TPLP: Page 1–19.
doi:10.1017/S1471068425100215

Complexity of Faceted Explanations in Propositional Abduction[*]

JOHANNES SCHMIDT
Jönköping University, Jönköping, Sweden
(*e-mail:* johannes.schmidt@ju.se)

MOHAMED MAIZIA
Jönköping University, Jönköping, Sweden
Linköping University, Linköping, Sweden
(*e-mail:* mohamed.maizia@ju.se)

VICTOR LAGERKVIST and JOHANNES K. FICHTE
Linköping University, Linköping, Sweden
(*e-mails:* victor.lagerkvist@liu.se, johannes.fichte@liu.se)

submitted 24 July 2025; revised 24 July 2025; accepted 27 July 2025

Abstract

Abductive reasoning is a popular non-monotonic paradigm that aims to explain observed symptoms and manifestations. It has many applications, such as diagnosis and planning in artificial intelligence and database updates. In propositional abduction, we focus on specifying knowledge by a propositional formula. The computational complexity of tasks in propositional abduction has been systematically characterized – even with detailed classifications for Boolean fragments. Unsurprisingly, the most insightful reasoning problems (counting and enumeration) are computationally highly challenging. Therefore, we consider reasoning between decisions and counting, allowing us to understand explanations better while maintaining favorable complexity. We introduce facets to propositional abductions, which are literals that occur in some explanation (relevant) but not all explanations (dispensable). Reasoning with facets provides a more fine-grained understanding of variability in explanations (heterogeneous). In addition, we consider the distance between two explanations, enabling a better understanding of heterogeneity/homogeneity. We comprehensively analyze facets of propositional abduction in various settings, including an almost complete characterization in Post's framework.

KEYWORDS: propositional abduction, computational complexity, Post's framework, fine-grained reasoning

[*] Author names are stated in reverse alphabetical order.

1 Introduction

Pedro is a passionate sailor. Today is Wednesday and he is about to go on the usual race to enjoy some waves and get a good challenge. But for some reason, nobody is out there for a race (a *manifestation*). He looks up into the sky and is a bit undecided, could the weather forecast have predicted calm winds or an unexpected storm resulting in calling off the race (a *hypothesis*) trying to explain his observation by finding appropriate causes. This type of backward reasoning is called abductive reasoning, one of the fundamental reasoning techniques that is commonly believed to be naturally used by humans when searching for diagnostic explanations. Abduction has many applications (Dai *et al.* 2019; Dellsén 2024; Ignatiev *et al.* 2019; Yu *et al.* 2023; Yu *et al.* 2023; Hu *et al*. 2025) and is well-studied in the areas of artificial intelligence, knowledge representation, and non-monotonic reasoning (Minsky 1974; Kakas *et al.* 1992; Miller 2019).

Qualitative reasoning problems like deciding whether an explanation exists or whether a proposition is relevant or necessary are computationally hard but still within range of modern solving approaches. More precisely, these problems are located on the second level of the polynomial hierarchy (PH) in the general case (Eiter and Gottlob 1995). However, asking for relevance or necessary propositions does not provide much insight into the variability of explanations. Enumeration and counting allow for more fine-grained reasoning but are computationally extremely expensive (Hermann and Pichler 2010; Creignou *et al.* 2019). Instead, we turn our attention to the world between *relevant* (belongs to some explanations) and *necessary* propositions (belongs to all explanations) and consider propositions that are *relevant but not necessary*, called *facets*.

In this work, we study the computational complexity of problems involving facets. To this end, we work in the universal algebraic setting by restricting the types of clauses/relations that are allowed (e.g., only Horn-clauses, or only 2-CNF). The resulting sets can be described by functions called *polymorphisms* and in the Boolean domain form a lattice known as *Post's lattice* (Post 1941). This setting makes it possible to obtain much more fine-grained complexity results and two prominent and early results are Lewis' dichotomy for propositional satisfiability (Lewis 1979) and Schaefer's dichotomy for Boolean constraint satisfaction (Schaefer 1978). However, this approach has been applied to many more problems (Creignou and Vollmer 2008), including non-monotonic reasoning and several variants of abduction (Nordh and Zanuttini 2008). We follow this line of research for faceted abduction.

Our **main contributions** are the following:

1. We introduce facets to propositional abduction thereby enabling a better understanding of propositions in explanations.
2. We establish a systematic complexity characterization for deciding facets in propositional abduction illustrated in Figure 1a.
   (i) Our classification provides a *complete picture* in Post's framework for all fragments, which can be described via clauses, for example, Horn, 2CNF, and dualHorn. Only two open cases remain: relations definable as Boolean linear equations of even length (with, or without, unit clauses).

https://doi.org/10.1017/S1471068425100215 Published online by Cambridge University Press

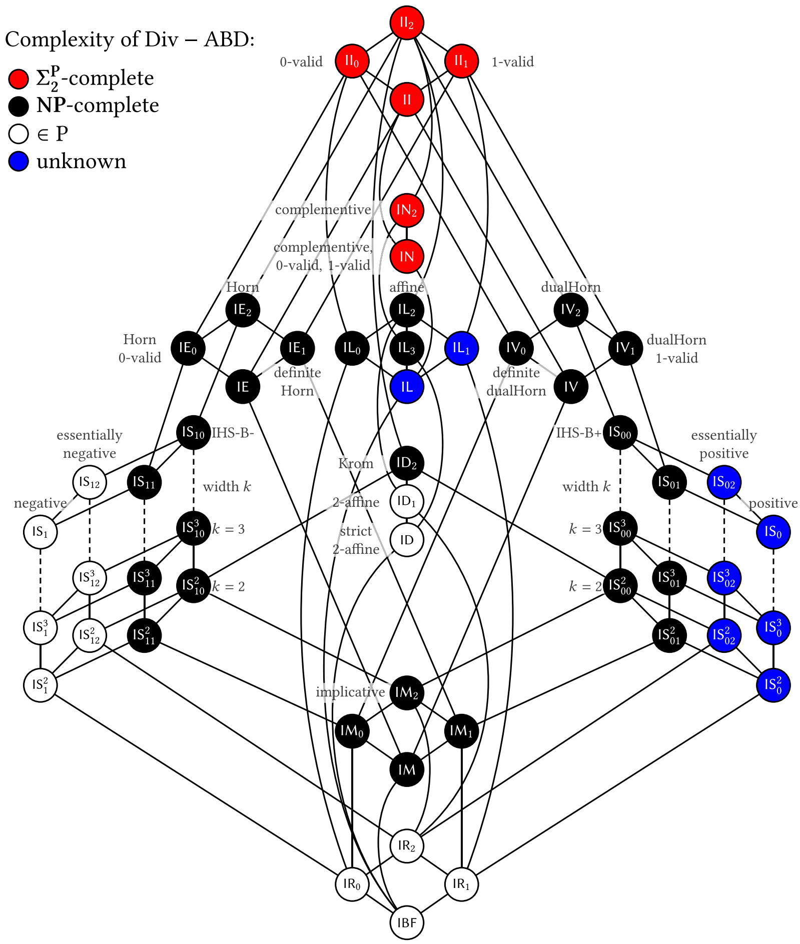

Fig. 1. Illustration of complexity results via Post's lattice.

(ii) Deciding facets is often not much harder than deciding explanation existence. In some surprising cases, the complexity increases.

(iii) Our facet results imply a corresponding classification for the problem of deciding relevance, which has received significant attention in the classical abduction literature and mentioned as an open question.

3. We study the related problem of determining whether there exist two explanations of sufficiently high *diversity*. This metric can be precisely related to the existence of facets and several notions from our facet classification carry over. However, this problem is provably much harder and becomes NP-hard already for a small fragment of Horn consisting of just implication $(x \rightarrow y)$.

1.1 Related works

The computational complexity of *propositional* abduction is well studied. Selman and Levesque (1990) showed that explanation existence is tractable for Horn theories for

https://doi.org/10.1017/S1471068425100215 Published online by Cambridge University Press

special cases of propositional abduction, but NP-hard when checking for an explanation that contains a particular proposition. Bylander *et al.* (1991) generalized the results to "best" explanations (most plausible combination of hypotheses that explains all the data) and present a tractable sub-class of abduction problems. Eiter and Gottlob (1995) proved Σ_2^{P}-completeness for propositional abduction. Creignou and Zanuttini (2006) established a precise trichotomy ($\mathrm{P}, \mathrm{NP}, \Sigma_2^{\mathrm{P}}$) in Schaefer's framework where inputs are restricted to generalized conjunctive normal form or subsets. Nordh and Zanuttini (2008) lifted results to propositional knowledge bases and establish a tetrachotomy (P, NP, co-NP, Σ_2^{P}). Creignou *et al.* (2010) showed a complete complexity classification for all considerable sets of Boolean functions. Relevance and dispensability are comparably speaking not as well understood. An early result by Friedrich *et al.* (1990) show that it is NP-hard to determine relevance over (definite) Horn theories, and Eiter and Gottlob (1995) additionally prove that it is Σ_2^{P}-hard to decide relevance if the knowledge base is an arbitrary propositional formula. Zanuttini (2003) later asks if there is a simple relationship between deciding relevance and the complexity of the underlying abduction problem. We find it surprising that Zanuttini's question (later repeated by Nordh and Zanuttini (2008)) still remains unanswered given that we by now have a complete understanding of the classical complexity of virtually all propositional abduction problems. In addition, counting and enumeration complexity is also well studied (Creignou *et al.* 2010, 2019; Hermann and Pichler 2010). Fellows *et al.* (2012); Pfandler *et al.* (2013); and Mahmood *et al.* (2021) included semantical structural restrictions (parameterized complexity) in the complexity study. The concept of facets has originally been introduced in the context of answer-set programming to enforce/forbid atoms in solutions and systematically investigate solutions without counting or enumeration (Alrabbaa *et al.* 2018; Fichte *et al.* 2022a). The complexity of ASP facets for tight, normal, and disjunctive programs was established very recently (Rusovac *et al.* 2024). Speck *et al.* (2025) lifted facets to symbolic planning for reasoning faster on the plan space, and Fichte *et al.* (2025) introduced facets to abstract argumentation. Eiter and Geibinger (2023) studied justifications for the presence, or absence, of an atom in the context of answer-set programming including so-called contrastive explanations. They provide a basic complexity theoretical characterization. Diversity has been considered in the literature on propositional satisfiability and logic programming, for example (Misra *et al.* 2024; Böhl *et al.* 2023). Abductive logic programming (ALP) combines logic programming with abductive reasoning, which then allows for generating hypotheses to explain observed facts or goals. Eiter *et al.* (1995) studied the complexity of ALP regarding consistency, relevance, and necessity but focusing on normal and disjunctive programs and commonly used semantics (well-founded, stable).

2 Preliminaries

We follow standard notions in computational complexity theory (Papadimitriou 1994; Arora and Barak 2009), propositional logic (Kleine Büning and Lettmann 1999), and propositional abduction (Bylander *et al*. 1991). Below, we briefly state relevant notations.

https://doi.org/10.1017/S1471068425100215 Published online by Cambridge University Press

2.1 Computational complexity

Let Σ and Σ' be some finite alphabets. We call $I \in \Sigma^*$ an *instance* and $||I||$ denotes the size of I. A *decision problem* is some subset $L \subseteq \Sigma^*$. Recall that P and NP are the complexity classes of all deterministically and non-deterministically polynomial-time solvable decision problems (Cook 1971). A polynomial-time many-to-one reduction ($\leq_m^{\mathrm{P}}$) from L to L' is a function $r : \Sigma^* \to \Sigma'^*$ such that for all $I \in \Sigma^*$ we have $I \in L$ if and only if $r(I) \in L'$ and r are computable in time $\mathcal{O}(||I|| \cdot c)$ for some constant c. In other words, a polynomial-time many-to-one reduction transforms instances of the decision problem L into instances of decision problem L' in polynomial time. We also need the PH (Stockmeyer and Meyer 1973; Stockmeyer 1976; Wrathall 1976). In particular, $\Delta_0^{\mathrm{P}} := \Pi_0^{\mathrm{P}} := \Sigma_0^{\mathrm{P}} := \mathrm{P}$ and $\Delta_{i+1}^{\mathrm{P}} := P^{\Sigma_i^p}$, $\Sigma_{i+1}^{\mathrm{P}} := \mathrm{NP}^{\Sigma_i^{\mathrm{P}}}$, and $\Pi_{i+1}^{\mathrm{P}} := \mathrm{coNP}^{\Sigma_i^{\mathrm{P}}}$ for $i > 0$ where C^D is the class C of decision problems augmented by an oracle for some complete problem in class D.

2.1.1 Propositional logic

A *literal* is a variable x or its negation $\neg x$. A *clause* is a disjunction of literals, often represented as a set. A clause of arity 1, that is, either (x) or $(\neg x)$, is a *unit clause*. We work in a general setting where atoms can be expressions of the form $R(x_1, \ldots, x_r)$ for variables $x_1, \ldots, x_r$ and an r-ary relation $R \subseteq \{0, 1\}^r$. A function $f \colon \{x_1, \ldots, x_r\} \to \{0, 1\}$ is then said to satisfy an atom $R(x_1, \ldots, x_r)$ if $(f(x_1), \ldots, f(x_r)) \in R$. A (conjunctive) *propositional formula* φ is a conjunction of atoms and we write $\mathrm{var}(\varphi)$ for its set of variables. A mapping $\sigma \colon \mathrm{var}(\varphi) \mapsto \{0, 1\}$ is called an *assignment* to the variables of φ and a *model* of a formula φ is an assignment to $\mathrm{var}(\varphi)$ that satisfies φ. For two formulas ψ and φ, we write $\psi \models \varphi$ if every model of ψ also satisfies φ.

2.2 Restrictions of constraint languages

As alluded in Section 1, we work in a fine-grained setting where not all possible relations are allowed. Formally, we say that a *constraint language* Γ is a set of Boolean relations, and a Γ-*formula* is a propositional formula φ where $R \in \Gamma$ for each atom $R(x_1, \ldots, x_r)$. For a constraint language Γ, we write $SAT(\Gamma)$ for the problem of deciding if a given Γ-formula admits at least one model. If Γ is naturally expressible as a set of clauses, we represent $R \in \Gamma$ in clausal form. This is the the case for most, but not all, cases that we consider in this paper. Usually, we do not distinguish between the relation, its defining clause, or an atom involving the clause. For example, we simply write (x) for the unary relation $\{(1)\}$, $(\neg x)$ for $\{(0)\}$, $(x_1 \to x_2)$ or $(\neg x_1 \vee x_2)$ for $\{(0,0), (0,1), (1,1)\}$, and so on. The empty set $\emptyset$ is the (nullary) relation that is always false, and we write $(x_1 = x_2)$ for the equality relation $\{(0,0), (1,1)\}$. For a constraint language Γ and $k \geq 1$, we often use the notation k-Γ for the set of relations/clauses of arity at most k. Thus, 2-CNF contains all 1/2-clauses, and 2-affine contains the unary/binary relations definable as equations mod 2. Additionally, for a language Γ we, let (1) $\Gamma^- = \Gamma \setminus \{(x), (\neg x)\}$ be Γ without the two unit clauses, and (2) $\Gamma^+ = \Gamma \cup \{(x), (\neg x)\}$ be Γ expanded with the two unit clauses. A language Γ is b-*valid* for $b \in \{0, 1\}$, if $(b, \ldots, b) \in R$ for each $R \in \Gamma$. We introduce the most important constraint languages for the purpose of this paper in Table 1. This

https://doi.org/10.1017/S1471068425100215 Published online by Cambridge University Press

Table 1. *Constraint languages and their corresponding co-clones. Here, Pos(c) and Neg(c) denote the number of positive and negative literals in a clause c, respectively. For more details, we refer to the work by Böhler et al. (2005) and the table in the supplemental material to this paper*

Name	Definition	Corresponding co-clone
CNF	$\{c \mid c \text{ is a clause}\}$	BR (II_2)
Horn	$\{c \mid c \text{ is a clause}, \text{Pos}(c) \leq 1\}$	IE_2
dualHorn	$\{c \mid c \text{ is a clause}, \text{Neg}(c) \leq 1\}$	IV_2
EN	$\{c \mid c \text{ is a clause}, \text{Pos}(c) = 0\} \cup \{(x), (x = y)\}$	IS_{12}
EP	$\{c \mid c \text{ is a clause}, \text{Neg}(c) = 0\} \cup \{(\neg x), (x = y)\}$	IS_{02}
affine	$(x_1 \oplus \ldots \oplus x_k) = b,\ k \geq 1,\ b \in \{0, 1\}$	IL_2

only covers a small number of the possible constraint languages. For many applications, including the facet classification in this paper, doing an exhaustive case analysis of *all* possible constraint languages is too complicated, and one needs simplifying assumptions. Here, it is known that each constraint language Γ can equivalently well be described as a set of functions closed under functional composition and containing all *projections* $\pi_i^n(x_1, \ldots x_n) = x_i$, *clones*. Thus, each clone groups together many similar constraint languages and the Boolean clones form a lattice known as *Post's lattice* when ordered by set inclusion (Post 1941). Many classification tasks become substantially simpler via Post's lattice, and it is well known that each clone corresponds to a dual relational object called a *co-clone*, which in turn induces a useful closure property on relations. In this paper we only need a small fragment of this algebraic theory and define this closure property via so-called *primitive positive definitions* (pp-definitions) and say that an r-ary relation R has a pp-definition over Γ if

$$R(x_1, \ldots, x_r) := \exists y_1, \ldots, y_n \,.\, \varphi(x_1, \ldots, x_r, y_1, \ldots, y_n)$$

where φ is a $(\Gamma \cup \{(x = y)\})$-formula. Thus, put otherwise, R can be defined as the set of models of $\exists y_1, \ldots, y_n \,.\, \varphi(x_1, \ldots, x_r, y_1, \ldots, y_n)$ with respect to the free variables $x_1, \ldots, x_r$. The reason for allowing the equality relation $(x = y)$ as an atom is that it leads to a much simpler algebraic theory. However, we sometimes need the corresponding definability notion without (free) equality. A pp-definition where φ is a Γ-formula is called an *equality-free primitive positive definition* (efpp-definition).

Definition 2.1.
For a constraint language Γ, we let $\langle\Gamma\rangle$ and $\langle\Gamma\rangle_{\neq}$, resp., be the smallest set of relations containing Γ and where $R \in \Gamma$ for any (ef)pp-definable relation R over Γ.

The set Γ is in this context said to be a *base*, and it is known that all co-clones can be defined in this way. For details, we refer to the work by Böhler *et al.* (2005) and the Table in the supplemental material to this paper. These notions generalize and unify many types of reductions and definability notions in the literature.

Example 2.2.
For example, consider the classical reduction from 4-SAT to 3-SAT by splitting a 4-clause $(\ell_1 \vee \ell_2 \vee \ell_3 \vee \ell_4)$ into two 3-clauses $(\ell_1 \vee \ell_2 \vee x)$ and $(\ell_3 \vee \ell_4 \vee \neg x)$ where x is a

https://doi.org/10.1017/S1471068425100215 Published online by Cambridge University Press

fresh variable. This can be viewed as a pp-definition $(\ell_1 \vee \ell_2 \vee \ell_3 \vee \ell_4) \equiv \exists x \,.\, (\ell_1 \vee \ell_2 \vee x) \wedge (\ell_3 \vee \ell_4 \vee \neg x)$, *and we conclude that* k-CNF $\subseteq \langle$3-CNF$\rangle$ *for any* $k \geq 1$.

A table of all Boolean co-clones is available in the supplemental material as well as another more elaborate example.

2.3 Propositional abduction

Let Γ be a constraint language, for example, a set of clauses. An instance I of the *positive propositional* abduction problem over Γ, $ABD(\Gamma)$ for short, is a tuple $I = (\mathrm{KB}, H, M)$ with KB being a Γ-formula over a finite set of Boolean variables called the *knowledge base* (or *theory*), $H \subseteq \mathrm{var}(\mathrm{KB})$ called *hypotheses*, $M \subseteq \mathrm{var}(\mathrm{KB})$ called *manifestations*. Since we have defined a Γ-formula as a conjunctive formula with atoms from Γ we sometimes take the liberty of viewing the knowledge base as a set rather than as a formula. A *positive explanation* E, *explanation* for short, is a subset $E \subseteq H$ such that (i) $\mathrm{KB} \wedge E$ is satisfiable and (ii) $\mathrm{KB} \wedge E \models M$. An explanation E is *(subset-)minimal* if no other set $E' \subsetneq E$ is an explanation of I.

The problem $ABD(\Gamma)$ asks whether there is an explanation, which in the decision context is the same as asking whether there is a minimal explanation. If Γ is arbitrary, we omit Γ from the problem and write ABD. Note that the complexity of $ABD(\Gamma)$ is completely determined (Nordh and Zanuttini 2008) and illustrated in Figure 1b.

We write $\mathcal{E}(I)$ to refer to the set of all explanations and $\mathcal{E}_M(I)$ for the set of all subset-minimal explanations. A variable $x \in H$ is *relevant* if x belongs to some subset-minimal explanation $E \in \mathcal{E}_M(I)$ and *necessary* if x belongs to all subset-minimal explanations $E \in \mathcal{E}_M(I)$. We abbreviate the sets of all relevant and necessary variables by $\mathcal{R}\mathrm{el}\mathcal{E}(I)$ and $\mathcal{N}\mathrm{ec}\mathcal{E}(I)$, respectively.

Example 2.3.
Consider our abduction example from the introduction, which we slightly extend. Therefore, let the ABD instance $I = (\mathrm{KB}, H, M)$ *consists of the knowledge base*

$$\mathrm{KB} = \{\underline{\boldsymbol{w}}\boldsymbol{ednesday} \rightarrow \underline{\boldsymbol{r}}\boldsymbol{aining},\ \underline{\boldsymbol{w}}\boldsymbol{ednesday} \wedge \underline{\boldsymbol{c}}\boldsymbol{alm} \rightarrow \underline{\boldsymbol{n}}\boldsymbol{o\text{-}race},$$

$$\underline{\boldsymbol{w}}\boldsymbol{ednesday} \wedge \underline{\boldsymbol{s}}\boldsymbol{torm} \rightarrow \underline{\boldsymbol{n}}\boldsymbol{o\text{-}race}\},$$

the manifestation n, *and the hypothesis* $\{w, c, s, r\}$.

The set of all explanations $\mathcal{E}(I)$ *consists of the explanations* $\{w, c\}$, $\{w, c, s\}$, $\{w, c, r\}$, $\{w, s\}$, $\{w, s, c\}$, $\{w, s, r\}$, *and* $\{w, s, c, r\}$. *The explanations* $\{w, c\}$ *and* $\{w, s\}$ *are subset-minimal and hence constitute the set* $\mathcal{E}_M(I)$. *When considering the elements of* $\mathcal{E}_M(I)$, *we immediately observe that the set of relevant propositions* $\mathcal{R}\mathrm{el}\mathcal{E}(I)$ *is formed of* w, c, *and* s. *Whereas the only element in the necessary propositions* $\mathcal{N}\mathrm{ec}\mathcal{E}(I)$ *is* w.

3 Facets in explanations

In non-monotonic reasoning, we commonly consider knowledge bases with multiple possible solutions, each leading to different conclusions. Central decision-based reasoning

https://doi.org/10.1017/S1471068425100215 Published online by Cambridge University Press

problems consider all possible solutions and consider how a variable relates to all solutions. When asking for brave (credulous) or cautious (skeptical) reasoning, we decide whether a variable belongs to one solution or all solutions, respectively. The underlying idea is that brave reasoning allows for multiple potential conclusions from a knowledge base, that is, a knowledge base may have uncertain outcomes. In contrast, skeptical reasoning requires a guaranteed outcome. This concept is also known in abductive reasoning with relevant and necessary explanations and has been considered in the literature, for example (Friedrich *et al.* 1990; Eiter and Gottlob 1995; Eiter *et al.* 1997; Zanuttini 2003; Nordh and Zanuttini 2008). However, a detailed complexity classification in Post's lattice is open to date. In this section, we consider reasoning between relevant and necessary explanations, so-called facets. Intuitively, a variable p is a facet if it is (i) part of *some* explanation (relevant), but (ii) not included in *every* explanation (dispensible). We start by defining facets formally.

Definition 3.1 (Facets).
Let $I = (\mathrm{KB}, H, M)$ be an ABD instance. A variable $x \in H$ is a facet in the instance I if $x \in \mathcal{R}\mathrm{el}\mathcal{E}(I) \setminus \mathcal{N}\mathrm{ec}\mathcal{E}(I)$.

Based on this definition, we define a computational problem whose task is to decide whether a given variable is a facet or not.

IsFacet(Γ)
Given: $I = (\mathrm{KB}, H, M, x)$ where (KB, H, M) is an ABD(Γ) instance and $x \in H$.
Task: Is x a facet in I?

The following example illustrates facets in the sailing scenario.

Example 3.2 (Cont.).
*Consider our ABD instance from Example 2.3. Since $\mathcal{R}\mathrm{el}\mathcal{E}(I) = \{\underline{\boldsymbol{w}}\boldsymbol{ednesday}, \underline{\boldsymbol{c}}\boldsymbol{alm}, \underline{\boldsymbol{s}}\boldsymbol{torm}\}$ and $\mathcal{N}\mathrm{ec}\mathcal{E}(I) = \{\underline{\boldsymbol{w}}\boldsymbol{ednesday}\}$, we observe that c and s are facets allowing for a variability in explanations whereas **w̲ednesday** occurs in all explanations and is thus not a facet. Note that without minimality in the relevance definition **r̲aining** would (contrary to intuition) qualify as relevant and also as a facet.*

The dispensability condition can be checked fairly easy in many cases. To this end, we test, given (KB, H, M, x), whether $(\mathrm{KB}, H \setminus \{x\}, M)$ admits an explanation or not. In particular, if $ABD(\Gamma)$ is in P, we can check this in polynomial time. Thus, the interesting computational aspect of IsFacet(Γ) is to decide when *both* relevance and dispensability can be decided without a major blow up in complexity. The core parts of our proofs give an immediate classification for relevance as well. Overall, we obtain an almost complete classification of IsFacet(Γ).

Theorem 3.3.
The classification of IsFacet(Γ) *in Figure 1a is correct.*

Before we prove Theorem 3.3 systematically, we observe that for most complexity questions of IsFacet(Γ) it is sufficient to consider the efpp-closure $\langle\Gamma\rangle_{\neq}$ of Γ.

Lemma 3.4.
Let Γ and Γ' be two constraint languages. If $\Gamma' \subseteq \langle\Gamma\rangle_{\neq}$, then IsFacet($\Gamma'$) $\leq_m^{\mathrm{P}}$ IsFacet(Γ).

https://doi.org/10.1017/S1471068425100215 Published online by Cambridge University Press

Proof (Idea).
We omit details, since the construction is exactly the same as in previous work ((Nordh and Zanuttini 2008), Lemma 22), but the basic idea is simply to replace each relation by the set of constraints prescribed by the efpp-definition, and introducing fresh variables (kept outside the hypothesis) for any existentially quantified variables. This exactly preserves the set of (minimal) explanations. □

Lifting Lemma 3.4 to pp-definability does not appear to be possible in general: the classical trick when encountering an equality constraint $(x = y)$ is to identify the two variables throughout the instance. But consider, for example, an instance with knowledge base $\{(x = y), (x \rightarrow m), (y \rightarrow m)\}$, $H = \{x, y\}$, and $M = \{m\}$. Then, x and y are both facets since $\{x\}$ and $\{y\}$ are both minimal explanations, but if we identify y with x and remove the equality constraint, we obtain the instance $\{x \rightarrow m\}$ where x is *not* a facet, since there is only one minimal explanation $\{x\}$. However, the loss of the equality relation in the efpp-closure $\langle\Gamma\rangle_{\neq}$ turns out to be manageable. We explain, in the proof of Theorem 3.3, why our results in the next two sections extend to all co-clones. To obtain the systematic cases, we require numerous lemmas, which we establish in the following.

3.1 Computational upper bounds

Recall from Figure 1b that $ABD(\Gamma)$ is always either in (i) P, (ii) (co)NP, or (iii) Σ_2^P. Hence, our first task is to identify the corresponding classes for the IsFacet(Γ) problem. Ideally, one could hope that IsFacet(Γ) can be solved without a large increase in complexity, for example, going from P to being NP-hard. We will see that this can often, but not always, be achieved. We begin by analyzing the simple language $\{x \rightarrow y\}$ where the only allowed constraint is an implication between two variables. From Figure 1b, we know that $ABD(\{x \rightarrow y\})$ is in P, and this can be extended to IsFacet($\{x \rightarrow y\}$) via a more involved algorithm.

Lemma 3.5.
IsFacet($\{x \rightarrow y\}$) $\in P$.

Proof.
Let (KB, H, M, x) be an instance of IsFacet($\{x \rightarrow y\}$), that is, KB only consists of implications. Note that KB is 1-valid. Consequently, there is an explanation if and only if H is an explanation. To see this, we observe that H is always consistent with KB. Thus, it has "maximal" entailment power. To explain a single $m \in M$, a single $h \in H$ is always sufficient. For $m \in M$, we denote by $h(m) = \{h \in H \mid \mathrm{KB} \land h \models m\}$, that is, all hypotheses from H that *explain* m.

Now, we observe that $h(m)$ can be computed in polynomial time, for each $m \in M$. Denote by $M_x \subseteq M$ the manifestations from M that are explained by x alone, that is, $M_x = \{m \in M \mid \mathrm{KB} \land x \models m\}$. The set M_x can also be computed in polynomial time by repeatedly checking whether $\mathrm{KB} \land x \models m$. Since M_x is explained by x, we make x "relevant" by finding an $E \subseteq H \setminus \{x\}$ that avoids explaining at least one $m \in M_x$. A maximal candidate for this is $H \setminus h(m)$. We can accomplish this as follows:

https://doi.org/10.1017/S1471068425100215 Published online by Cambridge University Press

```
E ← 'none'
for m ∈ M_x do
    if KB ∧ H \ h(m) ⊨ M \ M_x then
        E ← H \ h(m) # candidate found
    end if
end for
if E = 'none' then
    return False # x can not be made relevant
end if
return KB ∧ H \ {x} ⊨ M # is there an explanation without x?
```

This runs in p-time, since entailment for $SAT(x \to y, x, \bar{x})$ is in p-time (Schaefer 1978). □

We continue with dualHorn where ABD is also in P. Here, membership in P for IsFacet(dualHorn) is less obvious. Given that ABD(Horn) is NP-complete, we could suspect that checking for relevance and dispensability is computationally more expensive. First, we need the following technical lemma, where we recall that dualHorn⁻ = dualHorn $\setminus \{x, \neg x\}$ is the set obtained from dualHorn by removing the two unit clauses.

Lemma 3.6 (⋆[1]).
IsFacet(*dualHorn*) $\leq_m^{\mathrm{P}}$ IsFacet(*dualHorn⁻*).

Next, we show that the result for IsFacet($\{x \to y\}$) can be extended to IsFacet(dualHorn⁻), and, thus, also to IsFacet(dualHorn) via Lemma 3.6.

Lemma 3.7 (⋆).
IsFacet(dualHorn⁻) ∈ P.

Our second major tractability result concerns 2-affine, that is, either unit clauses or relations definable by $(x \oplus y = 0)$ (equality) or $(x \oplus y = 1)$ (inequality).

Lemma 3.8.
IsFacet(2-affine) ∈ P.

Proof.
Let (KB, H, M, x) be an instance of IsFacet(2-affine). We assume that each relation in KB is represented by precisely one linear equation of arity at most 2, see (Creignou *et al.* 2011) and (Mahmood *et al.* 2021). First, if KB is not satisfiable we answer no. Second, we propagate all unit clauses as in Lemma 3.6. Each remaining equation then expresses either equality or inequality between two variables. With the transitivity of the equality relation and the fact that in the Boolean case $a \neq b \neq c$ implies $a = c$, we can identify equivalence classes of variables such that each two classes are either independent or they must have contrary truth values. We call a pair of dependent equivalence classes (X, Y) a *cluster*, that is X and Y must take contrary truth values. Denote by $X_1, \ldots, X_p$ the equivalence classes that contain variables from M such that $X_i \cap M \neq \emptyset$. Denote by $Y_1, \ldots, Y_p$ the equivalence classes such that for each i the pair (X_i, Y_i) represents a cluster. We make the following stepwise observations: (1) there is an explanation if and only if $H \cap X_i \neq \emptyset$ for every $1 \leq i \leq p$, (2) a subset-minimal explanation is constructed by taking exactly one

[1] We prove statements marked by "⋆" in the supplemental material.

https://doi.org/10.1017/S1471068425100215 Published online by Cambridge University Press

representative from each X_i. (3) x can be made relevant if $x \in X_i$, for some i. (4) x is a facet if additionally each X_i contains at least one representative different from x. These checks can be done in polynomial time. We conclude that IsFacet(2-affine) $\in$ P. □

We continue with the corresponding membership questions for complexity classes above P. Here, we make a case distinction of whether the underlying satisfiability problem is in P, and in particular whether $SAT(\Gamma^+)$ is in P. We begin with the following lemma.

Lemma 3.9 (⋆).
If $SAT(\Gamma^+) \in \mathrm{P}$, *then there is a polynomial time algorithm to determine whether a given* $E \subseteq H$ *is an explanation for a given abduction instance* (KB, H, M).

In particular, we obtain the following general statement, which shows that the complexity of IsFacet(Γ) for many natural cases does not jump to Σ_2^{P}.

Lemma 3.10 (⋆).
For any constraint language Γ, $SAT(\Gamma^+) \in \mathrm{P} \Rightarrow$ IsFacet$(\Gamma) \in \mathrm{NP}$.

This covers a substantial number of cases since $SAT(\Gamma^+)$ is in P when Γ is *Schaefer*, that is, contained in IV_2 (dualHorn), IE_2 (Horn), IL_2 (affine), or ID_2 (2-CNF). Our last major tractable case concerns the set of essentially negative clauses EN.

Lemma 3.11.
IsFacet(EN) $\in P$.

Proof.
We assume an arbitrary instance of IsFacet(EN): (KB, H, M, x). We first apply unit propagation, exactly as in Lemma 3.6. We can now assume that the instance only contains negative clauses of size ≥ 2 and equality clauses. We organize all variables which are equal to each other into equivalence classes as in Lemma 3.8, with the exception that all classes are independent in this case. If some equivalence class that contains an $m_i \in M$ does not contain variables from H, this m_i cannot be entailed. Thus, the abduction problem has no solutions and no facets.

Otherwise, if all classes that contain an $m_i \in M$ contain at least one variable from H, we set all variables in these classes to true and check if this is consistent with KB. This, guarantees the existence of abduction solutions. If this is the case, we can check if x is a facet. For this, we need two conditions: let $x \in C$ where C is an equivalence class. First, there must be at least one manifestation $m_i \in C$, else x cannot imply any m_i and thus is never needed in a subset-minimal solution (x would not be relevant). Second, there must be at least one variable $x_i \in C$ different from x, otherwise x will always be needed to explain m_i and there can be no solution without it (x would be necessary). □

3.2 Computational lower bounds

We begin with a general result that implies that the facet problem is always at least as hard as the underlying abduction problem, provided the Boolean equality relation can be expressed.

Lemma 3.12.
$ABD(\Gamma) \leq_m^{\mathrm{P}}$ IsFacet(Γ) *if* $(x = y) \in \Gamma$ *for any constraint language* Γ.

https://doi.org/10.1017/S1471068425100215 Published online by Cambridge University Press

Proof.
Given an instance (KB, H, M) of $ABD(\Gamma)$ we let x, y, m be fresh variables. We define the instance $(\mathrm{KB}', H', M', x)$ of IsFacet(Γ) as $\mathrm{KB}' = \mathrm{KB} \cup \{(x=m), (y=m)\}$, $H' = H \cup \{x, y\}$, $M' = M \cup \{m\}$. We claim that (KB, H, M) admits an explanation if and only if x is a facet in (KB', H', M'). ("$\Rightarrow$"): assume that $E \subseteq H$ is a subset-minimal explanation. Then, $E \cup \{x\}$ and $E \cup \{y\}$ are both subset-minimal explanations in (KB', H', M'), so x is a facet. ("$\Leftarrow$"): assume that x is a facet in (KB', H', M'). Then, there exists a subset minimal explanation $E' \subseteq H'$ where $x \in E'$, and it follows that $E' \setminus \{x\}$ is a (subset-minimal) explanation for (KB, H, M). □

By combining this with Lemmas 3.4 and 3.13, we inherit all hardness results from ABD. All non-polynomial cases of $ABD(\Gamma)$ satisfy $\Gamma \not\subseteq \mathsf{IS}_{12}$ and $\Gamma \not\subseteq \mathsf{IS}_{02}$, that is, such languages are not essentially negative and not essentially positive (Nordh and Zanuttini 2008).

Lemma 3.13.
((Mahmood et al. 2021), Lemma 9) *Let* Γ *be a constraint language. If* $\Gamma \not\subseteq \mathsf{IS}_{12}$ *and* $\Gamma \not\subseteq \mathsf{IS}_{02}$, *then* $(x = y) \in \langle\Gamma\rangle_{\neq}$ *and* $\langle\Gamma\rangle = \langle\Gamma\rangle_{\neq}$.

However, the facet problem IsFacet is generally even harder than ABD. We present a technical lemma, providing us the unit clause (x) for free. Then, we will present languages where ABD is polynomial and IsFacet is NP-hard, as well as languages where ABD is coNP-complete, while IsFacet is Σ_2^P-complete.

Lemma 3.14 (⋆).
For any constraint language Γ, it holds that IsFacet$(\Gamma \cup \{(x)\}) \leq_m^{\mathrm{P}}$ IsFacet(Γ).

We are now ready to state a crucial reduction, which is at the heart of the increased complexity of IsFacet *versus* ABD. The IsFacet-problem allows to simulate negative unit clauses, provided that the language is 1-valid and can express implication $(x \to y)$.

Lemma 3.15.
If Γ *is 1-valid, then* $ABD(\Gamma \cup \{(\neg x)\}) \leq_m^{\mathrm{P}}$ IsFacet$(\Gamma \cup \{x \to y\})$.

Proof.
Let (KB, H, M) be an instance of $ABD(\Gamma \cup \{(\neg x)\})$. If KB contains two unit clauses $\neg x$ and $\neg y$ for distinct variables x and y, we may simply identify x with y and obtain an equivalent instance. Thus, we may wlog assume that $\mathrm{KB} = \varphi \wedge (\neg z)$, $z \in \mathrm{var}(\varphi)$, for a Γ-formula φ. Let x, y, m denote fresh variables and define $V = \mathrm{var}(\varphi) \cup H \cup M \cup \{x, y, m\}$. We define the target instance $(\mathrm{KB}', H', M', x)$ as

$$\mathrm{KB}' = \varphi \wedge \bigwedge_{x_i \in V} (z \to x_i) \wedge (x \to m) \wedge (y \to m), \quad H' = H \cup \{x, y\}, \quad M' = M \cup \{m\}.$$

Note that KB' is a $\Gamma \cup \{x \to y\}$-formula, as required.

In the following, we prove correctness formally. Observe first that for $z = 0$, KB and φ have exactly the same models (upto the fresh variables x, y, m). For $z = 1$, φ may admit additional models. However, due to KB' containing the construct $\bigwedge_{x_i \in V}(z \to x_i)$, the only additional model is the all-1 model.

https://doi.org/10.1017/S1471068425100215 Published online by Cambridge University Press

Correctness:

("$\Rightarrow$"): Be $E \subseteq H$ an explanation for (KB, H, M). Then, with the above observation that the only additional model is the all-1 model (which satisfies M and m), it is easily observed that

1. $E' = E \cup \{x\}$ constitutes an explanation for (KB', H', M')
2. $E' \setminus \{x\}$ is *no* explanation for (KB', H', M')
3. there is an explanation without x, namely $E \cup \{y\}$

In summary, x is a facet.

("$\Leftarrow$"): Be x a facet for (KB', H', M'). Then there is a set $E' \subseteq H'$ such that

1. E' is an explanation for (KB', H', M')
2. $E' \setminus \{x\}$ is *no* explanation for (KB', H', M')
3. there is an explanation for (KB', H', M') without x

From the construction it is easily observed that E' must be of the form $E' = E \cup \{x\}$, for an $E \subseteq H$. Since $E' \setminus \{x\} = E$ fails as explanation for (KB', H', M'), and E is obviously consistent with (1-valid) KB', we conclude that E fails due to $\mathrm{KB}' \wedge E$ not entailing $M' = M \cup \{m\}$. That is,

$$\mathrm{KB}' \wedge E \not\models M \cup \{m\} \tag{1}$$

Further, since $E \cup \{x\}$ is an explanation, we know that $\mathrm{KB}' \wedge E \cup \{x\}$ *does* entail $M' = M \cup \{x\}$. Since by construction, x can not be responsible for entailing M, we conclude that $\mathrm{KB}' \wedge E$ entails M. That is,

$$\mathrm{KB}' \wedge E \models M \tag{2}$$

From (1) and (2) we conclude that that $\mathrm{KB}' \wedge E \not\models m$. From this we conclude that $\mathrm{KB}' \wedge E$ admits models where $z = 0$ (otherwise, $\mathrm{KB}' \wedge E$ would entail m, due to KB' containing $z \to m$). Therefore, we conclude that $\mathrm{KB}'[z=0] \wedge E$ admits models (is consistent) and entails M. Since $\mathrm{KB}'[z=0] \equiv \mathrm{KB}$ (upto the "irrelevant" variables x, y, m) we conclude that 1) $\mathrm{KB} \wedge E$ is consistent, and 2) $\mathrm{KB} \wedge E \models M$. That is, E is an explanation for (KB, H, M). □

We are now ready to derive the hardness results in a series of short, technical lemmas.

Lemma 3.16 (⋆).
If $\mathsf{II}_1 = \langle \Gamma \rangle$ *or* $\mathsf{IE}_1 = \langle \Gamma \rangle$, *then* $ABD(\Gamma \cup \{(\neg x)\}) \leq_m^{\mathrm{P}} \textsc{IsFacet}(\Gamma)$.

Lemma 3.17 (⋆).
If $\mathsf{IN} \subseteq \langle \Gamma \rangle$, *then* $\textsc{IsFacet}(\Gamma)$ *is* Σ_2^{P}*-hard.*

Lemma 3.18 (⋆).
If $\mathsf{IE} \subseteq \langle \Gamma \rangle$, *then* $\textsc{IsFacet}(\Gamma)$ *is* NP*-hard.*

We remark that Friedrich *et al.* (1990) prove NP-hardness for the relevance problem for IE_1. While this proof can be adapted to our setting it does *not* generalize to the other cases in this section. By combining all results, we obtain the main result of the paper.

https://doi.org/10.1017/S1471068425100215 Published online by Cambridge University Press

Proof of Theorem 3.3.
First, we observe that the each language Γ considered in Lemma 3.7, Lemma 3.8, or Lemma 3.11 either contains or can define equality $(x = y)$. Hence, Lemma 3.4 is applicable and proves tractability for any language Δ such that $\Delta \subseteq \langle\Gamma\rangle_{\neq} = \langle\Gamma\rangle$. This covers all tractable cases in Figure 1a.

For intractability, all NP-complete cases *except* IE_1 and IE follow from Lemma 3.10 (since $SAT(\Gamma^+)$ is in P for every such Γ), Lemma 3.12, and Lemma 3.13. The former two cases are instead proven to be NP-hard in Lemma 3.18, and inclusion in NP follows from Lemma 3.10. Last, Σ_2^{P}-hardness for all remaining cases follow from Lemma 3.17, and inclusion in Σ_2^{P} is straightforward via arguments similar to Lemma 3.10. □

We view the two missing cases IL (even linear equations) and IL_1 (even linear equations, and unit clauses) as interesting future research questions. However, via Lemma 3.10 we may at least observe that $\textsc{IsFacet}(\Gamma) \in \mathrm{NP}$ for any base Γ of IL or IL_1. Hence, the only question remaining is whether these problems are in P, NP-complete, or — unlikely but still possible — NP-intermediate.

We conclude this section with the observation that all membership and hardness proofs, which we have given for the IsFacet problem are also applicable to the relevance problem (deciding if $x \in H$ is relevant).

Theorem 3.19.
The classification of IsFacet(Γ) *in Figure 1a also describes the complexity of the relevance problem.*

Proof.
Recall that an instance of the relevance problem is given by $I = (\mathrm{KB}, H, M, x)$ and the question is whether $x \in \mathcal{R}\mathrm{el}\mathcal{E}(I)$, that is, whether x belongs to a subset-minimal explanation. We revisit now all membership and hardness proofs for IsFacet and observe that they can easily be adapted to the relevance problem.

Membership in P. First observe that the reduction of Lemma 3.6 to get rid of unit clauses can be performed analogously on the relevance problem. Next we observe that all algorithms showing P-membership, that is, Lemmas 3.5, 3.7, 3.8, and 3.11, first decide whether the given x is relevant, and then in a second (independent) step decide whether x is dispensable (not necessary). By dropping the dispensability check, we obtain a polynomial time algorithm to decide the relevance problem.

Membership in NP. Analogously to the P-membership algorithms we drop the step of the dispensability check: in Lemma 3.10 omit guessing an $E_2 \subseteq H \setminus \{x\}$ and verifying that E_2 is an explanation.

Membership in Σ_2^{P}. Inclusion in Σ_2^{P} is straightforward via arguments similar to Lemma 3.10 using an NP-oracle.

NP-*hardness and* Σ_2^{P}-*hardness*. First observe that Lemma 3.4 is also applicable to relevance, since the underlying reduction preserves the exact set of explanations. Next observe that the reduction from Lemma 3.12 carries over one-to-one to relevance. It is optional to simplify the proof by removing the clause $(y = m)$ and the variable y (whose only purpose was to assure that x is not necessary). Lemma 3.14 is easily observed to carry over one-to-one to relevance. Lemma 3.15 carries over one-to-one, again it is optional to

https://doi.org/10.1017/S1471068425100215 Published online by Cambridge University Press

simplify the proof by removing y and the clause $(y \rightarrow m)$. Finally, Lemmas 3.16, 3.17, and 3.18 hold analogously, since all used lemmas carry over, as shown above.

In summary, this theorem is proven analogously to Theorem 3.3. □

4 Diverse explanations

In the previous section on facets, we considered whether there exists a variable that is relevant in explanations but dispensable. Thereby, we obtain a notion on flexibility on one variable belonging to explanations. Now, we lift flexibility from one variable to a set of variables in explanations and ask whether there exist two explanations of sufficiently high *diversity*. It turns out that this metric can be precisely related to the existence of facets and several notions from our facet classification carry over. However, measuring the distance is provably much harder and becomes NP-hard already for the small fragment of Horn consisting of a single implication $(x \rightarrow y)$. Before, we illustrate our complexity results in detail, we define our distance measure and computational problem.

Definition 4.1.
Let $I = (\mathrm{KB}, H, M)$ be an ABD instance, and $E_1 \subseteq H$ and $E_2 \subseteq H$ be two sets of variables over the hypotheses H. Then, the distance $d(E_1, E_2)$ between E_1 and E_2 is the cardinality of their symmetric difference. More formally,

$$\begin{aligned} d(E_1, E_2) &:= |\, E_1 \triangle E_2 \,| = |\, (E_1 \cup E_2) \setminus (E_1 \cap E_2) \,| \\ &= |\, \{x \in H \mid x \in E_1 \text{ and } x \notin E_2 \text{ or } x \notin E_1 \text{ and } x \in E_2\} \,|. \end{aligned}$$

If E_1 and E_2 are explanations, i.e., $E_1, E_2 \in \mathcal{E}(I)$, and $d(E_1, E_2) \geq k$, then we call E_1 and E_2 k-diverse explanations.

Note that the maximum distance is $|H|$, which is reached by $d(H, \emptyset)$. Our notion of distance is in line with the corresponding notion for $SAT(\Gamma)$ (Misra *et al.* 2024) and many other diversity problems studied in AI. Note that we do *not* require that the two explanations are minimal, since the distance notion does not require this.

Next, we define the *diversity problem for abduction*.

Div-ABD(Γ)
Given: An ABD(Γ) instance $I = (\mathrm{KB}, H, M)$ and $k \geq 0$.
Task: Does I have two k-diverse explanations E_1 and E_2?

We have the following relationship to facets.

Proposition 4.2.
Let $I = (\mathrm{KB}, H, M)$ be an instance of ABD(Γ) and $E_1, E_2 \in \mathcal{E}_M(I)$. Then, every $x \in E_1 \triangle E_2$ is a facet.

Proof.
Let $x \in E_1 \triangle E_2 = (E_1 \cup E_2) \setminus (E_1 \cap E_2)$. We observe that if x is *not* a facet then either (1) $x \in E_1 \cap E_2$ since it is part of *every* explanation, or (2) $x \notin E_1 \cup E_2$ since it is not included in *any* subset-minimal explanation. Hence, $x \notin E_1 \triangle E_2$, meaning that $x \in E_1 \triangle E_2$ is only possible if x is a facet. □

https://doi.org/10.1017/S1471068425100215 Published online by Cambridge University Press

From the relationship between facets and the distance notion, which we establish in Proposition 4.2, we can suspect similarities between the computational problems IsFacet(Γ) and Div−abd(Γ). First, we establish that all hardness results are inherited from ABD, analogously to Lemma 3.12 and Lemma 3.4 for IsFacet(Γ).

Lemma 4.3 (⋆).
$ABD(\Gamma) \leq_m^{\mathrm{P}} \text{Div-ABD}(\Gamma)$ *if* $(x = y) \in \Gamma$.

Also, Σ_2^{P}-hardness for 1-valid and complementive languages is obtained analogously.

Lemma 4.4 (⋆).
If $\mathsf{IN} \subseteq \langle\Gamma\rangle$, *then* Div-ABD(Γ) *is* Σ_2^{P}*-hard.*

However, the problem Div-ABD is generally harder than IsFacet. Below, we establish NP-hardness for simple implicative languages. We reduce from the problem Div-Pos2SAT where we are given a positive 2-CNF formula φ and an integer k. Therefore, we require two models of Hamming distance at least k, which is NP-hard (Misra *et al.* 2024).

Lemma 4.5 (⋆).
$\text{Div-Pos2SAT} \leq_m^{\mathrm{P}} \text{Div-ABD}(\{x \rightarrow y\})$.

Thus, despite the similarities between IsFacet(Γ) and Div-ABD(Γ), the latter seems to be significantly harder. However, we observe two tractable cases.

Lemma 4.6 (⋆).
Div-ABD(2-affine), Div-ABD(EP) $\in$ P.

5 Conclusion

In this paper, we introduce faceted reasoning to propositional abduction. We illustrate that this reasoning allows more fine-grained decisions than previously explored notions such as relevance and necessity/dispensability. We relate facets to the problem of finding diverse explanations. We establish an almost complete complexity classification in Post's lattice. In many cases, facets can be found without a major blow-up in complexity. This is particularly interesting, given that *counting* minimal explanations is almost always substantially harder (Hermann and Pichler 2010). Our facet classification also implies a corresponding classification for the relevance problem, thus answering an open question (Zanuttini 2003; Nordh and Zanuttini 2008). For diversity, our results are less conclusive, but any major tractable cases seem unlikely, since it is hard already for the fragment $(x \rightarrow y)$.

5.1 Completing the trichotomy

The two open cases, affine equations of odd length with, or without, unit clauses, could be interesting to resolve. It seems unlikely that IsFacet(Γ) could be tractable for such languages, but, at the same time, any hardness proof likely needs to involve significant new ideas. Note that affine languages were also absent in earlier counting complexity classifications (Hermann and Pichler 2010). Hence, there is a blind spot for complexity of abduction. One possible way forward could be to first classify the complexity of Div-ABD(Γ) for all affine languages, which seems likely to be hard.

5.2 Parameterized complexity and diversity

It is reasonable that Div-ABD(Γ) can be fully classified, but without any large, tractable cases. However, more fine-grained investigations involving problem structure (parameter) remains interesting. For Div-ABD(Γ) a natural parameter is the maximum allowed distance. A systematic classification could not only open up new efficiently solvable cases but could also prove to be a useful framework for proving hardness for other types of diversity problems, especially, since the classical complexity of abduction is much richer than the one for SAT.

5.3 Abductive logic programming (ALP)

Since detailed complexity results on logic programs (Truszczyński 2011) and results on the complexity of ALP regarding consistency, relevance, and necessity (Eiter *et al.* 1995) exist, it could be interesting to extend our results to stable models in ALP where the input is given in form of rules.

5.4 Applications

Faceted reasoning can aid the search for heterogeneous explanations, which could be valuable in any domain with classical applications of abduction, for example, diagnosis, and explainable AI. More concretely, a practical application of facets are logistics applications where solutions need to be explained such as in the Beluga AI Competition (Gnad *et al.* 2025). There, several tasks aim at flexibility in explanations or alternatives. Finally, we hope that our results on diverse explanations will spark interest into a deeper complexity study and exploration to use these for tasks like model debugging or decision support. Beyond abductive reasoning, we expect that facets and diversity could be interesting for epistemic logic programs (Eiter *et al.* 2024), default logic (Fichte *et al.* 2024), and probabilistic reasoning (Fichte *et al.* 2022b).

Acknowledgments

The work has received funding from the Swedish research council under grant VR-2022-03214 and ELLIIT funded by the Swedish government.

Supplementary material

The supplementary material for this article can be found at https://doi.org/10.1017/S1471068425100215.

References

Alrabbaa, C., Rudolph, S. and Schweizer, L. 2018. Faceted answer-set navigation. In *Proc. RuleML+RR'18*. Springer, 211–225.

Arora, S. and Barak, B. 2009. *Computational Complexity - A Modern Approach*. Cambridge University Press.

Böhl, E., Gaggl, S. A. and Rusovac, D. 2023. Representative answer sets: Collecting something of everything. In *ECAI*. IOS Press, 271–278.

Böhler, E., Reith, S., Schnoor, H. and Vollmer, H. 2005. Bases for Boolean co-clones. *Information Processing Letters* 96, 2, 59–66.

Bylander, T., Allemang, D., Tanner, M. C. and Josephson, J. R. 1991. The computational complexity of abduction. *Artificial Intelligence* 49, 1, 25–60.

Cook, S. A. 1971. The complexity of theorem-proving procedures. In *STOC 1971*. ACM, 151–158.

Creignou, N., Kröll, M., Pichler, R., Skritek, S. and Vollmer, H. 2019. A complexity theory for hard enumeration problems. *Discrete Applied Mathematics* 268, 191–209.

Creignou, N., Olive, F. and Schmidt, J. 2011. Enumerating all solutions of a Boolean CSP by non-decreasing weight. In *Proc. SAT*. Springer, Vol. 6695, 120–133.

Creignou, N., Schmidt, J. and Thomas, M. 2010. Complexity of propositional abduction for restricted sets of Boolean functions. In *Proc. KR*. AAAI Press, 8–16.

Creignou, N. and Vollmer, H. 2008. *Boolean Constraint Satisfaction Problems: When Does Post's Lattice Help?*. Springer, 3–37.

Creignou, N. and Zanuttini, B. 2006. A complete classification of the complexity of propositional abduction. *SIAM Journal on Computing* 36, 1, 207–229.

Dai, W.-Z., Xu, Q., Yu, Y. and Zhou, Z.-H. 2019. Bridging machine learning and logical reasoning by abductive learning. In *Proc. NeurIPS 2019*. Curran Associates, Inc, Vol, 32.

Dellsén, F. 2024. Abductive reasoning in science. In *Elements in the Philosophy of Science*. Cambridge University Press.

Eiter, T., Fichte, J. K., Hecher, M. and Woltran, S. 2024. Epistemic logic programs: Non-ground and counting complexity. In *Proc. IJCAI*. ijcai.org, 3333–3341.

Eiter, T. and Geibinger, T. 2023. Explaining answer-set programs with abstract constraint atoms. In *Proc. IJCAI*. International Joint Conferences on Artificial Intelligence Organization, 3193–3202.

Eiter, T. and Gottlob, G. 1995. The complexity of logic-based abduction. *Journal of the ACM* 42, 1, 3–42.

Eiter, T., Gottlob, G. and Leone, N. 1995. Complexity results for abductive logic programming. In *Proc. LPNMR*. Springer, 1–14.

Eiter, T., Gottlob, G. and Leone, N. 1997. Semantics and complexity of abduction from default theories. *Artificial Intelligence* 90, 1, 177–223.

Fellows, M. R., Pfandler, A., Rosamond, F. A. and Rümmele, S. 2012. The parameterized complexity of abduction. In *Proc. AAAI*. AAAI Press, 743–749.

Fichte, J. K., Fröhlich, N., Hecher, M., Lagerkvist, V., Mahmood, Y., Meier, A. and Persson, J. 2025. Facets in argumentation: A formal approach to argument significance. In *Proc. IJCAI 2025*. ijcai.org, In press.

Fichte, J. K., Gaggl, S. A. and Rusovac, D. 2022a. Rushing and strolling among answer sets – navigation made easy. In *Proc. AAAI*. AAAI Press, 5651–5659.

Fichte, J. K., Hecher, M. and Nadeem, M. A. 2022b. Plausibility reasoning via projected answer set counting - A hybrid approach. In *Proc. IJCAI*. ijcai.org, 2620–2626.

Fichte, J. K., Meier, A. and Schindler, I. 2024. Strong backdoors for default logic. *ACM Transactions on Computational Logic* 25, 3, 1–24.

Friedrich, G., Gottlob, G. and Nejdl, W. 1990. Hypothesis classification, abductive diagnosis and therapy. In *ESE 1990*. Springer, 69–78.

Gnad, D., Hecher, M., Gaggl, S., Rusovac, D., Speck, D. J. and Fichte, J. K. 2025. Interactive exploration of plan spaces. In *Proc. KR 2025*. To appear. IJCAI Organization, In press.

Hermann, M. and Pichler, R. 2010. Counting complexity of propositional abduction. *Journal of Computer and System Sciences* 76, 7, 634–649.

Hu, W.-C., Dai, W.-Z., Jiang, Y. and Zhou, Z.-H. 2025. Efficient rectification of neuro-symbolic reasoning inconsistencies by abductive reflection. In *Proc. AAAI 2025*, 17333–17341.

Ignatiev, A., Narodytska, N. and Marques-Silva, J. 2019. Abduction-based explanations for machine learning models. In *Proc. AAAI 2019*. AAAI Press, 1511–1519.

Kakas, A. C., Kowalski, R. A. and Toni, F. 1992. Abductive logic programming. *Journal of Logic and Computation* 2, 6, 719–770.

Kleine Büning, H. and Lettmann, T. 1999. *Propositional Logic – Deduction and Algorithms*, Vol. 48 of Cambridge tracts in theoretical computer science. Cambridge University Press.

Lewis, H. 1979. Satisfiability problems for propositional calculi. *Mathematical Systems Theory* 13, 45–53.

Mahmood, Y., Meier, A. and Schmidt, J. 2021. Parameterized complexity of abduction in Schaefer's framework. *Journal of Logic and Computation* 31, 1, 266–296.

Miller, T. 2019. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence* 267, 1–38.

Minsky, M. 1974. A framework for representing knowledge. Technical report, Massachusetts Institute of Technology, USA. https://dspace.mit.edu/bitstream/handle/1721.1/6089/AIM-306.pdf?sequence=2&isAllowed=y.

Misra, N., Mittal, H. and Rai, A. 2024. On the parameterized complexity of diverse SAT. In *ISAAC 2024*. Dagstuhl, 50:1–50:18.

Nordh, G. and Zanuttini, B. 2008. What makes propositional abduction tractable. *Artificial Intelligence* 172, 10, 1245–1284.

Papadimitriou, C. H. 1994. *Computational Complexity*. Addison-Wesley.

Pfandler, A., Rümmele, S. and Szeider, S. 2013. Backdoors to abduction. In *Proc. IJCAI 2013*. IJCAI/AAAI, 1046–1052.

Post, E. 1941. The two-valued iterative systems of mathematical logic. *Annals of Mathematics Studies* 5, 1–122.

Rusovac, D., Hecher, M., Gebser, M., Gaggl, S. A. and Fichte, J. K. 2024. Navigating and Querying Answer Sets: How Hard Is It Really and Why? In *Proc. KR 2024*. IJCAI Organization, 642–653.

Schaefer, T. J. 1978. The complexity of satisfiability problems. In *Proc. STOC 1978*. ACM, 216–226.

Selman, B. and Levesque, H. J. 1990. Abductive and default reasoning: A computational core. In *Proc. AAAI 1990*. AAAI Press / The MIT Press, 343–348.

Speck, D., Hecher, M., Gnad, D., Fichte, J. K. and Corrêa, A. B. 2025. Counting and reasoning with plans. In *Proc. AAAI 2025*. AAAI Press, 26688–26696.

Stockmeyer, L. J. 1976. The polynomial-time hierarchy. *Theoretical Computer Science* 3, 1, 1–22.

Stockmeyer, L. J. and Meyer, A. R. 1973. Word problems requiring exponential time: Preliminary report. In *STOC 1973*. ACM, 1–9.

Truszczyński, M. 2011. Trichotomy and dichotomy results on the complexity of reasoning with disjunctive logic programs. *Theory and Practice of Logic Programming* 11, 6, 881–904.

Wrathall, C. 1976. Complete sets and the polynomial-time hierarchy. *Theoretical Computer Science* 3, 1, 23–33.

Yu, J., Ignatiev, A., Stuckey, P. J., Narodytska, N. and Marques-Silva, J. 2023. Eliminating the impossible, whatever remains must be true: On extracting and applying background knowledge in the context of formal explanations. In *Proc. AAAI 2023*. AAAI Press, 4123–4131.

Zanuttini, B. 2003. New polynomial classes for logic-based abduction. *Journal of Artificial Intelligence Research* 19, 1–10.