Polynomial delay Hybrid algorithms to enumerate candidate keys for a relation

Karima Ennaoui, Lhouari Nourine

Polynomial delay Hybrid algorithms to enumerate candidate keys for a relation

Karima Ennaoui, Lhouari Nourine

Université Clermont-Auvergne, CNRS, Mines de Saint-Étienne, Clermont-Auvergne-INP, LIMOS, France

Abstract

We investigate the problem of candidate keys enumeration of a relational schema. The notion of candidate keys is also known as minimal generators in lattice or FCA terminologies or as implicates in logic. Given an implicational base or a set of functional dependencies, Lucchesi and Osborn gave in [11] an incremental polynomial time algorithm to enumerate all candidate keys of a relation schema. Using the state of the art of enumeration technics (see Elbassioni [5]), however, it turns out to be a polynomial delay algorithm and exponential space (see Ennaoui [6] and Bérczi *et al.*[3]).

In this paper we exploit the presence of unary functional dependencies that define a partial order over the set of attributes. We use a bijection between key-ideal sets (ideal associated to a key) and candidate keys, and we show that the number of key-ideal sets can be exponential in the number of minimal key-ideal sets. Moreover, if there is a polynomial delay and space algorithm to enumerate minimal key-ideal sets then there is one for all candidate keys. We also give a polynomial delay algorithm to enumerate all minimal key-ideal sets. As a consequence, we derive a polynomial delay hyprid[1] algorithm to enumerate all candidate keys using space bounded by the number of minimal key-ideal sets.

Keywords: Candidate keys, Minimal generators, Enumeration problem, Posets

Email addresses: karima.ennaoui@uca.fr (Karima Ennaoui), lhouari.nourine@isima.fr (Lhouari Nourine)

[1]Hybrid means that two algorithms are combined to enumerate candidate keys

1. Introduction

This paper is an extended paper of BDA'16 [7] and Ennaoui's phd thesis [6].

We investigate the problem of enumerating all candidate keys of a relational database schema. A candidate key of a set of functional dependencies, also known as minimal generator in closure systems or FCA terminologies [13], is a minimal subset of attributes that identifies uniquely every tuple of the relation. Listing candidate keys is related to normalization for relational databases, and has other applications in different fields (for instance formal context analysis [10], systems security [15]).

Enumerating all candidate keys has been studied in the literature by considering as input a relation instance or a set of functional dependencies. Whenever the input is given by a relation instance, the enumeration of all candidate keys is equivalent to the enumeration of all minimal transversal of an hypergraph [2, 12], or to the enumeration of minimal dominating sets of a graph [9]. This problem is also known as the enumeration of all minimal generators of a concept in Datamining and FCA communities [16]. The enumeration of all candidate keys from a relation instance can be solved using a quasi-polynomial incremental algorithm [8]. When the input is a set of functional dependencies, the best known result is given by Lucchesi and Osborn [11] with an incremental polynomial time algorithm and an exponential space. Using the state of the art [5] of enumeration technics, however, it turns out to be a polynomial delay algorithm and exponential space. Saiedian and Spencer in [14] use the notion of attribute graph of the set of functional dependencies and show that candidate keys are union of candidate keys of strongly connected components of the attribute graph. Ennaoui and Nourine [6, 7] improve the algorithm given in [11] by providing a polynomial delay algorithm to enumerate candidate keys when the input is a set of functional dependencies. Recently, Bérczi *et al.*[3] show also that candidate keys can be enumerated in polynomial delay.

In this paper, we investigate the problem of candidate keys enumeration of a relational database given by a set of $\mathcal{A}$ of attributes and an implicational base:

Keys Enumeration (K-Enum)
Input: An implicational base Σ on a set $\mathcal{A}$ of attributes.
Output: The set of all candidate keys $\mathcal{K}$.

Our approach is to exploit the presence of unitary functional dependencies that define a partially ordered set P_Σ (poset for short) over the attributes set $\mathcal{A}$. For each candidate key K, we associate the smallest ideal of P_Σ containing K, that we call a key-ideal.

The key-ideal sets family $\mathcal{I}^K$ is then partitioned into equivalence classes, where each class is represented by a special type of key-ideal sets called minimal. We point out that the number of key-ideal sets $|\mathcal{I}^K|$ may be exponential in the number of minimal key-ideal sets $|\mathcal{I}^K_{min}|$. By establishing a one-to-one correspondence between $\mathcal{I}^K$ and candidate keys $\mathcal{K}$, we show that if there is a polynomial delay and polynomial space algorithm to enumerate minimal key-ideal sets $\mathcal{I}^K_{min}$ then there is one to enumerate all candidate keys. We also give a polynomial delay algorithm to enumerate all minimal key-ideal sets. As a consequence, there is a polynomial delay algorithm to enumerate candidate keys where space is bounded by the number of minimal key-ideal sets $|\mathcal{I}^K_{min}|$. This bound can be smaller than the number of candidate keys, but they are equal when no unit functional dependency is present.

Our main results can be sumarized in the following:

Theorem. *If there is a polynomial delay and space algorithm to enumerate minimal key-ideal sets, then there is one to enumerate all key-ideal sets in polynomial delay and polynomial space.*

Theorem. *There is a polynomial delay algorithm to enumerate candidate keys, where space is bounded by the number of minimal key-ideal sets.*

2. Preliminaries

Let $\mathcal{A}$ be a finite set of attributes. A partially ordered set (poset) $P = (\mathcal{A}, \preceq)$ is a set $\mathcal{A}$ together with a binary relation $\preceq$ that is reflexive, antisymmetric and transitive. A subset I of $\mathcal{A}$ is called an *ideal* of P, if $x \in I$ and $y \preceq x$ imply $y \in I$. We denote $Max(I) = \{x \in I \mid \forall y \in I, x \not\prec y\}$ and $\mathcal{I}(P)$ the set of all ideal sets of P.

An implicational base Σ over $\mathcal{A}$ is defined by a set of implications (or functional dependencies) $L \to R$ with $(L, R) \in 2^{\mathcal{A}} \times 2^{\mathcal{A}}$. The Σ-closure of a set $X \subseteq \mathcal{A}$ is the smallest set denoted by X^Σ containing X and verifying for every $L \to R \in \Sigma$ that if $L \subseteq X^\Sigma$ then $R \subseteq X^\Sigma$. An implication $L \to R$ is called *unitary* if $|L| = 1$. We denote the set of all unitary implications by Σ_u and $\Sigma_{nu} = \Sigma \setminus \Sigma_u$ the set of non-unitary implications. Without loss of generality, we suppose that $|R| = 1$ for every implication $L \to R$ in Σ.

A *key* K of Σ over a set $\mathcal{A}$, is a subset of $\mathcal{A}$ verifying that $K^{\Sigma} = \mathcal{A}$. A key K is called *candidate* [4, 11, 14] if none of its proper subsets is a key of Σ. We denote by $\mathcal{K}$ the set of all candidate keys of Σ, also referred to as minimal keys in the literature.

First, we show that a candidate key cannot contain two equivalent attributes and they are interchangeable, where two attributes a and b are said to be *equivalent*, if $b \in a^{\Sigma_u}$ and $a \in b^{\Sigma_u}$.

Lemma 1. *Let a and b be two equivalent attributes in $\mathcal{A}$ and K a candidate key containing an attribute a. Then $b \notin K$ and $(K \setminus \{a\}) \cup \{b\}$ is a candidate key.*

Proof. Let a and b be two equivalent attributes in $\mathcal{A}$ and K a candidate key containing an attribute a. Then $b \in a^{\Sigma_u}$, and since $a \in K$ we have $b \in K^{\Sigma}$ and $b \notin K$, otherwise K is not minimal. Now, let $K' = (K \setminus \{a\}) \cup \{b\}$. Since $a \in b^{\Sigma_u}$ then $a \in K'^{\Sigma}$, and thus K' is a key. Moreover any proper subset $S \subseteq K'$ is not a key. Indeed, either S or $((S \setminus \{b\}) \cup \{a\})$ is a subset of K and they are both keys which contradicts the fact that K is a candidate key. □

Whenever Σ contains equivalent attributes, then we only keep one representative attribute for each set of equivalent attributes and consider a reduced implicational base Σ' obtained from Σ by replacing attributes of each equivalent set by its representative.

Example 1. *Consider the implicational base $\Sigma = \{a \to b, b \to c, c \to a, bd \to ace, ce \to abd\}$ on the set $\mathcal{A} = \{a, b, c, d, e\}$. Then, the attribute set $\{a, b, c\}$ are equivalent and, if we choose a as a representative, we obtain $\Sigma' = \{ad \to e, ae \to d\}$. From lemma 1, any candidate key K of Σ' containing the representative a then $(K \setminus \{a\}) \cup \{b\}$ and $(K \setminus \{a\}) \cup \{c\}$ are also candidate keys.*

Hence, the keys of Σ' are $\{ad, ae\}$, and the keys of Σ are $\{ad, be, bd, ce, ad, ce\}$.

In the rest of the paper, we assume that Σ_u does not contain equivalent attributes which is also known as acyclic and corresponds to a poset $P_{\Sigma} = (\mathcal{A}, \leq)$, where $a \leq b$ iff $a \in b^{\Sigma_u}$.

Definition 2. *We call a key-ideal set of Σ, every ideal I of P_{Σ} such that $Max(I)$ is a candidate key of Σ. A key-ideal set is called minimal if it does not contain a proper key-ideal set.*

We denote $\mathcal{I}^{\mathcal{K}}$ (resp. $\mathcal{I}^{\mathcal{K}}_{min}$) the family of all key-ideal sets (resp. minimal key-ideal sets).

Moreover, note that the number of minimal key-ideal sets of Σ can be significantly smaller than $|\mathcal{I}^{\mathcal{K}}|$. For instance, considering $\mathcal{A} = \{a_1, a_2, ..., a_{2p-1}, a_{2p}\}$ for some integer p, $\Sigma_{nu} = \{\{a_1, ..., a_p\} \to X\}$ and $\Sigma_u = \{a_{p+i} \to a_i, 1 \leq i \leq p\}$. Then there is a unique minimal key-ideal set $I = \{a_1, ..., a_p\}$ and 2^p key-ideal sets or candidate keys.

From Definition 2, there is a one-to-one mapping between $\mathcal{I}^{\mathcal{K}}$ and $\mathcal{K}$, which implies that the enumeration of candidate keys and key-ideal sets are polynomially equivalent.

Example 2. *Consider the following set of implications* Σ *with* $\Sigma_u = \{e \to ab, f \to bc, g \to c, h \to d\}$ *and* $\Sigma_{nu} = \{ac \to h, ad \to f, bd \to g, cd \to e\}$. *The set of all candidate keys is :*

$\mathcal{K} = \{ac, ad, bd, cd, af, ag, ce, eg, ah, de, eh, bh, df, fh, ch, dg, gh, ef\}$, *and the corresponding the set of key-ideal sets* $\mathcal{I}^{\mathcal{K}} = \{ac, ad, bd, cd, abcf, acg, abce, abceg, adh, abde, abdeh, bdh, bcdf, bcdfh, cdh, cdg, cdgh, abcef\}$, *and* $\mathcal{I}^{\mathcal{K}}_{min} = \{ac, ad, bd, cd\}$.

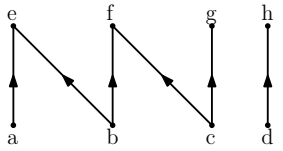

Figure 1: The poset P_Σ associated to Σ in Example 2

3. Enumeration of key-ideals

We begin by describing our approach for the enumeration of key-ideal sets. We assume that Σ does not contain equivalent attributes and a lexicographical order on the set of attributes $\mathcal{A} = \{a_1, ..., a_n\}$ corresponding to a linear extension of P_Σ, such that if $a_i \leq a_j$ in P_Σ then $i \leq j$. That is for any $b \in a^\Sigma$, b is lexicographically smaller or equal to a. We define the colexicographical order on sets by $X \leq_{col} Y$ if the largest attribute in $X \Delta Y$ (with Δ the symmetric difference) belongs to Y.

We define the function $\pi : \mathcal{I}^K \to \mathcal{I}^K$, with $\pi(J) = J$ if $J \in \mathcal{I}^K_{min}$, otherwise $\pi(J) = J'$ such that J' is the colexicographically largest proper subset of J and J' is a key-ideal.

We note that the function π is well defined for every non minimal key-ideal, since all key-ideal sets included in J are totally ordered under the colexicographical order.

Example 3. *Following Example 2, consider the key-ideal $J = abceg$. The set of all key-ideals included in J are ordered as follow: $ac \leq_{col} abce \leq_{col} acg$ which implies $\pi(J) = acg$. For $J = abdeh$, we have $ad \leq_{col} bd \leq_{col} abde \leq_{col} adh \leq_{col} bdh$ which implies $\pi(J) = bdh$.*

Lemma 3. *For a non minimal key-ideal $J \in \mathcal{I}^K$, there exists a unique key-ideal $J' = \pi(J)$.*

Proof. Let J be a key-ideal not in $\mathcal{I}^K_{min}$. Then $I_J = \{J' \in \mathcal{I}^K | J' \subset J\}$ is not empty. Since I_J is totally ordered under the colexicographical order, there is a unique Key-ideal J' in I_J with J' the colexicographically largest in I_J. □

We denote by $\pi^*(I)$ the iterative operator until stationary, i.e. it reaches a minimal key-ideal. It is worth noticing that for any $J \in \mathcal{I}^K$, there is a unique $I \in \mathcal{I}^K_{min}$ such that $\pi^*(J) = I$. Thus, the operator π^* induces an equivalence relation $\sim$ on the set of key-ideal sets $\mathcal{I}^K$ as follows: For two key-ideal sets $I, J \in \mathcal{I}^K$,

$$I \sim J \texttt{ iff } \pi^*(I) = \pi^*(J).$$

The equivalence relation $\sim$ induces a partition of $\mathcal{I}^K$, where each equivalence class corresponds to $[I] = \{J \in \mathcal{I}^K \mid I \sim J\}$ for some $I \in \mathcal{I}^K_{min}$. The set of all equivalence classes of $\sim$ on $\mathcal{I}^K$, denoted $A/\sim = \mathcal{I}^K_{min}$, is the quotient of the relation $\sim$.

In the rest of the paper, we first show that each equivalence class can be enumerated in polynomial delay and space, and then we give a polynomial delay algorithm to enumerate the quotient $\mathcal{I}^K_{min}$. By combining the two algorithms, we derive a polynomial delay algorithm to enumerate $\mathcal{I}^K$ and thus, candidate keys using $O(|\mathcal{I}^K_{min}|)$ space.

3.1. Enumeration of a Key-Ideal Class

Let I be a minimal key-ideal set. We give in this section an algorithm that enumerates $[I]$. If $J' = \pi(J)$ we call J a child of J' and J' a parent of J. We denote by $\mathcal{G}[I]$ the digraph whose vertex set is $[I]$ and the edge set is defined according to the parent-child relationship, i.e. (J, J') is an edge if $J' = \pi(J)$.

Proposition 4. *For $I \in \mathcal{I}^K_{min}$, $\mathcal{G}[I]$ is a tree rooted at I of depth at most $|\mathcal{A}|$.*

Proof. Let J be any non-minimal key-ideal in [I]. From Lemma 3, J has exactly one parent J' and I has itself as parent, then $\mathcal{G}[I]$ has $|[I]|-1$ edges. Moreover (J, J') is an edge in $\mathcal{G}[I]$, and we have $I \subseteq J' \subset J$. By recursively applying the parent function to J', we reach I in at most $|J' \setminus I|$ iterations, i.e $\pi^*(J) = \pi^*(J') = I$. Then $\mathcal{G}[I]$ is connected, and thus is a tree with root I. The iterator operator π^* finds the root in at most $|\mathcal{A}|$. □

Lemma 5 shows that any child of given key-deal J can be computed using $J \cup a^{\Sigma_u}$ for some $a \in \mathcal{A} \setminus J$.

Lemma 5. *Let J be a non minimal key-ideal set. Then $J = \pi(J) \cup a^{\Sigma_u}$ for some $a \in \mathcal{A} \setminus J$.*

Proof. Let a be the lexicographically smallest element in $Max(J)$ verifying that $J \setminus \{a\}$ is a key. We prove next that $J = \pi(J) \cup a^{\Sigma_u}$.

First of all, note that such a exists because J is non minimal, and therefore it strictly contains a key-ideal J', with $Max(J) \not\subseteq J'$.

Second, we prove that there exists a key-ideal J' such that $(Max(J) \setminus \{a\}) \subset Max(J')$.

To do so, we begin by proving that there exists a key-ideal J' verifying $Max(J') \subseteq Max(J \setminus \{a\})$: We know that $J \setminus \{a\}$ is a key and an ideal (because J is an ideal and $a \in Max(J)$), hence $Max(J \setminus \{a\})$ is a key. Therefore $Max(J \setminus \{a\})$ contains a candidate key $Max(J')$.

We suppose now that $(Max(J) \setminus \{a\}) \not\subseteq Max(J')$, then there exists $b \in (Max(J) \setminus \{a\})$ and $b \notin Max(J')$. However, since $Max(J') \subseteq Max(J \setminus \{a\})$, $Max(J') \setminus \{b\}$ is a key, it implies that $Max(J \setminus \{a\}) \setminus \{b\}$ is also a key. In addition, since $(Max(J) \setminus Max(J \setminus \{a\}))$ is a subset of a^{Σ_u} and $a \in Max(J)$ and $b \notin \{a\}^{\Sigma_u}$, then $(Max(J \setminus \{a\}) \setminus \{b\})^{\Sigma}$ is a subset of $(Max(J) \setminus \{b\})^{\Sigma}$. We conclude that $Max(J) \setminus \{b\}$ is a key, which contradicts the fact that J is a key-ideal .

We conclude that there exists a key-ideal J' verifying that $(Max(J) \setminus \{a\}) \subseteq Max(J')$.

Third, let us prove that $\pi(J)$ contains $Max(J) \setminus \{a\}$. To do so, we consider K^+_a (respectively K^-_a) be the set of elements in $Max(J)$ that are strictly greater than (respectively smaller than) a in the lexicographical order. Note that $K^+_a \cup K^-_a = Max(J) \setminus \{a\}$. We distinguish the following cases:

- Case 1: $K_a^+ \not\subseteq \pi(J)$. By construction of the considered lexicographical order, every element $b \in J$ is inferior or equal to an element in K_a^+. And since K_a^+ is a subset of J', than $K_a^+ \not\subseteq \pi(J)$ implies that J' is colexicographically greater than $Max(\pi(J))$. Which contradicts the definition of $\pi(J)$. We conclude that K_a^+ is a subset of $\pi(J)$.

- Case 2: $K_a^- \not\subseteq \pi(J)$. By definition of a, every b in K_a^- verifies that $J \setminus \{b\}$ is not a key, hence every key that is a subset of J, contains K_a^-. Therefore, $K_a^- \not\subseteq \pi(J)$ contradicts the fact that $\pi(J)$ is a key. We conclude that K_a^- is a subset of $\pi(J)$.

Finally, we deduce from all above that $J = \pi(J) \cup \{a\}^{\Sigma_u}$ where a is the lexicographically smallest element in $Max(J)$ verifying that $J \setminus \{a\}$ is a key. □

Example 4. *Figure 2 illustrates trees associated to all equivalence classes of key-ideal sets of* Σ *given in Example 2.*

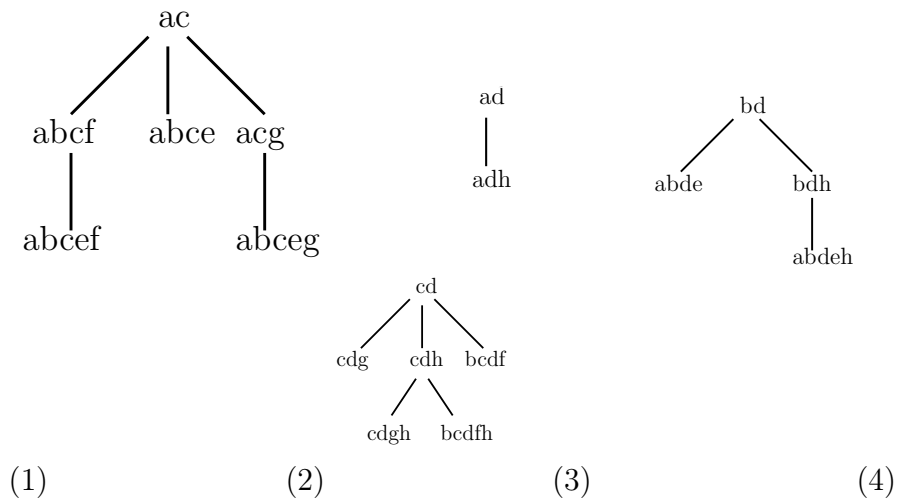

Figure 2: The trees of all equivalence classes of Example 2.

Following the general idea of reverse search [1], we search the tree $\mathcal{G}[I]$ by computing all potential children of a given key-ideal $J \in [I]$, i.e. $\{J \cup \{a\}^{\Sigma_u}, a \in \mathcal{A}\}$ and check those that have J as parent, i.e. $J = \pi(J \cup \{a\}^{\Sigma_u})$. First, we give an algorithm to compute the parent $\pi(J)$ for a key-ideal J.

Algorithm 1: $\pi(J)$

Output: The colexicographically largest key-ideal $\pi(J) \subseteq J \in \mathcal{I}^K$

```
begin
    if J is a minimal key-ideal then
        return J
    else
        Let a be the lexicographical smallest attribute in Max(J)
          such that (J \ {a})^Σ = A;
        J = J \ {a};
        while J is not a key-ideal do
            Let a be the lexicographical smallest attribute in Max(J)
              such that (J \ {a})^Σ = A;
            J = J \ {a};
```

Lemma 6. *For every key-ideal $J \in \mathcal{I}^K$, Algorithm 1 computes $\pi(J)$.*

Proof. Let $J \in \mathcal{I}^K$. Let us suppose that J is not a minimal key-ideal set, otherwise the lemma is obvious. First, Algorithm 1 removes the smallest attribute with the property that $(J \setminus \{a\})^\Sigma = \mathcal{A}$, i.e. it is a key. All other attributes removed by the While loop, are attributes in $\{a\}^{\Sigma_u}$, since, according to the chosen ordering of attributes, they are smaller than a in this ordering. From Lemma 5, we know that $(Max(J) \setminus \{a\}) \subseteq \pi(J)$ and $\pi(J)$ contains some other attributes from $\{a\}^{\Sigma_u}$. Since the removed attributes from $\{a\}^{\Sigma_u}$ are the smallest ones, then the obtained $\pi(J)$ is the colexicographically largest key-ideal set contained in J. □

We give a polynomial delay and space algorithm to enumerate all key-ideal sets in the equivalence class $[I]$ for some minimal key-ideal set I. Given $I \in \mathcal{I}^K_{min}$, Algorithm List-Class follows the depth first search traversal of the tree $\mathcal{G}[I]$ defined in proposition 4. It uses a stack Q to store key-ideal sets in $[I]$ that are not visited.

Algorithm 2: LIST-CLASS($I \in \mathcal{I}^K_{min}$)

Output: The key-ideal sets in the equivalence classe $[I]$
begin
  Insert I to Q; {Q a stack}
1 **while** Q *not empty* **do**
    Pull I from Q;
    Output I;
2   Let $S = \{b \in \mathcal{A} \mid a \in Max(I), a \in b^{\Sigma_u}\}$;
3   **while** S *is not empty* **do**
4     $x :=$ the lexicographically smallest element in S;
5     $J = I \cup \{x\}^{\Sigma_u}$;
6   **if** $I = \pi(J)$ **then**
        Push J to Q;

Proposition 7. *For a minimal key-ideal I in $\mathcal{I}^K_{min}$, Algorithm* LIST-CLASS *outputs all key-ideal sets of the equivalence $[I]$ in polynomial delay and space.*

Proof. Let I be minimal key-ideal. Each step of the While loop in line 1 of Algorithm LIST-CLASS, outputs I and computes all children (at most $|\mathcal{A}|$) of I, and inserts them in the stack Q. In addition, since $\mathcal{G}[I]$ is a tree of depth at most $|\mathcal{A}|$, we first conclude that the size of the stack Q is bounded by $O(|\mathcal{A}|^2)$. So the space used by the algorithm is polynomial. The cost of one iteration of the while loop is dominated by the computation of the set S and checking $\pi(J)$. Clearly S are the successor set in the poset P_Σ which can be computed in polynomial time. On the other side, $\pi(J)$ is also computed in polynomial time by Algorithm 1. So the total time complexity is polynomial in the size of the input. The correctness follows from Lemma 5 since each key-ideal J inserted in Q and output is a child of I. □

3.2. Enumeration of minimal Key-Ideal sets

In the following we give an algorithm to enumerate all minimal key-ideal sets in polynomial delay and for each minimal key-ideal found, we apply algorithm LIST-CLASS to enumerate the key-ideal sets in its class. The strategy of our algorithm is inspired from Lucchesi and Osborn algorithm [11], that enumerates candidate keys of a set of functional dependencies Σ. It starts with a random candidate key, then generates new ones using functional dependencies from Σ: substitute an element a from the current key with a

subset L such that $L \rightarrow a$ is in Σ_{nu}, then minimize using recursively Algorithm 1. Each time a new minimal key-ideal set is generated, a queue of already enumerated minimal key-ideal sets is updated and stored to avoid redundancy.

First, we prove in Theorem 8 that we can enumerate only minimal key-ideal sets in the same way as algorithm in [11].

Theorem 8. *Let $\mathcal{I}$ be a non-empty subset of $\mathcal{I}^K_{min}$. $\mathcal{I} \neq \mathcal{I}^K_{min}$ if and only if $\mathcal{I}$ contains a minimal key-ideal set I and Σ_{nu} contains an implication $L \rightarrow a$ such that $(L \cup I \setminus \{a\}^{\Sigma_u})^{\Sigma_u}$ does not include any key-ideal set in $\mathcal{I}$.*

Proof. We first prove that the stated condition is sufficient to have $\mathcal{I} \neq \mathcal{I}^K_{min}$.

We suppose then that $\mathcal{I}$ contains a minimal key-ideal set I and Σ contains an implication $L \rightarrow a$ such that $(L \cup I \setminus \{a\}^{\Sigma_u})^{\Sigma_u}$ does not include any key-ideal set in $\mathcal{I}$. Since $L \rightarrow a$ is in Σ and a^{Σ_u} is a subset of I, then $(L \cup I)^{\Sigma} = (L \cup I \setminus \{a\}^{\Sigma_u})^{\Sigma}$. Furthermore, $L \cup I$ is a key (because I is key-ideal), hence $(L \cup I \setminus \{a\}^{\Sigma_u})$ is also a key. Which means that $(L \cup I \setminus \{a\}^{\Sigma_u})^{\Sigma_u}$ is a key and an ideal, thus it contains a minimal key-ideal set that does not figure in $\mathcal{I}$ according to the condition.

Second, to prove that the condition is necessary, we assume that there exists a minimal key-ideal set $I' \notin \mathcal{I}$. Let S be an ideal in $\mathcal{I}(\mathcal{P}_{\Sigma})$ verifying: (1) $I' \subseteq S$; (2) for every I'' in $\mathcal{I}$, $I'' \not\subseteq S$ and (3) for every element $a \notin S$, $S \cup \{a\}$ is either not an ideal, or it contains a key-ideal set in $\mathcal{I}$. Note that S is a strict subset of $\mathcal{A}$ and a key, since $\mathcal{I}$ is not empty.

Since $S \neq S^{\Sigma} = X$, then Σ contains $L \rightarrow a$ in Σ_{nu} with L included in S and $a \notin S$. Then according to (3), $S \cup \{a\}$ is either not an ideal or contains a key-ideal set in $\mathcal{I}$. If the latter then we take $a' = a$, else we consider $a' = min_{\preceq}\{b \in a^{\Sigma_u} | b \notin S\}$. Then we have $S \cup \{a'\}$ is an ideal and $a' \notin S$, which implies according to (3) that $S \cup \{a'\}$ contains a key-ideal I in $\mathcal{I}$.

Now we prove that $(L \cup I \setminus \{a\}^{\Sigma_u})^{\Sigma_u}$ is a subset of S. First, $I \setminus a'$ is a subset of S by construction. And since, a' is in $\{a\}^{\Sigma_u}$, then $I \setminus a^{\Sigma_u}$ is a subset of S. Third, S contains L. Then, $L \cup I \setminus \{a\}^{\Sigma_u}$ is a subset of S. In addition, since S is an ideal, then the closure over Σ_u is also in S. We conclude that $(L \cup I \setminus \{a\}^{\Sigma_u})^{\Sigma_u}$ does not include any key-ideal from $\mathcal{I}$. □

We define the super graph of minimal key-ideal sets $\mathcal{G}[\Sigma]$ whose vertices are minimal key-ideal sets and edges represent (I, J) if $J = \pi^*((L \cup I) \setminus \{a\}^{\Sigma_u})$ for some $L \rightarrow a$ in Σ. Recall that $\pi^*(I)$ computes the unique minimal key-ideal included in I.

Proposition 9. *$\mathcal{G}[\Sigma]$ is strongly connected.*

Proof. Let I_0 be a minimal key-ideal set. We show that any other key-ideal set J is accessible from I_0, i.e. there is a path in $\mathcal{G}[\Sigma]$ from I_0 to J. We start with $\mathcal{I} = \{I_0\}$ which is accessible from I_0 in $\mathcal{G}[\Sigma]$. Inductively suppose $\mathcal{I} \neq \mathcal{I}^K_{min}$, using Theorem 8 there is a key-ideal set $I \in \mathcal{I}$ and $L \to a \in \Sigma$ such that $J = \pi^*((L \cup I) \setminus \{a\}^{\Sigma_u})$ is a minimal key-ideal ideal set not in $\mathcal{I}$. Then there exists an edge between I and J and by hypothesis there is a path from I_0 to I, so there is a path from I_0 to J. When Theorem 8 cannot be applied then $\mathcal{I} = \mathcal{I}^K_{min}$ and thus every key-ideal is accessible from I_0 in $\mathcal{G}[\Sigma]$.

Since I_0 is chosen arbitrary, then there is a path from any two key-ideal sets and thus $\mathcal{G}[\Sigma]$ is strongly connected.

□

Example 5. *Following Example 1, Figure 3 illustrates the super graph $\mathcal{G}[\Sigma]$. For example, applying the functional dependency $bd \to g$ to the minimal key-ideal cd, we obtain bd.*

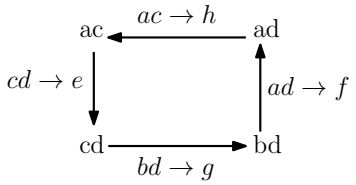

Figure 3: The super graph associated to $\mathcal{I}^K_{min}$ of Σ

Given Σ, Algorithm All-Minimal-Key-ideals follows the depth first search traversal of a tree rooted at $\pi(\mathcal{A})$ of the super graph defined in proposition 9.

Algorithm 3: ALL-MINIMAL-KEY-IDEALS(Σ)

Output: All key-ideal sets $\mathcal{I}^K$ of Σ
begin
  Let $I = \pi^*(\mathcal{A})$;
  $Mark(I) = unmarked$;
1 Output I;
  Insert I to Q;
2 **while** *there is an unmarked* I *in* Q **do**
    $Mark(I) = marked$;
3   **for** $L \to a \in \Sigma_{nu}$ **do**
      $J = (L \cup I) \setminus \{a\}^{\Sigma_u}$;
      $J = \pi^*(J^{\Sigma_u})$;
4   **if** $J \notin Q$ **then**
        $Mark(J) = unmarked$;
5   Output I;
        Insert J to Q;
  Return Q;

Proposition 10. *Algorithm* ALL-MINIMAL-KEY-IDEALS *outputs all minimal key-ideal sets of an implicational base* Σ *in polynomial delay using* $O(|\mathcal{A}|.|\mathcal{I}^{\mathcal{K}}_{min}|)$ *space memory.*

Proof. Algorithm ALL-MINIMAL-KEY-IDEALS follows a depth first search (DFS) of the super graph $\mathcal{G}[\Sigma]$ with root $I = \pi^*(\mathcal{A})$. Each minimal key-ideal set is inserted only one time using the marking, and according to Proposition 9, all the neighbors of a minimal key-ideal set are visited (line 3). Thus minimal key-ideal sets are all output without redundancy.

The time complexity of each iteration of the while loop takes polynomial time to compute all the neighbors of a minimal key-ideal set I.

Each key-ideal set inserted one time in the queue and the size of each one is bounded by $|\mathcal{A}|$. Hence, the total space used by the algorithm is bounded by $O(|\mathcal{A}|.|\mathcal{I}^{\mathcal{K}}_{min}|)$. □

Corollary 11. *There is a polynomial delay algorithm to generate all minimal key-ideal sets and candidate keys using* $O(|\mathcal{A}|.|\mathcal{I}^{\mathcal{K}}_{min}|)$ *space memory.*

Proof. It suffices to replace Output() in lines (in lines 1 and 5) by LIST-CLASS() to compute all key-ideal sets in $[I]$ by searching the tree $\mathcal{G}[I]$. □

The remained open question is whether there exists a polynomial delay and space algorithm to enumerate all minimal key-ideal sets of a relational schema.

Another exciting question is to replace $\Sigma_u \subseteq \Sigma$ by a $\Sigma_{uk} \subseteq \Sigma$ where Σ_{uk} (uk for unit key) corresponds to a closure system having only one or few minimal keys. This is a kind of generalization of distributive lattices to more general lattices, which may be a way to overcome the space complexity to enumerate candidate keys.

Acknowledgment. The authors are grateful to Arnaud Mary for his suggestions and remarks. This research is supported by the French government IDEXISITE initiative 16-IDEX-0001 (CAP 20-25).

References

[1] David Avis and Komei Fukuda. Reverse search for enumeration. *Discrete Applied Mathematics*, 65(1):21–46, 1996. First International Colloquium on Graphs and Optimization.

[2] C. Beeri, M. Dowd, R. Fagin, and R. Statman. On the structure of armstrong relations for functional dependencies. *Journal of the ACM (JACM)*, 31(1):30–46, 1984.

[3] Kristóf Bérczi, Endre Boros, Ondrej Cepek, Petr Kucera, and Kazuhisa Makino. Unique key horn functions. *Theor. Comput. Sci.*, 922:170–178, 2022.

[4] Edgar F Codd. A relational model of data for large shared data banks. *Communications of the ACM*, 13(6):377–387, 1970.

[5] Khaled M. Elbassioni. A polynomial delay algorithm for generating connected induced subgraphs of a given cardinality. *J. Graph Algorithms Appl.*, 19(1):273–280, 2015.

[6] Karima Ennaoui. *Computational aspects of infinite automata simulation and closure system related issues. (Aspects de complexité du problème de composition des services web).* PhD thesis, University of Clermont Auvergne, France, 2018.

[7] Karima Ennaoui and Lhouari Nourine. Hybrid algorithms for candidate keys enumeration for a relational schema. In *Base de Données Avancées 2016, Futuroscope, Poitiers - France.*, 2022.

[8] M. L. Fredman and L. Khachiyan. On the complexity of dualization of monotone disjunctive normal forms. *J. Algorithms*, 21(3):618–628, 1996.

[9] Mamadou Moustapha Kanté, Vincent Limouzy, Arnaud Mary, and Lhouari Nourine. On the enumeration of minimal dominating sets and related notions. *SIAM J. Discrete Math.*, 28(4):1916–1929, 2014.

[10] Stéphane Lopes, Jean-Marc Petit, and Lotfi Lakhal. Functional and approximate dependency mining: database and FCA points of view. *Journal of Experimental & Theoretical Artificial Intelligence*, 14(2-3):93–114, April 2002.

[11] Cláudio L. Lucchesi and Sylvia L. Osborn. Candidate keys for relations. *Journal of Computer and System Sciences*, 17(2):270 – 279, 1978.

[12] Heikki Mannila and Kari-Jouko Raiha. Algorithms for inferring functional dependencies from relations. *Data and Knowledge Engineering*, 12(1):83 – 99, 1994.

[13] L. Nourine and J.-M. Petit. Extending set-based dualization: Application to pattern mining. In *ECAI 2012*, pages 630–635, 2012.

[14] H. Saiedian and T. Spencer. An efficient algorithm to compute the candidate keys of a relational database schema. *The Computer Journal*, 39(2):124–132, 1996.

[15] Nicolas Veyrat-Charvillon, Benoît Gérard, Mathieu Renauld, and François-Xavier Standaert. An optimal key enumeration algorithm and its application to side-channel attacks. In *International Conference on Selected Areas in Cryptography*, pages 390–406. Springer, 2012.

[16] M. Wild. The joy of implications, aka pure Horn formulas: mainly a survey. To appear in *Theoretical Computer Science*.