Reihe Informatik
3 / 92

A definition of redundancy in relational databases

Mila E. Majster-Cederbaum
Peter Peters
Fakultät für Mathematik und Informatik
Universität Mannheim
Seminargebäude A5
6800 Mannheim 1

Mai 1992

0. Introduction

The relational data model as proposed by Codd is a well-established method for data abstraction. Two essential aspects in this model are the definition of the data structure via the relation scheme and the data semantics via data dependencies. Various classes of data dependencies have been studied in the past [5, 6, 7, 8, 9, 13, 15]. In the presence of data dependencies "update dependencies" (or anomalies) and "redundancy" may occur as first observed by Codd in [5, 6]. Normal forms have been proposed as a means to control update anomalies and redundancy [6, 14]. But as the notion of redundancy has never been formally defined, one cannot make any precise statement concerning the presence or absence of redundancy for a given design.

In this paper we attempt to provide a formal definition of the notion of redundancy for the case of a single relation respectively relation scheme. We first give a static semantic definition of redundancy and then present an operational analogue. Intuitively speaking a relation r contains redundancy, if some "part" of the information given in r can be "determined" from the "rest" of r. And a relation scheme with a given set of data dependencies admits redundancy if there is a relation belonging to this scheme that contains redundancy.

The paper is organized in six sections. Section 1 contains the definition of the relational model that we use. We make use of partial "relations" that are built from constants and variables. In section 2 we present the semantic definition of redundancy. Section 3 introduces a class of data dependencies, i.e. implicational dependencies and a chase procedure for partial relations. Section 4 gives an operational characterization of redundancy. The main theorem in this section is theorem 4.3. It states that a relation r in a class of relations $sat(D)$ contains redundancy if there exists a partial relation q that "contains less information" than r and for which $chase_D(q) = r$, i.e. the missing information can be "derived". In section 5 we treat the special cases of functional dependencies and multivalued dependencies. It is shown that in the case of functional dependencies BCNF is a necessary and sufficient condition for a relation scheme not to admit redundancy. The analogous result is established for 4NF in the case of multivalued dependencies. In section 6 we discuss some aspects of the case where a set of relation schemes instead of a single relation scheme is considered.

1. The relational model

In contrast to the original relational model we allow the use of variables in relations.

Definition 1.1: A *relation scheme* is given by its name R, a set of n *attributes* $\{A_1, \ldots, A_n\}$ called the *universe* of R and a set of m $(m \leq n)$ *domain symbols* $\{D_1, \ldots, D_m\}$ which are related to attributes by a mapping

$$domsym : \{A_1, \ldots, A_n\} \longrightarrow \{D_1, \ldots, D_m\}.$$

The *type* of a relation scheme is given by an expression

$$R(A_1|D_{i_1}, \ldots, A_n|D_{i_n}),$$

where $D_{i_j} = domsym(A_j)$ $(j = 1, \ldots, n)$. n is called the *degree* of the relation scheme. $R(A_1|D_{i_1}, \ldots, A_n|D_{i_n})$ is often abbreviated by $R(A_1, \ldots, A_n)$ or R. We also write $attr(R) = \{A_1, \ldots, A_n\}$ and $deg(R) = n$. R is said to be of *discrete type* if

$$domsym(A) \neq domsym(B) \text{ for all } A, B \in attr(R) \text{ with } A \neq B.$$

For every domain symbol D a set of *constants* $dom(D)$ and a set of *variables* $var(D)$, both countable infinite and with $dom(D) \cap var(D) = \emptyset$, are given. We demand that $dom(D_1) \cap dom(D_2) = \emptyset$ and $var(D_1) \cap var(D_2) = \emptyset$ for any two domain symbols D_1, D_2 of R.
For every domain symbol D we define a mapping

$$ord : var(D) \cup dom(D) \longrightarrow I\!N_0$$

that maps each variable to its *index* in $I\!N$ and each constant to zero. We request that no two variables in D obtain the same index.

For convenience we introduce the following notations :
For $A \in attr(R)$ we will use the notations $dom(A)$, $var(A)$ as well as $dom(R)$, $var(R)$ in the obvious way.

Similarly, for $x \in var(R) \cup dom(R)$ we want to use $domsym(x)$ and $attr(x)$, where in the non discrete case we get

$$attr(x) = \{A \in attr(R); domsym(x) = domsym(A)\},$$

while $attr(x)$ consists of a single attribute in the discrete case.

Definition 1.2: For a relation scheme $R(A_1, \ldots, A_n)$ of degree n we define a *partial tuple* t as a mapping

$$t : attr(R) \longrightarrow dom(R) \cup var(R),$$

where $t(A) \in dom(A) \cup var(A)$ for each $A \in attr(R)$.
Sometimes we write $t(a_1, \ldots, a_n)$ or just $(a_1, \ldots, a_n)$ where $a_i = t(A_i)$ for $i = 1, \ldots, n$.
$def(t)$ is the set of all attributes A in R for which $t(A)$ is a constant and

$$\begin{aligned} var(t) &= t(attr(R) \setminus def(t)) \\ dom(t) &= t(def(t)) \end{aligned}$$

A partial tuple for R is called a *tuple* for R if t does not contain any variables.
Finally the set of all partial tuples for R is given by $tup(R)$.

Definition 1.3: Let R be a relation scheme. A *partial relation* r for R is a finite set of partial tuples for R. $var(r)$ is the set of all variables of r and $dom(r)$ is the set of all constants of r.
A partial relation for R is a *relation* for R if it does not contain any variables.
The set of all partial relations for R is denoted by $rel(R)$.

Partial relations are used in [10] for handling the problem of null values.

Definition 1.4: Let R be a relation scheme. A mapping

$$d : var(R) \cup dom(R) \longrightarrow var(R) \cup dom(R),$$

where $d(var(A)) \subseteq var(A) \cup dom(A)$ and $d\big|_{dom(R)} = id$ is called a *domain mapping* for R.
The set of all *domain mappings* for R is denoted by $DOM(R)$.

Definition 1.5: Let R be a relation scheme and $d \in DOM(R)$. Furthermore let $I \subset var(R) \times (var(R) \cup dom(R))$ such that
i) $(x, c) \in I : \ \Rightarrow domsym(x) = domsym(c)$
ii) $(x, c_1) \in I$ and $(x, c_2) \in I \quad \Rightarrow c_1 = c_2$.
We then define

$$d_I(x) = \begin{cases} y & \text{if } (x, y) \in I \\ d(x) & \text{otherwise} \end{cases}$$

In case $I = \{(x, c)\}$ we write $d_{(x,c)}$ for d_I.

Definition 1.6: Let R be a relation scheme, $t \in tup(R)$ and $d \in DOM(R)$. We define a *substitution* of t by d by

$$t\langle d\rangle = d \circ t.$$

A *substitution* for a partial relation $r \in rel(R)$ by d is given by

$$r\langle d\rangle = \{t\langle d\rangle; t \in r\}.$$

Obviously $r\langle f \circ g\rangle = r\langle g\rangle\langle f\rangle$ holds for arbitrary $f, g \in DOM(R)$ and $r \in rel(R)$.

Definition 1.7: Let R be a relation scheme, $u, v \in tup(R)$ and $q, r \in rel(R)$.
We say, u *subsumes* v, if there exists a domain mapping $d \in DOM(R)$ such that $u = v\langle d\rangle$. Then we write $u \geq_d v$ or just $u \geq v$. If $u \geq v$ holds but not $v \geq u$, we write $u > v$.
Similarly, for partial relations we say, r *subsumes* q, if there exists a domain mapping $d \in DOM(R)$ such

that $r \supseteq q\langle d \rangle$. We write as before $r \geq_d q$ or just $r \geq q$. If $q \geq_e r$ for $e \in DOM(R)$ holds in addition we say that q and r are equivalent and write $q \mathrel{\widehat{=}} r$. In the case $r \geq q$ and $r \not\equiv q$ we write $r > q$.

The term ***subsumes*** appears first in [10], where it is introduced in the context of partial relations with unmarked null values.

Theorem 1.1: Let R be a relation scheme. The subsumption $\geq$ is a partial ordering and $\mathrel{\widehat{=}}$ is an equivalence relation on $rel(R)$.

Proof: The proof is a straight forward argument using the properties of $\subseteq$ and the fact that $r\langle f \rangle\langle g \rangle = r\langle g \circ f \rangle$ for all partial relations $r \in rel(R)$ and $g, f \in DOM(R)$.□

Definition 1.8: Let R be a relation scheme and $q \in rel(R)$. By

$$comp(q) = \{r \in rel(R); r \text{ is a relation and } r \geq q\}$$

we denote the set of all *completions* of q in $rel(R)$.

Theorem 1.2: Let R be a relation scheme. For $r_1, r_2 \in rel(R)$

$$comp(r_1) \subseteq comp(r_2) \text{ iff } r_1 \geq r_2.$$

Proof:
i) Let $comp(r_1) \subseteq comp(r_2)$.
If r_1 is a relation then $r_1 \in comp(r_1)$ and therefore $r_1 \in comp(r_2)$. thus $r_1 \geq r_2$.
If r_1 contains variables then let $var(r_1) = \{x_1, \ldots, x_k\}$. We select a set $\{c_1, \ldots, c_k\} \subseteq dom(R)$ of k distinct constants that is disjoint to either of $dom(r_1)$ and $dom(r_2)$ and satisfies $domsym(c_i) = domsym(x_i)$.
We set

$$I = \{(x_i, c_i); 1 \leq i \leq k\}$$

and get

$$r_1\langle id_I \rangle \in comp(r_1) \subseteq comp(r_2).$$

Thus there exists $e \in DOM(R)$ with $r_1\langle id_I \rangle \supseteq r_2\langle e \rangle$.
Finally we define $d \in DOM(R)$ by

$$d(x) = \begin{cases} e(x) & \text{if } e(x) \notin \{c_1, \ldots, c_k\} \\ x_i & \text{if } e(x) = c_i, 1 \leq i \leq k \end{cases}$$

and get $r_1 \supseteq r_2\langle d \rangle$, thus $r_1 \geq r_2$.
ii) Let $r_1 \geq r_2$, then there exists $d \in DOM(R)$ with $r_1 \supseteq r_2\langle d \rangle$. For each $r \in comp(r_1)$ there is a mapping $e \in DOM(R)$ with $r \supseteq r_1\langle e \rangle$. Thus we obtain

$$r_1\langle e \rangle \supseteq r_2\langle d \rangle\langle e \rangle \supseteq r_2\langle e \circ d \rangle$$

and so $r \supseteq r_2\langle e \circ d \rangle$ hence $r \in comp(r_2)$ holds.□

2. A semantic definition of redundancy

Intuitively some information is redundant if it is somehow represented in more than one way in our system. Hence we might drop parts of our description while still being able to obtain the same information as before.

Definition 2.1: Let R be a relation scheme and $sat \subseteq rel(R)$ a set of so called valid relations and $r \in rel(R)$ a partial relation, then

$$comp_{sat}(r) = comp(r) \cap sat$$

is the set of all valid completions of r.

Definition 2.2: Let R be a relation scheme and $sat \subseteq rel(R)$ a set of valid relations. The set of all partial relations that are consistent with sat is denoted by

$$rel_{sat}(R) = \{r; r \in rel(R) \text{ and } comp_{sat}(r) \neq \emptyset\}.$$

Definition 2.3: Let R be a relation scheme and $sat \subseteq rel(R)$ a set of relations. If $r \in rel(R)$ is a partial relation then the set of all minimal valid completions is denoted by

$$min_{sat}(r) = \{q \in comp_{sat}(r); \tilde{q} \in comp_{sat}(r) \text{ and } \tilde{q} \subseteq q \text{ implies } \tilde{q} = q\}.$$

Lemma 2.1: Let R be a relation scheme, $sat \subseteq rel(R)$ a set of relations and $q_1, q_2 \in rel_{sat}(R)$ partial relations. Then

$$comp_{sat}(q_1) = comp_{sat}(q_2)$$

holds iff

$$min_{sat}(q_1) = min_{sat}(q_2)$$

Proof: i) Let $comp_{sat}(q_1) = comp_{sat}(q_2)$ and choose

$$r \in min_{sat}(q_1) \subseteq comp_{sat}(q_1) = comp_{sat}(q_2).$$

For each

$$s \in comp_{sat}(q_2) \text{ with } s \subseteq r$$

we get

$$s \in comp_{sat}(q_1)$$

and therefore $s = r$, because $r \in min_{sat}(q_1)$. Thus

$$min_{sat}(q_1) \subseteq min_{sat}(q_2).$$

ii) Let $min_{sat}(q_1) = min_{sat}(q_2)$ and $s \in comp_{sat}(q_1)$. Since s is finite $r \in min_{sat}(q_1)$ exists with $r \subseteq s$. Clearly, $s \in sat$ and $r \in min_{sat}(q_2)$. Therefore $q_2 \leq r \leq s$ holds and finally $s \in comp_{sat}(q_2)$. Thus $comp_{sat}(q_1) \subseteq comp_{sat}(q_2)$ holds. □

Lemma 2.2: Let R be a relation scheme, $sat \subseteq rel(R)$ a set of relations and $q \in rel_{sat}(R)$ a partial relation. If $min_{sat}(q)$ consists of a single relation then

$$min_{sat}(q) = \bigcap(r; r \in comp_{sat}(q)) \in sat$$

omitting parentheses.

Proof:

i) Because $min_{sat}(q)$ consists of a single relation, each completion in $comp_{sat}(q)$, including $min_{sat}(q)$, is a superset of $min_{sat}(q)$. Therefore

$$min_{sat}(q) = \bigcap(r; r \in comp_{sat}(q)).$$

ii) By definition $min_{sat}(q) \in comp_{sat}(q)$, thus $min_{sat}(q) \in sat$. □

Definition 2.4: Let R be a relation scheme and $sat \subseteq rel(R)$ a set of relations. Furthermore let $r \in sat$ be a relation. Then r *contains redundancy* wrt. sat, if there exists a partial relation q with

$$r > q, |r| \geq |q| \text{ and } comp_{sat}(r) = comp_{sat}(q).$$

Lemma 2.3: Let R be a relation scheme and $sat \subseteq rel(R)$ a set of relations. Furthermore let $r \in sat$ be a relation. Then r contains redundancy wrt. sat, if there exists a partial relation q with

$$r > q, |r| \geq |q| \text{ and } r = min_{sat}(q).$$

3. Implicational dependencies and the chase

In the previous section we used a set *sat* to describe the set of those relations which we want to consider valid in a given situation. Using *sat* we gave a semantic, i.e. static, definition of redundancy. We are now looking for an operational, i.e. algorithmic counterpart. For this we have to give some more information about how such a set *sat* might look like. We use implicational dependencies as introduced in [2, 4, 9], to characterize sets of valid relations.

Implicational dependencies (ID) fall into two classes, the class of *equality generating dependencies* (EGDs) and the class of *total tuple generating dependencies* (TTGDs). EGDs are generalizations of FDs and TTGDs are generalizations of MVDs.

Definition 3.1: Let R be a relation scheme of discrete type.
An EGD is an expression of the form $\langle (a/b); U\rangle$ where $U \in rel(R)$ with $dom(U) = \emptyset$ and $a, b \in var(U)$.
A TTGD is an expression of the form $\langle v; U\rangle$ where $U \in rel(R)$ with $dom(U) = \emptyset$ and $v \in tup(R)$ with $var(v) \subseteq var(U)$ and $def(v) = \emptyset$.
Henceforth we assume that for every TTGD we have $ord(v(A)) \leq ord(x)$ for all $x \in var(A)$ and for all $A \in attr(R)$. In a similar way we demand for all EGDs that $ord(a) \leq ord(b)$ and $ord(a) \leq ord(x)$ for all $x \in var(attr(a))$ holds.

Example 3.1: For $R(A, B, C)$ the FD $AB \rightarrow C$ and the MVD $B \rightarrow\rightarrow C$ are given.
$AB \rightarrow C$ is the EGD $\langle (C_1/C_2); \{(A_1, B_1, C_1), (A_1, B_1, C_2)\}\rangle$ and $B \rightarrow\rightarrow C$ is the $TTGD$ $\langle (A_1, B_1, C_1); \{(A_1, B_1, C_2), (A_2, B_1, C_1)\}\rangle$.

A	B	C
A_1	B_1	C_1
A_1	B_1	C_2
$C_1 = C_2$		

A	B	C
A_1	B_1	C_2
A_2	B_1	C_1
A_1	B_1	C_1

EGD and $TTGD$ as tables

Definition 3.2: Let R be a relation scheme of discrete type and D a set of IDs for R. A relation r for R is valid for D if for all IDs F in D the following holds:

(1) If $F = \langle v; U\rangle$ is a $TTGD$ then for each $d \in DOM(R)$ with $U\langle d\rangle \subseteq r$, $v\langle d\rangle \in r$ holds.
(2) If $F = \langle (a/b); U\rangle$ is a EGD then for each $d \in DOM(R)$ with $U\langle d\rangle \subseteq r$, $d(a) = d(b)$ holds.

The set of all valid relations for D is denoted by $sat(D)$. $rel_{sat(D)}(R)$ is defined as in Definition 2.2.

In [4] a decision procedure for the implication problem of IDs, the *chase* is given, using so-called EGD– and $TTGD$–rules. Our rules differ from those in [4] as we also have to deal with constant values.

Definition 3.3: Let R be a relation scheme of discrete type. We introduce a relation Ξ that is not yet in $rel(R)$. The relation Ξ has the role of an error relation. This error relation is the result of the *chase* if an EGD–rule fails.
Let $r \in rel(R)$ be a partial relation.
EGD–rules : If we have the EGD $\langle (a/b); U\rangle$ and $d \in DOM(R)$ so that $U\langle d\rangle \subseteq r$ holds we change r to r', where

(1) if $d(a), d(b) \in var(R)$ then
$$r' = \begin{cases} r\langle id_{(d(a),d(b))}\rangle & , \quad ord(d(a)) \geq ord(d(b)) \\ r\langle id_{(d(b),d(a))}\rangle & , \quad ord(d(a)) < ord(d(b)), \end{cases}$$
(2) if $d(a) \in dom(R)$, $d(b) \in var(R)$ then
$r' = r\langle id_{(d(b),d(a))}\rangle$,
(3) if $d(b) \in dom(R)$, $d(a) \in var(R)$ then
$r' = r\langle id_{(d(a),d(b))}\rangle$,
(4) if $d(a), d(b) \in dom(R)$ then
$$r' = \begin{cases} r & , \quad d(a) = d(b) \\ \Xi & , \quad d(a) \neq d(b). \end{cases}$$

In this case we write $\langle(a/b);U\rangle_d : r \longrightarrow r'$, which means that we transform r to r' under $\langle(a/b);U\rangle$ with d by applying an EGD–rule.
$TTGD$–rules : If we have a $TTGD$ $\langle v;U\rangle$ and $d \in DOM(R)$, so that $U\langle d\rangle \subseteq r$ holds, we change r to r' where $r' = r \cup \{v\langle d\rangle\}$.
In this case we write $\langle v;U\rangle_d : r \longrightarrow r'$, which means that we transform r to r' under $\langle v;U\rangle$ with d by applying a $TTGD$–rule.

Definition 3.4: Let R be a relation scheme of discrete type and D a set of IDs for R. Furthermore let $r \in rel(R)$ be a partial relation.
A *generating sequence* for r under D is a sequence $r_0, r_1, \ldots, r_n, \ldots$ of partial relations, where $r = r_0$ and every partial relation r_{i+1} with $0 \leq i$ is generated by applying an EGD– resp. a $TTGD$–rule to r_i. Only IDs from D are used. Furthermore we demand that $r_i \neq r_{i+1}$, $i = 0, 1, \ldots$.
If a generating sequence has a last element r_n, i.e. no further EGD– resp. $TTGD$–rules can be applied, then r_n is called a chase of r under D. $chase_D(r)$ denotes the set of all last elements of r under D.

Lemma 3.1: Let R be a relation scheme of discrete type and D a set of IDs for R. The chase procedure terminates and for every partial relation r there exists a global upper bound for the length of all generating sequences of r under D.

Proof: Since a partial relation is finite it contains only a finite set of variables and constants. EGD– and $TTGD$–rules can produce only a finite number of different partial relations, because these rules do not create new constants or variables.
If in each generating sequence no partial relation occurs more than once, termination of the chase procedure is proved. A global upper bound is the number of all different partial relations which may originate from r by applying EGD– and $TTGD$–rules. We have to show that no partial relation occurs more than once.
Let r_i, r_j be partial relations of some generating sequence for r where $i < j$.
If somewhere in $r_i, \ldots, r_j$ an EGD–rule is used then r_i contains a variable that does not belong to r_j, thus $r_i \neq r_j$.
If somewhere in $r_i, \ldots, r_j$ a $TTGD$–rule is used, either r_j consists of more tuples than r_i or after the application of this $TTGD$–rule a following EGD–rule reduces the number of tuples to that of r_i and our previous argument applies. □

4. Some properties of the chase and an operational characterization of redundancy

For the rest of the paper we will write $min_D(r)$ for $min_{sat(D)}(r)$ and $comp_D(r)$ for $comp_{sat(D)}(r)$ for a set D of implicational dependencies.

Definition 4.1: Let R be a relation scheme of discrete type and D a set of IDs for R. The set of all partial relations which are not affected by the applications of the chase is given by

$$sat^*(D) = \{r \in rel(R); chase_D(r) = \{r\}\}.$$

Definition 4.2: Let R be a relation scheme of discrete type and D a set of IDs for R. The set of all partial relations for which the chase does not fail is given by

$$rel_D(R) = \{r \in rel(R); \Xi \notin chase_D(r)\}.$$

Lemma 4.1: Let R be a relation scheme of discrete type and D a set of IDs for R. Furthermore let $r \in rel_D(R)$. Then $chase_D(r) \subseteq sat^*(D)$.

Proof: Because each partial relation $r^* \in chase_D(r)$ is a last element of some generating sequence for r under D, by construction, no EGD–, resp. $TTGD$–rule is applicable to r^*. Thus $chase_D(r^*) = \{r^*\}$ and therefore $r^* \in sat^*(D)$. □

Lemma 4.2: Let R be a relation scheme of discrete type and D a set of IDs for R. Then

(1) $sat(D) \subseteq sat^*(D) \subseteq rel_D(R)$ and
(2) $sat(D) = \{r \in sat^*(D); r \text{ is a relation }\}$.

Proof: The two statements follow directly from the definition of $sat(D)$, $sat^*(D)$ resp. $rel_D(R)$. □

Lemma 4.3: Let R be a relation scheme of discrete type and D a set of IDs for R. Furthermore, let $r \in sat^*(D)$. If there is e in $DOM(R)$ such that

(1) $e(var(r)) \subseteq dom(R) \setminus dom(r)$ and
(2) $e\big|_{var(r)}$ is injective,

then $r\langle e\rangle \in sat(D)$.

Proof: Because of the choice of e, $r\langle e\rangle$ is a non partial version of r. □

Theorem 4.1: Let R be a relation scheme of discrete type and D a set of IDs for R. Furthermore let $r \in rel_D(R)$ and $\bar{r} \in chase_D(r)$. Then there is a unique $d_{(D,r,\bar{r})} \in DOM(R)$ satisfying

(1) $r\langle d_{(D,r,\bar{r})}\rangle \subseteq \bar{r}$,
(2) $var(r\langle d_{(D,r,\bar{r})}\rangle) = var(\bar{r})$,
(3) $d_{(D,r,\bar{r})}\big|_{var(R)\setminus var(r)} = id$ and
(4) $d^2_{(D,r,\bar{r})} = d_{(D,r,\bar{r})}$.

Here $d_{(D,r,\bar{r})}$ depends on r, $\bar{r}$ and on D. In addition, $ord(d_{(D,r,\bar{r})}(x)) \leq ord(x)$ holds for every $x \in var(R)$.

Proof: Let $F_{1_{f_1}} : r \longrightarrow r_1, \ldots, F_{n_{f_n}} : r_{n-1} \longrightarrow r_n = \bar{r}$ be a generating sequence for r. We prove the assertion by induction on the length of the generating sequence.
Let $n = 0$. Under the assumption that $d_{(D,r,\bar{r})}$ exists we first show that $d_{(D,r,\bar{r})}$ is unique. We have $r = \bar{r}$, thus

$$var(r\langle d_{(D,r,\bar{r})}\rangle) = var(\bar{r}) = var(r).$$

Therefore $d_{(D,r,\bar{r})}$ is a one-to-one mapping from $var(r)$ to $var(r)$. Because of the conditon $d^2_{(D,r,\bar{r})} = d_{(D,r,\bar{r})}$ we have

$$\text{if } y = d_{(D,r,\bar{r})}(x) \text{ then } d_{(D,r,\bar{r})}(y) = d^2_{(D,r,\bar{r})}(x) = d_{(D,r,\bar{r})}(x) = y\ .$$

Consequently, $d_{(D,r,\bar{r})}$ is the identity. Obviously the mapping $d_{(D,r,\bar{r})} = id$ satisfies all conditions.
Now let $r \in rel_D(R)$ be a partial relation with a generating sequence of length n resulting in $\bar{r} \in chase_D(r)$. Consequently, the partial relation r_1 has a generating sequence of length $n-1$ resulting in $\bar{r}_1 = \bar{r}$. By assumption, there is a mapping $d_{(D,r_1,\bar{r}_1)}$ satisfying the properties of the theorem. We first prove uniqueness under the assumption that $d_{(D,r,\bar{r})}$ exists.
If $F_1 = \langle v; U\rangle$ is a $TTGD$, then

$$U\langle f_1\rangle \subseteq r \text{ and } r_1 = r \cup \{v\langle f_1\rangle\}$$

holds. Thus we have

$$U\langle d_{(D,r,\bar{r})} \circ f_1\rangle = U\langle f_1\rangle\langle d_{(D,r,\bar{r})}\rangle \subseteq r\langle d_{(D,r,\bar{r})}\rangle \subseteq \bar{r}$$

and due to $\bar{r} \in sat^*(D)$

$$v\langle f_1\rangle\langle d_{(D,r,\bar{r})}\rangle \in \bar{r}.$$

This means

$$r_1\langle d_{(D,r,\bar{r})}\rangle \subseteq \bar{r},$$

thus $r\langle d_{(D,r,\bar{r})}\rangle \subseteq r_1\langle d_{(D,r,\bar{r})}\rangle \subseteq \bar{r} = \bar{r}_1$. From this we get

$$r_1\langle d_{(D,r,\bar{r})}\rangle \subseteq \bar{r}_1 \text{ and } var(r_1\langle d_{(D,r,\bar{r})}\rangle) = var(\bar{r}_1).$$

As $var(r) = var(r_1)$ we get

$$d_{(D,r,\bar{r})}\big|_{var(R)\setminus var(r_1)} = d_{(D,r,\bar{r})}\big|_{var(R)\setminus var(r)} = id$$

and by assumption $d^2_{(D,r,\bar{r})} = d_{(D,r,\bar{r})}$.
$d_{(D,r,\bar{r})}$ fulfills all properties of $d_{(D,r_1,\bar{r}_1)}$, but $d_{(D,r_1,\bar{r}_1)}$ is unique among all mappings with these properties. We have shown that $d_{(D,r,\bar{r})} = d_{(D,r_1,\bar{r}_1)}$ holds, and by that the uniqueness of $d_{(D,r,\bar{r})}$ follows from the uniqueness of $d_{(D,r_1,\bar{r}_1)}$.
If $F_1 = \langle (a/b); U\rangle$ is an EGD, then

$$U\langle f_1\rangle \subseteq r \text{ and } r_1 = r\langle id_{(x,y)}\rangle$$

where

$$(x,y) = \begin{cases} (f_1(a), f_1(b)), & f_1(b) \in dom(R) \text{ or} \\ & f_1(b) \in var(R) \text{ and } ord(f_1(a)) \geq ord(f_1(b)) \\ (f_1(b), f_1(a)), & \text{otherwise} \end{cases}$$

Because of $r\langle d_{(D,r,\bar{r})}\rangle \subseteq \bar{r}$, it follows

$$U\langle d_{(D,r,\bar{r})} \circ f_1\rangle \subseteq U\langle f_1\rangle\langle d_{(D,r,\bar{r})}\rangle \subseteq r\langle d_{(D,r,\bar{r})}\rangle \subseteq \bar{r}$$

and due to $\bar{r} \in sat^*(D)$ we have

$$d_{(D,r,\bar{r})}(x) = d_{(D,r,\bar{r})}(y).$$

From $x \notin var(\bar{r})$ and $d = d_{(D,r,\bar{r})(x,x)}$ we conclude $d_{(D,r,\bar{r})} = d \circ id_{(x,y)}$. Now we have

$$r_1\langle d\rangle = r\langle id_{(x,y)}\rangle\langle d\rangle = r\langle d_{(D,r,\bar{r})}\rangle \subseteq \bar{r} = \bar{r}_1.$$

This yields $var(r_1\langle d\rangle) = var(\bar{r}_1)$ and $r_1\langle d\rangle \subseteq \bar{r}_1$. Furthermore we know

$$d^2(z) = d^2_{(D,r,\bar{r})}(z) = d_{(D,r,\bar{r})}(z) = d(z) \text{ for } z \neq x$$

and

$$d^2(x) = x = d(x).$$

Finally we have

$$d\big|_{var(R)\setminus var(r_1)} = id$$

because of

$$var(r_1) = var(r) \setminus \{x\} \text{ and } d(x) = x.$$

As before, d fulfills all properties of $d_{(D,r_1,\bar{r}_1)}$. We know that $d_{(D,r_1,\bar{r}_1)}$ is unique hence $d = d_{(D,r_1,\bar{r}_1)}$. Thus $d_{(D,r,\bar{r})} = d_{(D,r_1,\bar{r}_1)} \circ id_{(x,y)}$ is unique. It is easy to see that $d_{(D,r,\bar{r})}$ fulfills the demanded properties. Finally we can conclude from the choice of (x,y) that the additional property holds. □

Lemma 4.4: Let R be a relation scheme of discrete type and D a set of IDs for R. If

$$r \in rel(R),\ r^* \in sat^*(D) \text{ and } d \in DOM(R)$$

such that

$$r\langle d\rangle \subseteq r^*$$

then for every $TTGD$– and EGD-rule

$$F_f : r \longrightarrow \hat{r}$$

we have

$$F_{d\circ f} : r\langle d\rangle \longrightarrow \hat{r}\langle d\rangle$$

and additionally

$$r\langle d\rangle \subseteq \hat{r}\langle d\rangle \subseteq r^*.$$

If F is an EGD and $\hat{r} = r\langle id_{(x,y)}\rangle$ it follows that $d(x) = d(y)$.

Proof: If $F = \langle v; U\rangle$ is a $TTGD$ we have

$$\langle v; U\rangle_f : r \longrightarrow \hat{r}$$

and hence

$$U\langle f\rangle \subseteq r \text{ and } \hat{r} = r \cup \{v\langle f\rangle\}.$$

Thus we get

$$U\langle d \circ f\rangle = U\langle f\rangle\langle d\rangle \subseteq r\langle d\rangle.$$

In addition

$$\hat{r}\langle d\rangle = (r \cup \{v\langle f\rangle\})\langle d\rangle = r\langle d\rangle \cup \{v\langle d \circ f\rangle\}.$$

This yields

$$F_{d\circ f} : r\langle d\rangle \longrightarrow \hat{r}\langle d\rangle \text{ and } r\langle d\rangle \subseteq \hat{r}\langle d\rangle \subseteq r^*$$

because of

$$r^* \in sat^*(D).$$

If $F = \langle (a/b); U\rangle$ we have

$$\langle (a/b); U\rangle_f : r \longrightarrow \hat{r}$$

and hence

$$U\langle f\rangle \subseteq r \text{ and } \hat{r} = r\langle id_{(x,y)}\rangle$$

where

$$(x, y) = \begin{cases} (f(a), f(b)), & f(b) \in dom(R) \text{ or} \\ & f(b) \in var(R) \text{ and } ord(f(a)) \geq ord(f(b)) \\ (f(b), f(a)), & \text{otherwise} \end{cases}.$$

As

$$U\langle d \circ f\rangle = U\langle f\rangle\langle d\rangle \subseteq r\langle d\rangle \subseteq r^* \in sat^*(D)$$

we get

$$d(x) = d(y).$$

We may write

$$d \circ id_{(x,y)} = d = id \circ d = id_{(d(x),d(y))} \circ d$$

and get

$$\begin{aligned} \hat{r}\langle d\rangle &= r\langle id_{(x,y)}\rangle\langle d\rangle = r\langle d \circ id_{(x,y)}\rangle = r\langle id_{(d(x),d(y))} \circ d\rangle \\ &= r\langle d\rangle\langle id_{(d(x),d(y))}\rangle = r\langle d\rangle. \end{aligned}$$

This means

$$\langle (a/b); U\rangle_{d\circ f} : r\langle d\rangle \longrightarrow \hat{r}\langle d\rangle \text{ and } r\langle d\rangle = \hat{r}\langle d\rangle \subseteq r^*. \square$$

Lemma 4.5: Let R be a relation scheme of discrete type and D a set of IDs for R. If

$$r \in rel_D(R),\ r^* \in sat^*(D) \text{ and } \bar{r} \in chase_D(r).$$

and

$$d \in DOM(R) \text{ with } r\langle d\rangle \subseteq r^*$$

then

$$r\langle d\rangle \subseteq \bar{r}\langle d\rangle \subseteq r^*.$$

Proof: The proof is a simple induction on the length of generating sequences using the previous lemma. □

Corollary: Let R be a relation scheme of discrete type and D a set of IDs for R. If

$$q \in rel(R),\ r \in rel_D(R) \text{ and } q \leq r$$

then

$$q \in rel_D(R).$$

Proof: We must show that $\Xi \notin chase_D(q)$. Let $\bar{q} \in chase_D(q)$. We know from $r \in rel_D(R)$ that

$$chase_D(r) \subseteq sat^*(D).$$

We choose

$$\bar{r} \in chase_D(r) \text{ with } \bar{r} \in sat^*(D).$$

We also know that a $\bar{d} \in DOM(R)$ exists with

$$r\langle \bar{d} \rangle \subseteq \bar{r}.$$

From $q \leq r$ it follows that a mapping $d \in DOM(R)$ exists with

$$q\langle d \rangle \subseteq r$$

and so

$$q\langle \bar{d} \circ d \rangle \subseteq \bar{r} \in sat^*(D).$$

From Lemma 4.5 we get

$$\bar{q}\langle \bar{d} \circ d \rangle \subseteq \bar{r}.$$

Thus,

$$\bar{q} \in sat^*(D) \text{ and } \Xi \notin chase_D(q)$$

because $\bar{q}$ was arbitrarily choosen. □

The following theorem states that if we work with *IDs* then the chase consists only of a single partial relation, in each case where the partial relation is consistent with the given *IDs*, and is the special relation Ξ, otherwise. Similar results are usually proved via the Church-Rosser property. Here we give an alternative proof.

Theorem 4.2: Let R be a relation scheme of discrete type and D a set of *IDs* for R. For every partial relation $r \in rel(R)$, $chase_D(r)$ is a singleton set, say $\{\bar{r}\}$. Thus only one $d_{(D,r,\bar{r})}$ exists if $r \in rel_D(R)$.
(Henceforth we write $chase_D(r) = \bar{r}$ for $chase_D(r) = \{\bar{r}\}$ and $d_{(D,r)}$ for $d_{(D,r,\bar{r})}$.)

Proof:

First let $r \in rel(R) \setminus rel_D(R)$. Assume there is

$$\bar{r} \in chase_D(r) \text{ with } \bar{r} \neq \Xi.$$

It follows that

$$r \leq \bar{r} \text{ and } \bar{r} \in rel_D(R).$$

As

$$r \in rel_D(R),$$

we obtain a contradicton.
Now let $r \in rel_D(R)$ and choose $\bar{r}_1, \bar{r}_2 \in chase_D(r)$. Then

$$\bar{d}_{(D,r,\bar{r}_1)}, \bar{d}_{(D,r,\bar{r}_2)} \in DOM(R)$$

exist by Theorem 4.1. We have

$$r\langle \bar{d}_{(D,r,\bar{r}_1)} \rangle \subseteq \bar{r}_1 \text{ and } r\langle \bar{d}_{(D,r,\bar{r}_2)} \rangle \subseteq \bar{r}_2.$$

From Lemma 4.5 we conclude

$$r\langle \bar{d}_{(D,r,\bar{r}_1)} \rangle \subseteq \bar{r}_2\langle \bar{d}_{(D,r,\bar{r}_1)} \rangle \subseteq \bar{r}_1$$

and

$$r\langle \bar{d}_{(D,r,\bar{r}_2)} \rangle \subseteq \bar{r}_1\langle \bar{d}_{(D,r,\bar{r}_2)} \rangle \subseteq \bar{r}_2.$$

Because of the properties of $\bar{d}_{(D,r,\bar{r}_1)}$ and $\bar{d}_{(D,r,\bar{r}_2)}$ we have

$$var(r\langle\bar{d}_{(D,r,\bar{r}_1)}\rangle) = var(\bar{r}_2\langle\bar{d}_{(D,r,\bar{r}_1)}\rangle) = var(\bar{r}_1)$$

and

$$var(r\langle\bar{d}_{(D,r,\bar{r}_2)}\rangle) = var(\bar{r}_1\langle\bar{d}_{(D,r,\bar{r}_2)}\rangle) = var(\bar{r}_2)$$

thus

$$\begin{aligned} var(\bar{r}_1) &= var(\bar{r}_2\langle\bar{d}_{(D,r,\bar{r}_1)}\rangle) = var(\bar{r}_1\langle\bar{d}_{(D,r,\bar{r}_2)}\rangle\langle\bar{d}_{(D,r,\bar{r}_1)}\rangle) \\ &= var(\bar{r}_1\langle\bar{d}_{(D,r,\bar{r}_1)} \circ \bar{d}_{(D,r,\bar{r}_2)}\rangle). \end{aligned}$$

Therefore

$$\bar{d}_{(D,r,\bar{r}_1)} \circ \bar{d}_{(D,r,\bar{r}_2)}\Big|_{var(\bar{r}_1)}$$

is bijective and so

$$\bar{d}_{(D,r,\bar{r}_2)}\Big|_{var(\bar{r}_1)}.$$

Now we conclude

$$|\bar{r}_1\langle\bar{d}_{(D,r,\bar{r}_2)}\rangle| = |\bar{r}_1|$$

and in a similar way

$$|\bar{r}_2\langle\bar{d}_{(D,r,\bar{r}_1)}\rangle| = |\bar{r}_2|.$$

This leads to

$$|\bar{r}_1| = |\bar{r}_1\langle\bar{d}_{(D,r,\bar{r}_2)}\rangle| \leq |\bar{r}_2| = |\bar{r}_2\langle\bar{d}_{(D,r,\bar{r}_1)}\rangle| \leq |\bar{r}_1|$$

thus

$$|\bar{r}_1| = |\bar{r}_2|$$

and we get

$$\bar{r}_2\langle\bar{d}_{(D,r,\bar{r}_1)}\rangle = \bar{r}_1 \text{ and } \bar{r}_1\langle\bar{d}_{(D,r,\bar{r}_2)}\rangle = \bar{r}_2$$

$$\bar{r}_1 = \bar{r}_2\langle\bar{d}_{(D,r,\bar{r}_1)}\rangle = \bar{r}_1\langle\bar{d}_{(D,r,\bar{r}_2)}\rangle\langle\bar{d}_{(D,r,\bar{r}_1)}\rangle = \bar{r}_1\langle\bar{d}_{(D,r,\bar{r}_1)} \circ \bar{d}_{(D,r,\bar{r}_2)}\rangle.$$

For $\bar{d}_{(D,r,\bar{r}_1)}$ and $\bar{d}_{(D,r,\bar{r}_2)}$ we have from Theorem 4.1

$$ord(\bar{d}_{(D,r,\bar{r}_1)}(x)) \leq ord(x) \text{ and } ord(\bar{d}_{(D,r,\bar{r}_2)}(x) \leq ord(x)$$

for all $x \in var(R)$. Thus

$$\bar{d}_{(D,r,\bar{r}_2)}\Big|_{var(r_1)} = id$$

and hence

$$\bar{r}_1 = \bar{r}_2.\square$$

Lemma 4.6: Let R be a relation scheme of discrete type and D a set of IDs for R. If

$$r \in rel(R),\ r^* \in sat^*(D) \text{ and } d \in DOM(R)$$

such that

$$r\langle d\rangle \subseteq r^*$$

then

$$r\langle d\rangle \subseteq chase_D(r)\langle d\rangle \subseteq chase_D(r\langle d\rangle) \subseteq r^*.$$

Proof: Because of $r\langle d\rangle \subseteq r^*$ it follows from Lemma 4.5 that

$$r\langle d\rangle \subseteq chase_D(r\langle d\rangle) \subseteq r^*$$

holds and also from Lemma 4.5 we conclude

$$r\langle d\rangle \subseteq chase_D(r)\langle d\rangle \subseteq chase_D(r\langle d\rangle)$$

as

$$chase_D(r\langle d\rangle) \in sat^*(D).\square$$

Corollary: Let R be a relation scheme of discrete type and D a set of IDs for R. Let

$$r_1, r_2 \in rel_D(R) \text{ and } r_1 \leq r_2$$

then

$$chase_D(r_1) \leq chase_D(r_2)$$

holds.

Lemma 4.7: Let R be a relation scheme of discrete type and D a set of IDs for R.

$$rel_D(R) = rel_{sat(D)}(R)$$

Proof: If $r \in rel_D(R)$, we have

$$chase_D(r) = r^* \neq \Xi \text{ and } r^* \in sat^*(D).$$

From Lemma 4.3 we know that there exists a mapping $e \in DOM(R)$ with

$$r^*\langle e\rangle \in sat(D).$$

As

$$r\langle e \circ d_{(D,r)}\rangle = r\langle d_{(D,r)}\rangle\langle e\rangle \subseteq r^*\langle e\rangle$$

we have

$$r^*\langle e\rangle \in comp_D(r)$$

hence

$$r \in rel_{sat(D)}(R).$$

If $r \in rel_{sat(D)}(R)$, then

$$comp_D(r) \neq \emptyset.$$

Thus there is a relation $r^* \in sat(D)$ with

$$r \leq r^*.$$

From $r^* \in sat(D)$ it follows that

$$r^* \in rel_D(R)$$

and therefore by the corollary of Lemma 4.5

$$r \in rel_D(R).\square$$

Lemma 4.8: Let R be a relation scheme of discrete type and D a set of IDs for R. If $r \in rel_D(R)$ then

(1) $comp_D(r) = comp_D(chase_D(r))$
(2) $chase_D(r) = chase_D(r\langle d_{(D,r)}\rangle)$.

Proof:

1. (a) Because of $r \leq chase_D(r)$ we conclude by Theorem 1.2 that $comp(r) \supseteq comp(chase_D(r))$ and therefore $comp_D(r) \supseteq comp_D(chase_D(r))$.
   (b) For each $\bar{r} \in comp_D(r) \subseteq sat(D) \subseteq sat^*(D)$, $r\langle d\rangle \subseteq \bar{r}$ for some $d \in DOM(R)$ holds. By Lemma 4.6 it follows that $chase_D(r)\langle d\rangle \subseteq \bar{r}$ and therefore $\bar{r} \in comp_D(chase_D(r))$.

2. Because of $r\langle d_{(D,r)}\rangle \subseteq chase_D(r)$ and $chase_D(r) \in sat^*(D)$ we conclude by Lemma 4.6 that

$$\begin{aligned} r\langle d_{(D,r)}\rangle &\subseteq chase_D(r)\langle d_{(D,r)}\rangle \\ &\subseteq chase_D(r\langle d_{(D,r)}\rangle) \\ &\subseteq chase_D(r). \end{aligned}$$

Since

$$var(r\langle d_{(D,r)}\rangle) = var(chase_D(r))$$

and

$$var(chase_D(r)) = var(chase_D(r)\langle d_{(D,r)}\rangle)$$

we get

$$chase_D(r) = chase_D(r)\langle d_{(D,r)}\rangle,$$

and finally

$$chase_D(r) = chase_D(r\langle d_{(D,r)}\rangle)$$

holds.□

We are now ready to state our first main theorem. It provides an operational characterization of redundancy.

Theorem 4.3: Let R be a relation scheme of discrete type and D a set of IDs for R. A relation $r \in sat(D)$ contains redundancy wrt. D iff a partial relation q exists with

$$q < r,\ |q| \leq |r| \text{ and } chase_D(q) \mathrel{\widehat{=}} r.$$

Proof:

i) Let $r \in sat(D)$ contain redundancy wrt. D. Hence there exists a partial relation $q \in rel_D(R)$ with

$$q < r,\ |q| \leq |r| \text{ and } min_D(q) = r.$$

We prove that $chase_D(q) \mathrel{\widehat{=}} r$ holds. Let $q^* = chase_D(q)$.
We know from Lemma 4.8 that

$$comp_D(q) = comp_D(q^*).$$

Moreover we know that

$$r = min_D(q) \text{ is minimal in } comp_D(q)$$

and hence r is minimal in $comp_D(q^*)$. Now we choose $e \in DOM(R)$ with

$$e(var(q^*)) \subseteq dom(R) \setminus dom(r)$$

and

$$e\Big|_{var(q^*)} \text{ is injective.}$$

Then $q^*\langle e\rangle \in sat(D)$ holds and so

$$q^*\langle e\rangle \in comp_D(q^*).$$

Since r is minimal we get

$$r \subseteq q^*\langle e\rangle$$

and because of our choice of e it follows that

$$r \subseteq q^* = chase_D(q).$$

Since $q < r$ we also have

$$chase_D(q) \leq r.$$

Hence

$$chase_D(q) \mathrel{\widehat{=}} r.$$

ii) Let $chase_D(q) \mathrel{\widehat{=}} r$ for some partial relation $q \in rel_D(R)$, $r \in sat(D)$ where $q < r$ and $|q| \leq |r|$. Then

$$comp_D(chase_D(q)) = comp_D(q) = comp_D(r)$$

by Lemma 4.8 and therefore we get

$$min_D(chase_D(q)) = min_D(q) = min_D(r) = r$$

hence r is a relation in $sat(D)$. □

Lemma 4.9: Let R be a relation scheme of discrete type and D a set of IDs for R. Furthermore, let $r \in rel_D(R)$ a partial relation, $r^* = chase_D(r)$ and $e \in DOM(R)$ a mapping with

(1) $e(var(r^*)) \subseteq dom(R) \setminus dom(r^*)$,
(2) $e\big|_{var(r^*)}$ is injective and
(3) $e\big|_{var(R)\setminus var(r^*)} = id$.

Then

$$chase_D(r)\langle e\rangle = chase_D(r\langle e\rangle) \in sat(D).$$

Proof:
i) $r\langle d_{(D,r)}\rangle \subseteq chase_D(r)$ by Theorem 4.1 and therefore

$$r\langle e \circ d_{(D,r)}\rangle = r\langle d_{(D,r)}\rangle\langle e\rangle \subseteq chase_D(r)\langle e\rangle.$$

Because of $e \circ d_{(D,r)} = e \circ d_{(D,r)} \circ e$ it follows that

$$\begin{aligned} r\langle e\rangle\langle e \circ d_{(D,r)}\rangle &= r\langle e \circ d_{(D,r)} \circ e\rangle = r\langle e \circ d_{(D,r)}\rangle = r\langle d_{(D,r)}\rangle\langle e\rangle \\ &\subseteq chase_D(r\langle d_{(D,r)}\rangle)\langle e\rangle = chase_D(r)\langle e\rangle \end{aligned}$$

by Lemma 4.6.
This yields $r\langle e\rangle \in rel_D(R)$, thus $chase_D(r) \neq \Xi$.
ii) Since $r\langle e\rangle \in rel_D(R)$, $d_{(D,r\langle e\rangle)} \in DOM(R)$ exists and $r\langle e\rangle\langle d_{(D,r\langle e\rangle)}\rangle \subseteq chase_D(r\langle e\rangle)$. By Lemma 4.6 we get

$$\begin{aligned} r\langle e\rangle\langle d_{(D,r\langle e\rangle)}\rangle &\subseteq chase_D(r)\langle e\rangle\langle d_{(D,r\langle e\rangle)}\rangle = chase_D(r)\langle e\rangle \\ &\subseteq chase_D(r\langle e\rangle), \end{aligned}$$

hence $var(chase_D(r)\langle e\rangle) = \emptyset$.
This implies $chase_D(r)\langle e\rangle \subseteq chase_D(r\langle e\rangle)$.
iii) As shown in ii),

$$r\langle e\rangle\langle d_{(D,r\langle e\rangle)}\rangle \subseteq chase_D(r)\langle e\rangle$$

holds. A further application of Lemma 4.6 leads to

$$\begin{aligned} r\langle e\rangle\langle d_{(D,r\langle e\rangle)}\rangle &\subseteq chase_D(r\langle e\rangle)\langle d_{(D,r\langle e\rangle)}\rangle \\ &\subseteq chase_D(r\langle e\rangle\langle d_{(D,r\langle e\rangle)}\rangle) \\ &\subseteq chase_D(r)\langle e\rangle, \end{aligned}$$

thus $chase_D(r\langle e\rangle) \subseteq chase_D(r)\langle e\rangle$. □

Definition 4.3: Let R be a relation scheme of discrete type and D a set of IDs for R. D admits (resp. does not admit) redundancy if there exists (resp. does not exist) $r \in sat(D)$ that contains redundancy.

Theorem 4.4: Let R be a relation scheme of discrete type and D a set of IDs for R. D admits redundancy, iff there exists a partial relation $q \in rel_D(R) \setminus sat^*(D)$ with

$$|q| \leq |chase_D(q)|.$$

Proof:
i) Let D admit redundancy.
Then there exist a relation $r \in sat(D)$ and a partial relation $q \in rel_D(R)$ with

$$q < r,\ |q| \leq |r| \text{ and } chase_D(q) \mathrel{\hat{=}} r.$$

Since $q < r$ and $chase_D(q) \mathrel{\hat{=}} r$ we have $q \neq chase_D(q)$ thus

$$q \in rel_D(R) \setminus sat^*(D).$$

Because of $chase_D(q) \mathrel{\hat{=}} r$ and $r \in sat(D)$ we conclude

$$r \subseteq chase_D(q)$$

hence

$$|r| \leq |chase_D(q)|$$

and therefore

$$|q| \leq |chase_D(q)|.$$

ii) Let us assume that we have a partial relation

$$q \in rel_D(R) \setminus sat^*(D) \text{ with } |q| \leq |chase_D(q)|.$$

If we set $r^* = chase_D(q)$ and define $e \in DOM(R)$ as in the previous lemma then we get

$$chase_D(q\langle e\rangle) = r^*\langle e\rangle$$

by this lemma.
Set $\bar{r} = r^*\langle e\rangle$.
Since e is injective on $var(r^*)$ and $e(x) = x$ for all $x \in var(q) \setminus var(r^*)$ we know that

$$e \text{ is injective on } var(q).$$

From this and the assumption we get

$$|q\langle e\rangle| \leq |chase_D(q)\langle e\rangle| = |\bar{r}|.$$

Now, from $chase_D(q\langle e\rangle) = \bar{r}$ we conclude that

$$q\langle e\rangle \leq \bar{r}.$$

We must show that $q\langle e\rangle < \bar{r}$ holds.
We assume that $\bar{r} \leq q\langle e\rangle$ holds.
Since $\bar{r} \in sat(D)$ we have

$$\bar{r} \subseteq q\langle e\rangle$$

and because of $|\bar{r}| \geq |q\langle e\rangle|$ we get

$$\bar{r} = q\langle e\rangle.$$

Thus, e evaluates all variables in q. Therefore

$$var(q) \subseteq var(chase_D(q))$$

because e evaluates at most all variables from $chase_D(q)$. The converse

$$var(chase_D(q)) \subseteq var(q)$$

holds always, hence

$$var(chase_D(q)) = var(q).$$

Consequently the chase-procedure does not replace variables in q. It follows that

$$d_{(D,q)} = id$$

and hence

$$q \subseteq chase_D(q).$$

By this and $chase_D(q)\langle e\rangle = q\langle e\rangle$ we get

$$q = chase_D(q).$$

But we know from the assumption that

$$q \neq chase_D(q).$$

This yields a contradiction and we know that $q\langle e\rangle < \bar{r}$ holds.
Now let us define $\bar{q} = q\langle e\rangle$. Then

$$\bar{q} < \bar{r},\ |\bar{q}| \leq |\bar{r}| \text{ and } chase_D(\bar{q}) \mathrel{\hat{=}} \bar{r}. \square$$

5. Redundancy and Normal Forms

Notation: Let R be a relation scheme of discrete type. For all attributes $A \in attr(R)$ we select variables $a_A, b_A \in var(R)$ where $ord(a_A) = 1$ and $ord(b_A) = 2$.
For $X \subseteq attr(R)$ we define $u_X \in tup(R)$ as

$$u_X(A) = \begin{cases} a_A & , A \in X \\ b_A & , A \notin X \end{cases}$$

and then $u_R = u_{attr(R)}$.
FDs $X \longrightarrow A$ resp. $MVDs$ $X \longrightarrow\!\!\!\!\rightarrow Y$ can be described as IDs $\langle(a_A/b_A); \{u_R, u_X\}\rangle$ resp. $\langle u_R; \{u_{XY}, u_{\overline{Y}}\}\rangle$ where $\overline{Y} = attr(R) \setminus Y$.
The definition of the notion "logically implies" ($\models$) is found e.g. in [4, 14].

Theorem 5.1: [4] Let R be a relation scheme of discrete type and D a set of IDs for R.

(1) D logically implies $\langle v; U\rangle$ iff $v \in chase_D(U)$.
(2) D logically implies $\langle(a/b); U\rangle$ iff $b \notin chase_D(U)$.

Using Lemma 8 in [4], we get the following theorem.

Theorem 5.2: Let R be a relation scheme of discrete type and D_1, D_2 two sets of IDs for R. $D_1 \models\!\!\dashv D_2$ iff $rel_{D_1}(R) = rel_{D_2}(R)$ and for all $q \in rel_{D_1}(R)$, $chase_{D_1}(q) = chase_{D_2}(q)$ holds.

Notation: Let R be a relation scheme of discrete type and D a set of IDs for R.

$key(D) \quad = \{X \to A;\ X \text{ is a key for } R \text{ wrt. } D\}$
$supkey(D) = \{X \to A;\ X \text{ is a superkey for } R \text{ wrt. } D\}$

As a simple consequence we get $key(D) \models\!\!\dashv supkey(D)$.

Definition 5.1: [8]
Let R be a relation scheme of discrete type and D a set of FDs for R. The relation scheme R together with D is in $BCNF$ iff

$$key(D) \models D$$

holds.

Theorem 5.3: Let R be a relation scheme of discrete type and D a set of FDs for R.
R together with D is in $BCNF$ iff D does not admit redundancy.

Proof: Let R in $BCNF$.
Thus we have $key(D) \models\!\!\dashv D$ and therefore $chase_D(q) = chase_{key(D)}(q)$ for $q \in rel_D(R)$. We must show that D does not admit redundancy.
Let q be an arbitrarily chosen partial relation from $rel_D(R) \setminus sat^*(D)$. It follows that $chase_{key(D)}(q) \neq q$. Thus we need at least one EGD-rule in the chase of q, say

$$F_f : q \longrightarrow \hat{q}.$$

Because we select F from $key(D)$, the left hand side of F, say K, is a key for D. However, by application of F, we can apply all FDs in $key(D)$, having K as left hand side directly to the two tuples concerned. Consequently, by the chase-procedure, we identify these both tuples. Since we do not use $TTGD$-rules we get $|chase_D(q)| < |q|$.
If D does not admit redundancy then we must prove that $key(D) \models D$. Hence let K be the left hand side of an arbitrary but nontrivial FD from D. We know that $|chase_D(U_K)| < |U_K|$ holds, where $U_K = \{u_K, u_R\}$, because $U_K \in rel_D(R) \setminus sat^*(D)$. Actually we have $chase_D(U_K) = u_R$, so that for all

$\langle (a_C/b_C); U_K\rangle$ with $C \in attr(R) \setminus K$, $a_C, b_C \in var(C)$ with $ord(a_C) = 1$ as well as $ord(b_C) = 2$, we have $b_C \notin var(chase_D(U_K))$. Accordingly we know from Theorem 5.1., that $D \models K \to C$ holds for all $C \in attr(R)$. Thus K is a superkey for R wrt. D. Hence $key(D) \models D$.□

Definition 5.2: [8]
Let R a be a relation scheme of discrete type and D a set of FDs and $MVDs$ for R. The relation scheme R together with D is in 4NF iff

$$key(D) \models D$$

holds.

Theorem 5.4: Let R be a relation scheme of discrete type and D a set of FDs and $MVDs$ for R. R together with D is in $4NF$ iff D does not admit redundancy.

Proof: If R is in $4NF$ then we show in the same way as in the previous theorem, that D does not admit redundancy. In order to prove the opposite direction of the proposition we will show

$$key(D) \models supkey(D) \models D.$$

If $F \in D$ is a FD we have already shown that $key(D) \models F$ holds. Therefore let now $F = X \to\to Y$ an arbitrary but nontrivial MVD from D. The associated $TTGD$ is $\langle u_R; \{u_{XY}, u_{XZ}\}\rangle$, where $Z = \overline{XY}$. Since F is nontrivial we have $\{u_{XY}, u_{XZ}\} \in rel_D(R) \setminus sat^*(D)$ and because D does not admit redundancy we get $|chase_D(\{u_{XY}, u_{XZ}\})| < |\{u_{XY}, u_{XZ}\}|$ thus $chase_D(\{u_{XY}, u_{XZ}\}) = u_R$. The latter holds because, during the chase-procedure, we only substitute variables with higher index by variables with lower index. However, this means that for all $G_C = \langle (a_C/b_C); \{u_{XY}, u_{XZ}\}\rangle$ we have $D \models G_C$.
Since $G_C \models\dashv X \to C$ for all $C \in \overline{X}$, X is a superkey for R wrt. D.
Hence for every nontrivial MVD $X \to\to Y$ from D, X is a superkey for R wrt. D. Thus we have $supkey(D) \models D$ and obviously $key(D) \models supkey(D)$, so $key(D) \models D$. Hence $key(D) \models\dashv D$. □

We conclude this section by considering another class of dependencies, the join dependencies (JD).
Join dependencies were introduced in [12]. In [1] Beeri et al. showed that a subclass of join dependencies, the so called *acyclic* JDs, are equivalent to a set of $MVDs$. Hence in the case of acyclic join dependencies Theorem 5.4 applies as well. If, however, we allow for cyclic join dependencies then it may happen that the dependencies do not admit redundancy but $key(D) \not\models D$.

Definition 5.3: Let $R(A_1|D_1, \ldots, A_n|D_n)$ be a relation scheme. If $\{A_{i1}, \ldots, A_{ik}\} \subseteq \{A_1, \ldots, A_n\}$ then

$$S = S(A_{i1}|D_{i1}, \ldots, A_{ik}|D_{ik}\}$$

is a subscheme of R with degree k.

Definition 5.4: Let R be a relation scheme and S a subscheme of R. If $r \in rel(R)$ is a partial relation for R, then by

$$\Pi_S(r) = \{t_S \in tup(S);\ \text{there is } t_R \in r \text{ with } t_S = t_R\mid_{attr(S)}\}$$

the *projection* from r to S is given.

Definition 5.5: Let R be a relation scheme and S_1, S_2 be two subschemes of R where $attr(R) = attr(S_1) \cup attr(S_2)$. If $s_1 \in rel(S_1)$ and $s_2 \in rel(S_2)$ then

$$r = s_1 \bowtie s_2 = \{t_r \in tup(R);\ \text{there is } t_{s_1} \in s_1, t_{s_2} \in s_2 \text{ with } t_r\mid_{attr(S_1)} = t_{s_1} \text{ and } t_r\mid_{attr(S_2)} = t_{s_2}\},$$

denotes the (*natural*) *join* for s_1 and s_2.

Definition 5.6: Let R be a relation scheme of discrete type and $X_1, \ldots, X_k \subseteq attr(R)$ sets of attributes of R where $attr(R) = \bigcup(X_i; 1 \leq i \leq k)$, then

$$\bowtie [X_1, \ldots, X_k]$$

denotes a *join dependency* (JD).
Let S_i with $attr(S_i) = X_i$, $i = 1, \ldots, k$, be subschemes of R. A relation $r \in rel(R)$ is valid for $\bowtie [X_1, \ldots, X_k]$ if

$$r = \Pi_{S_1}(r) \bowtie \cdots \bowtie \Pi_{S_k}(r).$$

Remark: Let R be a relation scheme of discrete type and $\bowtie [X_1, \cdots, X_k]$ a JD for R. As described in [13] one can formulate $\bowtie [X_1, \cdots, X_k]$ as a $TTGD$ in the following way :

For all $1 \leq i \leq k$ create a partial tuple u_i that has a variable with index 1 in all X_i-components and a variable that occurs in none of the other u_j, $(1 \leq j \neq j \leq k)$, otherwise. This yields the $TTGD$

$$\langle t; \{u_1, \cdots, u_k\}\rangle,$$

where t is a partial tuple that consists only of variables with index 1.

Example 5.1: Let $R(A, B, C)$ be a relation scheme of discrete type, $AB \rightarrow C$, $BC \rightarrow A$ FDs for R and $\bowtie [AB, BC, AC]$ a JD for R.
Let $D = \{AB \rightarrow C, BC \rightarrow A, \bowtie [AB, BC, AC]\}$.
The dependencies in D are equivalent to the $EGDs$ F_1, F_2 as depicted below.

$$\begin{array}{ccc} A & B & C \\ \hline A_1 & B_2 & C_1 \\ A_1 & B_1 & C_2 \\ A_2 & B_1 & C_1 \\ \hline \multicolumn{3}{c}{A_1 = A_2} \\ \hline \end{array} \qquad \begin{array}{ccc} A & B & C \\ \hline A_1 & B_2 & C_1 \\ A_1 & B_1 & C_2 \\ A_2 & B_1 & C_1 \\ \hline \multicolumn{3}{c}{C_1 = C_2} \\ \hline \end{array}$$

$EGD \quad F_1$ $\qquad$ $EGD \quad F_2$

Because of $D \models \{F_1, F_2\} = E$ we get $chase_D(r) = chase_E(r)$ for all partial relations $r \in rel_D(R)$. Either the application of F_1 (resp. F_2) identifies two tuples or after that the application of F_2 (resp. F_1) identifies two tuples.
From this it follows for all $r \in rel_D(r) \setminus sat^*(D)$ that

$$|chase_D(r)| = chase_E(r)| < |r|.$$

Thus D does not admit redundancy.
Furthermore

$$key(D) = \{AB \rightarrow C, BC \rightarrow A\}$$

holds but

$$key(D) \models D.$$

does not hold.
Thus we found a set of IDs that does not admit redundancy and yet is not logically implied from keys.

6. Discussion

In the previous section we gave a static and a dynamic characterization of redundancy for the case of a single relation scheme. Without giving a formal definition for update anomalies as discussed in [5, 6], we argue now that a relation scheme that admits redundancy with respect to a set *sat* of relations will also cause update anomalies and viceversa if we request that updating of a relation in *sat* results in a relation in *sat*. Informally a relation that contains redundancy with respect to *sat* contains some information more than once and hence updating of this information affects more than one item. Viceversa if in a relation update anomalies with respect to *sat* may occur, i.e. more than one item is affected if some piece of information is to be changed, then this means some form of redundancy is contained. As a very simple example consider $sat = \{r_1, r_2, r_3\}$ where

$$r_1 = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} \quad r_2 = \begin{pmatrix} a & b & c \\ d & e & f \\ g' & h' & i' \end{pmatrix} \quad r_3 = \begin{pmatrix} a & b & c \\ d & e & f \\ g'' & h'' & i'' \end{pmatrix}$$

An update operation on r_1 modifying g to g' causes an update dependency as h and i have to be changed as well. Clearly r_1, r_2, r_3 exibit redundancy. In each case a partial relation q_i exists that satisfies $|q_i| \leq |r_i|, q_i < r_i$ and $r_i = \min_{sat}(q_i)$.

We showed before that in the case where *sat* is characterized by some set of functional (resp. multivalued) dependencies, $BCNF$ (resp. $4NF$) is a necessary and sufficient condition to avoid redundancy (and thus update anomalies).

Let us now assume that we have to deal with a relation scheme R and a set F of functional dependencies, that admits redundancy (and hence update anomalies). One way to handle this situation could be to construct a loss-less join decomposition of the scheme into $BNCF$. Let us assume we found such a decomposition that also preserves the functional dependencies, as e.g. in the example of [6], where

$$R = (Empl\ \#, Tel\ \#, Project\ \#)$$

with $Empl\ \# \rightarrow Tel\ \#$. Then the $BCNF$ decomposition

$$\rho = (R_1, R_2)$$

where $R_1 = (Empl\ \#, Tel\ \#)$ und $R_2 = (Empl\ \#, Project\ \#)$ indeed gets rid of the redundant storing of the telephone number for employees that work in more than one project. So substituting a relation r for R that satisfies $Empl\ \# \rightarrow Tel\ \#$ by the two projected relations $\Pi_{R_1}(r)$ and $\Pi_{R_2}(r)$ solves the problem of reduncancy and update anomalies. However, one may easily construct examples where a desired decomposition into $BCNF$ exists but the update anomaly problem is even worse after decomposition. Such an example is given in the following: We consider a relation scheme $BCEF$ and dependencies $BC \rightarrow F, CE \rightarrow F$. We consider the decomposition $(R_1, R_2, R_3) = (BCE, BCF, CEF)$. Then each R_i is in $BCNF$, the decomposition is a lossless-join decomposition and all functional dependencies are preserved. Let us now consider the relation r and its projections on R_i as follows:

$$r = \begin{array}{c} \begin{array}{cccc} B & C & E & F \end{array} \\ \hline \begin{pmatrix} b & c & e & f \\ b & c & e' & f \\ b' & c & e' & f \end{pmatrix} \end{array} \quad r_1 = \begin{array}{c} \begin{array}{ccc} B & C & E \end{array} \\ \hline \begin{pmatrix} b & c & e \\ b & c & e' \\ b' & c & e' \end{pmatrix} \end{array} \quad r_2 = \begin{array}{c} \begin{array}{ccc} B & C & F \end{array} \\ \hline \begin{pmatrix} b & c & f \\ b' & c & f \end{pmatrix} \end{array} \quad r_3 = \begin{array}{c} \begin{array}{ccc} C & E & F \end{array} \\ \hline \begin{pmatrix} c & e & f \\ c & e' & f \end{pmatrix} \end{array}$$

Now changing f to $\tilde{f}$ in the second tuple of r forces us to change f also in the other tuples of r, i.e. 3 values are affected. In the decomposed case, we have to modify all f's in r_2 and r_3!

The above examples show that there are cases where a $BCNF$ decomposition may help in getting rid of the redundancy (update anomaly) problem and cases where it does not help. It is an interesting problem to provide a criterion that tells us when decomposition is useful and when it is not. For this a formal framework has to be given, in which we may talk about redundancy and update anomalies for a set of relation schemes.

References

[1] C. BEERI, R. FAGIN, D. MAIER, and M. YANNAKAKIS. On the Desirability of Ayclic Database Schemes. *Journal of the ACM*, 30(3):479 – 513, July 1983.

[2] C. BEERI and M. Y. VARDI. The Implication Problem for Data Dependencies. *Lecture Notes in Computer Science*, vol. 115; 73 – 85. Springer Verlag, New York, 1981.

[3] C. BEERI and M. Y. VARDI. Formal Systems for Tupel and Equality Generating Dependencies. *SIAM Journal of Computing*, 13(1):76 – 98, February 1984.

[4] C. BEERI and M. Y. VARDI. A Proof Procedure for Data Dependencies. *Journal of the ACM*, 31(4):718 – 741, October 1984.

[5] E. F. CODD. A Relational Model of Data for Large Shared Data Banks. *Communications of the ACM*, 13(6):377 – 387, June 1970.

[6] E. F. CODD. Further Normalization of the Data Base Relational Model. In: *Rustin, R.: Data Base Systems*, Courant Inst. Comp. Sci. Symp. 6, 33 – 64, *Prentice-Hall, Englewood Cliffs, N.Y.*, 1971.

[7] R. FAGIN. Multivalued Dependencies and a New Normal Form for Relational Databases. *ACM Transactions on Database Systems*, 2(3):262 – 278, September 1977.

[8] R. FAGIN. Normal Forms and Relational Database Operators. In: *Proc. ACM SIGMOD*, 153–160, 1979.

[9] R. FAGIN. Horn Clauses and Database Dependencies. *Journal of the ACM*, 29(4):952 – 985, October 1982.

[10] D. MAIER. The Theory of Relational Databases. *Computer Science Press*, Rockville, 1983.

[11] J. A. MAKOWSKY. Characterizing Data Base Dependencies. *Lecture Notes in Computer Science*, vol. 115; 86 – 97. Springer Verlag, New York, 1981.

[12] J. RISSANEN. Theory of Relations for Databases — a Tutorial Survey. *Lecture Notes in Computer Science*, vol. 64; 537 – 551. Springer Verlag, New York, 1978.

[13] E. SCIORE. A Complete Axiomatization of Full Join Dependencies. *Journal of the ACM*, 29(2):373 – 393, April 1982.

[14] J. D. ULLMAN. Principles of Database Systems. *Computer Science Press*, Rockville, 1983.

[15] M. YANNAKAKIS and C. H. PAPADIMITRIOU. Algebraic dependencies. *Journal of Computer and System Siences*, 25:2 – 41, 1982.