# Pseudo-Closed Family Verification is NP-Complete (Or: How Claude Helped Tackle Bernhard's Problem)

Sebastian Rudolph[⋆]

TU Dresden and ScaDS.AI Dresden/Leipzig
sebastian.rudolph@tu-dresden.de

**Abstract.** Every closure operator on a finite set admits a canonical minimum implication basis – the *stem base* or *Duquenne–Guigues base* – whose premises are the *pseudo-closed sets*. The *Pseudo-Closed Family* problem (PCF) asks whether a given family of sets equals the pseudo-closed sets of some closure operator. We prove that PCF is NP-complete, negatively answering a question posed by Bernhard Ganter at CONCEPTS'25, which essentially asked for "easy" ways to verify PCF.
As the large language model Claude Opus 4.6 was instrumental in establishing the NP-hardness part, this paper also discusses the methodology applied and observations made in the process.

## 1 Introduction

During the CONCEPTS'25 conference, Bernhard Ganter posed a problem to the audience. The problem revolved around one of the arguably most mysterious and misunderstood notions of FCA: a closure operator's *pseudo-closed sets*. Given that pseudo-closed sets are a key notion when it comes to describing and understanding the attribute logic of formal contexts, they have been a subject of intense research in FCA, in particular regarding their representational succinctness and computational properties [13,9,10,15,11,20,19,3,1,16].

The question posed was strikingly simple: Given a family of subsets of some finite set $S$, how can we find out if it corresponds to the pseudo-closed sets of some closure operator on $S$? While the problem is clearly decidable by brute-force methods (assuming a finite base set), it turned out surprisingly tricky to come up with necessary and sufficient conditions that can be easily checked. Ganter provided two necessary conditions (see Fig. 1), but the conjecture that they are also sufficient was refuted pretty much on the spot, by another attendee who provided a counterexample. Consequently, the matter remained unresolved.

In the aftermath of the conference, the author spent some time thinking about the problem from a complexity-theoretic perspective. Clearly, the existence of an easily verifiable characterization would require that the underlying

[⋆] It is firm consensus among academic publishers and ethics organizations that large language models must not be listed as co-authors. If that weren't so, the author would have considered this – which would have raised the question of Claude's surname.

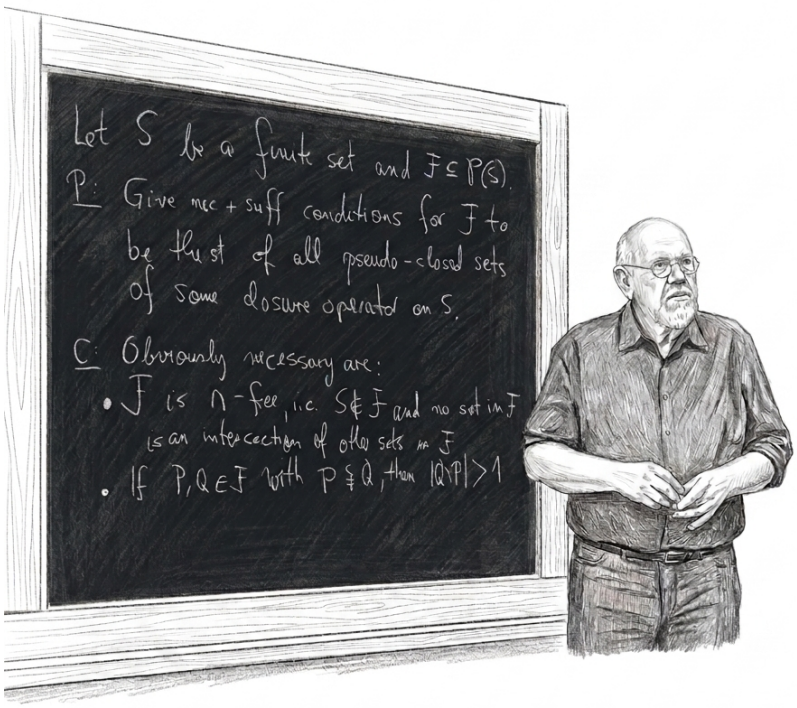


**Fig. 1.** Impression of Bernhard Ganter presenting the problem discussed in this paper on 12 September 2025 during CONCEPTS'25 in Cluj-Napoca, Romania.

computational decision problem is tractable. Some tractable computational tools for related problems already exist, most notably closure computation [4] and "stembasification" [15] of an implication set. Yet, even with these at hand, the best the author could do was to show that the problem is in NP. Attempts to complement this upper complexity bound with a lower one by showing NP-hardness failed, putting the problem in tractability limbo between P and NP.

In view of the usual shortage of time available for uninterrupted research and made curious by the latest news about large language models solving non-trivial mathematical problems,[1] the author decided to try and solicit the help of a large language model (Claude Opus 4.6 by Anthropic) to tackle the problem. The ensuing exchange spanning a few days turned out to be an exciting journey with new insights into the opportunities and challenges of human-LLM collaboration in the mathematical domain. In the end it was possible with joint forces to settle the complexity problem by showing NP-hardness, which amounts to the insight that no easily (in the sense of tractably) verifiable conditions exist that would be both necessary and sufficient.

The first part of the paper provides a full formal treatment of the established results, the second reflects on the experience of the unusual collaborative process.

[1] cf. `https://www-cs-faculty.stanford.edu/~knuth/papers/claude-cycles.pdf`

## 2 Preliminaries

In this section, we set the stage by introducing the terminology relevant to our treatise. Notably, formal contexts and formal concepts – the foundational notions in Formal Concept Analysis – are not required at all, as the topic primarily revolves around closure operators.

**Definition 1 (closure operator, closed set)** *A* closure operator *on a finite set $S$ is a function $\varphi\colon 2^S \to 2^S$ satisfying, for all $A, B \subseteq S$:*

- $A \subseteq \varphi(A)$ *(extensivity),*
- $A \subseteq B \implies \varphi(A) \subseteq \varphi(B)$ *(monotonicity),*
- $\varphi(\varphi(A)) = \varphi(A)$ *(idempotency).*

*A set $C \subseteq S$ with $\varphi(C) = C$ is called* closed.

While closure operators are often defined over arbitrary sets, in our investigation, we require $S$ to be a finite set. This condition is justified by the fact that any complexity-theoretic consideration requires finite problem inputs; it also avoids some complications related to the recursive definitions of pseudo-closed sets.

The following straightforward consequence of monotonicity will be used repeatedly: If $C$ is closed and $A \subseteq C$, then $\varphi(A) \subseteq C$.

Now we introduce implications as a key means of expressing closure operators.

**Definition 2 (implication, closure under an implication set)** *An* implication *on $S$ is a pair $A \to B$ with $A, B \subseteq S$. A set $C \subseteq S$* respects *$A \to B$ if $A \subseteq C$ implies $B \subseteq C$.*

*Given a finite set $\mathfrak{I}$ of implications on $S$, the* $\mathfrak{I}$-closure *of $X \subseteq S$, denoted $X^{\mathfrak{I}}$, is the smallest superset of $X$ respecting every implication in $\mathfrak{I}$.*

The term "$\mathfrak{I}$-closure" is justified by the fact that the map $X \mapsto X^{\mathfrak{I}}$ is a closure operator. The $\mathfrak{I}$-closure can be computed iteratively: Starting from $C = X$, whenever some $A \to B \in \mathfrak{I}$ satisfies $A \subseteq C$ whereas $B \not\subseteq C$, we replace $C$ by $C \cup B$, repeating this process until $C$ stabilizes. The time required for this naive approach is quadratic in the size of $\mathfrak{I}$, but with more elaborate algorithms a linear runtime can be achieved [4]. We now introduce the central notion of this paper. The definition seems to be notoriously hard to comprehend for students (and even some established researchers), due to its alleged circular nature.

**Definition 3 (pseudo-closed set)** *A set $P \subseteq S$ is* pseudo-closed *for a closure operator $\varphi$ on $S$ if:*

*(i) $\varphi(P) \neq P$ (i.e., $P$ is not closed), and*
*(ii) for every pseudo-closed $Q \subsetneq P$, we have $\varphi(Q) \subseteq P$.*

*This recursive definition is well-founded by strict inclusion.*

One rather straightforward observation about pseudo-closed sets deserves to be made explicit, as it will be instrumental later on.

**Fact 4 (sandwich property)** *If $\varphi$ is a closure operator with pseudo-closed sets $Q$ and $P$ such that $Q \subsetneq P$, then $Q \subsetneq \varphi(Q) \subsetneq P$.*

Note the sandwich property also immediately gives rise to the second necessary condition observed by Ganter (see Fig. 1): for any two pseudo-closed sets $P$ and $Q$ with $Q \subsetneq P$ we must have $|P \setminus Q| > 1$.

The first necessary condition of Ganter, essentially stating that the intersection of two inclusion-incomparable pseudo-closed sets must never be pseudo-closed, is justified by another easy observation presented next.

**Fact 5 (intersection-free)** *If $\varphi$ is a closure operator with pseudo-closed sets $P_1$ and $P_2$ such that $P_1 \not\subseteq P_2$ and $P_2 \not\subseteq P_1$, then $P_1 \cap P_2$ is not pseudo-closed.*

The fact directly generalizes to intersections of more than two pseudo-closed sets. We proceed by providing another central notion, which in fact is the main reason for the existence of the notion of pseudo-closed sets: it warrants the definition of a canonical, minimal way of representing closure operators by means of implications.

**Definition 6 (canonical base[2])** *Let $\varphi$ be a closure operator. The* canonical base $\mathfrak{S}_\varphi$ *of $\varphi$ is the implication set $\{P \to \varphi(P) \mid P \text{ pseudo-closed}\}$.*

**Theorem 7 (Guigues & Duquenne [6,5]).** *Given a closure operator $\varphi$, the canonical base $\mathfrak{S}_\varphi$ is a basis for $\varphi$, that is, it satisfies $(\cdot)^{\mathfrak{S}_\varphi} = \varphi$ and none of the implications of $\mathfrak{S}_\varphi$ follows from the others. Among all such bases, $\mathfrak{S}_\varphi$ has minimum cardinality.*

The Stembase algorithm [2,15] converts any finite set $\mathfrak{I}$ of implications on $S$ into the canonical base of the closure operator $(\cdot)^{\mathfrak{I}}$ in polynomial time.

**Algorithm** Stembase($\mathfrak{I}$)
**Input:** A finite set $\mathfrak{I}$ of implications on $S$.
**Output:** The canonical base $\mathfrak{S}$ of $(\cdot)^{\mathfrak{I}}$.
1. Set $\mathfrak{S} := \emptyset$.
2. For every $A \to B \in \mathfrak{I}$: substitute $A \to B$ by $A \to (A \cup B)^{\mathfrak{I}}$.
3. As long as $\mathfrak{I} \neq \emptyset$:
   (a) select an $A \to B$ from $\mathfrak{I}$,
   (b) delete $A \to B$ from $\mathfrak{I}$,
   (c) if $A^{\mathfrak{I} \cup \mathfrak{S}} \neq B$ then add $A^{\mathfrak{I} \cup \mathfrak{S}} \to B$ to $\mathfrak{S}$.
4. Output $\mathfrak{S}$ and terminate.

**Theorem 8 ([15, Theorem 2]).** *The algorithm Stembase computes a canonical base for $(\cdot)^{\mathfrak{I}}$. Specifically:*
*(a) $(\cdot)^{\mathfrak{I}} = (\cdot)^{\mathfrak{S}}$ (the closure operator is preserved), and*
*(b) the premises of $\mathfrak{S}$'s implications are exactly the pseudo-closed sets for $(\cdot)^{\mathfrak{S}}$.*

As computing the $\mathfrak{I}$-closure can be done in linear time [4], the algorithm runs in time quadratic in the size of $\mathfrak{I}$ (the naive implicational closure algorithm described above would yield cubic runtime, which is still polynomial). Crucially, the premises of $\mathfrak{S}$ thus computed are exactly the pseudo-closed sets of $(\cdot)^{\mathfrak{I}}$.

[2] In the literature, the canonical base is also referred to as stem base or Duquenne–Guigues base. We follow Vincent Duquenne's wish to discontinue both terms in favor of *canonical base*, except for the already established name Stembase *algorithm*.

## 3 The Pseudo-Closed Family Problem

It is a worthwhile question how the definition of pseudo-closedness constrains the way in which pseudo-closed sets can co-exist. In particular, is it possible to capture such co-existence constraints in an exact way, so that for each family of subsets of some set $S$ it can be straightforwardly verified if it corresponds to the pseudo-closed sets of some closure operator? These contemplations give rise to the computational problem at the core of this paper, which we now introduce.

**Definition 9 (PCF)** *An instance of the* Pseudo-Closed Family *problem (PCF) consists of a finite set $S$ and a family $\mathcal{F} \subseteq 2^S$ of subsets of $S$. The question is whether there exists a closure operator on $S$ whose pseudo-closed sets are exactly $\mathcal{F}$. If such a closure operator exists, we say $(S, \mathcal{F})$ is* valid. *If $S$ is clear from the context, we may sometimes simply say $\mathcal{F}$ is* valid.

*Example 1.* Let us first consider the instance of PCF with $S_1 = \{a, b, c, d\}$ and $\mathcal{F}_1 = \{\{a\}, \{b\}, \{c\}, \{a, b, c\}\}$, noting that the instance satisfies both of Ganter's necessary conditions: $\mathcal{F}_1$ is intersection-free and any two comparable sets differ in more than one element. Still, $(S_1, \mathcal{F}_1)$ is not valid: Any fitting closure operator $\varphi_1$ would need to map each singleton element of $\mathcal{F}_1$ to a two-element subset of $\{a, b, c\}$. W.l.o.g. (exploiting symmetry), fix $\varphi_1(\{a\}) = \{a, b\}$, which forces $\varphi_1(\{b\}) = \{a, b\}$ by monotonicity and idempotency, because $\{b\} \subseteq \{a, b\} = \varphi_1(\{a, b\})$. But then, there is no appropriate choice left for $\varphi_1(\{c\})$: Picking $\varphi_1(\{c\}) = \{a, c\}$ would mean $\{a\} \subseteq \varphi_1(\{c\})$ which, by monotonicity and idempotency yields $\varphi_1(\{a\}) \subseteq \varphi_1(\varphi_1(\{c\})) = \varphi_1(\{c\})$, which contradicts the earlier choice of $\varphi_1(\{a\})$. Picking $\varphi_1(\{c\}) = \{b, c\}$ would produce the same issue. ◊

*Example 2.* To contrast the previous example, consider the PCF instance with $S_2 = \{a, b, c, d, e\}$ and $\mathcal{F}_2 = \{\{a\}, \{b\}, \{c\}, \{d\}, \{a, b, c, d\}\}$. We will argue that $(S_2, \mathcal{F}_2)$ is valid. In order to do so, we have to describe a corresponding closure operator $\varphi_2$. We choose to describe $\varphi_2$ by means of an implication set $\mathfrak{I}_2$, letting $\varphi_2 = (\cdot)^{\mathfrak{I}_2}$. Let now $\mathfrak{I}_2 = \{a \to ab,\ b \to ab,\ c \to cd,\ d \to cd,\ abcd \to abcde\}$ (using simplified notation for better readability). It can be verified, using the Stembase algorithm, that $\mathfrak{I}_2$ is already a canonical base and thus its premises are the pseudo-closed sets, which coincide with the elements of $\mathcal{F}_2$, as desired. ◊

In what follows, we are going to analyze the computational worst-case complexity of PCF. Sadly, our results imply that no polynomial time algorithm (thus: easily verifiable necessary and sufficient conditions) can exist, unless P = NP, since we will establish the following theorem:

**Theorem 10.** *PCF is NP-complete.*

We establish the easier result of NP-membership in Section 4 and the somewhat more involved NP-hardness in Section 5. The impacts of this complexity result on "easy expressibility" of necessary and sufficient conditions for PCF validity will be discussed in Section 6.

## 4 Membership in NP

As usual, to exactly pin down a problem's complexity class, one has to show membership and hardness. The former means that, using a certain underlying computational model, the problem can be solved within certain space or time bounds depending on the size of the input. The class NP contains all problems which can be solved in polynomial time on a nondeterministic Turing machine. A helpful alternative characterization of NP is through a "Guess and Verify" method: Using this model, the algorithm is allowed to "guess" a so called *certificate* – a piece of data the size of which is polynomially bounded by the size of the input. After guessing, the algorithm verifies the combined input and certificate in deterministic polynomial time. The crux here is that the algorithm will "guess right" whenever an appropriate certificate exists. For further and more formal treatment of the matter, we refer the interested reader to standard textbooks on complexity theory [14,21].

We will now proceed to establish NP-membership of PCF. While this falls short of establishing membership in P, it might also not be immediately clear why the complexity is not higher, say, further up in the polynomial hierarchy or even PSpace, if not ExpTime or NExpTime: The prime choice for a certificate would certainly be a "witnessing" closure operator $\varphi$, that is, one whose pseudo-closed sets are exactly the elements of the family $\mathcal{F}$ given as input. Yet, representing such a closure operator explicitly as a function $2^S \to 2^S$ would require exponential space (and hence too much to be of any use as an NP certificate). Luckily, as one might fathom upon studying the argument in Example 2, this issue can be overcome by representing $\varphi$ by means of its canonical base, which both gives us direct access to the pseudo-closed sets and has size directly bounded by the size of $S$ and $\mathcal{F}$.

**Proposition 11** *PCF is in NP.*

*Proof.* Given an instance $(S, \mathcal{F})$ of PCF, we specify a "Guess and Verify" type algorithm consisting of four steps:

*Guess.* For each $P \in \mathcal{F}$, nondeterministically choose a set $B_P \subseteq S$ with $B_P \supsetneq P$. The intuitive goal is to guess, if possible, the appropriate right-hand side for the implication with $P$ on the left in the canonical base.
*Build.* Form the implication set $\mathfrak{I} = \{P \to B_P \mid P \in \mathcal{F}\}$.
*Compute.* Run the Stembase algorithm (Theorem 8) on input $\mathfrak{I}$. This produces a canonical base $\mathfrak{S}$ for the closure operator $(\cdot)^{\mathfrak{I}}$ in quadratic time. By Theorem 8(b), the premises of $\mathfrak{S}$ are exactly the pseudo-closed sets of $(\cdot)^{\mathfrak{I}}$.
*Verify.* Accept exactly if the set of premises of $\mathfrak{S}$ equals $\mathcal{F}$.

We now argue that the algorithm is correct; it accepts if and only if $\mathcal{F}$ is valid. If the algorithm accepts, then $(\cdot)^{\mathfrak{I}}$ is a closure operator whose pseudo-closed sets are exactly $\mathcal{F}$, so $\mathcal{F}$ is indeed valid. Conversely, suppose $\mathcal{F}$ is valid, witnessed by some closure operator $\varphi$. By Theorem 7, the canonical base of $\varphi$ is $\mathfrak{S}_\varphi = \{P \to \varphi(P) \mid P \in \mathcal{F}\}$. Guessing $B_P = \varphi(P)$ for each $P \in \mathcal{F}$ gives $\mathfrak{I} = \mathfrak{S}_\varphi$.

Since $\mathfrak{I}$ is already a canonical base, the STEMBASE algorithm produces a canonical base with the same closure operator and hence the same pseudo-closed sets, namely $\mathcal{F}$. So this guess leads to acceptance.

Considering the complexity, we find that the guess has size $O(|\mathcal{F}| \cdot |S|)$, so it is clearly of polynomial size regarding the input. We now consider the time complexities of the three subsequent steps. Building $\mathfrak{I}$ is in $O(|\mathcal{F}| \cdot |S|)$. Running STEMBASE takes $O(|\mathcal{F}|^2 \cdot |S|)$ time and comparing the premises of $\mathfrak{S}$ with $\mathcal{F}$ takes $O(|\mathcal{F}| \cdot |S|)$ time. Therefore, the total is polynomial in $|S|$ and $|\mathcal{F}|$. □

This result also gives an (although certainly unsatisfactory) answer to Ganter's question: a necessary and sufficient condition for PCF is the existence of appropriate right-hand sides of the implications in the canonical base. Once they are provided, verification is easy. Clearly, a brute-force implementation checking all possibilities would be highly inefficient and much more goal-directed algorithms can be conceived, exploiting known properties of closure operators to reduce the search space and using highly optimized solvers for NP-complete problems (such as CSP or SAT solvers), but this would not improve the worst-case complexity of the problem (as shown by the hardness proof hereafter), and corresponding considerations are outside the scope of this paper.

The characterization of positive PCF instances as left-hand sides of canonical bases also gives rise to a noteworthy side observation: if $(S, \mathcal{F})$ is a positive PCF instance, then so is $(S, \mathcal{G})$ for every $\mathcal{G} \subseteq \mathcal{F}$. This follows from the insight that every subset of a canonical base is a canonical base (for another closure operator), since applying the STEMBASE algorithm would not lead to any changes.

## 5 NP-Hardness

We now turn to the NP-hardness part, essentially showing that PCF is intrinsically hard and no deterministic polynomial time algorithm can be found that solves it (unless P = NP). The common way to establish such a result is through reduction from another problem, whose NP-hardness has already been established. After identifying such an appropriate problem, one shows that any instance of said problem can be transformed in deterministic polynomial time into an acceptance-equivalent instance of the problem whose hardness is to be established. Intuitively, an easy way to solve the latter problem would imply existence of an easy way to solve the former, which would lead to a contradiction. Again we refer to the standard literature for more details [14,21].

In our case we need to identify a known NP-complete problem admitting a polynomial time translation to PCF, such that both acceptance and nonacceptance are preserved. The available options are plentiful, but there are some "usual suspects" to be tried first. One of these is *3SAT*, that is, checking satisfiability of a propositional formula that takes the form of a big conjunction over three-way disjunctions (called clauses) over literals (negated or unnegated atoms). Attempts to use this problem directly turned out to be unsuccessful. However, numerous variations of 3SAT have been discussed in the literature and

classified complexity-wise. Among those is *NAE-3SAT* (where NAE stands for "not all equal"), refining 3SAT by requiring that the satisfying valuation must not evaluate all three literals of any clause to the same truth value simultaneously. A variant of NAE-3SAT is Monotone NAE-3SAT, which considers only negation-free formulas (that is, the only occurring literals are unnegated atoms).

**Definition 12 (Monotone NAE-3SAT)** *An instance* $\Psi$ *of* Monotone NAE-3SAT *consists of a finite set* $\{x_1, \ldots, x_n\}$ *of* variables *and a finite set* $\{C_1, \ldots, C_k\}$ *of* clauses*, where each clause is a set* $\{x_a, x_b, x_c\}$ *of three distinct variables. We ask whether there exists a truth assignment* $\sigma : \{x_1, \ldots, x_n\} \to \{\mathbf{true}, \mathbf{false}\}$*, such that* $\{\sigma(x_\ell) \mid x_\ell \in C_i\} = \{\mathbf{true}, \mathbf{false}\}$ *for every* $i \in \{1, \ldots, k\}$*, i.e., every clause contains at least one* **true** *and at least one* **false** *variable.*

Schaefer proved that Monotone NAE-3SAT is NP-complete as part of his dichotomy theorem for generalized satisfiability [18].[3] In our investigations, it turned out that Monotone NAE-3SAT is best suited for establishing a reduction to PCF (to the best of our knowledge). Thus we will proceed as follows: First, we specify a translation from a Monotone NAE-3SAT instance $\Psi$ into a PCF instance $(S_\Psi, \mathcal{F}_\Psi)$ (Section 5.1). Then we show that the translation preserves non-acceptance (Section 5.2) and acceptance (Section 5.3).

### 5.1 The Reduction

Let $\Psi$ be a Monotone NAE-3SAT instance with variables $x_1, \ldots, x_n$ ($n \geq 2$) and distinct clauses $C_1, \ldots, C_k$. Let $C_j = \{x_{a_j}, x_{b_j}, x_{c_j}\}$, with pairwise distinct indices inside each clause. We define the corresponding PCF instance $(S_\Psi, \mathcal{F}_\Psi)$.

*Ground set* $S_\Psi$. For each variable $x_i$, we introduce a "complement element" $y_i$. Additionally, we introduce a fresh element $z$. Thus we define

$$S_\Psi = T \cup \{z\} \qquad \text{with} \qquad T = \{x_1, \ldots, x_n,\ y_1, \ldots, y_n\}.$$

Elements $x_i$ will be referred to as *x-type*; elements $y_i$ as *y-type*.

*Family* $\mathcal{F}_\Psi$. We define three types of subsets of $S_\Psi$:

$$\begin{aligned}
V_i &= T \setminus \{x_i, y_i\} && \text{for } 1 \leq i \leq n && \text{(assignment indicators)},\\
D_j^+ &= T \setminus \{x_{a_j}, x_{b_j}, x_{c_j}\} && \text{for } 1 \leq j \leq k && \text{(all-}\mathbf{true}\text{ detectors)},\\
D_j^- &= T \setminus \{y_{a_j}, y_{b_j}, y_{c_j}\} && \text{for } 1 \leq j \leq k && \text{(all-}\mathbf{false}\text{ detectors)}.
\end{aligned}$$

Now we set $\mathcal{R} = \{V_i\} \cup \{D_j^+\} \cup \{D_j^-\}$ and let $\mathcal{F}_\Psi = \mathcal{R} \cup \{T\}$. Clearly, the reduction runs in polynomial time: $|\mathcal{F}_\Psi| = n + 2k + 1$ and $|S_\Psi| = 2n + 1$.

[3] Interestingly, one of the reviewers pointed out that this problem is structurally equivalent to deciding whether a given 3-uniform hypergraph is 2-colorable, proven to be NP-complete by László Lovász [12] as early as 1973. The original publication seems to have a glitch that is easy to fix: `https://cstheory.stackexchange.com/questions/39665/reduction-of-graph-chromatic-number-to-hypergraph-2-colorability`

It might be helpful to provide some upfront intuition about the proposed reduction. Compared to $T$, each assignment indicator $V_i$ misses the pair $\{x_i, y_i\}$. Then any witnessing closure operator must map $V_i$ to a set strictly between $V_i$ and $T$, adding exactly one of $x_i$ or $y_i$. This choice is meant to encode the truth value of $x_i$. The detectors $D_j^+$ and $D_j^-$ enforce the NAE condition: if all three variables in a clause receive the same truth value, the corresponding detector becomes closed, contradicting pseudo-closedness. We conclude this section with a straightforward observation.

**Fact 13 (incomparability)** *All sets in $\mathcal{R}$ are incomparable under inclusion.*

*Proof.* For each set $Q \in \mathcal{R}$, the "gap" $T \setminus Q$ has a characteristic type signature: $T \setminus V_i = \{x_i, y_i\}$ (one $x$-type, one $y$-type); $T \setminus D_j^+ = \{x_a, x_b, x_c\}$ (three $x$-type); $T \setminus D_j^- = \{y_a, y_b, y_c\}$ (three $y$-type). Containment $A \subseteq B$ requires $T \setminus B \subseteq T \setminus A$, which fails in every pair by comparing types and cardinalities. □

### 5.2 Backward direction: $\mathcal{F}_\Psi$ valid implies $\Psi$ satisfiable

Now we prove that the translation from Monotone NAE-3SAT to PCF indeed preserves non-acceptance, that is, if $\Psi$ does not have an NAE-satisfying variable assignment, then $\mathcal{F}_\Psi$ is not valid. We will show the contrapositive. To this end, assume $\mathcal{F}_\Psi$ is valid, witnessed by some closure operator $\varphi$. The following easily observed fact legitimizes the subsequent specification of a variable assignment.

**Fact 14 (variable partition)** *For each $i$, $\varphi(V_i)$ contains exactly one of $\{x_i, y_i\}$.*

*Proof.* $\mathcal{F}_\Psi$ being valid means $V_i$ and $T$ are pseudo-closed. Then, by the sandwich property: $V_i \subsetneq \varphi(V_i) \subsetneq T$. Since $T \setminus V_i = \{x_i, y_i\}$, the set $\varphi(V_i)$ contains at least one but not both of these elements. □

With this guarantee we can now define the variable assignment $\sigma$ by letting $\sigma(x_i) = \textbf{true}$ if $x_i \notin \varphi(V_i)$ (which is equivalent to $\varphi(V_i) = T \setminus \{x_i\}$), and **false** otherwise – in which case we have $y_i \notin \varphi(V_i)$, or equivalently $\varphi(V_i) = T \setminus \{y_i\}$.

**Lemma 15 (NAE condition)** *$\sigma$ satisfies every clause of $\Psi$ as required by Monotone NAE-3SAT.*

*Proof.* For any clause $C_j = \{x_a, x_b, x_c\}$ of $\Psi$, we verify that it contains at least one variable mapped to **true** and one to **false**, through a proof by contradiction.

*Not all* **true**. Suppose $\sigma(x_a) = \sigma(x_b) = \sigma(x_c) = \textbf{true}$. Then $\varphi(V_i) = T \setminus \{x_i\}$ must hold for each $i \in \{a, b, c\}$. Now observe that for each such $i$ we get $D_j^+ = T \setminus \{x_a, x_b, x_c\} \subseteq T \setminus \{x_i\} = \varphi(V_i)$. Since each $\varphi(V_i)$ is closed, we obtain $\varphi(D_j^+) \subseteq \varphi(V_i)$ by monotonicity and idempotency, as discussed earlier. Intersecting all these three upper bounds under set inclusion gives us:

$$\varphi(D_j^+) \subseteq \bigcap\nolimits_{i \in \{a,b,c\}} (T \setminus \{x_i\}) = T \setminus \{x_a, x_b, x_c\} = D_j^+.$$

Together with extensivity, we would obtain $\varphi(D_j^+) = D_j^+$, and therefore conclude that $D_j^+$ is closed. Yet, this contradicts pseudo-closedness of $D_j^+$.

*Not all* **false**. Assume $\sigma(x_a) = \sigma(x_b) = \sigma(x_c) = \mathbf{false}$. Then we obtain $\varphi(V_i) = T \setminus \{y_i\}$ for $i \in \{a, b, c\}$, and $D_j^- = T \setminus \{y_a, y_b, y_c\} \subseteq \varphi(V_i)$ for each such $i$. Thus, the analogous argument to the one above yields $\varphi(D_j^-) = D_j^-$, contradicting pseudo-closedness. □

We have thus shown that every valid $\mathcal{F}_\Psi$ (or rather the corresponding witnessing closure operator $\varphi$) gives rise to an assignment $\sigma$ witnessing $\Psi$'s satisfiability. By contraposition, our construction of $\mathcal{F}_\Psi$ from $\Psi$ preserves non-acceptance.

### 5.3 Forward direction: $\Psi$ satisfiable implies $\mathcal{F}_\Psi$ valid

What is left is to show that the proposed translation also preserves acceptance. To this end, assume $\Psi$ is satisfiable and let $\sigma$ be an NAE-satisfying assignment. We need to construct a closure operator whose pseudo-closed sets are exactly the elements of $\mathcal{F}_\Psi$. It will be convenient to describe said closure operator in terms of an implication set. The subsequent definition provides us with such an implication set, and also establishes the instrumental notion of witness elements.

**Definition 16** *Given an NAE-satisfying assignment $\sigma$ for $\Psi$, we construct an implication set $\mathfrak{I}$ over $S_\Psi$ as follows. First, for each $Q \in \mathcal{F}_\Psi$, choose an element $s_Q \in S_\Psi \setminus Q$ such that:*
- *$s_{V_i} = y_i$ if $\sigma(x_i) = \mathbf{true}$; $s_{V_i} = x_i$ if $\sigma(x_i) = \mathbf{false}$.*
- *$s_{D_j^+} = x_\ell$ for some $x_\ell$ in clause $C_j$ with $\sigma(x_\ell) = \mathbf{false}$ (exists by NAE).*
- *$s_{D_j^-} = y_h$ for some $x_h$ in clause $C_j$ with $\sigma(x_h) = \mathbf{true}$ (exists by NAE).*
- *$s_T = z$.*

*Then define $\mathfrak{I} = \{Q \to Q \cup \{s_Q\} \mid Q \in \mathcal{F}_\Psi\}$ and let $\varphi = (\cdot)^\mathfrak{I}$.*

*Moreover, for each $i \in \{1, \ldots, n\}$, we define the corresponding* witness *$w_i$ such that $w_i \in \{x_i, y_i\} \setminus \{s_{V_i}\}$, that is, the witness is the element not chosen as $s_{V_i}$. Consequently, $w_i = x_i$ if $\sigma(x_i) = \mathbf{true}$ and $w_i = y_i$ if $\sigma(x_i) = \mathbf{false}$.*

Note that there is a certain leeway in the construction of $\mathfrak{I}$ as the choices of $s_Q$ are not all uniquely determined for the "detectors". The witness elements, on the other hand, are uniquely defined. A rough intuition behind the witnesses is to "shield" $T$ from being "reached" by the closure of any of its subsets from $\mathcal{F}_\Psi$. The subsequent development of lemmas makes this intuition formal.

**Lemma 17 (witnesses are never chosen)** *For each $i \in \{1, \ldots, n\}$ and every $Q \in \mathcal{R}$ we have $s_Q \neq w_i$.*

*Proof.* We consider any $w_i$ and proceed by cases.
Case $\sigma(x_i) = \mathbf{true}$ (therefore $w_i = x_i$). We go through all choices for $Q$:
- For $Q = V_i$, we obtain $s_{V_i} = y_i \neq x_i$.
- Also, clearly for $Q = V_j$ with $j \neq i$, we have $s_{V_j} \in \{x_j, y_j\} \not\ni x_i$.
- For $Q = D_j^+$, we find that $s_{D_j^+} = x_\ell$ with $\sigma(x_\ell) = \mathbf{false}$, so $x_\ell \neq x_i$;
- For $Q = D_j^-$, we see that $s_{D_j^-} = y_h$ is $y$-type, so $s_{D_j^-} \neq x_i$.

Case $\sigma(x_i) = \mathbf{false}$ (therefore $w_i = y_i$). Again, we go through all choices for $Q$:

- For $Q = V_i$, we obtain $s_{V_i} = x_i \neq y_i$.
- Also, clearly for $Q = V_j$ with $j \neq i$, we have $s_{V_j} \in \{x_j, y_j\} \not\ni y_i$.
- For $Q = D_j^+$, we see that $s_{D_j^+} = x_\ell$ is $x$-type, so $s_{D_j^+} \neq y_i$.
- For $Q = D_j^-$, we find that $s_{D_j^-} = y_h$ with $\sigma(x_h) = \mathbf{true}$, which requires $h \neq i$ and hence implies $y_h \neq y_i$. □

**Lemma 18 (witness complements are closed)** *For each $i \in \{1, \ldots, n\}$, the set $T \setminus \{w_i\}$ is a closed set of $\varphi$.*

*Proof.* Let $C = T \setminus \{w_i\}$. We check that $C$ respects every implication in $\mathfrak{I}$. Thus we have to go through all $Q \in \mathcal{F}_\Psi$ with $Q \subseteq C$.

Obviously, the case $Q = T$ is impossible, since $w_i \in T$ but $w_i \notin C$.

If $Q \in \mathcal{R}$, we need to ensure that $s_Q \in C$, which just means that $s_Q \neq w_i$ must hold (all other possible $s_Q$ are already trivially contained in $T$). And indeed, $s_Q \neq w_i$ holds by Lemma 17. □

**Lemma 19 (no closure reaches $T$)** *For every $Q \in \mathcal{R}$ we have $Q^\mathfrak{I} \subsetneq T$.*

*Proof.* For any such $Q$, we exhibit a witness $w_i$ with $w_i \notin Q$, so that $Q \subseteq T \setminus \{w_i\}$. Since $T \setminus \{w_i\}$ is closed by Lemma 18, we can then conclude $Q^\mathfrak{I} \subseteq T \setminus \{w_i\} \subsetneq T$.

- For $Q = V_i$, we obtain $w_i \in \{x_i, y_i\} = T \setminus V_i$, so $w_i \notin V_i$.
- For $Q = D_j^+$, consider $C_j = \{x_a, x_b, x_c\}$. Since $\sigma$ is an NAE-satisfying assignment, there must be some variable $x_i \in \{x_a, x_b, x_c\}$ with $\sigma(x_i) = \mathbf{true}$. But then $w_i = x_i \notin D_j^+$.
- For $Q = D_j^-$, again consider $C_j = \{x_a, x_b, x_c\}$. As $\sigma$ is an NAE-satisfying assignment, there must also be some variable $x_i \in \{x_a, x_b, x_c\}$ with $\sigma(x_i) = \mathbf{false}$. Then we obtain $w_i = y_i \notin D_j^-$. □

Finally, the insights obtained through this chain of lemmas can be leveraged to make the final step and establish that the closure operator $\varphi$ realized by the implication set $\mathfrak{I}$ has exactly the elements of $\mathcal{F}_\Psi$ as pseudo-closed sets.

**Proposition 20** *The pseudo-closed sets of $\varphi$ are exactly $\mathcal{F}_\Psi$.*

*Proof.* We show for each subset $F \subseteq S_\Psi$ the following correspondence: $F$ is contained in $\mathcal{F}_\Psi$ if and only if it is pseudo-closed. We use strong induction on $|F|$. For the base case, $F = \emptyset \notin \mathcal{F}_\Psi$, the statement holds since $\varphi(\emptyset) = \emptyset$.

Now assume $F \in \mathcal{F}_\Psi$, whence we need to show that $F$ is pseudo-closed. We show $\varphi(F) \neq F$ by observing $F^\mathfrak{I} \supseteq F \cup \{s_F\} \supsetneq F$. In order to show the second condition, let $Q$ be pseudo-closed with $Q \subsetneq F$. By the induction hypothesis this means that $Q \in \mathcal{F}_\Psi$. In case $F = T$, we therefore get $Q \in \mathcal{R}$ and can indeed conclude $Q^\mathfrak{I} \subsetneq T$ thanks to Lemma 19. In case $F \in \mathcal{R}$ we know by Fact 13 that no element of $\mathcal{F}_\Psi$ is a proper subset of $F$, thus condition (ii) is vacuously true.

Now assume $F \notin \mathcal{F}_\Psi$. We need to show that $F$ is not pseudo-closed. Toward a contradiction, suppose $F$ were pseudo-closed. Then $F^\mathfrak{I} \supsetneq F$ would need to hold (due to $F$ not being closed). Therefore, some $P \in \mathcal{F}_\Psi$ with $P \subseteq F$ must have $s_P \notin F$. Now since $P \neq F$ (due to $F \notin \mathcal{F}_\Psi$) we get $P \subsetneq F$. By the induction hypothesis, $P$ must be pseudo-closed, so condition (ii) of pseudo-closedness for $F$ requires $P^\mathfrak{I} \subseteq F$. However $s_P \in P^\mathfrak{I}$ which clearly contradicts $s_P \notin F$. □

This concludes our argument that for every satisfiable Monotone NAE-3SAT instance $\Psi$, the corresponding NAE-satisfying variable assignment can be exploited to create a closure operator – expressed through an implication set – whose pseudo-closed sets coincide with $\mathcal{F}_\Psi$'s elements, ensuring that $\mathcal{F}_\Psi$ is valid. Consequently, our construction of $\mathcal{F}_\Psi$ from $\Psi$ preserves acceptance.

## 6 What This Means for Ganter's Original Question

The reader might (and, in fact, one of this paper's reviewers *did*) ask what the obtained result has to do with Ganter's original question, which did not ask for the complexity of PCF at all, but rather for necessary and sufficient conditions for it. Fortunately, the field of *descriptive complexity theory* [8] has established connections between computational complexity and expressivity which we can exploit to arrive at an answer that might be considered more informative.

**Definition 21 (easy property)** *Given an instance $(S, \mathcal{F})$ of PCF, let us define the corresponding first-order structure $\mathcal{I}_{(S,\mathcal{F})}$ as follows: The domain of $\mathcal{I}_{(S,\mathcal{F})}$ is set to $D = S \cup \mathcal{F} \cup \{S\} \cup \{\mathcal{F}\}$; that is, it contains all elements of $S$ and all elements of $\mathcal{F}$ as first-class citizens, plus one element representing the full set $S$ and one representing the family $\mathcal{F}$; we assume $S$ does not itself contain sets as elements. Beyond that, $\mathcal{I}_{(S,\mathcal{F})}$ contains two constants* $\mathtt{S}$ *and* $\mathtt{F}$ *(interpreted by $S$ and $\mathcal{F}$, respectively) as well as one binary predicate $\in$ (written in infix notation), which is interpreted in the obvious way: $\in^{\mathcal{I}_{(S,\mathcal{F})}} = \{(d, d') \mid d \in d'\}$.*

*We call a property that a PCF instance may or may not have,* easy*, if it is first-order expressible, that is, if there exists a first-order logic sentence $\chi$ such that $\mathcal{I}_{(S,\mathcal{F})} \models \chi$ exactly if $(S, \mathcal{F})$ has the property.*

Arguably, properties that are easy in the sense formalized above are preferable as structural conditions – they are both rather straightforward to grasp cognitively and computationally verifiable with low effort.

*Example 3.* Using the notational macros $t \notin s$ for $\neg(t \in s)$, as well as $t \subseteq s$ for $\forall w.(w \in t \Rightarrow w \in s)$, and $t \subset s$ for $t \subseteq s \wedge \neg(s \subseteq t)$, Ganter's two necessary conditions can be expressed by the following two first-order sentences:

$$\mathtt{S} \notin \mathtt{F} \wedge \forall y.\big(y \in \mathtt{F} \Rightarrow \exists x.x \notin y \wedge \forall z.(y \subset z \Rightarrow x \in z)\big)$$

$$\forall y, z.\big((y \in \mathtt{F} \wedge z \in \mathtt{F} \wedge y \subset z) \Rightarrow \exists x, x'.(x \neq x' \wedge x \in z \wedge x' \in z \wedge x \notin y \wedge x' \notin y)\big)$$

that is, both of these properties (and thus also their conjunction) are easy in the sense defined above. ◊

**Theorem 22.** *There is no easy property characterizing PCF validity.*

*Proof (Sketch).* For space reasons, we resort to a high-level argument and assume familiarity with circuit complexity. If PCF validity were first-order expressible over $\mathcal{I}_{(S,\mathcal{F})}$, it would be decidable in $\mathrm{AC}_0$ via descriptive complexity theory [8]. On the other hand, it is easy to show that PCF is NP-complete even under $\mathrm{AC}_0$ reductions. Since $\mathrm{PARITY} \in \mathrm{NP}$, composition would then yield $\mathrm{AC}_0$ circuits for PARITY, contradicting the lower bound established by Håstad [7]. □

Note that this non-expressibility result is independent of the P vs. NP question. It essentially asserts that no first-order statement using the usual set operations and comparisons will work to characterize PCF validity, unless we allow to quantify over *all* subsets of $S$ (which the definition of our setting excludes).

## 7 Contemplation: LLMs for Mathematical Research

Given that the methodology to arrive at the hardness result deviates from a mathematician's traditional working style "thinking hard while looking out of the window", the author finds it opportune to deliberate on his use of the large language model Claude Opus 4.6 accessed via `you.com` through a university-wide subscription. The full chat history of the exchange is made available online [17]. Note that the collaboration by itself was not a well-planned, controlled experiment, but rather an ad hoc attempt. Consequently, the reported observations and experiences have to be taken as anecdotal evidence and subjective impressions, which may include the tendency to anthropomorphize the used system, and be it only for the sake of a more relatable reporting.

The impression that the PCF problem might serve as a good test case to try this new approach arose from the consideration that

- the problem seemed easy to describe in a self-contained way,
- the type of argument needed (PTime algorithm or NP reduction) is quite standard and occurs frequently in the literature (ergo: the training data),
- a final argument would likely not be overly intricate, once discovered,
- the author had investigated the issue and thus developed a good intuition and capability of double checking, correcting, and steering the model's activities.

### 7.1 Method

To get started, the author provided Claude with a PDF of "Some Notes on Pseudo-Closed Sets" [15], providing many of the necessary formal preliminaries. To trigger engagement with the material, we let Claude create an in-depth summary of the technical content. Thereafter, the actual problem was presented, asking for an idea about the complexity. This being unsuccessful, Claude was presented the author's previously established argument for NP membership, which it[4] appeared to understand (and even to enthusiastically embrace). In the ensuing exchange on a rather scholarly level, the author deemed it useful to follow certain interaction patterns discussed next.

*Enforcing rigor.* The author repeatedly and persistently exhorted Claude to be rigorous and meticulous in its investigation and output. It was explicitly discouraged from using vague statements and hand-waving of any kind, and it was immediately called out whenever it did (way more harshly than the author would ever address a fellow human being). Whenever the model engaged in so-called *scratching* (indicated by phrases like "Hm...", "Wait!", "This does not work.", or "Let me try another approach."), it was afterwards told to produce a

[4] To withstand anthropomorphization, we use "it" rather than "he", "she" or "they".

clear and streamlined presentation of the findings obtained. At times, the model was asked to produce a full LaTeX document, for a technical and stylistic review.

*Mode switching.* For a long time in the process, it was unclear if PCF was indeed NP-hard, or if a PTime algorithm did exist after all. The model repeatedly encountered problems establishing the hardness proof. In such cases, the author would switch the mode of investigation by asking Claude to leverage the obstacles encountered when trying to show NP-hardness for the creation of a PTime algorithm. When after continued attempts to come up with such an algorithm Claude reported to have hit another barrier, it was asked to switch the mode again and exploit the insights gained for a hardness argument. Several such mode switches occurred in the course of the exchange, which seemed to result in "peeling off" peripheral problems and getting to the core of the issue.

*Use of examples.* Whenever Claude was of the opinion to have established a correct PTime algorithm (which happened several times), the author advised it to verify the algorithm's correctness against several PCF instances – both positive and negative ones and both self-created ones as well as provided by the author. The same strategy was pursued whenever Claude was stuck trying to prove an important lemma towards the result. Very often glitches in the arguments could be found and assumed lemmas refuted by this technique.

*Suggesting where to look.* Occasionally, Claude had identified a subproblem which was crucial for the overall investigation. Sometimes the author was able to provide guidance and pointers to related problems or problem classes. Claude typically welcomed such suggestions and steered its investigation accordingly.

*Pushing for Simplification.* In the lengthy process of the theory development, many auxiliary notions were defined and helper lemmas proven. Some directions were abandoned. When asked to produce a coherent document, it became apparent that some of this material was irrelevant to the actual line of argumentation. For a clearer picture it was sometimes helpful to prompt Claude to critically verify which parts of the development were still necessary and which ones could be deleted. Likewise, toward the end of the exchange, Claude had established a twofold reduction: from Monotone NAE-3SAT to an intermediate problem (which Claude itself had created to describe the "hard core" of the task) and from there to PCF. While this was correct, it required another human intervention to turn this into a direct reduction, in the course of which another significant simplification and shortening of the material could be achieved.

Once a firm and reliable result was established and verified, Claude was prompted one last time to produce LaTeX output for a document. This document served as a starting point for this paper's technical part. What followed was a last in-depth check of all the arguments by the author and a very thorough rewrite, including re-arranging the material to follow a better narrative, adding "prose" to provide more intuition and smoothing out arguments in the proofs. Section 6 was added to the paper's final version (without any involvement of Claude) as a response to a concern brought up by one reviewer.

### 7.2 Observations regarding Claude

Claude makes errors on all levels. It may even occasionally confuse $\subseteq$ with $\supseteq$, but it seems such occurrences can be reduced by relentless appeal to mathematical rigor and the advice to continuously double-check itself. When called out on errors, it was usually quick in admitting and correcting them. On the other hand, Claude can be credited with a good knowledge of general proof strategies, as well as problem candidates for complexity reductions – it was indeed Claude which picked Monotone NAE-3SAT as appropriate "yardstick problem". Claude also sometimes surprised in providing good intuition and higher-level reasoning. It even verbalized encountered problems using catchy metaphors – Claude was the one to actually introduce certain names such as the "sandwich property" or "all-**true**-detector". Claude is not immune against the infamous phenomenon of reference fabrication; somewhat amusingly, it claimed the PCF problem had been posed in a 2024 technical report entitled "On pseudo-closed sets and their recognition" authored by a certain S. Krötzsch together with some H. Reiter.

### 7.3 Observations regarding Sebastian

One self-observation by the author is that the model somehow manages to keep the user "hooked", likely by conveying the impression that the final solution is just around the corner (using phrases like "the final solution" or "the breakthrough"), with the consequence of working extra hours unintentionally. This made the author wonder, if he could not have solved the problem just by himself with the same time investment – probably a clear case of *hindsight bias*.

An impression the author has mixed feelings about is that his role in the process felt quite "managerial" and thematically detached. This took away quite some of the usual fun of engaging with the matter in a hands-on fashion, and it also left the feeling of only having too loose a grasp of the matter, a feeling that only dissolved during the intense post-hoc reworking of the material.

## 8 Conclusion

Proving NP-completeness of the Pseudo-Closed Family problem, we negatively answered a question by Bernhard Ganter for straightforwardly verifiable necessary and sufficient criteria. The hardness result was established in the course of an intense mathematical exchange with Claude Opus 4.6, which also provided an initial draft of part of this paper's LaTeX sources. All technical results were carefully vetted by the (human) author, who also takes full responsibility for any remaining errors. The exchange has mellowed the author's sceptical stance toward the usefulness of large language models for serious mathematical work, but he would like to believe that the most impactful progress can be made only through collaborations between humans and AI agents.

**Acknowledgments.** We thank the anonymous reviewers of this paper for their valuable comments. This work was funded by BMFTR in ScaDS.AI and by BMFTR and DAAD in project 57616814 (SECAI).

## References


1. Babin, M.A., Kuznetsov, S.O.: Recognizing pseudo-intents is conp-complete. In: Proc. 7th Int. Conf. on Concept Lattices and Their Applications (CLA). CEUR Workshop Proceedings, vol. 477, pp. 294–301. CEUR-WS.org (2010)
2. Day, A.: The lattice theory of functional dependencies and normal decompositions. Int. J. Algebra Comput. **2**(4), 409–432 (1992)
3. Distel, F.: Hardness of enumerating pseudo-intents in the lectic order. In: Proc. 8th Int. Conf. on Formal Concept Analysis (ICFCA). LNCS, vol. 5986, pp. 124–137. Springer (2010)
4. Dowling, W.F., Gallier, J.H.: Linear-time algorithms for testing the satisfiability of propositional horn formulae. J. Log. Program. **1**(3), 267–284 (1984)
5. Ganter, B., Wille, R.: Formal Concept Analysis - Mathematical Foundations, Second Editiontion. Springer (2024)
6. Guigues, J.L., Duquenne, V.: Familles minimales d'implications informatives résultant d'un tableau de données binaires. Math. Sci. Hum. **95**, 5–18 (1986)
7. Håstad, J.: Almost optimal lower bounds for small depth circuits. In: Proc. 18th Annual ACM Symposium on Theory of Computing. pp. 6–20. ACM (1986)
8. Immerman, N.: Descriptive complexity. Graduate texts in computer science, Springer (1999)
9. Kuznetsov, S.O.: On the intractability of computing the Duquenne-Guigues base. J. Univers. Comput. Sci. **10**(8), 927–933 (2004)
10. Kuznetsov, S.O., Obiedkov, S.A.: Counting pseudo-intents and #P-completeness. In: Proc. 4th Int. Conf. on Formal Concept Analysis (ICFCA). LNCS, vol. 3874, pp. 306–308. Springer (2006)
11. Kuznetsov, S.O., Obiedkov, S.A.: Some decision and counting problems of the Duquenne-Guigues basis of implications. Discret. Appl. Math. **156**(11), 1994–2003 (2008)
12. Lovász, L.: Coverings and colorings of hypergraphs. In: Proc. 4th Southeastern Conf. on Combinatorics. pp. 3–12. Utilitas Math. (1973)
13. Mannila, H., Räihä, K.J.: Design of Relational Databases. Addison-Wesley (1992)
14. Papadimitriou, C.H.: Computational complexity. Addison-Wesley (1994)
15. Rudolph, S.: Some notes on pseudo-closed sets. In: Proc. 5th Int. Conf. on Formal Concept Analysis (ICFCA). LNCS, vol. 4390, pp. 151–165. Springer (2007)
16. Rudolph, S.: Succinctness and tractability of closure operator representations. Theor. Comput. Sci. **658**, 327–345 (2017)
17. Rudolph, S.: Exchange with Claude Opus 4.6 towards proving NP-completeness of the Pseudo-Closed Family problem (PCF) (May 2026). `https://doi.org/10.5281/zenodo.20307509`
18. Schaefer, T.J.: The complexity of satisfiability problems. In: Proc. 10th Annual ACM Symposium on Theory of Computing (STOC). pp. 216–226 (1978)
19. Sertkaya, B.: Some computational problems related to pseudo-intents. In: Proc. 7th Int. Conf. on Formal Concept Analysis (ICFCA). LNCS, vol. 5548, pp. 130–145. Springer (2009)
20. Sertkaya, B.: Towards the complexity of recognizing pseudo-intents. In: Proc. 17th Int. Conf. on Conceptual Structures (ICCS). LNCS, vol. 5662, pp. 284–292. Springer (2009)
21. Sipser, M.: Introduction to the theory of computation. PWS Publishing (1997)