Dresden University of Technology
Institute for Theoretical Computer Science
Chair for Automata Theory

# LTCS–Report

# Some Computational Problems Related to Pseudo-intents

Barış Sertkaya

LTCS-Report 08-06

Lehrstuhl für Automatentheorie
Institut für Theoretische Informatik
TU Dresden
http://lat.inf.tu-dresden.de

Noethnitzer Str. 46
01187 Dresden
Germany

# Some Computational Problems Related to Pseudo-intents

Barış Sertkaya [⋆]

TU Dresden, Germany
sertkaya@tcs.inf.tu-dresden.de

**Abstract.** We investigate the computational complexity of several decision, enumeration and counting problems related to pseudo-intents. We show that given a formal context and a set of its pseudo-intents, checking whether this context has an additional pseudo-intent is in coNP and it is at least as hard as checking whether a given simple hypergraph is saturated. We also show that recognizing the set of pseudo-intents is also in coNP and it is at least as hard as checking whether a given hypergraph is the transversal hypergraph of another given hypergraph. Moreover, we show that if any of these two problems turns out to be coNP-hard, then unless P = NP, pseudo-intents cannot be enumerated in output polynomial time. We also investigate the complexity of finding subsets of a given Duquenne-Guigues Base from which a given implication follows. We show that checking the existence of such a subset within a specified cardinality bound is NP-complete, and counting all such minimal subsets is #P-complete.

## 1 Introduction

Pseudo-intents are of significant importance in Formal Concept Analysis [8]. They are the premises of the implications in the Duquenne-Guigues Base [10], which is a minimum cardinality set of implications, from which all valid implications of the underlying formal context follow. Since their introduction, computational problems related to pseudo-intents has attracted attention among the researchers in the community.

One central computational problem related to pseudo-intents is determining whether a given set is a pseudo-intent of a given formal context. It has been shown in [14, 15] that this problem is in coNP. Unfortunately, the lower complexity bound for this problem is still open. One other natural problem is enumerating the pseudo-intents of a given formal context. The most well known algorithm

[⋆] Supported by German Research Foundation (DFG, BA 1122/12-1)

for this purpose is the algorithm known as *next closure* [7]. However, this algorithm always enumerates the concept intents as well as a by product. Recently, an *attribute incremental* algorithm that computes the pseudo-intents by processing a single attribute at a single step has been introduced in [17]. In [18], an algorithm for checking whether a set is pseudo-intent, has been presented. This algorithm runs in time exponential in the size of the attribute set of the given context. There, it has also been shown that this algorithm can be optimized by using it on the reduced context. Another problem related to pseudo-intents is given a formal context, determining the number of its pseudo-intents. In [13], it has been shown that this counting problem is #P-hard, and in addition to this, the number of pseudo-intents can be exponential in the size of the incidence relation of the formal context. Given this fact, it is clearly not possible to enumerate all pseudo-intents of a formal context in time polynomial time in the size of its incidence relation. In complexity theory, for analyzing the performance of enumeration algorithms where the number of solutions can be exponential in the size of the input, one considers other measures. One such measure is to take into account not only the size of the input, but also the size of the output. An algorithm is said to run in *output polynomial* time [12] if it enumerates the solutions in time polynomial in the size of the input *and the output*. One advantage of an output polynomial algorithm is that it runs in polynomial time (in the size of the input) when there are only polynomially many solutions.

In the present work we investigate whether pseudo-intents can be enumerated in output polynomial time. To this purpose we formulate two decision problems that are of significant importance to answer this question. In Section 3 we show that given a formal context $\mathbb{K}$ and a set of its pseudo-intents $\mathcal{P}$, checking whether $\mathbb{K}$ has an additional pseudo-intent, i.e., a pseudo-intent that does not already appear in $\mathcal{P}$, is in coNP and it is at least as hard as checking whether a given simple hypergraph is saturated [3], which is a prominent open problem in hypergraph theory [2]. In Section 4 we show that recognizing the set of pseudo-intents, i.e., given a formal context $\mathbb{K}$ and a set $\mathcal{P}$ of subsets of its attribute set, checking whether $\mathcal{P}$ is precisely the set of pseudo-intents of $\mathbb{K}$ is also in coNP and it is at least as hard as checking whether a given hypergraph is the transversal hypergraph of another given hypergraph [3], which is also an open problem. Moreover, we show that if any of these two problems turns out to be coNP-hard, then unless P = NP, pseudo-intents cannot be enumerated in output polynomial time. In Section 5 we investigate the complexity of finding subsets of a given Duquenne-Guigues Base from which a given implication follows. We show that checking the existence of such a subset within a specified cardinality bound is NP-complete, and counting all such minimal subsets is #P-complete.

## 2 Preliminaries

We briefly introduce basic notions of Formal Concept Analysis [8]. Given a *formal context* $\mathbb{K} = (G, M, I)$ with the *derivation operator* $(\cdot)'$, and an *implication* $P \to Q$, where $P, Q \subseteq M$, we say that $P \to Q$ *holds* in $\mathbb{K}$ if the *objects* that have the *attributes* in $P$ also have the attributes in $Q$, i.e., $P' \subseteq Q'$. A set $A \subseteq M$ *respects* an implication $P \to Q$ if $P \not\subseteq A$ or $Q \subseteq A$. An implication $P \to Q$ *follows semantically* from a set of implications $\mathcal{L}$ (written $\mathcal{L} \models P \to Q$) if each subset of $M$ respecting the implications in $\mathcal{L}$ also respects $P \to Q$. We denote the *implicational theory* of $\mathcal{L}$, i.e, the set of all implications that follow from $\mathcal{L}$, with $Imp(\mathcal{L})$.

In [10], a minimum cardinality set of implications from which all valid implications of a given context follow has been characterized. It is called the *Duquenne-Guigues Base* of a formal context. The left handsides of the implications in a Duquenne-Guigues Base are called the pseudo-intents of the underlying formal context. A set $P \subseteq M$ is a *pseudo-intent* if $P \neq P''$ and $Q'' \subsetneq P$ holds for every pseudo-intent $Q \subsetneq P$. Equivalently, a set $P \subseteq M$ is a pseudo-intent if $P \neq P''$, it is a quasi-intent, and for every quasi-intent $Q \subsetneq P$, $Q'' \subsetneq P$ holds, where a *quasi-intent* is defined as a set $Q \subseteq M$ that satisfies $R'' \subseteq Q$ or $R'' = Q''$ for any $R \subseteq Q$.

### 2.1 Hypergraphs and related problems

A *hypergraph* [2] $\mathcal{H} = (V, \mathcal{E})$ consists of a set of *vertices* $V = \{v_i \mid 1 \leq i \leq n\}$, and a set of *(hyper)edges* $\mathcal{E} = \{E_j \mid 1 \leq j \leq m\}$ where $E_j \subseteq V$. A set $W \subseteq V$ is called a *transversal* of $\mathcal{H}$ if it intersects all edges of $\mathcal{H}$, i.e., $\forall E \in \mathcal{E}.\ E \cap W \neq \emptyset$. A transversal is called *minimal* if no proper subset of it is a transversal. The set of all minimal transversals of $\mathcal{H}$ constitute another hypergraph on $V$ called the *transversal hypergraph* of $\mathcal{H}$, which is denoted by $Tr(\mathcal{H})$. Generating $Tr(\mathcal{H})$ is an important problem which has applications in many fields of computer science. It is defined as follows:

**Problem:** TRANSVERSAL ENUMERATION (TRANS-ENUM)
*Input:* A hypergraph $\mathcal{H} = (V, \mathcal{E})$ on a finite set $V$.
*Output:* The edges of the transversal hypergraph $Tr(\mathcal{H})$.

The well known decision problem associated to this computation problem is defined as follows:

**Problem:** TRANSVERSAL HYPERGRAPH (TRANS-HYP)
*Input:* Two hypergraphs $\mathcal{H} = (V, \mathcal{E}_{\mathcal{H}})$ and $\mathcal{G} = (V, \mathcal{E}_{\mathcal{G}})$.
*Question:* Is $\mathcal{G}$ the transversal hypergraph of $\mathcal{H}$, i.e., does $Tr(\mathcal{H}) = \mathcal{G}$ hold?

Computational complexity of these problems have now been extensively studied [3, 5, 6] and many important applications of these problems have been identified in logic and artificial intelligence [4], databases [16] and data mining [11]. TRANS-HYP is known to be in coNP, but so far neither a polynomial time algorithm has been found, nor has it been proved to be coNP-complete. Similarly, it is an open problem whether TRANS-ENUM can be solved in output polynomial time. We say that a decision problem $\Pi$ is TRANS-HYP-hard if TRANS-HYP can be reduced to $\Pi$ by a standard polynomial transformation. In the following we are often going to make use of this notion for reducing some problems on hypergraphs to some problems on pseudo-intents.

## 3 Complexity of enumerating pseudo-intents

For enumerating pseudo-intents, unfortunately no output polynomial algorithm is known currently. The most well known algorithm *next closure* [7] for enumerating the pseudo-intents always enumerates the concept intents as well, i.e, its running time depends not only on the number of pseudo-intents but also on the number of concept intents. Since the number of concept intents can be exponential in the number of pseudo-intents, this algorithm in general does not run in output polynomial time. Similarly, the *attribute-incremental algorithm* in [17] has also time complexity depending on both the number of pseudo-intents and the number of concept intents. In the light of our current knowledge, it is not even clear whether there can be an algorithm at all that enumerates pseudo-intents in output polynomial time. In order to investigate this further, let us first formally define the problem.

**Problem:** PSEUDO-INTENT ENUMERATION (PIE)
*Input:* A formal context $\mathbb{K}$.
*Output:* The set of pseudo-intents of $\mathbb{K}$.

For solving this enumeration problem, the following decision problem has crucial importance:

**Problem:** ADDITIONAL PSEUDO-INTENT (API)
*Input:* A formal context $\mathbb{K} = (G, M, I)$, and a set $\mathcal{P}$ of pseudo-intents of $\mathbb{K}$, i.e., $\mathcal{P} = \{P \mid P \subseteq M,\ P \textit{ pseudo-intent of } \mathbb{K}\}$.
*Question:* Is there an additional pseudo-intent, i.e., $Q \subseteq M$ s.t. $Q$ is a pseudo-intent of $\mathbb{K}$ and $Q \notin \mathcal{P}$?

Because, as Proposition 1 below shows, if this problem cannot be decided in polynomial time, then unless P = NP, PIE cannot be solved in output polynomial time.

**Proposition 1.** *If* API *cannot be decided in polynomial time, then unless* P = NP*,* PIE *cannot be solved in output-polynomial time.*

*Proof.* Assume that we have an algorithm $\mathcal{A}$ that solves PIE in output-polynomial time. Let its runtime be bounded by a polynomial $p(IS, OS)$ where $IS$ denotes the size of the input context and $OS$ denotes the size of the output, i.e., the set of all pseudo-intents of the input context.

In order to decide API for an instance given by the context $\mathbb{K}$ and a set $\mathcal{P}$ of pseudo-intents of $\mathbb{K}$, we construct another algorithm $\mathcal{A}'$ that works as follows: It runs $\mathcal{A}$ on $\mathbb{K}$ for at most $p(|\mathbb{K}|, |\mathcal{P}|)$-many steps. If $\mathcal{A}$ terminates within $p(|\mathbb{K}|, |\mathcal{P}|)$-many steps, it means that $\mathcal{P}$ contains all pseudo-intents of $\mathbb{K}$, i.e., there is no additional pseudo-intent. So $\mathcal{A}'$ returns *no.* If $\mathcal{A}$ does not terminate after $p(|\mathbb{K}|, |\mathcal{P}|)$-many steps, this implies that there is an additional pseudo-intent that is not contained in $\mathcal{P}$, so $\mathcal{A}'$ returns *yes*. It is easy to see that the runtime of $\mathcal{A}'$ is bounded by a polynomial in $|\mathbb{K}|$ and $|\mathcal{P}|$, that is $\mathcal{A}'$ decides API in time polynomial in the size of the input. □

The proposition shows that determining the complexity of API is indeed crucial for determining the complexity of PIE. In the following we show that API is in coNP, and it is at least as hard as the complement of a prominent open problem on hypergraphs. However, whether API is coNP-hard remains unfortunately open.

**Proposition 2.** API *is in co*NP.

*Proof.* Given an instance of API with the input $\mathbb{K}$ and $\mathcal{P}$, construct the set of implications $\mathcal{L} = \{P \to P'' \mid P \in \mathcal{P}\}$ and nondeterministically guess a set $Q \subseteq M$. We can verify in polynomial time that $Q \to Q''$ *does not* follow from $\mathcal{L}$, i.e., the complement of the problem is in NP, thus API is in coNP. □

Before we can continue with the proof of lower bound, we need to introduce some more notions from hypergraphs. A hypergraph $\mathcal{H} = (V, \mathcal{E})$ is called *saturated* [3] if every subset of $V$ is contained in at least one of the edges of $\mathcal{H}$, or it contains at least one edge of $\mathcal{H}$, i.e., for every $W \subseteq V$, $W \subseteq E$ holds, or $E \subseteq W$ holds for some $E \in \mathcal{E}$. It has been shown in [3] that checking whether a hypergraph is saturated is coNP-complete. There, a special case of the problem where the given hypergraph is restricted to be simple, has also been considered. A hypergraph is called *simple* if no edge contains another edge.

**Problem:** SIMPLE HYPERGRAPH SATURATION (SIMPLE-H-SAT)
*Input:* A simple hypergraph $\mathcal{H} = (V, \mathcal{E})$, i.e., $\forall E, E' \in \mathcal{E}. E \subseteq E' \Rightarrow E = E'$.
*Question:* Is $\mathcal{H}$ saturated, i.e., is it true that for every $W \subseteq V$, $W \subseteq E$ holds or $E \subseteq W$ holds for some $E \in \mathcal{E}$?

It is not difficult to see that this problem is in coNP. However, up to now there has neither been a proof that it is coNP-hard, nor a proof that it is in P. It has been shown in [3] that this problem is under polynomial transformations computationally equivalent to TRANS-HYP, which as mentioned before is a prominent open problem in hypergraph theory. In the following we show that our problem API is at least as hard as the complement of SIMPLE-H-SAT:

**Theorem 1.** API *is co*SIMPLE-H-SAT-*hard.*

*Proof.* Let an instance of SIMPLE-H-SAT be given with the simple hypergraph $\mathcal{H} = (V, \mathcal{E})$ where $\mathcal{E} = \{E_1, \ldots, E_n\}$. From $\mathcal{H}$ we construct the formal context $\mathbb{K}_{\mathcal{H}} = (G, M, I)$ where $M = V$, and $G$ and $I$ are defined as follows: For every $E_i$, $1 \leq i \leq n$, we create the following objects: For every $D \subsetneq E_i$ such that $|D| = |E_i| - 1$, we create an object with the intent $D$. $E_i$ has $|E_i|$-many such subsets. We name these objects as $g_{ij}$ where $1 \leq i \leq n$ and $1 \leq j \leq |E_i|$. In total, $G$ contains $\sum_{i=1}^{n} |E_i|$ objects. We construct $\mathcal{P}$ by just taking the the edges of $\mathcal{H}$, i.e, $\mathcal{P} = \{E_1, \ldots, E_n\}$. Obviously, both $\mathbb{K}_{\mathcal{H}}$ and $\mathcal{P}$ can be constructed in time polynomial in the size of $\mathcal{H}$.

Note that $\mathbb{K}_{\mathcal{H}}$ has the following property: Since $\mathcal{H}$ is simple no edge is contained in another edge, and obviously not in strict subsets of any other edge. Then, for every $i$ such that $1 \leq i \leq n$, $E_i' = \emptyset$ and $E_i'' = M$. That is $E_i$ is not closed. Moreover its all strict subsets are closed. Because for every $D \subsetneq E_i$ either there is an object whose intent is $D$, or there is a set of objects such that the intersection of their intents is $D$. This is due to the objects $g_{ij}$, where $1 \leq j \leq |E_i|$, whose intents are strict subsets of $E_i$ with cardinality $|E_i| - 1$. Thus, the edges $E_i$ are pseudo-intents of $\mathbb{K}_{\mathcal{H}}$, which means that $\mathbb{K}_{\mathcal{H}}$ and $\mathcal{P}$ indeed form an instance of API. We claim that $\mathcal{H}$ is *not* saturated if and only if $\mathbb{K}_{\mathcal{H}}$ has an additional pseudo-intent.

($\Rightarrow$) Assume $\mathcal{H}$ is not saturated. Then, there exists a $W \subseteq V$ such that for every $i$ such that $1 \leq i \leq n$, $W \nsubseteq E_i$ holds and $E_i \nsubseteq W$ holds. Assume without loss of generality that $W$ is minimal with respect to property $W \nsubseteq E_i$. Since $W$ is not contained in any $E_i$, and obviously not contained in any strict subset of any $E_i$, $W' = \emptyset$ and $W'' = M$. That is $W$ is not closed. Take any $X \subsetneq W$. Since $W$ is minimal, $X \subseteq E_i$ holds for some $1 \leq i \leq n$. We know that $E_i \nsubseteq W$, then $X = E_i$ cannot hold, thus $X$ satisfies $X \subsetneq E_i$. Since all strict subsets of $E_i$ are closed, $X$ is closed. We have shown that $W$ is not closed but its all strict subsets are closed, thus $W$ is a pseudo-intent. Moreover, it is an additional pseudo-intent since $W \neq E_j$, for all $1 \leq j \leq n$.

($\Leftarrow$) Assume $\mathbb{K}_{\mathcal{H}}$ has an additional pseudo-intent, i.e., a pseudo-intent $Q$ such that $Q \neq E_i$ for every $1 \leq i \leq n$. Since strict subsets of $E_i$ are closed, $Q$ cannot be a strict subset of any $E_i$. Thus $Q \not\subseteq E_i$ for every $1 \leq i \leq n$. Moreover, by definition $Q$ contains the closure of strictly smaller pseudo-intents. We know that for every $1 \leq i \leq n$, $E_i$ is a pseudo-intent, and $E_i'' = M$. Since $Q$ does not strictly contain $M$, it cannot strictly contain any $E_i$ either. Together with $Q \neq E_i$, this implies that $E_i \not\subseteq Q$. We have shown that there exists a $Q \subseteq V$ such that $Q \not\subseteq E_i$ and $E_i \not\subseteq Q$ for every $1 \leq i \leq n$, thus $\mathcal{H}$ is not saturated. □

The following is an immediate consequence of Theorem 1 above and Theorem 4.12 in [3]:

**Corollary 1.** API *is co*TRANS-HYP-*hard.*

Theorem 1 has some interesting consequences. The formal context we have constructed in the proof has a special property; namely, subsets of object intents also closed in this formal context. The proof suggests that for the formal contexts of this form, the problem API and the complement problem of SIMPLE-H-SAT are computationally equivalent problems, i.e., API is coSIMPLE-H-SAT-complete. For such formal contexts, in addition to the reduction given in the proof, one can also easily reduce API to the complement of SIMPLE-H-SAT, i.e, take an instance of API given with such a context and a set of pseudo-intents of this context, construct an instance of SIMPLE-H-SAT and show that there is an additional pseudo-intent if and only if the constructed simple hypergraph is not saturated. It would definitely be interesting to investigate whether formal contexts of this form are natural in some application domains.

One other point that should be noted here is that SIMPLE-H-SAT lies at the boundary of intractability. As mentioned before, for arbitrary graphs it is coNP-complete [3]. The proof of Theorem 1 depends on the fact that the given hypergraph is simple. Whether this restriction can be eliminated and thus the intractability result carries over to API for arbitrary formal contexts, is definitely an interesting question that should be investigated.

## 4 Complexity of recognizing the set of pseudo-intents

Next we consider another problem about pseudo-intents, namely recognizing the set of pseudo-intents. More precisely, given a formal context $\mathbb{K} = (G, M, I)$ and a set $\mathcal{P} \subseteq \mathscr{P}(M)$ it is the problem of deciding whether $\mathcal{P}$ is precisely the set of pseudo-intents of $\mathbb{K}$. Clearly, this problem can also be formulated as: Given a formal context $\mathbb{K}$ and a set of implications $\mathcal{L}$, decide whether $\mathcal{L}$ is the Duquenne-Guigues Base of $\mathbb{K}$. In the following we are going to investigate its computational complexity. We start with defining the problem formally:

**Problem:** PSEUDO-INTENTS (PIS)
*Input:* A formal context $\mathbb{K} = (G, M, I)$, and a set $\mathcal{P} \subseteq \mathscr{P}(M)$.
*Question:* Is $\mathcal{P}$ precisely the set of pseudo-intents of $\mathbb{K}$?

The following proposition shows that like computational complexity of API, the complexity of PIS has also crucial importance for the solvability of PIE in output polynomial time.

**Proposition 3.** *If* PIS *cannot be decided in polynomial time, then unless* P = NP, PIE *cannot be solved in output-polynomial time.*

*Proof.* The proof is almost the same as the proof of Proposition 1. Again we assume that we have an algorithm $\mathcal{A}$ that solves PIE in output-polynomial time and construct another algorithm $\mathcal{A}'$ that runs $\mathcal{A}$ for at most $p(|\mathbb{K}|, |\mathcal{P}|)$-many steps. The only difference is that, if $\mathcal{A}$ terminates within $p(|\mathbb{K}|, |\mathcal{P}|)$-many steps, then $\mathcal{A}'$ first compares the output of $\mathcal{A}$ with $\mathcal{P}$ and then returns *yes* if and only if they are equal. If they are not equal, or if $\mathcal{A}$ has not yet terminated, then $\mathcal{A}'$ returns *no*. Thus if PIE can be solved in output polynomial time, PIS can be decided in polynomial time. □

In the following we show that just like in the case of API, PIS is also in coNP, and it is at least as hard as TRANS-HYP. However, whether PIS is polynomial, or it is coNP-hard also remains open.

**Proposition 4.** PIS *is in co*NP.

*Proof.* Given an instance with the input $\mathbb{K} = (G, M, I)$ and $\mathcal{P}$, an algorithm that decides PIS for this instance first checks whether the elements of $\mathcal{P}$ are pseudo-intents of $\mathbb{K}$. If it encounters an element that is not a pseudo-intent, it terminates and returns *no*. If every $P \in \mathcal{P}$ is a pseudo-intent, then it continues with the second step. This step is the same as the algorithm in the proof of Proposition 2. The algorithm constructs the set of implications $\mathcal{L} = \{P \to P'' | P \in \mathcal{P}\}$ and non-deterministically guesses a set $Q \subseteq M$. Obviously the implication $Q \to Q''$ holds in $\mathbb{K}$, thus if $\mathcal{L}$ is a base for $\mathbb{K}$ then $Q \to Q''$ follows from $\mathcal{L}$. Then the algorithm verifies that this is *not* the case.

It is not difficult to see that this is a coNP algorithm. In the first step the algorithm performs polynomially-many checks each of which can be done in coNP by using the algorithm in [14]. In the second step the algorithm non-deterministically guesses a $Q$ and in polynomial time verifies that $Q \to Q''$ does *not* follow from $\mathcal{L}$, which means that $\mathcal{L}$ is *not* a base, which implies that $\mathcal{P}$ is *not* the set of all pseudo-intents of $\mathbb{K}$. This step can be performed in coNP as well, thus the whole algorithm is a coNP algorithm. □

**Theorem 2.** PIS *is* TRANS-HYP-*hard.*

*Proof.* Let an instance of TRANS-HYP be given by the hypergraphs $\mathcal{H} = (V, \mathcal{E}_{\mathcal{H}})$ and $\mathcal{G} = (V, \mathcal{E}_{\mathcal{G}})$, where $\mathcal{E}_{\mathcal{H}} = \{h_i \mid 1 \leq i \leq n\}$ and $\mathcal{E}_{\mathcal{G}} = \{g_i \mid 1 \leq i \leq m\}$. From $\mathcal{H}$ we construct the context $\mathbb{K}_{\mathcal{H}} = (G, M, I)$ where $M = V$, and $G$ and $I$ are defined as follows: for every edge $h_i \in \mathcal{E}_{\mathcal{H}}$, create an object whose intent is the complement of $h_i$, i.e., $M \setminus h_i$. Let us denote this with $\overline{h_i}$. Moreover, for each set $f \subsetneq \overline{h_i}$ such that $|f| = |\overline{h_i}| - 1$, create an object with the intent $f$. $\overline{h_i}$ has $|\overline{h_i}|$-many such subsets. That is, for every edge $h_i \in \mathcal{E}_{\mathcal{H}}$ we create $|\overline{h_i}| + 1$ objects, which means that $\mathbb{K}_{\mathcal{H}}$ contains $\sum_{i=1}^{n} |\overline{h_i}| + n$ objects in total. From $\mathcal{G}$ we construct $\mathcal{P}_{\mathcal{G}}$ by simply defining $\mathcal{P}_{\mathcal{G}} = \mathcal{E}_{\mathcal{G}}$. It is easy to see that this construction indeed creates an instance of PIS and the context $\mathbb{K}_{\mathcal{H}}$ as well as the set $\mathcal{P}_{\mathcal{G}}$ can be constructed in time polynomial in the sizes of $\mathcal{H}$ and $\mathcal{G}$. Note that $\mathbb{K}_{\mathcal{H}}$ has the following property: $(*)$ If $B \subseteq M$ is an object intent, then any $A \subseteq B$ is closed since every such $A$ can be expressed as the intersection of some object intents. We claim that $\mathcal{G}$ is the transversal hypergraph of $\mathcal{H}$ if and only if $\mathcal{P}_{\mathcal{G}}$ is precisely the set of pseudo-intents of $\mathbb{K}_{\mathcal{H}}$.

($\Rightarrow$) Assume $\mathcal{G}$ is the transversal hypergraph of $\mathcal{H}$. Take an edge of $\mathcal{G}$, say $g$. $g$ is a minimal transversal of $\mathcal{H}$. By definition, for every $h_i \in \mathcal{E}_{\mathcal{H}}$, $g$ satisfies $g \cap h_i \neq \emptyset$, which is equivalent to $g \not\subseteq \overline{h_i}$. This means that $g$ is not closed in $\mathbb{K}_{\mathcal{H}}$. Because $g$ is not contained in any object intent, hence $g'' = M$. Now take any $f \subsetneq g$. Since $g$ is a *minimal* transversal, $f$ will not be a transversal. That is, for some $h_i \in \mathcal{E}_{\mathcal{H}}$, $f \cap h_i = \emptyset$, which is equivalent to $f \subseteq \overline{h_i}$. Due to Property ($*$), such $f$ are closed. This means that $g$ is not closed in $\mathbb{K}_{\mathcal{H}}$, but its all proper subsets $f$ are closed, which implies that $g$ is a pseudo-intent of $\mathbb{K}_{\mathcal{H}}$. Thus we have shown that if $\mathcal{G}$ is the transversal hypergraph of $\mathcal{H}$, then $\mathcal{P}_{\mathcal{G}}$ is precisely the set of pseudo-intents of $\mathbb{K}_{\mathcal{H}}$.

($\Leftarrow$) Assume $\mathcal{P}_{\mathcal{G}}$ is precisely the set of pseudo-intents of $\mathbb{K}_{\mathcal{H}}$. Take any pseudo-intent $p \in \mathcal{P}_{\mathcal{G}}$. By definition, $p$ is not closed. Due to Property ($*$), $p$ is not contained in any object intent, i.e., $p \not\subseteq \overline{h_i}$, and thus $p'' = M$. This means that $p$ satisfies $p \cap h_i \neq \emptyset$ for every edge $h_i \in \mathcal{E}_{\mathcal{H}}$, i.e., $p$ is a transversal of $\mathcal{H}$. Moreover, $p$ is minimal. Assume it were not. Then there would be another transversal $q \subsetneq p$, and $q$ would satisfy $q \not\subseteq \overline{h_i}$ for every $h_i \in \mathcal{E}_{\mathcal{H}}$ as well. This would mean that $q$ is not closed in $\mathbb{K}_{\mathcal{H}}$ and has the same closure as $p$ which is $M$. This contradicts the fact that $p$ is a pseudo-intent. Thus, $p$ is indeed a minimal transversal of $\mathcal{H}$. We have shown that if $\mathcal{P}_{\mathcal{G}}$ is precisely the set of pseudo-intents of $\mathbb{K}_{\mathcal{H}}$, then $\mathcal{G}$ is the transversal hypergraph of $\mathcal{H}$, which completes the proof of our claim. □

Theorem 2 has the following consequences: For the type of formal contexts used in the reduction, i.e., where subsets of object intents are also closed sets, PIS and TRANS-HYP are computationally equivalent with respect to polynomial transformations, that is PIS is TRANS-HYP-complete. One can take an instance of PIS given with such a formal context and easily reduce it to TRANS-HYP. In this case, enumerating pseudo-intents (PIE) and enumerating hypergraph transversals (TRANS-ENUM) also become computationally equivalent problems. In order to solve an instance of PIE, one can construct the corresponding hypergraph and solve TRANS-ENUM on this hypergraph for instance by using the algorithm in [6] by Fredman and Khachiyan. The minimal transversals of this hypergraph will be the pseudo-intents of the original formal context.

# 5 Finding explanations in the Duquenne-Guigues Base

In the present section, we investigate the problem of *finding explanations* in a Duquenne-Guigues Base, in other words, finding subsets of a given Duquenne-Guigues Base that has a given implication as consequence. In logic, for an arbitrary set of axioms, this problem is known as *axiom pinpointing.* In [1] Baader et. al. have considered this problem for a Description Logic called $\mathcal{EL}^+$ and shown that even for propositional Horn logic a given consequence can have exponentially many minimal explanations, and finding a minimum cardinality explanation is NP-complete.

From a logical point of view, our implications in FCA are also propositional Horn clauses. However, here we consider the above problem when the given set of implications is not an arbitrary set of implications, but it is the Duquenne-Guigues Base of a formal context. Our motivation for considering the problem under this restriction can be explained with the following scenario: Consider a domain expert that explores a context with the attribute exploration method and works with the resulting Duquenne-Guigues Base as a compact representation of the implications holding in her formal context. She notices that from this base, an implication that actually is not true in her application domain follows. This means that during attribute exploration she has wrongly confirmed some implication questions, and needs to find which one of them lead to the erroneous implication. We start with the formal definition of an explanation in a Duquenne-Guigues Base:

**Definition 1.** *Let $\mathcal{L}$ be the Duquenne-Guigues Base of a formal context on the set of attributes $M$, and $P \to Q$ be an implication such that $\mathcal{L} \models P \to Q$. We say that a subset $\mathcal{J} \subseteq \mathcal{L}$ explains $P \to Q$ if $\mathcal{J} \models P \to Q$ is satisfied. In this case we call $\mathcal{J}$ an explanation of $P \to Q$. We say that $\mathcal{J}$ is a minimal explanation of $P \to Q$ if no proper subset of $\mathcal{J}$ explains $P \to Q$.*

In the following for a set of implications $\mathcal{L}$ we will sometimes abuse the terminology and say "the Duquenne-Guigues Base of $Imp(\mathcal{L})$" in order to mean the Duquenne-Guigues Base of the set of all implications that follow from $\mathcal{L}$. The following lemma gives a syntactic characterization of the Duquenne-Guigues Base that will later help us to recognize whether a given set of implications is a Duquenne-Guigues Base.

**Lemma 1.** *Let $\mathcal{L} = \{P_i \to Q_i \mid 1 \leq i \leq n\}$ be a set of implications such that $P_i, Q_i \subseteq M$ and $Q_i \not\subseteq P_i$. $\mathcal{L}$ is the Duquenne-Guigues Base of $Imp(\mathcal{L})$ if and only if for every $1 \leq i \leq n$ the following two conditions are satisfied:*

- *$P_i$ is closed under $\mathcal{L} \setminus \{P_i \to Q_i\}$, and*
- *$P_i \cup Q_i$ is closed under $\mathcal{L} \setminus \{P_i \to Q_i\}$.*

*Proof.* ($\Rightarrow$) If $\mathcal{L}$ is the Duquenne-Guigues Base of $Imp(L)$, then $P_1, \ldots, P_n$ are pseudo-closed sets of the closure system induced by $Imp(\mathcal{L})$. Take any $P_i$ . By definition $P_i$ contains the closure of all $P_j$ such that $P_j \subsetneq P_i$. Thus $P_i$ is closed under $\mathcal{L} \setminus \{P_i \to Q_i\}$. By the definition of Duquenne-Guigues Base, $P_i \cup Q_i$ is also closed under $\mathcal{L} \setminus \{P_i \to Q_i\}$.

($\Leftarrow$) Assume $\mathcal{L}$ is a set of implications that satisfies the two conditions. We claim that it is the Duquenne-Guigues Base of $Imp(\mathcal{L})$. In order to prove this we need to show:

*i*) $P_i$ are the pseudo-closed sets of the closure system induced by $Imp(\mathcal{L})$, where $1 \leq i \leq n$, and
*ii*) for every $1 \leq i \leq n$, $(Imp(\mathcal{L}))(P_i) = P_i \cup Q_i$ holds.

Since for any $X \subseteq M$, $\mathcal{L}(X) = (Imp(\mathcal{L}))(X)$ holds, we are going to show these for $\mathcal{L}$.

We start with *ii*): Take any $P_i \to Q_i$ and let $\mathcal{L}' = \mathcal{L} \setminus \{P_i \to Q_i\}$. We know that $P_i \cup Q_i$ is closed under $\mathcal{L}'$, i.e., $\mathcal{L}'(P_i \cup Q_i) = P_i \cup Q_i$. Then $P_i \cup Q_i$ is also closed under $\mathcal{L}$, i.e., $\mathcal{L}(P_i \cup Q_i) = P_i \cup Q_i$. Obviously $\mathcal{L}(P_i) = \mathcal{L}(P_i \cup Q_i)$ holds for every $P_i \to Q_i \in \mathcal{L}$. Then $\mathcal{L}(P_i) = P_i \cup Q_i$ holds. Thus we have shown *ii*.

In order to show *i* we need to show that:

1. $P_i$ is not closed, i.e., $P_i \neq \mathcal{L}(P_i)$,
2. $P_i$ is quasi-closed, i.e., for every $R \subseteq P_i$, $\mathcal{L}(R) \subseteq P_i$ holds or $\mathcal{L}(R) = \mathcal{L}(P_i)$ holds,
3. $P_i$ strictly contains the closure of every strictly contained quasi-closed set, i.e., for every quasi-closed set $R \subsetneq P_i$, $\mathcal{L}(R) \subsetneq P_i$ holds.

We start with 1: We are given that for every $1 \leq i \leq n$, $Q_i \nsubseteq P_i$. Then $P_i \neq \mathcal{L}(P_i)$ holds trivially. For showing 2, take any $P_i$ and some $R \subseteq P_i$. Let $\mathcal{L}' = \mathcal{L} \setminus \{P_i \to Q_i\}$. Since implicational closure is monotone, $\mathcal{L}'(R) \subseteq \mathcal{L}'(P_i)$. We are given that $P_i$ is closed under $\mathcal{L}'$, i.e., $\mathcal{L}'(P_i) = P_i$ hence $\mathcal{L}'(R) \subseteq P_i$. If $\mathcal{L}'(R) = P_i$, then $\mathcal{L}(R) = \mathcal{L}(P_i)$ and we are done. If $\mathcal{L}'(R) \subsetneq P_i$, then $\mathcal{L}'(R) = \mathcal{L}(R) \subsetneq P_i$ and we are done. Thus we have shown that $P_i$ is quasi-closed.

Now we are going to show 3: Take any $P_i$ and some quasi-closed set $R \subsetneq P_i$. Since implicational closure is extensive, $R \subseteq \mathcal{L}(R)$ holds. If $\mathcal{L}(R) = R$ then $\mathcal{L}(R) \subsetneq P_i$ and we are done. If $R \subsetneq \mathcal{L}(R)$, then there exists an implication $P_j \to Q_j$, where $1 \leq j \leq n$, such that $P_j \subseteq R$ and $Q_j \nsubseteq R$. Together with $R \subsetneq P_i$, this implies $P_j \subsetneq P_i$.

We know that $P_i$ is closed under $\mathcal{L} \setminus \{P_i \to Q_i\}$. Since $P_j \subsetneq P_i$, this implies $Q_j \subseteq P_i$, hence $P_j \cup Q_j \subseteq P_i$. Since $Q_i \nsubseteq P_i$, $Q_i \nsubseteq P_j \cup Q_j$. We know that $P_j \cup Q_j$ is closed under $\mathcal{L} \setminus \{P_j \to Q_j\}$. If $P_j \cup Q_j = P_i$ were satisfied, then $P_j \cup Q_j$ would not be closed under $\mathcal{L} \setminus \{P_j \to Q_j\}$ since $Q_i \nsubseteq P_j \cup Q_j$. Thus, $P_j \cup Q_j \subsetneq P_i$. By using *ii*, we can rewrite it as $\mathcal{L}(P_j) \subsetneq P_i$.

We know that $R$ is quasi-closed. Since $P_j \subseteq R$, $\mathcal{L}(P_j) \subseteq R$ holds or $\mathcal{L}(P_j) = \mathcal{L}(R)$ holds. By *ii* we know that $\mathcal{L}(P_j) = P_j \cup Q_j$. Since $Q_j \nsubseteq R$, $\mathcal{L}(P_j) \subseteq R$ cannot hold. Thus, $\mathcal{L}(P_j) = \mathcal{L}(R)$ holds. Together with $\mathcal{L}(P_j) \subsetneq P_i$ from above, this implies that $\mathcal{L}(R) \subsetneq P_i$. Thus we have shown 3, which completes the proof of *i*, which in turn completes the proof of our claim. □

Using Lemma 1, we can show that in the worst case, a given implication can have exponentially many minimal explanations in a given Duquenne-Guigues Base. The following example demonstrates this situation:

*Example 1.* Consider the set of implications

$$\mathcal{L} := \{\{x, b_{i-1}\} \to \{p_i, q_i\},\ \{y, p_i\} \to \{b_i\},\ \{y, q_i\} \to \{b_i\} \mid 1 \leq i \leq n\}$$

on the set of attributes $M = \{b_0, x, y, \} \cup \{b_i, p_i, q_i \mid 1 \leq i \leq n\}$. Note that none of the left handsides is contained in another left handside or in the union of left and right handsides of another implication, i.e., $\mathcal{L}$ satisfies the two conditions stated in Lemma 1 thus, $\mathcal{L}$ is the Duquenne-Guigues Base of $Imp(\mathcal{L})$.

Consider the implication $\{b_0, x, y\} \to \{b_n\}$ that follows from $\mathcal{L}$. A minimal explanation of this implication looks like either $\{\{b_0, x\} \to \{p_1, q_1\}, \{y, p_1\} \to \{b_1\}, \ldots\}$ or, $\{\{b_0, x\} \to \{p_1, q_1\}, \{y, q_1\} \to \{b_1\}, \ldots\}$. That is at every step $i$ such that $1 \leq i \leq n$ we have two choices since the attribute $b_i$ can be generated either by the implication $\{y, p_i\} \to \{b_i\}$, or by the implication $\{y, q_i\} \to \{b_i\}$. This means that there are $2^n$ minimal explanations. Since the size of $\mathcal{L}$ is linear in $n$, the example shows that there can be exponentially many minimal explanations in a given Duquenne-Guigues Base.

### 5.1 Minimum cardinality explanation

Although there can be exponentially minimal explanations, given a Duquenne-Guigues Base $\mathcal{L}$ and an implication $\psi$ that follows from it, it is not difficult to find one minimal explanation of $\psi$ in $\mathcal{L}$. We can just start with $\mathcal{L}$, iterate over the implications in $\mathcal{L}$ and remove an implication if $\psi$ still follows from the remaining set of implications. Clearly, this algorithm terminates since $\mathcal{L}$ is finite. It is correct since $\psi$ still follows from the remaining set of implications and none of the implications in the remaining set can be removed without destroying this property.

However, if we want an explanation that is not only minimal w.r.t. set inclusion, but also minimal w.r.t. cardinality, the problem becomes harder. In [1], Baader et. al. have shown that for an arbitrary set of implications (there called propositional Horn axioms) finding an explanation within a specified cardinality bound is NP-complete. Here we consider this problem for the case when the given set of implications is not arbitrary, but it is the Duquenne-Guigues Base of implications holding in a closure system. It turns out that under this restriction the problem does not become easier, i.e., it remains NP-complete.

**Problem:** MINIMUM CARDINALITY EXPLANATION (MCE)
*Input:* A Duquenne-Guigues Base $\mathcal{L}$, an implication $L \to R$ s.t. $\mathcal{L} \models L \to R$ and a natural number $n$.
*Question:* Is there an explanation of $L \to R$ in $\mathcal{L}$ with cardinality less than or equal to $n$, i.e., is there an $\mathcal{L}' \subseteq \mathcal{L}$ such that $\mathcal{L}' \models L \to R$ and $|\mathcal{L}'| \leq n$?

**Theorem 3.** MCE *is* NP*-complete.*

*Proof.* The problem is in NP. We can nondeterministically guess a subset $\mathcal{L}'$ of $\mathcal{L}$ with cardinality $n$, and in polynomial time check whether $L' \models L \to R$. This test can indeed be done in polynomial time by checking whether $R \subseteq \mathcal{L}'(L)$.

In order to show NP-hardness, we are going to give a reduction from the NP-complete problem VERTEX COVER [9]. Recall that a vertex cover of the graph $\mathcal{G} = (V, E)$ is a set $W \subseteq V$ such that for every edge $\{u, v\} \in E$, $u \in W$ holds, or $v \in W$ holds. The problem VERTEX COVER is defined as follows:

**Problem:** VERTEX COVER
*Input:* Graph $\mathcal{G} = (V, E)$, a natural number $n$.
*Question:* Is there a vertex cover of $\mathcal{G}$ of size less than or equal to $n$?

Consider an instance of the VERTEX COVER problem given by $\mathcal{G} = (V, E)$, where $V = \{v_1, \ldots, v_l\}$, $E = \{e_1, \ldots, e_k\}$, and edge $e_i = \{v_{i1}, v_{i2}\}$. We construct an instance of the MCE in the following way: For every vertex $v \in V$ we introduce an attribute $m_v$, for every edge $e_j$, $1 \leq j \leq k$, we introduce an attribute $m_{e_j}$, and finally two more additional attributes $m_a$ and $m_b$. Using these attributes we construct the following set of implications:

$$\begin{aligned}\mathcal{L} := \{&\{m_v\} \to \{m_{e_j} \mid v \in e_j, 1 \leq j \leq k\} \mid v \in V\} \cup \\ &\{\{m_a, m_{e_1}, \ldots, m_{e_k}\} \to \{m_b\}\}.\end{aligned}$$

Note that none of the implications in $\mathcal{L}$ contains the left handside of another implication in its left handside or in the union of its left and right handsides. Thus, due to Lemma 1, $\mathcal{L}$ is indeed the Duquenne-Guigues Base of $Imp(\mathcal{L})$. In addition to $\mathcal{L}$, we construct the following implication $\psi$ that follows from $\mathcal{L}$: $\psi : \ \{m_a\} \cup \{m_v \mid v \in V\} \to \{m_b\}$. It is not difficult to see that both $\mathcal{L}$ and $\psi$ can be constructed in time polynomial in the size of $\mathcal{G}$, and that $\psi$ follows from $\mathcal{L}$. We claim that $\mathcal{G}$ has a vertex cover of size less than or equal to $n$, where $n \leq |V|$, if and only if $\mathcal{L}$ has a subset $\mathcal{L}'$ that explains $\psi$, and the size of $\mathcal{L}'$ is polynomial in $n$.

($\Rightarrow$) Assume $W \subseteq V$ is a vertex cover of $\mathcal{G}$. Then the following set $\mathcal{L}' \subseteq \mathcal{L}$ constructed by using $W$ is an explanation of $\psi$:

$$\begin{aligned}\mathcal{L}' := \{&\{m_w\} \to \{m_{e_j} \mid w \in e_j, 1 \leq j \leq k\} \mid w \in W\} \cup \\ &\{\{m_a, m_{e_1}, \ldots, m_{e_k}\} \to \{m_b\}\}.\end{aligned}$$

Since $W$ is a vertex cover, it contains at least one vertex from every edge $e_j$, $1 \leq j \leq k$. Thus, $\{m_{e_1}, \dots m_{e_k}\} \subseteq \mathcal{L}'(\{m_w \mid w \in W\})$. Since $\{m_w \mid w \in W\} \subseteq \{m_v \mid v \in V\}$, this implies that $\{m_{e_1}, \dots m_{e_k}\} \subseteq \mathcal{L}'(\{m_v \mid v \in V\})$, which in turn implies that $\{m_b\} \subseteq \mathcal{L}'(\{m_a\} \cup \{m_v \mid v \in V\})$. Thus we have shown that $\mathcal{L}'$ is indeed an explanation of $\psi$, and that it contains exactly $n+1$ implications.

($\Leftarrow$) Now assume that $\mathcal{L}$ has a subset $\mathcal{L}'$ of size $m$ that is an explanation of $\psi$. $\mathcal{L}'$ should contain the implication $\{m_a, m_{e_1}, \dots, m_{e_k}\} \to \{m_b\}$, since otherwise the attribute $m_b$ cannot be generated. Moreover, since the premise of this implication contains the attributes $m_{e_1}, \dots, m_{e_k}$, $\mathcal{L}'$ should also contain implications of type $\{m_w\} \to \{m_{e_j} \mid w \in e_j\}$ such that every $m_{e_j}$, $1 \leq j \leq k$, is generated. This means that the set $W$ of such $w$ is indeed a vertex cover since it intersects every edge $e_j$, $1 \leq j \leq k$. Thus we have shown that $W$ is a vertex cover of $\mathcal{G}$ and it has size $m-1$. This finishes the proof of the claim that $\mathcal{G}$ has a vertex cover of size $n$ if and only if $\mathcal{L}$ has a subset of size $n+1$ that explains $\psi$. □

### 5.2 Counting minimal explanations

In applications where one is interested in all explanations that are minimal w.r.t. set inclusion, it might be useful to know in advance how many of them exist. Next we consider this counting problem. It turns out that it is hard for the counting complexity class #P [19], i.e., it is intractable.

**Problem:** #MINIMAL EXPLANATION (#ME)
*Input:* A Duquenne-Guigues Base $\mathcal{L}$, and an implication $L \to R$ s.t. $\mathcal{L} \models L \to R$.
*Output:* The number of all minimal explanations of $L \to R$, i.e., $|\{\mathcal{L}' \subseteq \mathcal{L} \mid \mathcal{L}' \models L \to R \textit{ and } \forall \mathcal{L}'' \subsetneq \mathcal{L}'.\mathcal{L}'' \not\models L \to R\}|$.

**Theorem 4.** #ME *is #P-complete.*

*Proof.* The problem is in #P. Given a Duquenne-Guigues Base $\mathcal{L}$, an implication $L \to R$ that follows from $\mathcal{L}$, and a set $\mathcal{L}' \subseteq \mathcal{L}$ we can in polynomial time verify whether $\mathcal{L}' \models L \to R$ just by checking whether $R \subseteq \mathcal{L}'(L)$ holds.

In order to show #P-hardness, we are going to give a parsimonious reduction from the #P-complete problem #MINIMAL VERTEX

COVER, which is the problem of counting the minimal vertex covers of a graph. It has been shown to be #P-complete in [20]. In our reduction we are going to use the same construction as in the proof of Theorem 3, i.e., from a given graph $\mathcal{G}$ we construct the same Duquenne-Guigues Base $\mathcal{L}$, and the same implication $\psi$ as in Theorem 3. What we additionally need to show here is that this construction establishes a bijection between minimal vertex covers of $\mathcal{G}$ and minimal explanations of $\psi$ in $\mathcal{L}$.

First we show that the construction in the proof of Theorem 3 establishes an injection: assume $W \subseteq V$ is a *minimal* vertex cover of $\mathcal{G}$, then the following set of implications is a *minimal* explanation of $\psi$ in $\mathcal{L}$:

$$\begin{aligned}\mathcal{L}' := \{&\{m_w\} \to \{m_{e_j} \mid w \in e_j, 1 \leq j \leq k\} \mid w \in W\} \cup \\ &\{\{m_a, m_{e_1}, \ldots, m_{e_k}\} \to \{m_b\}\}.\end{aligned}$$

In the proof of Theorem 3 we have already shown that $\mathcal{L}'$ is an explanation. Here we need to show that it is minimal as well. If $W$ is minimal, then removal of any vertex $w$ from $W$ will result in a $Y \subsetneq W$ such that $v_{j1} \notin Y$ and $v_{j2} \notin Y$ for some edge $e_j$. This implies that removal of the corresponding implication $\{m_w\} \to \{m_{e_j} \mid w \in e_j\}$ from $\mathcal{L}'$ will result in a $\mathcal{L}''$ such that the attribute $m_{e_j}$ does not appear on the right handside of any of the implications in $\mathcal{L}''$, which means that $\mathcal{L}''$ cannot explain $\psi$, i.e., $\mathcal{L}'$ is minimal.

Now we show that it establishes a surjection: assume $\mathcal{L}'$ is a minimal explanation. Then every $m_{e_j}$, $1 \leq j \leq k$, occurs at least once on the right handside of some implication of the form $\{m_w\} \to \{m_{e_j} \mid w \in e_j\}$, where $w \in W$, because otherwise $\mathcal{L}'$ cannot explain $\psi$. We have already shown in the proof of Theorem Theorem 3 that such a $W$ is a vertex cover. Moreover, removal of any implication of this form from $\mathcal{L}'$ results in a set of implications that is not an explanation. This is because $\mathcal{L}'$ is a minimal explanation. That is, removal of any $w$ from $W$ results in a $Y \subsetneq W$ such that $v_{j1} \notin Y$ and $v_{j2} \notin Y$ for some $1 \leq j \leq k$, i.e., $W$ is minimal. Thus we have shown that our construction establishes a bijection between minimal vertex covers and minimal explanations. □

### 5.3 Computing all minimal explanations

In Example 1 we have demonstrated that a given implication can have exponentially many minimal explanations in a given Duquenne-Guigues Base. Given this fact, it is clearly not possible to enumerate all minimal explanations in time polynomial in the size of the input. In this case one can investigate the existence of an output polynomial algorithm for this problem:

**Problem:** MINIMAL EXPLANATION ENUMERATION (MEE)
*Input:* A Duquenne-Guigues Base $\mathcal{L}$ and an implication $L \to R$ s.t. $\mathcal{L} \models L \to R$.
*Output:* The set of all minimal explanations of $L \to R$ in $\mathcal{L}$, i.e., $\{\mathcal{L}' \subseteq \mathcal{L} \mid \mathcal{L}' \models L \to R \textit{ and } \forall \mathcal{L}'' \subsetneq \mathcal{L}'.\ \mathcal{L}'' \not\models L \to R\}$.

In order to investigate the complexity of this enumeration problem, we need to investigate the following decision problem:

**Problem:** ADDITIONAL MINIMAL EXPLANATION (AME)
*Input:* A Duquenne-Guigues Base $\mathcal{L}$, an implication $L \to R$ s.t. $\mathcal{L} \models L \to R$, and a set of minimal explanations of $L \to R$ in $\mathcal{L}$, i.e, $\mathscr{J} = \{\mathcal{J}_i \mid \mathcal{J}_i \subseteq \mathcal{L}, \mathcal{J}_i \models L \to R \textit{ and } \forall \mathcal{J}' \subsetneq \mathcal{J}_i.\ \mathcal{J}' \not\models L \to R\}$
*Question:* Is there a minimal explanation that is not already listed in $\mathscr{J}$, i.e., $\mathcal{J} \subseteq \mathcal{L}$ such that $\mathcal{J} \models L \to R$, $\forall \mathcal{J}' \subsetneq \mathcal{J}.\ \mathcal{J}' \not\models L \to R$ and $\mathcal{J} \notin \mathscr{J}$?

Because if AME is not in P, there cannot be an algorithm that solves MEE in output polynomial time (unless P = NP). We can show it by the same argument used in the proofs of Propositions 1 and 3.

**Proposition 5.** *If* AME *cannot be decided in polynomial time, then* MEE *cannot be solved in output-polynomial time unless* P = NP.

*Proof.* Assume there exists an algorithm that solves MEE in output polynomial time. In order to solve an instance of AME given with the input $(\mathcal{L}, L \to R, \mathscr{J})$, we run this algorithm on the instance of MEE given with the input $(\mathcal{L}, L \to R)$ for at most $|\mathscr{J}|$ steps. If the algorithm terminates before $|\mathscr{J}|$ steps then we check whether the output of the algorithm is equal to $\mathscr{J}$. If the algorithm has not terminated after $|\mathscr{J}|$ steps, $\mathscr{J}$ is not the set of all minimal explanations. Since this algorithm runs in time polynomial in the size of the set of all minimal explanations, we can decide AME in polynomial time. □

It is not difficult to see that AME is in coNP. Given an instance of AME with the Duquenne-Guigues Base $\mathcal{L}$, the implication $\psi$ and a set of minimal explanations $\mathscr{J}$, we can nondeterministically guess a minimal subset of $\mathcal{L}$ that is not already contained in $\mathscr{J}$ and in polynomial time verify that this subset *does not* explain $\psi$. Unfortunately we do not know the lower bound of this problem at the moment. It is definitely an interesting question whether this problem, like API and PIS, is also related to the decision problems SIMPLE-H-SAT and TRANS-HYP from hypergraph theory.

## 6 Concluding remarks and future work

We have considered several decision, enumeration and counting problems related to pseudo-intents. Among them, PIE, the problem of enumerating pseudo-intents has been the central point of our interest. The question whether this problem can be solved in output polynomial time or not remains unfortunately open. However we have formulated two decision questions, namely API and PIS, that are crucial in determining the complexity of PIE. Some interesting consequences of our results can be summed up as follows:

- If any of the problems API, or PIS turns out to be coNP-hard, then unless P = NP, there cannot be an algorithm that solves PIE in output polynomial time (Proposition 1, Proposition 3).
- Showing that any of the problems API or PIS is polynomial implies that the open problems TRANS-HYP and SIMPLE-H-SAT are also polynomial (Theorem 1, Theorem 2, [3]).
- Even if TRANS-HYP and SIMPLE-H-SAT turn out to be polynomial, API and PIS can still be coNP-hard, thus it can still be the case that PIE is not solvable in output polynomial time.
- Even if API and PIS turn out to be polynomial, it can still be the case that PIE is not solvable in output polynomial time.

Moreover, if checking whether a given set if a pseudo-intent of a given formal context turns out to be coNP-hard, then PIE cannot be solved in output polynomial time, since solutions that have inherent complexity cannot be enumerated in output polynomial time unless P = NP [12]. We have also investigated the complexity of finding explanations, i.e., subsets from which a given implication follows, in a given Duquenne-Guigues Base. We have shown that finding a minimum cardinality one is NP-complete, and counting minimal explanations is #P-complete.

As future work, we are going to work on determining the exact complexity of the problems API and PIS. For API, we are going to investigate whether the hardness result [3] on hypergraph saturation for arbitrary graphs carries over to API on arbitrary formal contexts. For PIS, we are going to investigate the types of formal context where PIS and TRANS-HYP (and thus PIE and TRANS-ENUM) become computationally equivalent problems, and find out whether this type of formal contexts are natural in some applications, and how often they occur in practice. One other interesting question is of course the lower complexity bound for checking whether a set is pseudo-intent. We are going to investigate whether this problem is also related to some hypergraph problem. In addition to this, we are going to work on determining the exact complexity of counting pseudo-intents.Note that in [14, 15] it has been mentioned that this problem is in #P, but this is not true. The results there only imply that this problem is in #·coNP, which contains #P. On the explanations side, we are going to work on determining the exact complexity of AME.

## References

1. F. Baader, R. Peñaloza, and B. Suntisrivaraporn. Pinpointing in the description logic $\mathcal{EL}^+$. In J. Hertzberg, M. Beetz, and R. Englert, editors, *Proceedings of the 30th German Conference on Artificial Intelligence (KI2007)*, volume 4667 of *Lecture Notes in Artificial Intelligence*, pages 52–67. Springer-Verlag, 2007.
2. C. Berge. *Hypergraphs*. Elsevier Science Publishers B.V. (North Holland), 1989.
3. T. Eiter and G. Gottlob. Identifying the minimal transversals of a hypergraph and related problems. *SIAM Journal on Computing*, 24(6):1278–1304, 1995.
4. T. Eiter and G. Gottlob. Hypergraph transversal computation and related problems in logic and ai. In S. Flesca, S. Greco, N. Leone, and G. Ianni, editors, *Proceedings of the European Conference on Logics in Artificial Intelligence (JELIA 2002)*, volume 2424 of *Lecture Notes in Computer Science*, pages 549–564. Springer-Verlag, 2002.
5. T. Eiter, G. Gottlob, and K. Makino. New results on monotone dualization and generating hypergraph transversals. *SIAM Journal on Computing*, 32(2):514–537, 2003.

6. M. L. Fredman and L. Khachiyan. On the complexity of dualization of monotone disjunctive normal forms. *Journal of Algorithms*, 21(3):618–628, 1996.
7. B. Ganter. Two basic algorithms in concept analysis. Technical Report Preprint-Nr. 831, Technische Hochschule Darmstadt, Darmstadt, Germany, 1984.
8. B. Ganter and R. Wille. *Formal Concept Analysis: Mathematical Foundations.* Springer-Verlag, Berlin, Germany, 1999.
9. M. R. Garey and D. S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness.* W. H. Freeman & Company, New York, NY, USA, 1990.
10. J.-L. Guigues and V. Duquenne. Familles minimales d'implications informatives resultant d'un tableau de données binaries. *Mathématiques, Informatique et Sciences Humaines*, 95:5–18, 1986.
11. D. Gunopulos, R. Khardon, H. Mannila, and H. Toivonen. Data mining, hypergraph transversals, and machine learning. In *Proceedings of the Sixteenth Symposium on Principles of Database Systems, (PODS 97)*, pages 209–216, 1997.
12. D. S. Johnson, C. H. Papadimitriou, and M. Yannakakis. On generating all maximal independent sets. *Information Processing Letters*, 27(3):119–123, 1988.
13. S. O. Kuznetsov. On the intractability of computing the Duquenne-Guigues Base. *Journal of Universal Computer Science*, 10(8):927–933, 2004.
14. S. O. Kuznetsov and S. A. Obiedkov. Counting pseudo-intents and #P-completeness. In R. Missaoui and J. Schmid, editors, *Proceedings of the 4th International Conference on Formal Concept Analysis (ICFCA 2006)*, volume 3874 of *Lecture Notes in Computer Science*, pages 306–308, Dresden, Germany, 2006. Springer-Verlag.
15. S. O. Kuznetsov and S. A. Obiedkov. Some decision and counting problems of the duquenne-guigues basis of implications. *Discrete Applied Mathematics*, 156(11):1994–2003, 2008.
16. H. Mannila and K.-J. Räihä. Design by example: An application of armstrong relations. *Journal of Computer and System Sciences*, 33(2):126–141, 1986.
17. S. A. Obiedkov and V. Duquenne. Attribute-incremental construction of the canonical implication basis. *Annals of Mathematics and Artificial Intelligence*, 49(1-4):77–99, 2007.
18. S. Rudolph. Some notes on pseudo-closed sets. In S. O. Kuznetsov and S. Schmidt, editors, *Proceedings of the 5th International Conference on Formal Concept Analysis, (ICFCA 2007)*, volume 4390 of *Lecture Notes in Computer Science*, pages 151–165. Springer-Verlag, 2007.
19. L. G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8(2):189–201, 1979.
20. L. G. Valiant. The complexity of enumeration and reliability problems. *SIAM Journal on Computing*, 8(3):410–421, 1979.