Computers Math. Applic. Vol. 26, No. 4, pp. 43–55, 1993
Printed in Great Britain.

0898-1221/93 $6.00 + 0.00

Algorithms for Generating an Armstrong Relation and Inferring Functional Dependencies in the Relational Datamodel

J. DEMETROVICS AND V. D. THI
Computer and Automation Institute, Hungarian Academy of Sciences
P.O. Box 63, Budapest, Hungary, H-1502

(Received October 1992; accepted November 1992)

Abstract—The main purpose of this paper is to give some new combinatorial algorithms for generating an Armstrong relation from a given relation scheme S and inferring functional dependencies (FDs) which hold in a relation R. We estimate the time complexities of them. It is known that worst-case time complexities of generating an Armstrong relation and inferring the FDs are exponential. However, we show that our algorithms are effective in many cases, i.e., in these cases, they require polynomial time in the size of S (in a case generating an Armstrong relation), in the size of R (in a case inferring the FDs). For BCNF class of relations and relation schemes we also present two effective algorithms constructing an Armstrong relation and finding a set of FDs. We give a class of relations and relation schemes for which generating an Armstrong relation and inferring the FDs are solved in polynomial time. In this paper, we prove that if relations and relation schemes satisfy certain additional properties then the FD-relation equivalence problem is solved in polynomial time. The keys play important roles in logical and structural investigations for FD in the relational datamodel. This paper gives some results concerning keys of relation and relation scheme.

Keywords—Relation, Relational datamodel, Functional dependency, Relation scheme, Generating an Armstrong relation, Dependency inference, Closure, Closed set, Minimal generator, Key, Minimal key, Antikey.

1. INTRODUCTION

Generating an Armstrong relation from given relation scheme and inferring FDs that hold in a relation play very important roles in logical and structural investigation of the relational datamodel both in practice and design theory. Several algorithms for generating an Armstrong relation and inferring FDs are known. For example, Beeri *et al.* [1] give an algorithm for generating an Armstrong relation from a given relation scheme $S = \langle U, F\rangle$. This algorithm constructs a family of all closed sets $CL(F)$. After that, from $CL(F)$, it computes a minimal generator $GEN(F)$. Thus, in all cases, the time complexity of this algorithm is always exponential in the size of S. In this paper, we give an algorithm for generating an Armstrong relation and an algorithm for inferring FDs. We show that in many cases time complexities of these algorithms are polynomial. Several types of families of FDs are introduced under the name normal forms (NFs). The most desirable NF is Boyce-Codd normal form (BCNF) which has been investigated in a lot of papers. We present two algorithms for generating an Armstrong relation and for inferring FDs in BCNF.

The paper is structured as follows. In Section 2, we give an algorithm for generating an Armstrong relation from a given arbitrary relation scheme and an algorithm for inferring FDs

Research supported by Hungarian Foundation for Scientific Research Grant 2575. The authors are grateful to Uhrin Bela for useful comments to the first version of the manuscript.

Typeset by $\mathcal{A}_{\mathcal{M}}\mathcal{S}$-TEX

that hold in a relation, i.e., for a given relation R this algorithm finds a relation scheme S such that R is an Armstrong relation of S. Worst-case time complexites of these algorithms are estimated. Section 3 gives two combinatorial algorithms for generating an Armstrong relation and for inferring FDs in BCNF. In this section, we show that if relations and relation schemes satisfy certain additional properties, then generating an Armstrong relation, inferring FDs and FD-relation equivalence problem are solved in polynomial time. Section 4 gives some problems for further research. Let us give some necessary definitions that are used in next sections.

DEFINITION 1.1. *Let $R = \{h_1, \ldots, h_m\}$ be a relation over U, and $A, B \subseteq U$. Then we say that B functionally depends on A in R (denoted $A \xrightarrow[R]{f} B$) iff*

$$(\forall\, h_i, h_j \in R)\,(\forall\, a \in A)\,(h_i(a) = h_j(a)) \Longrightarrow (\forall\, b \in B)\,(h_i(b) = h_j(b))).$$

Let $F_R = \{(A, B) : A, B \subseteq U,\ A \xrightarrow[R]{f} B\}$. F_R is called the full family of functional dependencies of R. We write (A, B) or $A \to B$ for $A \xrightarrow[R]{f} B$ when R, f are clear from the context.

DEFINITION 1.2. *A functional dependency over U is a statement of the form $A \to B$, where $A, B \subseteq U$. The FD $A \to B$ holds in a relation R if $A \xrightarrow[R]{f} B$. We also say that R satisfies the FD $A \to B$.*

DEFINITION 1.3. *Let U be a finite set, and denote $P(U)$ its power set. Let $Y \subseteq P(U) \times P(U)$. We say that Y is an f-family over U iff for all $A, B, C, D \subseteq U$*

(1) $(A, A) \in Y$,
(2) $(A, B) \in Y,\ (B, C) \in Y \Longrightarrow (A, C) \in Y$,
(3) $(A, B) \in Y,\ A \subseteq C,\ D \subseteq B \Longrightarrow (C, D) \in Y$,
(4) $(A, B) \in Y,\ (C, D) \in Y \Longrightarrow (A \cup C, B \cup D) \in Y$.

Clearly, F_R is an f-family over U. It is known [2] that if Y is an arbitrary f-family, then there is a relation R over U such that $F_R = Y$.

DEFINITION 1.4. *A relation scheme S is a pair $\langle U, F\rangle$, where U is a set of attributes, and F is a set of FDs over U. Let F^+ be a set of all FDs that can be derived from F by the rules in Definition 1.3. Denote $A^+ = \{a : A \to \{a\} \in F^+\}$. A^+ is called the closure of A over S. It is clear that $A \to B \in F^+$ iff $B \subseteq A^+$.*

Clearly, if $S = \langle U, F\rangle$ is a relation scheme, then there is a relation R over U such that $F_R = F^+$ (see [2]). Such a relation is called an Armstrong relation of S.

DEFINITION 1.5. *Let R be a relation, $S = \langle U, F\rangle$ be a relation scheme, Y be an f-family over U and $A \subseteq U$. Then A is a key of R (a key of S, a key of Y) if $A \xrightarrow[R]{f} U$ ($A \to U \in F^+, (A, U) \in Y$). A is a minimal key of $R(S, Y)$ if A is a key of $R(S, Y)$, and any proper subset of A is not a key of $R(S, Y)$. Denote $K_R, (K_S, K_Y)$ the set of all minimal keys of $R(S, Y)$.*

Clearly, K_R, K_S, K_Y are Sperner systems over U.

DEFINITION 1.6. *Let K be a Sperner system over U. We define the set of antikeys of K, denoted by K^{-1}, as follows:*

$$K^{-1} = \{A \subset U : (B \in K) \Longrightarrow (B \not\subseteq A) \text{ and } (A \subset C) \Longrightarrow (\exists B \in K)\,(B \subseteq C)\}$$

It is easy to see that K^{-1} is also a Sperner system over U.

It is known [3] that if K is an arbitrary Sperner system, then there is a relation scheme S such that $K_S = K$.

In this paper we always assume that if a Sperner system plays the role of the set of minimal keys (antikeys), then this Sperner system is not empty (doesn't contain U). We consider the comparison of two attributes as an elementary step of algorithms. Thus, if we assume that subsets of U are represented as sorted lists of attributes, then a Boolean operation on two subsets of U requires at most $|U|$ elementary steps.

DEFINITION 1.7. *Let $I \subseteq P(U)$, $U \in I$, and $A, B \in I \Longrightarrow A \cap B \in I$. Let $M \subseteq P(U)$. Denote $M^+ = \{\cap M' : M' \subseteq M\}$. We say that M is a generator of I iff $M^+ = I$. Note that $U \in M^+$ but not in M, since it is the intersection of the empty collection of sets.*

Denote $N = \{A \in I : A \neq \cap\{A' \in I : A \subset A'\}\}$.

In [3], it is proved that N is the unique minimal generator of I. Thus, for any generator N' of I we obtain $N \subset N'$.

DEFINITION 1.8. *Let R be a relation over U, and E_R the equality set of R, i.e., $E_R = \{E_{ij} : 1 \leq i < j \leq |R|\}$, where $E_{ij} = \{a \in U : h_i(a) = h_j(a)\}$. Let $T_R = \{A \in P(U) : \exists\, E_{ij} = A,\ \nexists\, E_{pq} : A \subset E_{pq}\}$. Then T_R is called the maximal equality system of R.*

DEFINITION 1.9. *Let R be a relation, and K a Sperner system over U. We say that R represents K iff $K_R = K$.*

The following theorem is known [4].

THEOREM 1.10. *Let K be a non-empty Sperner system and R a relation over U. Then R represents K iff $K^{-1} = T_R$, where T_R is the maximal equality system of R.*

2. GENERATING AN ARMSTRONG RELATION AND INFERRING FUNCTIONAL DEPENDENCIES

In this section, we construct two combinatorial algorithms for generating an Armstrong relation and inferring functional dependencies from a given relation scheme and a relation. We estimate these algorithms.

First we give an algorithm that finds the set of antikeys from a given Sperner system.

ALGORITHM 2.1. FINDING A SET OF ANTIKEYS.

Input: Let $K = \{B_1, \dots, B_m\}$ be a Sperner system over U.

Output: K^{-1}.

Step 1: We set $K_1 = \{U - \{a\} : a \in B_1\}$. It is obvious that $K_1 = \{B_1\}^{-1}$.

Step $q+1$: ($q < m$) We assume that $K_q = F_q \cup \{X_1, \dots, X_{t_q}\}$, where $X_1, \dots, X_{t_q}$ containing B_{q+1} and $F_q = \{A \in K_q : B_{q+1} \not\subseteq A\}$. For all i ($i = 1, \dots, t_q$) we construct the antikeys of $\{B_{q+1}\}$ on X_i in an analogous way as K_1. Denote them by $A^i_1, \dots, A^i_{r_i}$ $(i = 1, \dots, t_q)$. Let

$$K_{q+1} = F_q \cup \{A^i_p : A \in F_q \Longrightarrow A^i_p \not\subset A,\ 1 \leq i \leq t_q,\ 1 \leq p \leq r_i\}.$$

We set $K^{-1} = K_m$.

THEOREM 2.2. *[5] For every q $(1 \leq q \leq m)$, $K_q = \{B_1, \dots, B_q\}^{-1}$, i.e., $K_m = K^{-1}$.*

It can be seen that K and K^{-1} are uniquely determined by one another and the determination of K^{-1} based on our algorithm does not depend on the order of $B_1, \dots, B_m$. Denote $K_q = F_q \cup \{X_1, \dots, X_{t_q}\}$ and l_q $(1 \leq q \leq m-1)$ is the number of elements of K_q.

PROPOSITION 2.3. *[5] The worst-case time complexity of Algorithm 2.1 is*

$$O\Big(|U|^2 \sum_{q=1}^{m-1} t_q\, u_q\Big),$$

where

$$u_q = \begin{cases} l_q - t_q, & \text{if } l_q > t_q, \\ 1, & \text{if } l_q = t_q. \end{cases}$$

Clearly, in each step of our algorithm K_q is a Sperner system. It is known [6] that the size of arbitrary Sperner system over U cannot be greater than $C_n^{[n/2]}$, where $n = |U|$. $C_n^{[n/2]}$ is asymptotically equal to $2^{n+1/2}/(\pi \cdot n^{1/2})$. From this, the worst-case time complexity of our algorithm cannot be more than exponential in the number of attributes. In cases for which $l_q \leq l_m (q = 1, \dots, m-1)$, it is easy to see that the time complexity of our algorithm is not greater than $O(|U|^2 |K| |K^{-1}|^2)$. Thus, in these cases, Algorithm 2.1 finds K^{-1} in polynomial time in $|U|$, $|K|$, and $|K^{-1}|$. It can be seen that if the number of elements of K is small then Algorithm 2.1 is very effective. It only requires polynomial time in $|U|$.

DEFINITION 2.4. *Let $S = \langle U, F \rangle$ be a relation scheme, $a \in U$. Denote $K_a = \{A \subseteq U : A \to \{a\}, \nexists B : (B \to \{a\})(B \subset A)\}$. K_a is called the family of minimal sets of the attribute a.*

Clearly, $U \notin K_a$, $\{a\} \in K_a$ and K_a is a Sperner system over U.

ALGORITHM 2.5. FINDING A MINIMAL SET OF THE ATTRIBUTE a.

Input: Let $S = \langle U = \{a_1, \dots, a_n\}, F \rangle$ be a relation scheme, $a = a_1$.

Output: $A \in K_a$.

Step 1: We set $L(0) = U$.

Step $i+1$: Set

$$L(i+1) = \begin{cases} L(i) - a_{i+1}, & \text{if } L(i) - a_{i+1} \to \{a\}, \\ L(i), & \text{otherwise.} \end{cases}$$

Then we set $A = L(n)$.

LEMMA 2.6. *$L(n) \in K_a$.*

PROOF. By the induction, it can be seen that $L(n) \to \{a\}$, and $L(n) \subseteq \cdots \subseteq L(0)$ (1). If $L(n) = a$, then by the definition of the minimal set of attribute a, we obtain $L(n) \in K_a$. Now we suppose that there is a B such that $B \subset L(n)$ and $B \neq \emptyset$. Thus, there exists a_j such that $a_j \notin B$, $a_j \in L(n)$. According to the construction of the algorithm, we have $L(j-1) - a_j \not\to \{a\}$. It is obvious that by (1) we obtain $L(n) - a_j \subseteq L(j-1) - a_j$ (2). It is clear that $B \subseteq L(n) - a_j$. From (1),(2) we have $B \not\to \{a\}$. The lemma is proved. ∎

Clearly, by the linear-time membership algorithm in [7], the time complexity of Algorithm 2.5 is $O(|U|^2 |F|)$.

LEMMA 2.7. *Let $S = \langle U, F \rangle$ be a relation scheme, $a \in U$, K_a be a family of minimal sets of a, $L \subseteq K_a, \{a\} \in L$. Then $L \subset K_a$ if and only if there are C, $A \to B$ such that $C \in L$ and $A \to B \in F$ and $\forall E \in L \Longrightarrow E \not\subseteq A \cup (C - B)$.*

PROOF. $\Longrightarrow$: We assume that $L \subset K_a$. Consequently, there exists $D \in K_a - L$. By $\{a\} \in L$ and K_a is a Sperner system over U, we can construct a maximal set Q such that $D \subseteq Q \subset U$ and $L \cup Q$ is a Sperner system. From the definition of K_a, we obtain $Q \to \{a\}$ (1) and $a \notin Q$ (2). If $A \to B \in F$ implies $(A \subseteq Q, B \subseteq Q)$ or $A \not\subseteq Q$ then $Q^+ = Q$. By (2) $Q \not\to \{a\}$. This conflicts with (1). Consequently, there is a FD $A \to B$ such that $A \subseteq Q$, $B \not\subseteq Q$. From the construction of Q there is C such that $C \in L$, $A \subseteq Q$, $C - B \subseteq Q$. It is obvious that $A \cup (C - B) \subseteq Q$. Clearly, $E \not\subseteq A \cup (C - B)$ for all $E \in L$.

$\Longleftarrow$: We assume that there are C, and $A \to B$ such that $C \in L, A \to B \in F$ and $E \not\subseteq A \cup (C - B)$ for all $E \in L$(3). By the definition of L we obtain $A \cup (C - B) \to \{a\}$. By $\{a\} \in L$ there is D such that $D \in K_a$, $a \notin D$, $D \subseteq A \cup (C - B)$. By (3) $D \in K_a - L$. Our proof is complete. ∎

Based on this lemma and Algorithm 2.5, we construct the following algorithm by induction.

ALGORITHM 2.8. FINDING A FAMILY OF MINIMAL SETS OF ATTRIBUTE a.

Input: Let $S = \langle U, F\rangle$ be a relation scheme, $a \in U$.

Output: K_a.

Step 1: Set $L(1) = E_1 = \{a\}$.

Step $i+1$: If there are C and $A \to B$ such that $C \in L(i)$, $A \to B \in F$, $\forall\, E \in L(i) \Longrightarrow E \not\subseteq A \cup (C - B)$, then by Algorithm 2.5 we construct an E_{i+1}, where $E_{i+1} \subseteq A \cup (C-B)$, $E_{i+1} \in K_a$. We set $K(i+1) = K(i) \cup E_{i+1}$. In the converse case, we set $K_a = L(i)$.

By Lemma 2.7, it is obvious that there exists a natural number t such that $K_a = L(t)$.

It can be seen that the worst-case time complexity of algorithm is $O(|U|\,|F|\,|K_a|\,(|U| + |K_a|))$. Thus, the time complexity of this algorithm is polynomial in $|U|, |F|$, and $|K_a|$.

Clearly, if the number of elements of K_a for a relation scheme $S = \langle U, F\rangle$ is polynomial in the size of S, then this algorithm is effective. Especially, when $|K_a|$ is small.

It is obvious that if for each $A \to B \in F$ implies $a \in A$ or $a \notin B$, then $K_a = \{a\}$.

REMARK 2.9. It is known [8] that if $S = \langle U, F\rangle$ is a relation scheme, $Z(F) = \{A : A^+ = A\}$ and $N(F)$ is a minimal generator of $Z(F)$, then

$$N(F) = MAX(F^+) = \bigcup_{a \in U} MAX(F^+, a),$$

where

$$MAX(F^+, a) = \{A \subseteq U : A \to \{a\} \notin F^+,\ A \subset B \Longrightarrow B \to \{a\} \in F^+\}.$$

Clearly, K_a is a Sperner system over U. It can be seen that $MAX(F^+, a)$ is a set of antikeys of K_a for all $a \in U$. Thus, $MAX(F^+, a) = K_a^{-1}$.

THEOREM 2.10. *[9] Let $R = \{h_1, \ldots, h_m\}$ be a relation, and F an f-family over U. Then $F_R = F$ iff for every $A \in P(U)$*

$$L_F(A) = \begin{cases} \bigcap\limits_{A \subseteq E_{ij}} E_{ij}, & \text{if } \exists\, E_{ij} \in E_R : A \subseteq E_{ij}, \\ U, & \text{otherwise,} \end{cases}$$

where $L_F(A) = \{a \in U : (A, \{a\}) \in F\}$ and E_R is the equality set of R.

Based on Remark 2.9, Theorem 2.10, Algorithms 2.1, 2.8, we construct an algorithm for generating an Armstrong relation from a given relation scheme, as follows:

ALGORITHM 2.11. GENERATING AN ARMSTRONG RELATION.

Input: Let $S = \langle U, F\rangle$ be a relation scheme .

Output: A relation R such that $F_R = F^+$.

Step 1: For each $a \in U$ by Algorithm 2.8 we compute K_a, and from Algorithm 2.1 construct the set of antikeys K_a^{-1}.

Step 2: $N = \bigcup\limits_{a \in U} K_a^{-1}$.

Step 3: Denote elements of N by $A_1, \ldots, A_t$, we construct a relation $R = \{h_0, h_1, \ldots, h_t\}$ as follows: For all $a \in U$, $h_0(a) = 0$, $\forall i = 1, \ldots, t$

$$h_i(a) = \begin{cases} 0, & \text{if } a \in A_i, \\ i, & \text{otherwise.} \end{cases}$$

By Remark 2.9 we obtain $N = N(F)$ and from Theorem 2.10 $F_R = F^+$ holds.

REMARK 2.12. Clearly, if we have $N(F)$, then we can directly construct R. The complexity of this construction depends on $|N(F)|$. It can be seen that the complexity of Algorithm 2.11 is the complexity of Step 1. By Proposition 2.3 and the estimation of Algorithm 2.8, it is easy to see that the worst-case time complexity of Algorithm 2.11 is

$$O\left(n \sum_{i=1}^{n}\left(n \sum_{q=1}^{m_i-1} t_{iq}\, u_{iq} + |F|\, m_i\,(m_i+n)\right)\right),$$

where $U = \{a_1, \ldots, a_n\}$, $m_i = |K_{a_i}|$ and

$$u_{iq} = \begin{cases} l_{iq} - t_{iq}, & \text{if } l_{iq} > t_{iq}, \\ 1, & l_{iq} = t_{iq}. \end{cases}$$

In cases for which $l_{iq} \leq l_{m_i} (\forall\, i, \forall\, q : 1 \leq q \leq m_i)$, the time complexity of our algorithm is $O\left(n \sum_{i=1}^{n} |K_{a_i}|\,(n|F| + |K_{a_i}|\,|F| + n\,|K_{a_i}^{-1}|^2)\right)$. Thus, the complexity of Algorithm 2.11 is polynomial in $|U|$, $|F|$, $|K_{a_i}|$, $|K_{a_i}^{-1}|$. Clearly, in these cases if $|K_{a_i}|$ and $|K_{a_i}^{-1}|$ are polynomial (especially, if they are small) in $|U|$ and $|F|$, then our algorithm is effective.

Now we use Algorithm 2.11 to construct an Armstrong relation for a relation scheme in the following example.

EXAMPLE 2.13. Let $S = \langle U, F\rangle$ be a relation scheme, where $U = \{a, b, c, d\}$ and $F = \{\{a, d\} \to U, \{a\} \to \{a, b, c\}, \{b, d\} \to \{b, c, d\}\}$.

By Algorithm 2.8, we obtain $K_a = \{a\}$, $K_b = \{\{a\}, \{b\}\}$, $K_c = \{\{a\}, \{b, d\}, \{c\}\}$, $K_d = \{d\}$.

Based on Algorithm 2.1, we have $K_a^{-1} = \{b, c, d\}$, $K_b^{-1} = \{c, d\}$, $K_c^{-1} = \{\{b\}, \{d\}\}$, $K_d^{-1} = \{a, b, c\}$.

Consequently, $N(F) = \{\{a, b, c\}, \{b, c, d\}, \{c, d\}, \{b\}, \{d\}\}$. Then we construct a relation R as follows:

a	b	c	d
0	0	0	0
0	0	0	1
2	0	0	0
3	3	0	0
4	0	4	4
5	5	5	0

Now we construct an algorithm for inferring FDs from a given relation.

ALGORITHM 2.14. [4] FINDING A MINIMAL KEY FROM A SET OF ANTIKEYS.

Input: Let K be a Sperner system, H a Sperner system, and $C = \{b_1, \ldots, b_m\} \subseteq U$ such that $H^{-1} = K$ and $\exists\, B \in K : B \subseteq C$.

Output: $D \in H$.

Step 1: Set $T(0) = C$.

Step $i+1$: Set $T = T(i) - b_{i+1}$.

$$T(i+1) = \begin{cases} T, & \text{if } \forall\, B \in K : T \not\subseteq B, \\ T(i), & \text{otherwise.} \end{cases}$$

We set $D = T(m)$.

LEMMA 2.15. *[4] If K is a set of antikeys, then $T(m) \in H$.*

LEMMA 2.16. *[4] Let H be a Sperner system over U, and $H^{-1} = \{B_1, \ldots, B_m\}$ be a set of antikeys of H, $T \subseteq H$. Then $T \subset H$, $T \neq \emptyset$ if and only if there is a $B \subseteq U$ such that $B \in T^{-1}$, $B \not\subseteq B_i (\forall\, i : 1 \leq i \leq m)$.*

Based on Lemma 2.16 and from Algorithm 2.14 we have the following algorithm.

ALGORITHM 2.17. FINDING A SET OF MINIMAL KEYS FROM A SET OF ANTIKEYS.

Input: Let $K = \{B_1, \ldots, B_k\}$ be a Sperner system over U.

Output: H such that $H^{-1} = K$.

Step 1: By Algorithm 2.14 we compute an A_1, set $K(1) = A_1$.

Step $i+1$: If there exists a $B \in K_i^{-1}$ such that $B \not\subseteq B_j (\forall\, j : 1 \leq j \leq k)$, then by Algorithm 2.14 we compute an A_{i+1}, where $A_{i+1} \in H$, $A_{i+1} \subseteq B$. Set $K(i+1) = K(i) \cup A_{i+1}$. In the converse case, we set $H = K(i)$.

PROPOSITION 2.18. *[10] The time complexity of Algorithm 2.17 is* $O\left(n\left(\sum_{q=1}^{m-1}(k\, l_q + n\, t_q\, u_q) + k^2 + n\right)\right)$, *where* $|U| = n$, $|K| = k$, $|H| = m$, *meaning of* l_q, t_q, u_q *see Proposition 2.3.*

Clearly, in cases for which $l_q \leq k\, (\forall\, q : 1 \leq q \leq m-1)$ *the time complexity of our algorithm is* $O(|U|^2\, |K|^2\, |H|)$. *It is easy to see that in these cases Algorithm 2.17 finds the set of minimal keys in polynomial time in the sizes of* U, K, H. *If* $|H|$ *is polynomial in* $|U|$ *and* $|K|$, *then our algorithm is effective. It can be seen that if the number of elements of* H *is small then Algorithm 2.17 is very effective.*

LEMMA 2.19. *Let* F *be an* f*-family over* U, $a \in U$. *Denote* $L_F(A) = \{a \in U : (A, \{a\}) \in F\}$, $Z_F = \{A : L_F(A) = A\}$. *Clearly,* $U \in Z_F$, $A, B \in Z_F \Longrightarrow A \cap B \in Z_F$. *Denote by* N_F *the minimal generator of* Z_F. *Set* $M_a = \{A \in N_F : a \notin A, \nexists B \in N_F : a \notin B, A \subset B\}$. *Then* $M_a = MAX(F, a)$, *where* $MAX(F, a) = \{A \subseteq U : A$ *is a nonempty maximal set such that* $(A, \{a\}) \notin F\}$.

PROOF. It is known [8] that $MAX(F, a) \subseteq N_F$ holds (1). Assume that $A \in M_a$. By $A \in N_F$, i.e., $L_F(A) = A$, and $a \notin A$, we obtain $(A, \{a\}) \notin F$. From (1) and according to the definition of M_a we have $A \in MAX(F, a)$. Conversely, if $A \in MAX(F, a)$ then by (1) $A \in N_F$ holds(2). By $(A, \{a\}) \notin F$ and from (2) we obtain $a \notin A$. According to the definition of $MAX(F, a)$ we have $A \in M_a$. Our proof is complete. ∎

Based on Algorithm 2.17 and Lemma 2.19, we construct the following algorithm for inferring FDs from a given relation.

ALGORITHM 2.20. INFERRING FDS FROM A GIVEN RELATION.

Input: R be a relation over U.

Output: $S = \langle U, F\rangle$ such that $F^+ = F_R$.

Step 1: From R we compute the equality set E_R.

Step 2: Set $N_R = \{A \in E_R : A \neq \cap\{B \in E_R : A \subset B\}\}$.

Step 3: For each $a \in U$ we construct $N_a = \{A \in N_R : a \notin A, \nexists B \in N_R : a \notin B, A \subset B\}$. After that, by Algorithm 2.17, we construct the family H_a $({H_a}^{-1} = N_a)$.

Step 4: Construct $S = \langle U, F\rangle$, where $F = \{A \to \{a\} : \forall\, a \in U,\ A \in H_a, A \neq \{a\}\}$.

PROPOSITION 2.21. $F_R = F^+$.

PROOF. Because F_R is an f-family over U, it can be seen that $N_{F_R} \subseteq E_R$, where N_{F_R} is the minimal generator of Z_{F_R}. By definition of the minimal generator, we obtain $N_R = N_{F_R}$. Consequently, $N_a = M_a$ holds. From the definition of the set of antikeys and according to the definition of K_a we have $H_a = K_a$. Hence, $F^+ \subseteq F_R$. Conversely, if $A \to B = \{b_1, \ldots, b_t\} \in F_R$ then by the construction of F we obtain $A \to \{b_i\} \in F^+$ for each $i = 1, \ldots, t$. Because there is not a trivial FD $\{a\} \to \{a\}$ in F, it can be seen that for all $i = 1, \ldots, t$, if there is no FD $B \to \{b_i\} \in F$, where $B \subseteq U - b_i$, then $b_i \in A$. Hence, $A \to B \in F^+$ holds. The proof is complete. ∎

It can be seen that $E_R, N_R, N_a\, \forall\, a \in U$ are constructed in polynomial time in the size of R. Clearly, the construction of F depends on the size of $H_a (\forall\, a \in U)$. Consequently, the worst-case time complexity of Algorithm 2.20 is

$$O\Big(n\sum_{i=1}^{n}\Big(\sum_{q=1}^{m_i-1}(k_i\, l_{iq} + n\, t_{iq}\, u_{iq}) + {k_i}^2 + n\Big)\Big),$$

where $U = \{a_1, \dots, a_n\}$, $|N_{a_i}| = k_i$, $|H_{a_i}| = m_i$, for meanings of l_{iq}, t_{iq}, u_{iq} see Propositions 2.3, 2.18. Clearly, if $l_{iq} \leq k_i$ ($\forall\, i$, $\forall\, q : 1 \leq q \leq m_i - 1$), then the time complexity of our algorithm is $O\Big(n^2 \sum_{i=1}^{n} {k_i}^2\, m_i\Big)$. Since k_i is polynomial in the size of R, in these cases if m_i is polynomial in the size of R then our algorithm is effective. The time complexity of this algorithm is polynomial in the size of R. Especially, $|H_a|$ is small.

Now, based on Algorithm 2.20, we construct a relation scheme $S = \langle U, F\rangle$ from a given relation in following example.

EXAMPLE 2.22. R is the following relation over $U = \{a, b, c, d\}$:

a	b	c	d
6	6	6	0
0	2	0	2
0	0	0	0
0	0	0	3
4	4	0	0
5	0	5	5
1	0	0	0

It can be seen that

$$\begin{aligned}
&E_R = \{\{a,b,c\}, \{b,c,d\}, \{a,c\}, \{b,c\}, \{c,d\}, \{b\}, \{c\}, \{d\}, \emptyset\},\\
&N_R = \{\{a,b,c\}, \{b,c,d\}, \{a,c\}, \{c,d\}, \{b\}, \{d\}\},\\
&N_a = \{b,c,d\},\ N_b = \{\{a,c\}, \{c,d\}\},\\
&N_c = \{\{b\}, \{d\}\},\ N_d = \{a,b,c\}.
\end{aligned}$$

We obtain $H_a = \{a\}$, $H_b = \{\{b\}, \{a,d\}\}$, $H_c = \{\{a\}, \{b,d\}, \{c\}\}$, $H_d = \{d\}$.

We construct $S = \langle U, F\rangle$, as follows: $U = \{a,b,c,d\}, F = \{\{a,d\} \to \{b\}, \{a\} \to \{c\}, \{b,d\} \to \{c\}\}$.

3. GENERATING AN ARMSTRONG RELATION AND INFERRING FDS IN BCNF

In this section, we present some results related to relations and relation schemes in Boyce-Codd normal form. We give two algorithms for generating an Armstrong relation and for inferring FDs in BCNF. We show that if relations and relation schemes satisfy certain additional properties, then generating an Armstrong relation, inferring FDs that hold in a relation, FD-relation equivalence problem are solved in polynomial time.

DEFINITION 3.1. *Let $S = \langle U, F\rangle$ be a relation scheme. We say that S is a k-relation scheme over U if $F = \{K_1 \to U, \dots, K_m \to U\}$, where $\{K_1, \dots, K_m\}$ is a Sperner system over U. It is easy to see that $K_S = \{K_1, \dots, K_m\}$.*

It can be seen that a relation scheme $S = \langle U, F\rangle$ (a relation R) is in BCNF iff $\forall\, A \subseteq U$ either $A^+ = A$ or $A^+ = U$ ($L_{F_R}(A) = A$ or $L_{F_R}(A) = U$). Clearly, if $S = \langle U, F\rangle$ is in BCNF then using the algorithm for finding a minimum cover in polynomial time we can construct a k-relation scheme $S' = \langle U, F'\rangle$ such that $F^+ = F'^+$, see [11]. Conversely, it can be seen that an arbitrary k-relation scheme is in BCNF. Consequently, we can consider a relation scheme in BCNF as a k-relation scheme.

REMARK 3.2. It is known [11] that $S = \langle U, F\rangle$ is in BCNF iff its minimum cover is a k-relation scheme.

Consequently, the BCNF property of S is polynomially recognizable.

Let R a relation over U. From R we compute E_R. After that, from E_R, we construct the maximal equality system T_R of R.

By Theorem 1.10, we obtain $T_R = K_R^{-1}$, where K_R is a set of all minimal keys of R.

Denote elements of T_R by $A_1, \dots, A_t$.

Set $M_R = \{A_i - a : a \in U, i = 1, \dots, t\}$.

Denote elements of M_R by $B_1, \dots, B_s$. We construct a relation $R' = \{h_0, h_1, \dots, h_s\}$ as follows: For all $a \in U$, $h_0(a) = 0$, for each $i = 1, \dots, s$ $h_i(a) = 0$ if $a \in B_i$, in the converse case we set $h_i(a) = i$.

By [11] R' is in BCNF and $K_R = K_{R'}$ (1).

It is easy to see that M_R and R' are constructed in polynomial time in the size of R.

Set $L_{F_R}(A) = \{a \in U : (A, \{a\}) \in F_R\}$.

Based on definition of BCNF relation and from (1), we can see that a relation R is in BCNF iff $N_{F_R} = N_{F_{R'}}$.

Because for an arbitrary relation R, N_{F_R} is computed in polynomial time, the BCNF property of R can be tested in polynomial time.

Let $S = \langle U, F\rangle$ be a relation scheme. Set $Z(F) = \{A : A^+ = A, A \subseteq U\}$ and $N(F)$ is the minimal generator of $Z(F)$.

We set $T_S = \{A \in N(F) : \nexists B \in N(F) : A \subset B\}$.

It is known [2] that for a given relation scheme S there is a relation R such that R is an Armstrong relation of S. On the other hand, by Theorem 1.10 and Theorem 2.10 the next proposition is clear.

PROPOSITION 3.3. *Let $S = \langle U, F\rangle$ be a relation scheme. Then $T_S = K_S^{-1}$.*

According to Theorem 1.10 and Proposition 3.3 it can be seen that $K_S^{-1} \subseteq N(F) = MAX(F^+)$ and $K_R^{-1} \subseteq N_{F_R} = MAX(F_R)$. It can be seen that there is a relation scheme $S = \langle U, F\rangle$ such that $|N(F)|$ is exponential in $|U|$, and $|F|$.

We can give an example which shows that $|K_S^{-1}|$ is linear, but $|N(F)|$ is exponential in $|U|$. On the other hand, for BCNF class of relation schemes $S = \langle U, F\rangle$ and relations R over U it is clear that $F_R = F^+$ iff $K_R = K_S$.

It is effective that in BCNF class of relations and relation schemes we directly compute the set of antikeys from a relation scheme S for generating an Armstrong relation and compute the set of minimal keys from a set of antikeys of relation R for inferring FDs. Consequently, based on Algorithms 2.1 and 2.17 we give two algorithms as follows.

ALGORITHM 3.4. GENERATING AN ARMSTRONG RELATION FOR BCNF RELATION SCHEME.

Input: $S = \langle U, F = \{K_1 \to U, \dots, K_m \to U\}\rangle$ be a k-relation scheme.

Output: Relation R such that $F_R = F^+$.

Step 1: From $K = \{K_1, \dots, K_m\}$ we construct $K^{-1} = \{B_1, \dots, B_t\}$ by Algorithm 2.1.

Step 2: Set $M = \{B_i - a : a \in U, i = 1, \dots, t\}$.

Step 3: Denote elements of M by $A_1, \dots, A_s$, construct a relation $R = \{h_0, h_1, \dots, h_s\}$ as follows: For all $a \in U : h_0(a) = 0$, for $i = 1, \dots, s$, set $h_i(a) = 0$ if $a \in A_i$, in the converse case $h_i(a) = i$.

By Remark 3.2, Theorem 2.2 we obtain $F_R = F^+$.

Clearly, the set M and the relation R are constructed in polynomial time in the size of K^{-1}.

Consequently, the time complexity of this algorithm is $O\left(|U|^3 \sum_{q=1}^{m-1} t_q u_q\right)$, for meanings of t_q, u_q, see Proposition 2.3.

In many cases, this algorithm requires polynomial time in the size of S (see Proposition 2.3). Based on Remark 3.2, Algorithms 2.11, and 3.4 we give a next algorithm.

ALGORITHM 3.5.

Input: $S = \langle U, F\rangle$ be a relation scheme.

Output: a relation R such that $F^+ = F_R$.

Step 1: From S find the minimum cover S'.

Step 2: If S' is a k-relation scheme, then, by Algorithm 3.4, we construct R' such that $F^+ = F_{R'}$, set $R = R'$. In the converse case by Algorithm 2.11, find R'' such that $F^+ = F_{R''}$, set $R = R''$.

Clearly, the time complexity of this algorithm depends on the time complexities of Algorithms 2.11 and 3.4.

ALGORITHM 3.6. INFERRING FDS FOR BCNF RELATION.

Input: Let R be a BCNF relation over U.

Output: $S = \langle U, F\rangle$ such that $F^+ = F_R$.

Step 1: From R, compute E_R.

Step 2: From E_R, compute the maximal equality system T_R.

Step 3: By Algorithm 2.17 we construct a set of minimal keys H of R.

Step 4: Denoting elements of H by $A_1, \ldots, A_m$, we construct a relation scheme as follows: $S = \langle U, F\rangle$, where $F = \{A_1 \to U, \ldots, A_m \to U\}$.

Based on Definition 3.1 and Remark 3.2, we have $F^+ = F_R$. It is clear that the time complexity of this algorithm is the time complexity of Algorithm 2.17. In many cases, this algorithm is very effective (see Proposition 2.18).

ALGORITHM 3.7.

Input: R a relation over U.

Output: $S = \langle U, F\rangle$ such that $F^+ = F_R$.

Step 1: From R, construct R' (in Remark 3.2.)

Step 2: If $N_{F_R} = N_{F_{R'}}$, then use Algorithm 3.6 to construct a relation scheme S', set $S = S'$.

In the converse case, by Algorithm 2.20, we construct a relation scheme S'', set $S = S''$. It is obvious that the time complexity of this algorithm depends on the time complexities of Algorithms 3.6 and 2.20.

THEOREM 3.8. *[12] Let F an f-family over U. N_F is the minimal generator of Z_F, where $Z_F = \{A : L_F(A) = A\}$, $L_F(A) = \{a : (A, \{a\}) \in F\}$. Denote $s(F) = \min\{m : |R| = m, F_R = F\}$. Then $(2|N_F|)^{1/2} \le s(F) \le |N_F| + 1$.*

REMARK 3.9. Let us take a partition $U = X_1 \cup \cdots \cup X_m \cup W$, where $m = [n/3]$, and $|X_i| = 3$ $(1 \le i \le m)$.

We set

$H = \{B : |B| = 2, B \subseteq X_i$ for some $i\}$, if $|W| = 0$,

$H = \{B : |B| = 2, B \subseteq X_i$ for some $i : 1 \le i \le m-1$ or $B \subseteq X_m \cup W\}$ if $|W| = 1$,

$H = \{B : |B| = 2, B \subseteq X_i$ for some $i : 1 \le i \le m$ or $B = W\}$, if $|W| = 2$.

It is easy to see that

$H^{-1} = \{A : |A \cap X_i| = 1, \forall i\}$ if $|W| = 0$,

$H^{-1} = \{A : |A \cap X_i| = 1, (1 \le i \le m-1)$ and $|A \cap (X_m \cup W)| = 1\}$ if $|W| = 1$,

$H^{-1} = \{A : |A \cap X_i| = 1, (1 \le i \le m)$ and $|A \cap W| = 1\}$ if $|W| = 2$.

If set $K = {H^{-1}}^{-1}$, i.e., H^{-1} is a set of minimal keys of K, then we have

$K = \{C : |C| = n-3, C \cap X_i = \emptyset$ for some $i\}$ if $|W| = 0$,

$K = \{C : |C| = n-3, C \cap X_i = \emptyset$ for some i $(1 \le i \le m-1)$ or $|C| = n-4, C \cap (X_m \cup W) = \emptyset\}$ if $|W| = 1$,

$K = \{C : |C| = n-3, C \cap X_i = \emptyset$ for some i $(1 \le i \le m)$ or $|C| = n-2, C \cap W = \emptyset\}$ if $|W| = 2$.

It is clear that $n - 1 \le |H| \le n + 2, 3^{[n/4]} < |H^{-1}|, |K| \le m + 1$. Based on this partition, Theorem 3.8, and Algorithms 3.4 and 3.6, we obtain the following propositions.

PROPOSITION 3.10. *In BCNF class of relations and relation schemes, the time complexity of generating an Armstrong relation for a given relation scheme S is exponential in the size of S.*

PROOF. Clearly, the worst-case time complexity of Algorithm 3.4 is exponential in the size of S. According to Theorem 3.8 we have $(2|N_{F^+}|)^{1/2} \le s(F^+)$. We construct a k-relation scheme $S = \langle U, F \rangle$, where $F = \{B \to U : B \in H\}$. It is obvious that $H^{-1} \subseteq N_{F^+}$. Hence, $(2^{1/2}3^{[n/8]}) \le s(F^+)$ holds. It can be seen that BCNF relation R that is constructed in Algorithm 3.4 has the number of rows at most $|U|\,|H^{-1}| + 1$. Thus, we always can construct a relation scheme S such that the number of rows of any Armstrong relation for S is exponential in the size of S. The proof is complete. ∎

Because BCNF class of relations and relation schemes is a special subfamily of the family of relations and relation schemes over U, the next corollary is obvious (it is known in [1]).

COROLLARY 3.11. *The time complexity of finding an Armstrong relation of a given relation scheme is exponential in the number of attributes.*

PROPOSITION 3.12. *In BCNF class of relations and relation schemes, the time complexity of inferring FDs for a given relation over U is exponential in the number of attributes.*

PROOF. It is clear that the worst-case time complexity of Algorithm 3.6 is exponential in the size of R. In Remark 3.9, we have $|K| \le m + 1$. We set $M = \{C - a : \forall\, a, C : a \in U, C \in K\}$. Denote elements of M by $C_1, \ldots, C_t$. Construct a relation $R = \{h_0, h_1, \ldots, h_t\}$ as follows: For all $a \in U$ $h_0(a) = 0$, for $i = 1, \ldots, t$ $h_i(a) = 0$ if $a \in C_i$, in the converse case $h_i(a) = i$. Clearly, $|R| \le (m + 1)|U| + 1$ holds. We construct a relation scheme $S = \langle U, F \rangle$ with $F = \{A \to U : A \in H^{-1}\}$. It is obvious that $3^{[n/4]} < |F|$, and $F_R = F^+$. Clearly, a minimum cover of any BCNF relation scheme is a k-relation scheme. Thus, we always can construct the BCNF relation R in which the number of rows of R is at most $(m+1)|U| + 1$ but for any relation scheme $S = \langle U, F \rangle$ such that $F_R = F^+$, the number of elements of F is exponential in the number of attributes. Our proof is complete. ∎

The following corollary is clear.

COROLLARY 3.13. *[13] Inferring FDs has exponential time.*

DEFINITION 3.14. *[6] Let K be a Sperner system over U. We say that K is saturated if for any $A \notin K$, $\{A\} \cup K$ is not a Sperner system.*

THEOREM 3.15. *[6] If K is a saturated Sperner system then $K = K_F$ uniquely determines F, where K_F is the set of all minimal keys of an f-family F.*

There are examples showing that there exists K (K^{-1}) such that K (K^{-1}) is saturated, but K^{-1} (K) is not saturated.

Now we define the next notion.

DEFINITION 3.16. *Let K be a Sperner system over U. We say that K is inclusive, if for every $A \in K$ there is a $B \in K^{-1}$ such that $B \subset A$. We call K embedded if for each $A \in K$ there exists a $B \in H : A \subset B$, where $H^{-1} = K$.*

THEOREM 3.17. *[5] Let K be a Sperner system over U. Denote H a Sperner system for which $H^{-1} = K$. The following facts are equivalent:*

(1) *K is saturated,*
(2) *K^{-1} is embedded,*
(3) *H is inclusive.*

Now we show a class of relations and relation schemes for which generating an Armstrong relation and inferring FDs are solved in polynomial time.

PROPOSITION 3.18.

(1) *Let $S = \langle U, F = \{K_1 \to U, \dots, K_m \to U\}\rangle$ be BCNF relation scheme. If $K = \{K_1, \dots, K_m\}$ is saturated then generating an Armstrong relation is solved in polynomial time in $|U|$ and $|F|$.*

(2) *Let R be a relation in BCNF. If K_R^{-1} is saturated then inferring FDs is solved in polynomial time in the size of R.*

PROOF. For (1), we set $M = \{K_i - \{a\} : a \in K_i,\ i = 1, \dots, m\}$ and $L = \{A \in M : A = A^+, \{A \cup a\}^+ = U \forall\, a \in U - A\}$, where A^+ is a closure of A. Denote elements of L by $A_1, \dots, A_s$. Set $N = \{A_i - a : a \in U, i = 1, \dots, s\}$. Assume that $N = \{B_1, \dots, B_t\}$. We construct a relation $R = \{h_0, h_1, \dots, h_t\}$, as follows: For each $a \in U$, $h_0(a) = 0$. For $i = 1, \dots, t$ $h_i(a) = 0$ if $a \in B_i$, in the converse case $h_i(a) = i$. By Remark 3.2 and based on Theorem 3.17, Definition 3.16 R is in BCNF and $K^{-1} = L = K_R^{-1}$. Hence, $F^+ = F_R$ holds. It can be seen that R is constructed in polynomial time in the size of S.

For (2): From R we compute K_R^{-1} in polynomial time in $|R|$. We set $P = \{T \cup \{a\} : a \in U, T \in K_R^{-1}\}$ and $Q = \{E \in P : (E, U) \in F_R, \forall A = E - a : a \in E, (A, U) \notin F_R\}$. Denote elements of Q by $E_1, \dots, E_l$. Clearly, Q is constructed in polynomial time. We set $S = \langle U, F\rangle$ with $F = \{E_1 \to U, \dots, E_l \to U\}$. By Theorem 3.17 and Definition 3.16 we obtain $Q = K_R$. Consequently, we have $F^+ = F_R$. Our proof is complete. ∎

DEFINITION 3.19. FD-RELATION EQUIVALENCE PROBLEM. *Let $S = \langle U, F\rangle$ be a relation scheme, and R a relation over U. Decide whether $F_R = F^+$.*

It is obvious that in cases of relations and relation schemes for which the time complexity of Algorithm 2.11 or Algorithm 2.20 is polynomial then the FD-relation equivalence problem is solved in polynomial time in the size of S or R. This is true for Algorithm 3.4 and Algorithm 3.6 in BCNF class.

Based on Proposition 3.18 it is obvious that $S = \langle U, F\rangle$ with $F = \{K_1 \to U, \dots, K_m \to U\}$, R are in BCNF, then if $\{K_1, \dots, K_m\}$ or K_R^{-1} are saturated, then the FD-relation equivalence problem is solved in polynomial time (it is known in [11].)

DEFINITION 3.20. *Let K_1, K_2 two Sperner system over U. We set $K = K_1 \cup K_2$ and $T_K = \{A \in K : \nexists B \in K : A \subset B\}$. We say that the union $K = K_1 \cup K_2$ is pseudo-saturated if T_K is a saturated Sperner system.*

Based on Definition 3.20 we give the next theorem related to relations and relation schemes the FD-relation equivalence problem of which is solved in polynomial time.

THEOREM 3.21. *Let $S = \langle U, F\rangle$ be a relation scheme in BCNF and R a relation over U in BCNF. $K_S = \{A_1, \dots, A_p\}$ ($K_R^{-1} = \{B_1, \dots, B_q\}$) is the set of minimal keys of S (the set of antikeys of R). Then if $K_S \cup K_R^{-1}$ is pseudo-saturated then the FD-relation equivalence problem is solved in polynomial time in the sizes of S and R.*

PROOF. Clearly, by Theorem 1.10 from R we compute K_R^{-1} in polynomial time in the size of R, and from S we find a k-relation scheme that is a minimum cover of S. The minimum cover is constructed in polynomial time in the size of S. If there is $A_i (1 \le i \le p)$ such that $A_i \subseteq B_j (1 \le j \le q)$, then $K_S \ne K_R$. Consequently, we can assume that $A_i \not\subseteq B_j$ for all i, j. For each $j = 1, \dots, q$, we compute B_j^+. It can be seen that for all $D \subseteq U$, D^+ is computed in polynomial time in the size of S. We set $M = \{B_j \cup \{a\} : a \in U - B_j\} = \{M_1, \dots, M_t\}$. It is obvious that M is computed in polynomial time. If $B_j^+ \ne U$ and for all $l = 1, \dots, t$, $M_l^+ = U$ hold then $B_j \in K_S^{-1}$ holds, otherwise we obtain $B_j \notin K_S^{-1}$. If there is a $B_j : B_j \notin K_S^{-1}$ then by the definition of antikeys $K_R \ne K_S$. We assume that for all j=1,...,q $B_j \in K_S^{-1}$. For each $i = 1, \dots, p$ we set $N = \{A_i - \{a\} : a \in A_i\} = \{N_1, \dots, N_s\}$. It can be seen that N is computed in polynomial time. If there is a $N_n (1 \le n \le s)$ such that $N_n \not\subseteq B_j$ for all $j = 1, \ldots, q$ then $A_i \notin K_R$ holds. In the converse case, we obtain $A_i \in K_R$. Clearly, if there is an $A_i \notin K_R$ then $K_S \neq K_R$. We assume that for each $i = 1, \ldots, p$ we have $A_i \in K_R$. We set

$$\begin{aligned}
Q &= \{A_i - \{a\} : a \in A_i, i = 1, \ldots, p\}, \\
P &= \{A \in Q : A = A^+, (A \cup \{a\})^+ = U, \forall\, a \in U - A\}, \\
J &= \{B_j \cup \{a\} : a \in U - B_j, j = 1, \ldots, q\}, \\
I &= \{B \in J : (B, U) \in F_R,\ (\{B - a\}, U) \notin F_R \forall\, a \in B\}.
\end{aligned}$$

Based on definition of K_S and definition of K_R^{-1} we can see that if there is either an $A \in P$ such that $A \notin K_R^{-1}$ or there exists a $B \in I$ but $B \notin K_S$ then $K_S \neq K_R$ holds. It can be seen that P, I are constructed in polynomial time in the sizes of S, R, K_S, K_R^{-1}. Finally, we see that if for all $i = 1, \ldots, p$, $j = 1, \ldots, q$, $A_i \in K_R$, $B_j \in K_S^{-1}$, $P \subseteq K_R^{-1}$, $I \subseteq K_S$ hold then by $K_S \cup K_R^{-1}$ is pseudo-saturated and according to the definition of set of minimal keys and definition of set of antikeys we obtain $K_R = K_S$. Since S, R are in BCNF we have $F_R = F^+$. The proof is complete. ∎

4. CONCLUSION

Our further research will concentrate on the following problems:

1. Find subfamilies of relations and relation schemes in which generating an Armstrong relation or inferring FDs are solved in polynomial time.
2. Given a relation scheme S and a relation R. What is the time complexity of deciding whether $K_S = K_R$.
3. What is the time complexity of finding a relation scheme S from a given relation R such that $K_S = K_R$.

REFERENCES

1. C. Beeri, M. Dowd, R. Fagin and R. Staman, On the structure of Armstrong relations for functional dependencies, *J. ACM* **31** (1), 30–46 (1984).
2. W.W. Armstrong, *Dependency Structures of Database Relationships*, Info. Processing 74, pp. 580–583, Holland Publ. Co., (1974).
3. J. Demetrovics, Relacios adatmodell logikai es strukturalis vizsgalata, MTA-SZTAKI Tanulmanyok, Budapest, 114, pp. 1–97, (1980).
4. J. Demetrovics and V.D. Thi, Relations and minimal keys, *Acta Cybernetic Hungary* **8** (3), 279–285 (1988).
5. V.D. Thi, Minimal keys and Antikeys, *Acta Cybernetica Hungary Tom.* **7** (4), 361–371 (1986).
6. G. Burosch, J. Demetrovics and G.O.H. Katona, The poset of closures as a model of changing databases, *Order* **4**, 127–142 (1987).
7. C. Beeri and P.A. Bernstein, Computational problems related to the design of normal form relational schemas, *ACM Trans. on Database Syst.* **4** (1), 30–59 (1979).
8. H. Mannila and K.J. Raiha, Design by example: An application of Armstrong relations, *J. Comput. Syst. Scien.* **33**, 126–141 (1986).
9. J. Demetrovics and V.D. Thi, Some results about functional dependencies, *Acta Cybernetica Hungary* **8** (3), 273–278 (1988).
10. V.D. Thi, Relacios adatmodell antikulcsairol, *Alkalmazott Matematikai Lapok* **12**, 111–124 (1986).
11. G. Gottlob and L. Libkin, Investigations on Armstrong relations, dependency inference, and excluded functional dependencies, *Acta Cybernetica Hungary* **9** (4), 385–402 (1990).
12. V.D. Thi, Funkcionalis fuggoseggel kapcsolatos nehany kombinatorikai jellegu vizsgalat a relacios adatmodellben, Ph.D. Dissertation, MTA-SZTAKI Tanulmanyok, Budapest, 191, pp. 1–157, (1986).
13. H. Mannila and K.J. Raiha, Algorithms for inferring functional dependencies, *Proc. of the Thirteenth International Conference on Very Large Data Bases*, Brighton, (September 1987).
14. C.L. Lucchesi and S.L. Osborn, Candidate keys for relations, *J. Comput. Syst. Scien.* **17** (2), 270–279 (1978).