



# Polynomial delay Hybrid algorithms to enumerate candidate keys for a relation

Karima Ennaoui, Lhouari Nourine

## ► To cite this version:

Karima Ennaoui, Lhouari Nourine. Polynomial delay Hybrid algorithms to enumerate candidate keys for a relation. BDA 2016, Nov 2016, Poitiers, France. hal-03124033v1

**HAL Id: hal-03124033**

**<https://hal.science/hal-03124033v1>**

Submitted on 28 Jan 2021 (v1), last revised 22 Dec 2023 (v2)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Polynomial delay Hybrid algorithms to enumerate candidate keys for a relation

Karima Ennaoui, Lhouari Nourine

<sup>a</sup>*Université Clermont Auvergne, CNRS, 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 [9] 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 [3]), however, it turns out to be a polynomial delay algorithm and exponential space.

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 hybrid<sup>1</sup> 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)

<sup>1</sup>Hybrid means that two algorithms are combined to enumerate candidate keys

## 1. Introduction

This paper is an extended paper of BDA'16[5] and Ennaoui's phd thesis [4].

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 lattice or FCA terminologies [11], 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 [8], systems security [13]).

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 [1, 10], or to the enumeration of minimal dominating sets of a graph [7]. This problem is also known as the enumeration of all minimal generators of a concept in Datamining and FCA communities [14]. The enumeration of all candidate keys from a relation instance can be solved using a quasi-polynomial incremental algorithm [6]. When the input is a set of functional dependencies, the best known result is given by Lucchesi and Osborn [9] with an incremental polynomial time algorithm and an exponential space. Using the state of the art [3] of enumeration technics, however, it turns out to be a polynomial delay algorithm and exponential space. Saiedian and Spencer in [12] 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.

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  $\Sigma$  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  $P_\Sigma$  (poset for short) over the attributes set  $\mathcal{A}$ . For each candidate key  $K$ , we associate the smallest ideal of  $P_\Sigma$  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 set 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  $|\mathcal{I}_{min}^K|$ . 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 space algorithm to enumerate minimal key-ideal sets  $\mathcal{I}_{min}^K$  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  $|\mathcal{I}_{min}^K|$ .

Our main results can be summarized in the following:

**Theorem.** *If there is a polynomial delay and space algorithm to enumerate  $\mathcal{I}_{min}^K$ , then there is one to enumerate all minimal keys in polynomial delay and 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  $Max(I) = \{x \in I \mid \forall y \in I, x \not\preceq y\}$  and  $\mathcal{I}(P)$  the set of all ideal sets of  $P$ .

An implicational base  $\Sigma$  over  $\mathcal{A}$  is defined by a set of functional dependencies (or implications)  $L \rightarrow R$  with  $(L, R) \in 2^{\mathcal{A}} \times 2^{\mathcal{A}}$ . The  $\Sigma$ -closure of a set  $X \subseteq \mathcal{A}$  is the smallest set denoted by  $X^\Sigma$  containing  $X$  and verifying for every  $L \rightarrow R \in \Sigma$  that if  $L \subseteq X^\Sigma$  then  $R \subseteq X^\Sigma$ . An implication  $L \rightarrow R$  is called *unitary* if  $|L| = 1$ . We denote the set of all unitary implication by  $\Sigma_u$  and  $\Sigma_{nu} = \Sigma \setminus \Sigma_u$  the set of non-unitary implication. Without loss of generality, we suppose that  $|R| = 1$  for every implication  $L \rightarrow R$  in  $\Sigma$ . A *key*  $K$  of  $\Sigma$  over a set  $\mathcal{A}$ , is a subset of  $\mathcal{A}$  verifying that  $K^\Sigma = \mathcal{A}$ . A key  $K$  is called *candidate* [2, 9, 12] if none of its proper subsets is a key of  $\Sigma$ . We denote by  $\mathcal{K}$  the set of all candidate keys of  $\Sigma$ , also referred to as minimal keys in the literature.

First, we show that a candidate key cannot contain two equivalent attributes and there 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.  $\square$

Whenever,  $\Sigma$  contains equivalent attributes, then we keep only one representative attribute for each set of equivalent attributes and consider a reduced implicational base  $\Sigma'$  obtained from  $\Sigma$  by replacing attributes of each equivalent set by its representative. For example, consider the implicational base  $\Sigma = \{a \rightarrow b, b \rightarrow c, c \rightarrow a, bd \rightarrow e, ce \rightarrow d\}$  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  $\Sigma' = \{ad \rightarrow e, ae \rightarrow d\}$ . From lemma 1, any candidate key  $K$  of  $\Sigma'$  containing the representative  $a$  then  $(K \setminus \{a\}) \cup \{b\}$  and  $(K \setminus \{a\}) \cup \{c\}$  are also candidate keys. Then the keys of  $\Sigma'$  is  $\{ad, ae\}$ , and the keys of  $\Sigma$  are  $\{ad, be, bd, ce, ad, ce\}$ .

In the rest of the paper, we assume that  $\Sigma_u$  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  $a \in b^{\Sigma_u}$ . We define  $\text{pred}(a) = \{b \in X \mid \nexists c \in X, \text{ such that } a < c < b\}$ .

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

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

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

From the previous definition, 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 equivalent.

### 3. Enumeration of key-ideals

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

Define the function  $\pi : \mathcal{I}^K \rightarrow \mathcal{I}^K$ , with  $\pi(J) = J'$  such that  $Max(J')$  is the colexicographically largest subset of  $J$  and  $J'$  is a key-ideal. Notice that the function  $\pi$  is well defined for every non minimal key-ideal, since the set of all key-ideal included in  $J$  can be totally ordered by the colexicographical order.

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

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

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

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  $I \in \mathcal{I}_{min}^K$ . The set of all equivalence classes of  $\sim$  on  $\mathcal{I}^K$ , denoted  $\mathcal{A}/\sim = \mathcal{I}_{min}^K$ , is the quotient of the relation  $\sim$ .

**Example 1.** *Consider the following set of functional dependencies  $\Sigma$  with  $\Sigma_u = \{e \rightarrow ab, f \rightarrow bc, g \rightarrow c, h \rightarrow d\}$  and  $\Sigma_{nu} = \{ac \rightarrow efgh, ad \rightarrow efgh, bd \rightarrow efgh, cd \rightarrow efgh\}$ . The set of all candidate keys are  $\mathcal{K}(\Sigma) = \{ac, ad, bd, cd, af, ag, ce, eg, ah, de, eh, bh, df, fh, ch, dg, gh\}$ .*

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 the quotient  $\mathcal{I}_{min}^K$ . By combining the two algorithms, we derive a polynomial delay algorithm to enumerate  $\mathcal{I}^K$  and thus, candidate keys using  $O(|\mathcal{I}_{min}^K|)$  space.

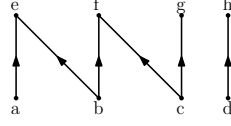


Figure 1: The poset associated to  $\Sigma$  in example 1

### 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  $[I]$ . We call  $J$  a child of  $J'$  if  $J' = \pi(J)$ . 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.**  $\mathcal{G}[I]$  is a tree rooted at  $I$ .

*Proof.* Let  $J$  be any non-minimal Key-ideal in  $[I]$ . Since  $J$  has exactly one parent and  $I$  has no parent,  $\mathcal{G}[I]$  has  $|[I]| - 1$  edges. In addition,  $(J, J')$  in  $\mathcal{G}[I]$ , we have  $I \subseteq J \subset J'$ . By recursively applying the parent function to  $J$  at most  $|J \setminus I|$  times we reach  $I$ . Then  $\mathcal{G}[I]$  is connected, and is a tree with root  $I$ .  $\square$

**Example 2.** Figure 1 shows the poset  $P_\Sigma$ , Figure 2 (a), (b), (c) and (d) illustrate trees associated to all equivalence classes of key-ideal sets.

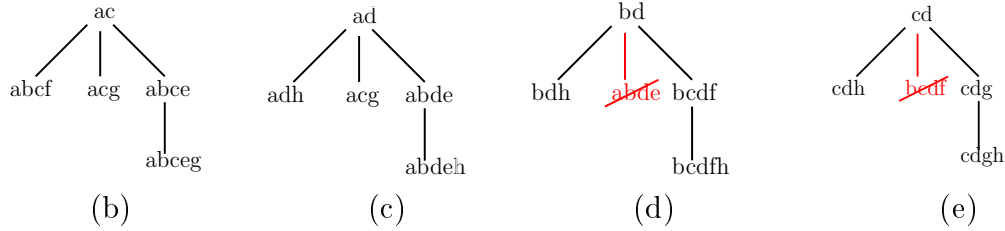


Figure 2:

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

**Lemma 5.** Let  $J$  and  $J'$  be two key-ideal sets. If  $J' = \pi(J)$  then exists  $a \in \mathcal{A} \setminus J'$  such that  $J = J' \cup a^{\Sigma_u}$ .

*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 have  $J \setminus \{a\}$  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 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  $a^{\Sigma_u}$  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 co-lexicographically 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 conclude 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.  $\square$



To traverse exactly the tree  $\mathcal{G}[I]$ , for each potential child  $J \cup a^{\Sigma_u}$  for some  $a \in \mathcal{A} \setminus J$ , we check if  $J = \pi(J \cup a^{\Sigma_u})$ .

---

**Algorithm 1:** PARENT ( $J \in \mathcal{I}^K$ )

---

```

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

```

---

**Lemma 6.** *For every key-ideal  $J$ , Algorithm 1 computes  $\pi(J)$ .*

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  $I \in \mathcal{I}_{min}^K$ , Algorithm LIST-CLASS follows the depth first search traversal of the tree defined in proposition 4. It uses a stack  $F$  to store key-ideal sets in  $[I]$  that are not visited.

---

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

---

**Output:** The key-ideal sets in the equivalence classe  $[I]$

**begin**

```

1   Insert  $I$  to  $F$ ;  $\{F$  a stack $\}$ 
   while  $F$  not empty do
       Pull  $I$  from  $F$ ;
       Output  $I$ ;
2   Let  $S = \{b \in \mathcal{A} \mid a \in b^{\Sigma_u}, a \in \text{Max}(I)\}$ ;
3   while  $S$  is not empty do
4        $a :=$  the lexicographically smallest element in  $S$ ;
5        $J = I \cup \{a\}^{\Sigma_u}$ ;
6       if  $I = \text{Parent}(J)$  then
           Push  $J$  to  $F$ ;

```

---

**Proposition 7.** *For a minimal key-ideal  $I$  in  $\mathcal{I}_{min}^K$ , Algorithm LIST-CLASS outputs all key-ideal sets of the equivalence  $[I]$  in delay  $O(|\mathcal{A}|^2 \cdot |\Sigma|)$  using  $O(|\mathcal{A}|^3)$  space memory.*

*Proof.* Each step of the While loop in line 2 of Algorithm LIST-CLASS, computes all children (at most  $n$ ) of a key-ideal  $I$  and insert them in the stack  $F$ .  $\square$

### 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 [9], that enumerates candidate keys of a set of functional dependencies  $\Sigma$ . It starts with a random candidate key, then generates new ones using functional dependencies from  $\Sigma$ : substitute an element  $a$  from the current key with a subset  $L$  such that  $L \rightarrow a$  is in  $\Sigma_{nu}$ , then minimize using recursively algorithm Parent. Each time a new minimal key-ideal is generated, the set of already enumerated keys is revisited to ensure that there is no redundancy.

First, we prove in theorem 8 that we can enumerate only minimal key-ideal sets in the same way as this algorithm.

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

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

Then,  $\mathcal{I}$  contains a minimal key-ideal set  $I$  and  $\Sigma$  contains an implication  $L \rightarrow a$  such that  $(L \cup I \setminus \{a\})^{\Sigma_u}$  does not include any key-ideal set in  $\mathcal{I}$ . Since  $L \rightarrow a$  is in  $\Sigma$  and  $a^{\Sigma_u}$  is a subset of  $I$ , then  $(L \cup I)^{\Sigma} = (L \cup I \setminus \{a\})^{\Sigma}$ . Moreover  $L \cup I$  is a key (because  $I$  is key-ideal), hence  $(L \cup I \setminus \{a\})^{\Sigma_u}$  is also a key. Which means that  $(L \cup I \setminus \{a\})^{\Sigma_u}$  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  $S \neq S^{\Sigma} = X$ , then  $\Sigma$  contains  $L \rightarrow a$  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 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  $(L \cup I \setminus \{a\})^{\Sigma_u}$  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  $S$ . And since  $S$  is an ideal, then the closure over  $\Sigma_u$  is also in  $S$ . We conclude that  $(L \cup I \setminus \{a\})^{\Sigma_u}$  does not include any key-ideal from  $\mathcal{I}$ .  $\square$

**Example 3.** *Following example 1, figure 3 illustrates how to find minimal key ideals using functional dependencies in  $\Sigma_{nu}$ . For example, applying the functional dependency  $bd \rightarrow c$ , we obtain  $bd$ .*

ALL-KEY-IDEALS is a polynomial delay algorithm that enumerates all key-ideal sets.

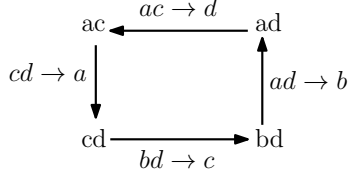


Figure 3: The super graph associated to  $\mathcal{I}_{min}^K$  of  $\Sigma$

---

**Algorithm 3:** ALL-KEY-IDEALS( $\Sigma$ )

---

**Output:** All key-ideal sets  $\mathcal{I}^K$  of  $\Sigma$

**begin**

    Let  $I = \text{Parent}^*(\mathcal{A})$ ;

    Insert  $I$  to  $Q$ ;

$\text{Mark}(I) = \text{unmarked}$ ;

**while** there is an unmarked  $I$  in  $Q$  **do**

$\text{Mark}(I) = \text{marked}$ ;

**for**  $L \rightarrow a \in \Sigma_{nu}$  **do**

$J = (L \cup I) \setminus \{a\}^{\Sigma_u}$ ;

$J = \text{Parent}^*(J^{\Sigma_u})$ ;

**if**  $J \notin Q$  **then**

$\text{Mark}(J) = \text{unmarked}$ ;

                Insert  $J$  to  $Q$ ;

                LIST-CLASS( $J$ );

1

    Return  $Q$ ;

---

**Proposition 9.** Algorithm ALL-KEY-IDEALS outputs all minimal key-ideal sets of an implicational base  $\Sigma$  in delay  $O(|X|^2 \cdot |\Sigma|^2)$  using  $O(|X| \cdot |\mathcal{I}_{min}^K|)$  space memory.

*Proof.* When the last key-ideal generated is minimal, it takes at most:

- $O(|\Sigma| \times \text{Complexity}(\text{LLM})) = O(|\Sigma|^2 \times |\mathcal{A}|^2)$ , and
- $O(|\mathcal{A}| \times |\Sigma|)$  to generate a non minimal key-ideal using Algorithm LIST-CLASS.

When the last key-ideal generated is not minimal, then it takes at most  $O(|\mathcal{A}| \times |\Sigma|)$  to generate a new solution.

As for the space, we only to stock the minimal key-ideal sets in this algorithm.  $\square$

**Corollary 10.** *Algorithm ALL-KEY-IDEALS outputs all key-ideal sets of an implicational base  $\Sigma$  in delay  $O(|X|^2 \cdot |\Sigma|)$  using  $O(|X| \cdot |\mathcal{I}_{min}^K|)$  space memory.*

The open remained question is whether exists a polynomial delay and space algorithm to enumerate all candidate keys of a relational schema.

## 4. Discussions

We give in this paper an efficient algorithm that enumerates candidate keys, when an order structure is present in the relational schema.

The open remained question is whether exists a polynomial delay and space algorithm to enumerate all candidate keys of a relational schema.

The proposed framework can also be improved using the idea in [12] where they consider acyclicity of  $\Sigma$ . Our work can be applied to each strongly connected component of  $\Sigma$ . It can also be applied to minimal generators in FCA terminology.

*Acknowledgment.* The authors are grateful to Arnaud Mary for his constructive suggestions and remarks. This work has been funded by the french national research agency (ANR Graphen, 2015-2018) and CNRS (Mastodons Qualisky project, 2016).

## References

- [1] 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.
- [2] Edgar F Codd. A relational model of data for large shared data banks. *Communications of the ACM*, 13(6):377–387, 1970.
- [3] 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.

- [4] 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.
- [5] Karima Ennaoui and Lhouari Nourine. Hybrid algorithms for candidate keys enumeration for a relational schema par et.
- [6] M. L. Fredman and L. Khachiyan. On the complexity of dualization of monotone disjunctive normal forms. *J. Algorithms*, 21(3):618–628, 1996.
- [7] 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.
- [8] 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.
- [9] Cláudio L. Lucchesi and Sylvia L. Osborn. Candidate keys for relations. *Journal of Computer and System Sciences*, 17(2):270 – 279, 1978.
- [10] Heikki Mannila and Kari-Jouko Raiha. Algorithms for inferring functional dependencies from relations. *Data and Knowledge Engineering*, 12(1):83 – 99, 1994.
- [11] L. Nourine and J.-M. Petit. Extending set-based dualization: Application to pattern mining. In *ECAI 2012*, pages 630–635, 2012.
- [12] 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.
- [13] Nicolas Veyrat-Charvillon, Benoît Gérard, Mathieu Renaud, 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.
- [14] M. Wild. The joy of implications, aka pure Horn formulas: mainly a survey. To appear in *Theoretical Computer Science*.