models \sigma$ ,
- (ii) there is a chasing sequence  $\overline{(\Sigma, \sigma)} = \sigma_0, \sigma_1, \dots$  of  $\sigma$  over  $\Sigma$  such that  $\text{chase}(\overline{(\Sigma, \sigma)})$  is trivial,
- (iii) there is a chasing sequence  $\overline{(\Sigma, \sigma)} = \sigma_0, \sigma_1, \dots$  of  $\sigma$  over  $\Sigma$  such that  $\sigma_n$  is trivial, for some  $n$ .

## 6 Completeness Theorem

In this section we show that the rules presented in Definition 4 are complete for the implication problem of embedded dependencies. Let us first illustrate the use of the axioms in the following simple example.

**Example 5.** Consider the implication problem  $\{\sigma, \sigma'\} \models \tau$  where  $\sigma, \sigma', \tau$  are illustrated in Fig. 3, e.g.,  $\sigma = (T, t)$  where  $T$  consists of the top two rows of  $\sigma$  and  $t$  is the bottom row. Note that  $\sigma$  and  $\tau$  are embedded multivalued dependencies of the form  $A \twoheadrightarrow B|C$  and  $A \twoheadrightarrow B|CD$ , respectively, and  $\sigma'$  is a functional dependency of the form  $C \rightarrow D$ . It is easy to see that the implication holds, and this can be also verified by a chasing sequence  $\tau_0, \tau_1, \tau_2$  of  $\tau$  over  $\{\sigma, \sigma'\}$  where  $\tau_2$  is trivial (Fig. 4). In the chasing sequence,  $\tau_0 = \tau$  and  $\tau_1$  is the result of applying  $\sigma$  to  $\tau_0$ . For this, note that there exists two valuations on  $T$  that embed  $T$  to  $\text{pr}_1(\tau_0)$  but has no extension that embeds  $t$  into  $\text{pr}_1(\tau_0)$ . These valuations are the identity and the function  $f$  that swaps the values of the top and bottom row of  $T$ . Then  $\tau_1$  is obtained by adding to  $\text{pr}_1(\tau_0)$   $\text{id}^*(t)$  and  $f^*(t)$  where  $\text{id}^*$  and  $f^*$  are distinct extensions of  $\text{id}$  and  $f$  to  $t$ , e.g.,  $\text{id}^* = \text{id}$  also on  $d_2$  and  $f^*$  maps  $d_2$  to  $d_3$ . Also,  $\tau_2$  is the result of applying  $\sigma'$  to  $\tau_1$  two times, i.e.,  $\tau_2$  is obtained from  $\tau_1$  by replacing  $d_3$  with  $d_0$  and  $d_2$  with  $d_1$ . Clearly  $\tau_2$  is trivial, and hence we obtain the claim by Proposition 1.

$\sigma =$	<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td><math>A</math></td><td><math>B</math></td><td><math>C</math></td><td><math>D</math></td></tr> <tr><td><math>a_0</math></td><td><math>b_0</math></td><td><math>c_0</math></td><td><math>d_0</math></td></tr> <tr><td><math>a_0</math></td><td><math>b_1</math></td><td><math>c_1</math></td><td><math>d_1</math></td></tr> <tr><td><math>a_0</math></td><td><math>b_0</math></td><td><math>c_1</math></td><td><math>d_2</math></td></tr> </table>	$A$	$B$	$C$	$D$	$a_0$	$b_0$	$c_0$	$d_0$	$a_0$	$b_1$	$c_1$	$d_1$	$a_0$	$b_0$	$c_1$	$d_2$
$A$	$B$	$C$	$D$														
$a_0$	$b_0$	$c_0$	$d_0$														
$a_0$	$b_1$	$c_1$	$d_1$														
$a_0$	$b_0$	$c_1$	$d_2$														

$\sigma' =$	<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td><math>A</math></td><td><math>B</math></td><td><math>C</math></td><td><math>D</math></td></tr> <tr><td><math>a_0</math></td><td><math>b_0</math></td><td><math>c_0</math></td><td><math>d_0</math></td></tr> <tr><td><math>a_1</math></td><td><math>b_1</math></td><td><math>c_0</math></td><td><math>d_1</math></td></tr> <tr><td></td><td></td><td><math>d_0 = d_1</math></td><td></td></tr> </table>	$A$	$B$	$C$	$D$	$a_0$	$b_0$	$c_0$	$d_0$	$a_1$	$b_1$	$c_0$	$d_1$			$d_0 = d_1$	
$A$	$B$	$C$	$D$														
$a_0$	$b_0$	$c_0$	$d_0$														
$a_1$	$b_1$	$c_0$	$d_1$														
		$d_0 = d_1$															

$\tau =$	<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td><math>A</math></td><td><math>B</math></td><td><math>C</math></td><td><math>D</math></td></tr> <tr><td><math>a_0</math></td><td><math>b_0</math></td><td><math>c_0</math></td><td><math>d_0</math></td></tr> <tr><td><math>a_0</math></td><td><math>b_1</math></td><td><math>c_1</math></td><td><math>d_1</math></td></tr> <tr><td><math>a_0</math></td><td><math>b_0</math></td><td><math>c_1</math></td><td><math>d_1</math></td></tr> </table>	$A$	$B$	$C$	$D$	$a_0$	$b_0$	$c_0$	$d_0$	$a_0$	$b_1$	$c_1$	$d_1$	$a_0$	$b_0$	$c_1$	$d_1$
$A$	$B$	$C$	$D$														
$a_0$	$b_0$	$c_0$	$d_0$														
$a_0$	$b_1$	$c_1$	$d_1$														
$a_0$	$b_0$	$c_1$	$d_1$														

Fig. 3. Dependencies  $\sigma, \sigma', \tau$

This procedure can now be simulated with our axioms as follows. First, with one application of [CS] we derive

$$(T, \text{id})[RS] \wedge a_0 b_0 c_0 d_0 \subseteq ABCD \wedge a_0 b_1 c_1 d_1 \subseteq ABCD$$

where  $T = \{t, t'\}$ ,  $R = \{A, B, C, D\}$ , and  $S = \{a_0, b_0, b_1, c_0, c_1, d_0, d_1\}$  is a set of values that are interpreted as new attributes. Here  $t(x)$  and  $t'(x)$ , for  $x \in S$ ,

$\tau_0 =$	<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td><math>A</math></td><td><math>B</math></td><td><math>C</math></td><td><math>D</math></td></tr> <tr><td><math>a_0</math></td><td><math>b_0</math></td><td><math>c_0</math></td><td><math>d_0</math></td></tr> <tr><td><math>a_0</math></td><td><math>b_1</math></td><td><math>c_1</math></td><td><math>d_1</math></td></tr> <tr><td><math>a_0</math></td><td><math>b_0</math></td><td><math>c_1</math></td><td><math>d_1</math></td></tr> </table>	$A$	$B$	$C$	$D$	$a_0$	$b_0$	$c_0$	$d_0$	$a_0$	$b_1$	$c_1$	$d_1$	$a_0$	$b_0$	$c_1$	$d_1$
$A$	$B$	$C$	$D$														
$a_0$	$b_0$	$c_0$	$d_0$														
$a_0$	$b_1$	$c_1$	$d_1$														
$a_0$	$b_0$	$c_1$	$d_1$														

$\tau_1 =$	<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td><math>A</math></td><td><math>B</math></td><td><math>C</math></td><td><math>D</math></td></tr> <tr><td><math>a_0</math></td><td><math>b_0</math></td><td><math>c_0</math></td><td><math>d_0</math></td></tr> <tr><td><math>a_0</math></td><td><math>b_1</math></td><td><math>c_1</math></td><td><math>d_1</math></td></tr> <tr><td><math>a_0</math></td><td><math>b_0</math></td><td><math>c_1</math></td><td><math>d_2</math></td></tr> <tr><td><math>a_0</math></td><td><math>b_1</math></td><td><math>c_0</math></td><td><math>d_3</math></td></tr> <tr><td><math>a_0</math></td><td><math>b_0</math></td><td><math>c_1</math></td><td><math>d_1</math></td></tr> </table>	$A$	$B$	$C$	$D$	$a_0$	$b_0$	$c_0$	$d_0$	$a_0$	$b_1$	$c_1$	$d_1$	$a_0$	$b_0$	$c_1$	$d_2$	$a_0$	$b_1$	$c_0$	$d_3$	$a_0$	$b_0$	$c_1$	$d_1$
$A$	$B$	$C$	$D$																						
$a_0$	$b_0$	$c_0$	$d_0$																						
$a_0$	$b_1$	$c_1$	$d_1$																						
$a_0$	$b_0$	$c_1$	$d_2$																						
$a_0$	$b_1$	$c_0$	$d_3$																						
$a_0$	$b_0$	$c_1$	$d_1$																						

$\tau_2 =$	<table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td><math>A</math></td><td><math>B</math></td><td><math>C</math></td><td><math>D</math></td></tr> <tr><td><math>a_0</math></td><td><math>b_0</math></td><td><math>c_0</math></td><td><math>d_0</math></td></tr> <tr><td><math>a_0</math></td><td><math>b_1</math></td><td><math>c_1</math></td><td><math>d_1</math></td></tr> <tr><td><math>a_0</math></td><td><math>b_0</math></td><td><math>c_1</math></td><td><math>d_1</math></td></tr> <tr><td><math>a_0</math></td><td><math>b_1</math></td><td><math>c_0</math></td><td><math>d_0</math></td></tr> <tr><td><math>a_0</math></td><td><math>b_0</math></td><td><math>c_1</math></td><td><math>d_1</math></td></tr> </table>	$A$	$B$	$C$	$D$	$a_0$	$b_0$	$c_0$	$d_0$	$a_0$	$b_1$	$c_1$	$d_1$	$a_0$	$b_0$	$c_1$	$d_1$	$a_0$	$b_1$	$c_0$	$d_0$	$a_0$	$b_0$	$c_1$	$d_1$
$A$	$B$	$C$	$D$																						
$a_0$	$b_0$	$c_0$	$d_0$																						
$a_0$	$b_1$	$c_1$	$d_1$																						
$a_0$	$b_0$	$c_1$	$d_1$																						
$a_0$	$b_1$	$c_0$	$d_0$																						
$a_0$	$b_0$	$c_1$	$d_1$																						

Fig. 4. Chasing sequence  $\tau_0, \tau_1, \tau_2$

	$A$	$B$	$C$	$D$	$a_0$	$b_0$	$b_1$	$c_0$	$c_1$	$d_0$	$d_1$
$t$	$a_0$	$b_0$	$c_0$	$d_0$							
$t'$	$a_0$	$b_1$	$c_1$	$d_1$							
id					$a_0$	$b_0$	$b_1$	$c_0$	$c_1$	$d_0$	$d_1$

**Fig. 5.**  $(T, \text{id})[RS]$ 

and  $A, B, C, D$  are interpreted as distinct values.  $(T, t)[RS]$  is illustrated in Fig. 5 where all the distinct values are hidden. Now with one application of [CR], letting  $f = \text{id}$ , we derive  $a_0b_0c_1d_2 \subseteq ABCD$  from

$$\sigma \wedge a_0b_0c_0d_0 \subseteq ABCD \wedge a_0b_1c_1d_1 \subseteq ABCD \quad (5)$$

Note that in this step,  $d_2$  is interpreted as a new attribute. Let then  $f$  be the valuation that is the identity on  $a_0, b_0, b_1, d_1$ , and otherwise maps  $a_1 \mapsto a_0$ ,  $c_0 \mapsto c_1$ , and  $d_0 \mapsto d_2$ . We notice that  $f(a_0b_0c_0d_0) = a_0b_0c_1d_2$  and  $f(a_1b_1c_0d_1) = a_0b_1c_1d_1$ . Hence, we may derive with one application of [CR]  $f(d_0) = f(d_1)$ , i.e.,  $d_2 = d_1$  from

$$\sigma' \wedge f(a_0b_0c_0d_0) \subseteq ABCD \wedge f(a_1b_1c_0d_1) \subseteq ABCD.$$

Then we apply [EE] and derive  $a_0b_0c_1d_1 \subseteq ABCD$  from

$$d_2 = d_1 \wedge a_0b_0c_1d_2 \subseteq ABCD$$

Finally, we may apply [CT] and derive  $\tau$  from  $(T, \text{id})[RS] \wedge a_0b_0c_1d_1 \subseteq ABCD$ .

The following lemma shows that the above technique extends to all chasing sequences. The proof is straightforward and hence omitted here (see Appendix of the arXiv version of the paper [17]).

**Lemma 3.** *Let  $(\overline{\Sigma}, \sigma) = \sigma_0, \sigma_1, \dots$  be a chasing sequence of  $\sigma$  over  $\Sigma$ , where  $\Sigma \cup \{\sigma\}$  is a finite set of egd's and tgd's over  $R$ , let  $\mathbf{A}$  be a sequence listing the attributes of  $R$ , let  $T := \text{pr}_1(\sigma)$  and  $T_i := \text{pr}_1(\sigma_i)$ , and let  $n \in \mathbb{N}$ . Then there exists a deduction from  $\Sigma$ , with attributes from  $R \cup \bigcup_{i \in \mathbb{N}} \text{Val}(T_i)$ , listing the following dependencies:*

- (i)  $(T^*, \text{id})[RS]$  where  $T^*|_R = T$ ,  $S = \text{Val}(T)$ , and  $T^*|_S$  consists of distinct values,
- (ii)  $f(x) = f(y)$ , for each application of  $(S, x = y)$  and  $f$  to  $\sigma_m$ , for  $m < n$ ,
- (iii)  $t(\mathbf{A}) \subseteq \mathbf{A}$ , for  $t \in T_m$  where  $m \leq n$ .

With the lemma, we can now show completeness.

**Theorem 2.** *Let  $\Sigma \cup \{\sigma\}$  be a finite set of egd's and tgd's over  $R$ . Then  $\Sigma \models \sigma \Leftrightarrow \Sigma \vdash \sigma$ .*

*Proof.* Assume that  $\Sigma \models \sigma$ , and let  $\mathbf{A}$  be a sequence listing  $R$ . Then by Proposition 1 there is a chasing sequence  $(\overline{\Sigma}, \overline{\sigma}) = \sigma_0, \sigma_1, \dots$  of  $\sigma$  over  $\Sigma$  such that  $\sigma_n$  is trivial for some  $n$ . Let  $D = (\tau_1, \dots, \tau_l)$  be a deduction from  $\Sigma$  obtained by Lemma 3, and let  $T := \text{pr}_1(\sigma)$  and  $T_i := \text{pr}_1(\sigma_i)$ .

Assume first that  $\sigma$  is an egd of the form  $(T, x = y)$ . Then  $\sigma_n$  is  $(T_n, z = z)$  where  $z = \rho_n(x) = \rho_n(y)$ . Now, either  $\rho_{i+1}(x)$  is  $\rho_i(x)$ , or the equality  $\rho_{i+1}(x) = \rho_i(x)$  (or its reverse) is listed in  $D$  by item (ii). Hence, using repeatedly [ES,ET] we may further on derive  $z = x$ . Since  $z = y$  is derivable analogously, we therefore obtain  $x = y$  by [ES,ET]. Then with one application of [CT], we derive  $(T, x = y)$  from  $(T^*, \text{id})[RS] \wedge x = y$  where  $T^*|_R = T$ . Note that the  $(T^*, \text{id})[RS]$  of the correct form is listed in  $D$  by item (i) of Lemma 3.

Assume then that  $\sigma$  is a tgd of the form  $(T, T')$ , and let  $T'_i := \text{pr}_2(\sigma_i)$ . Then  $\sigma_n$  is  $(T_n, T'_n)$ , and there is a valuation  $f$  on  $T'_n$  such that  $f$  is the identity on  $\text{Val}(T_n) \cap \text{Val}(T'_n)$  and  $f(T'_n) \subseteq T_n$ . Let  $t' \in T'$ . Then  $\rho_n \circ t' \in T'_n$  and by item (iii) of Lemma 3 we obtain that  $f \circ \rho_n \circ t'(\mathbf{A}) \subseteq \mathbf{A}$  is listed in  $D$ . For  $A \in R$ , we have then two cases :

- If  $t'(A) \in \text{Val}(T') \cap \text{Val}(T)$ , then we first notice that  $f \circ \rho_n \circ t'(A)$  is  $\rho_n \circ t'(A)$  since  $\rho_n \circ t'(A) \in \text{Val}(T'_n) \cap \text{Val}(T_n)$ . Also we notice that the equality  $\rho_n \circ t'(A) = t'(A)$  can be derived analogously to the egd case.
- If  $t'(A) \in \text{Val}(T') \setminus \text{Val}(T)$ , then  $f \circ \rho_n \circ t'(A) = f \circ t'(A)$  since by the definition of the chase  $\rho_n$  is the identity on  $\text{Val}(T') \setminus \text{Val}(T)$ .

Now, letting  $f^*$  be the mapping  $\text{Val}(T') \rightarrow \text{Att}$  which is the identity on  $\text{Val}(T') \cap \text{Val}(T)$  and agrees with  $f$  on  $\text{Val}(T') \setminus \text{Val}(T)$ , we can by the previous reasoning and using repeatedly [EE] derive  $f^* \circ t'(\mathbf{A}) \subseteq \mathbf{A}$  from  $f \circ \rho_n \circ t'(\mathbf{A}) \subseteq \mathbf{A}$ . Finally, we can then with one application of [CT] derive  $(T, T')$  from

$$(T^*, \text{id})[RS] \wedge \bigwedge_{t' \in T'} f^* \circ t'(\mathbf{A}) \subseteq \mathbf{A}. \quad \square$$

## 7 Typed Dependencies

Consider then the class of typed embedded dependencies. In this setting [CS] and [CT] can be replaced with rules that involve only embedded join dependencies (ejd's) and inclusion dependencies. We define ejd's over tuples of attributes as follows.

**Definition 6.** Let  $\mathbf{A}_1, \dots, \mathbf{A}_n$  be tuples of attributes listing  $R_1, \dots, R_n$ , respectively, and let  $R := \bigcup_{i=1}^n R_i$ . Then  $\bowtie(\mathbf{A}_i)_{i=1}^n$  is an embedded join dependency with the semantic rule

$$- r \models \bowtie(\mathbf{A}_i)_{i=1}^n \text{ if and only if } r|_R = r|_{R_1} \bowtie \dots \bowtie r|_{R_n}.$$

The two alternative rules for the chase are now the following. We call a relation typed if none of its values appears in two distinct columns.

CS\* Chase Start\*:

$$\bigwedge_{t \in T} \mathbf{A} \subseteq t(\mathbf{A}) \wedge \bowtie (t(\mathbf{A}))_{t \in T} \wedge \bigwedge_{t \in T} t(\mathbf{A}) \subseteq \mathbf{A}$$

where  $T$  is a typed relation and  $\text{Val}(T)$  is a set of *new attributes*.

CT\* Chase Termination\*:

$$\text{tgd} : \text{if } \bigwedge_{t \in T} \mathbf{A} \subseteq t(\mathbf{A}) \wedge \bowtie (t(\mathbf{A}))_{t \in T} \wedge \bigwedge_{t' \in T'} u \circ t'(\mathbf{A}) \subseteq \mathbf{A}, \text{ then } (T, T')[R],$$

$$\text{egd} : \text{if } \bigwedge_{t \in T} \mathbf{A} \subseteq t(\mathbf{A}) \wedge \bowtie (t(\mathbf{A}))_{t \in T} \wedge x = y, \text{ then } (T, x = y)[R],$$

where tgd:  $u$  is a mapping  $\text{Val}(T') \rightarrow \text{Att}$  that is the identity on  $\text{Val}(T') \cap \text{Val}(T)$ , and egd:  $x, y \in \text{Val}(T)$ .

The first rule is sound for typed dependencies since, for arbitrary  $r$  with  $\text{Dom}(r) \cap \text{Val}(T) = \emptyset$ , an instance of [CS\*] is satisfied by  $r \bowtie q(r)$  where  $q$  is the SPJR query

$$\rho_{t_1(\mathbf{A})/\mathbf{A}} \mathbf{A} \bowtie \dots \bowtie \rho_{t_n(\mathbf{A})/\mathbf{A}} \mathbf{A},$$

where  $\rho$  is the rename operator and  $T = \{t_1, \dots, t_n\}$ . However, a counter example for soundness can be easily constructed for untyped dependencies. If  $T$  and  $r$  are the relations illustrated in Fig. 6, then no extension  $r'$  of  $r$  to  $\text{Val}(T)$  satisfies  $\bigwedge_{t \in T} t(\mathbf{A}) \subseteq \mathbf{A}$ .

$$T = \begin{array}{c|cc} & A & B \\ \hline t & x & y \\ t' & y & x \end{array} \quad r = \begin{array}{c|cc} & A & B \\ \hline s & 0 & 1 \end{array}$$

**Fig. 6.** Relations  $T$  and  $r$

Soundness of [CT\*] is obtained analogously to that of [CT]. Also, completeness is obtained by deriving exactly in the same way as in the general case,  $\bigwedge_{t' \in T'} u \circ t'(\mathbf{A}) \subseteq \mathbf{A}$  (in the tgd case) or  $x = y$  (in the egd case) from  $\bigwedge_{t \in T} t(\mathbf{A}) \subseteq \mathbf{A}$ . Let us then write  $\Sigma \vdash^* \sigma$  if  $\sigma$  is deduced from  $\Sigma$  in the sense of Definition 5 and using rules [EE, CS\*, CR, CT\*] together with elimination and introduction of conjunction. Then we obtain the following theorem.

**Theorem 3.** *Let  $\Sigma \cup \{\sigma\}$  be a finite set of typed egd's and tgd's over  $R$ . Then  $\Sigma \models \sigma \Leftrightarrow \Sigma \vdash^* \sigma$ .*

**Acknowledgement.** The author was supported by grant 264917 of the Academy of Finland.

## References

1. Aho, A.V., Beeri, C., Ullman, J.D.: The theory of joins in relational databases. *ACM Trans. Database Syst.* **4**(3), 297–314 (1979)
2. Maier, D., Mendelzon, A.O., Sagiv, Y.: Testing implications of data dependencies. *ACM Trans. Database Syst.* **4**, 455–469 (1979)
3. Beeri, C., Vardi, M.Y.: A proof procedure for data dependencies. *J. ACM* **31**(4), 718–741 (1984)
4. Chandra, A.K., Vardi, M.Y.: The implication problem for functional and inclusion dependencies is undecidable. *SIAM J. Comput.* **14**(3), 671–677 (1985)
5. Mitchell, J.C.: The implication problem for functional and inclusion dependencies. *Inf. Control* **56**(3), 154–173 (1983)
6. Thalheim, B.: *Dependencies in Relational Databases*. Teubner-Texte zur Mathematik. Springer, New York (1991)
7. Mitchell, J.C.: Inference rules for functional and inclusion dependencies. In: Fagin, R., Bernstein, P.A. (eds.) (PODS), pp. 58–69. ACM (1983)
8. Hodges, W.: Compositional semantics for a language of imperfect information. *J. Interest Group Pure Appl. Logics* **5**(4), 539–563 (1997)
9. Galliani, P.: Inclusion and exclusion dependencies in team semantics: On some logics of imperfect information. *Ann. Pure Appl. Log.* **163**(1), 68–84 (2012)
10. Väänänen, J.: *Dependence Logic*. Cambridge University Press, Cambridge (2007)
11. Galliani, P., Hella, L.: Inclusion Logic and Fixed Point Logic. In: Rocca, S.R.D. (ed.) *Computer Science Logic 2013 (CSL 2013)*. Leibniz International Proceedings in Informatics (LIPIcs), vol. 23, pp. 281–295. Dagstuhl, Germany, Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik (2013)
12. Immerman, N.: Relational queries computable in polynomial time. *Inf. Control* **68**(1), 86–104 (1986)
13. Vardi, M.Y.: The complexity of relational query languages. In: *Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing*, pp. 137–146. ACM (1982)
14. Hannula, M., Kontinen, J.: A finite axiomatization of conditional independence and inclusion dependencies. In: Beierle, C., Meghini, C. (eds.) *FoIKS 2014*. LNCS, vol. 8367, pp. 211–229. Springer, Heidelberg (2014)
15. Yannakakis, M., Papadimitriou, C.H.: Algebraic dependencies. *J. Comput. Syst. Sci.* **25**(1), 2–41 (1982)
16. Abiteboul, S., Hull, R., Vianu, V.: *Foundations of Databases*. Addison Wesley, Reading (1995)
17. Hannula, M.: Reasoning about embedded dependencies using inclusion dependencies. *CoRR abs/1507.00655* (2015)