Explainability via Short Formulas: the Case of Propositional Logic with Implementation

REIJO JAAKKOLA, Mathematics Research Centre, Tampere University, Finland
TOMI JANHUNEN, Mathematics Research Centre, Tampere University, Finland
ANTTI KUUSISTO*, Mathematics Research Centre, Tampere University, Finland
MASOOD FEYZBAKHSH RANKOOH, University of Helsinki, Finland
MIIKKA VILANDER, Mathematics Research Centre, Tampere University, Finland

We conceptualize explainability in terms of logic and formula size, giving a number of related definitions of explainability in a very general setting. Our main interest is the so-called local explanation problem which aims to explain the truth value of an input formula in an input model. The explanation is a formula of minimal size that (1) obtains the same truth value as the input formula on the input model and (2) transmits that truth value to the input formula globally, i.e., on every model. As an important example case, we study propositional logic in this setting and show that the local explainability problem is complete for the second level of the polynomial hierarchy. The hardness result holds already for DNF-formulas. We also give parameterized versions of these problems leading to NP-completeness. The generality of our definitions allows us to lift complexity results also, e.g., to S5 modal logic and ensembles of decision trees. We also provide an implementation in answer set programming and investigate its capacity in relation to explaining answers to the n-queens and dominating set problems. Furthermore, we give an example of explaining the behavior of a black-box classifier.

JAIR Associate Editor: Tommy Meyer

JAIR Reference Format:
Reijo Jaakkola, Tomi Janhunen, Antti Kuusisto, Masood Feyzbakhsh Rankooh, and Miikka Vilander. 2025. Explainability via Short Formulas: the Case of Propositional Logic with Implementation. *Journal of Artificial Intelligence Research* 83, Article 8 (July 2025), 27 pages. DOI: 10.1613/jair.1.17422

1 Introduction

This paper investigates explainability in a general setting, with a goal of generalizing the most common approaches in the literature. The key in our approach is to relate explainability to formula size. We define several problems related to explainability for an arbitrary logic and show that these definitions instantiate well to the case of propositional logic. We differentiate between the *global* and *local explanation problems*. The global explanation problem for a logic L takes as input a formula $\varphi \in \mathrm{L}$ and outputs an equivalent formula of minimal size. Thus the objective is to explain the global behaviour of φ. For example, we can consider $\neg\neg\neg\neg\neg\neg p$, where the number of negations is even, to be globally explained by p. In contrast, the goal of the local explanation problem is to

*Corresponding Author.

Authors' Contact Information: Reijo Jaakkola, ORCID: https://orcid.org/0000-0003-4714-4637, jaakkolareijo@hotmail.com, Mathematics Research Centre, Tampere University, Tampere, Finland; Tomi Janhunen, ORCID: https://orcid.org/0000-0002-2029-7708, tomi.janhunen@tuni.fi, Mathematics Research Centre, Tampere University, Tampere, Finland; Antti Kuusisto, ORCID: https://orcid.org/0000-0003-1356-8749, antti.kuusisto@tuni.fi, Mathematics Research Centre, Tampere University, Tampere, Finland; Masood Feyzbakhsh Rankooh, ORCID: https://orcid.org/0000-0001-5660-3052, masood.feyzbakhshrankooh@helsinki.fi, University of Helsinki, Helsinki, Finland; Miikka Vilander, ORCID: https://orcid.org/0000-0002-7301-939X, miikka.vilander@tuni.fi, Mathematics Research Centre, Tampere University, Tampere, Finland.

DOI: 10.1613/jair.1.17422

explicate why an input formula ψ gets the truth value b in an input model (or instance) M. Given a tuple (M, ψ, b), the problem outputs a formula χ of minimal size such that (1) the formula χ obtains the same truth value b on M, and (2) on every model M' where χ gets the truth value b, also ψ gets that same truth value. Intuitively, this second condition states that the given truth value b of χ on a model M' causes ψ to be judged similarly. In summary, the local explanation problem gives reasons why a piece of data (or a model) is treated in a given way (i.e., obtains a given truth value) by a classifier (or a formula).

The two explanation problems are search problems and naturally give rise to corresponding decision problems we call explainability problems. The *global explainability problem* for a logic L asks, given a formula $\varphi \in$ L and $k \in \mathbb{N}$, whether there exists a formula equivalent to ψ of size at most k. The *local explainability problem* gets as input a tuple (M, ψ, b, k), and the task is then to check whether there exists a formula χ of size at most k satisfying the above conditions (1) and (2) of the local explanation problem. In addition to these two decision problems and the two search problems above, we also define some further generalizations allowing for uncertainty, approximation and explaining with a different logic than that of the input.

As an important particular case, we study the local explainability problem of propositional logic (PL) in detail. We show that for PL in particular, it suffices to consider local explanations which are (De-Morganizations of) subconjunctions of the input assignment. This coincides with many approaches in the literature, where subsets of the input assignment are the objects that are sought as explanations to begin with.

We prove that the local explainability problem for PL is Σ_2^p-complete. This is an important result whose usefulness lies in its implications that go far beyond PL itself. Indeed, the result gives a robust lower bound for various logics. We demonstrate this by establishing Σ_2^p-completeness of the local explainability problem of S5 modal logic, and we observe that, indeed, a rather wide range of logics with satisfiability in NP have Σ_2^p-complete local explainability problems.

We also show that the local explainability problem for PL can be reduced to the same problem for DNF-formulas, establishing that the Σ_2^p-completeness of local explainability holds already for DNF-formulas. This result again has far-reaching implications as DNF-formulas—due to their simple syntactic form—can particularly easily be translated into various machine learning classifiers. As an easy application, we show that explaining ensembles of decision trees via PL is Σ_2^p-complete.

As a further theoretical result, we prove that, when limiting to explaining only the positive truth value $\top$, the local explainability problem is only NP-complete for CNF-formulas of PL. As a corollary, we get NP-completeness of the problem for DNF-formulas in restriction to the truth value $\bot$. While theoretically interesting, these results are also relevant from the point of view of applications, as quite often real-life classification scenarios require explanations only in the case of one truth value. For example, explanations concerning automated insurance decisions are typically relevant only in the case of rejected applications.

In addition to global and local explanations, we also define a general version of the *counterfactual explanation problem* in the spirit of the related literature. The problem asks for the smallest change to the input that leads to a desired result. We instantiate the corresponding decision problem to PL and show that the PL version is NP-complete. Surprisingly this precise result does not seem to have been formulated for PL in the literature.

We provide an implementation of the local explainability problem of PL based on *answer-set programming* (ASP). Generally, ASP is a logic programming language based on the stable model semantics and propositional syntax, particularly Horn clauses. ASP is *especially suitable* and almost custom-made for implementing the local explainability problem of PL, as ASP is designed precisely for the complexity levels up to Σ_2^p. Indeed, while the disjunction-free fragments of ASP [23] cover the first level of the polynomial hierarchy in a natural way, proper disjunctive rules with cyclic dependencies become necessary to reach the second level of the hierarchy [9]. However, we exploit the expressive power of such disjunctions indirectly via the stable-unstable semantics [4] and oracles based on disjunction-free logic programs.

We test the implementation via experiments with benchmarks based on the *n-queens* and *dominating set problems*. The experiments provide concrete and compact explanations why a particular configuration of queens on the generalized chessboard or a particular set of vertices of a graph is or is not an acceptable solution to the involved problem. Runtime scale exponentially in the size of the instance and negative explanations are harder to compute than positive ones. We also use a modified version of the implementation to explain the behavior of a black-box classifier with several classification categories. Here we see that by varying the set of categories to be explained, we can obtain different information in terms of the lengths and generality of the explanations.

For global explainability, we use propositional logic to obtain approximate global explanations of data based on the percentage of misclassified points as the notion of error. We utilize an implementation previously published in [14].

Related work. While the literature on explainability is extensive, only a small part of it is primarily based on logic. One of the earliest logic-based notions of explanation was the *prime implicant explanation* of [21]. In the context of PL, a prime implicant of a formula φ is a subset-minimal set X of literals such that X implies φ. The notion of prime implicant has a long history in theoretical computer science, but [21] were the first to explicitly relate them to the currently active trend of research on explainable AI. Other closely related notions of logic-based explanations include *abductive explanations* [18, 2] and *sufficient reasons* [16, 1].

All of the above notions are clearly related to our notion of local explanations. However, there are some key differences. The notions in the literature are defined in terms of finding a minimal subset of features—or propositions—that suffice to explain the input instance of a Boolean decision function. Our definition of local explainability allows for any kind of formulas as outputs and generally imposes very few restrictions on the logics that can be considered. In the case of propositional logic, we separately prove that a De-Morganization (cf. 5.1) of a subconjunction of the literals in the input is always a possible output. We end up with a similar goal of removing propositions, but from a different starting point and in a different, general setting. The fact that our definition ends up similar to the standard one in the case of propositional logic can be considered—we believe—as evidence towards its naturalness. For other logics, such as FO and beyond, the resemblance to previous notions decreases. One benefit of our definition is that it is directly applicable to a wide variety of logics. For example, our approach immediately provides a suitable setting for studying the explainability problem of S5 modal logic, which we show Σ_2^p-complete. It is also worth noting that in Shih et al. (2022), Ji and Darwiche (2023), Marques-Silva and Ignatiev (2022) and Bassan and Katz (2023), the explanations are subset-minimal while we use globally formula-size minimal explanations. In Bassan and Katz (2023), the subset-minimal explanations are accompanied by a coefficient that estimates how close the explanation is in terms of size to a globally minimal explanation.

We also note that while most of the literature on logic-based explainability focuses on variants of explaining the decision of a classifier on a single input, we also define global explanation problems that aim to explain the entire behavior of a classifier. The simplest variant comes down to the previously studied problem of formula minimization [24], while the more general versions allow for approximation or formulating explanations in a logic different from the input logic.

Counterfactual explanations have no single fixed definition in the literature, but are generally related to (minimal) changes to the input to achieve some different outcome. Variants of this idea are studied in the literature, e.g., the *necessary reasons* of [16] and the *contrastive explanations* of [18]. Perhaps the closest to our definition is the problem called Minimum Change Required (MCR) in [1]. Our definitions are similar to the ones in the literature and generalized to an arbitrary logic.

We also mention some famous approaches to explainable AI that are not logic-based. The well-known LIME method [20] produces local approximate explanations in some neighborhood of the input model. Our generalized definitions in Section 3.1 also feature approximation but we focus on fully global and fully local explanations,

differing from the semi-local approach of LIME. In practice, LIME is often used as a feature importance measure; other such measures include e.g., Shapley values [17]. While feature importance as an approach to explainability can highlight relevant features, it can fail to recognize the relevance of combinations of features. Our approach on the other hand produces minimal formulas that can refer to multiple features and their combinations in a natural way.

We move on to work related to our complexity results. [24] shows that the *shortest implicant problem* is Σ_2^p-complete, thereby solving a long-standing open problem of Stockmeyer. An implicant of φ is a conjunction χ of literals such that $\chi \models \varphi$. The shortest implicant problem asks whether there is an implicant of φ with size at most k. Size is defined as the number of occurrences of literals. Below we prove that the local explainability problem for PL can be reduced to certain particular implicant problems. However, despite this, the work of Umans does not directly modify to give the Σ_2^p-completess result of the local explainability problem. The key issue is that the explainability problem requires an interpolant between a set χ of literals and a formula φ, where χ has precisely the same set of propositional symbols as φ. Thus we need to give an independent proof for the Σ_2^p lower bound. Also, formula size in [24] is measured in a more coarse way.

[1] consider a problem they call Minimum Sufficient Reason (MSR), which is similar to local explainability. They implicitly show that MSR for PL is Σ_2^p-complete, using Umans' work 2001 as a starting point. Our proof for PL is more direct and simpler and we also go further, showing that Σ_2^p-completeness holds already for DNF-formulas. Barceló et al. also do not provide any implementation of MSR.

Summary of contributions.

- General formulation of global, local and counterfactual explanations in the spirit of the related literature.
- Instantiations of these problems to propositional logic with proofs that link them to previous notions in the literature.
- Proof of Σ_2^p-completeness for the local explainability problem of PL, including its restriction to DNF-formulas.
- Proof of NP-completeness for the local explainability of CNF-formulas and DNF-formulas when restricted to specific truth values.
- Instantiations of local explainability for S5 and ensembles of decision trees with Σ_2^p-completeness results.
- Implementations and examples of the problems for PL with openly available code, showing that the problems work in a natural way in practice.

This article is an extension of the conference article [13] and the manuscript [12]. This version includes the following new additions compared to these works. We have further generalized the explainability problems, allowing for approximation, explanations via a different logic and explanations of sets of truth values rather than just single truth values. We also define counterfactual explainability problems in this version and show that the problem for PL is NP-complete. We improve upon our earlier result of Σ_2^p-completeness for the local explainability problem for PL by showing that the same problem restricted to DNF-formulas is already Σ_2^p-complete. We apply this result to show that explaining ensembles of decision trees using PL is likewise Σ_2^p-complete. On the experimental side, we have added an example on global explanations with approximation using a previously published implementation, as well as an example of explaining different sets of truth values of a four-valued black-box classifier using PL.

2 Preliminaries

Let Φ be a set of propositional symbols. The set $\mathrm{PL}(\Phi)$ of formulas of **propositional logic** PL over Φ is given by the grammar

$$\varphi := p \mid \top \mid \bot \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi)$$

where $p \in \Phi$. A *literal* is either an atom p or its negation $\neg p$, also known as *positive* and *negative* literals, respectively. We do not consider $\top$ and $\bot$ as literals for technical reasons having to do with some definitions below. A Φ-**assignment** is a function $s : \Phi \to \{0, 1\}$. When Φ is clear from the context or irrelevant, we simply refer to assignments rather than Φ-assignments. A Φ-assignment that is otherwise like s but sends $p \in \Phi$ to $b \in \{0, 1\}$ is denoted by $s(p/b)$. We define the semantics of propositional logic in the usual way, and we write $s \vDash \varphi$ if the assignment s **satisfies** the formula $\varphi \in \mathrm{PL}(\Phi)$. Alternatively, we can use the **standard valuation** function v_Φ defined such that $v_\Phi(s, \varphi) = 1$ if $s \vDash \varphi$ and otherwise $v_\Phi(s, \varphi) = 0$.

A formula $\psi \in \mathrm{PL}(\Phi)$ is a **logical consequence** of $\varphi \in \mathrm{PL}(\Phi)$, denoted $\varphi \vDash \psi$, if for every Φ-assignment s, $s \vDash \varphi$ implies $s \vDash \psi$. A formula $\chi \in \mathrm{PL}(\Phi)$ is an **interpolant** between φ and ψ if $\varphi \vDash \chi$ and $\chi \vDash \psi$. For a finite Φ, we say that a formula φ is a **maximal conjunction** w.r.t. Φ if φ is a conjunction of exactly one *literal* for each $p \in \Phi$. *A Φ-assignment s can be naturally identified with a maximal conjunction,* for example $\{(p, 1), (q, 0)\}$ identifies with $p \wedge \neg q$. A formula χ is a **subconjunction** of a conjunction of literals φ if χ is a conjunction of literals occurring in φ or if $\chi = \top$. We similarly define a **maximal disjunction** w.r.t. Φ as a disjunction of exactly one literal for each $p \in \Phi$ and a **subdisjunction** χ of a disjunction of literals φ as a disjunction of some of the literals in φ or $\chi = \bot$. The **size** of φ, denoted $\mathit{size}(\varphi)$, is the number of occurrences of propositional symbols, binary connectives and negations in φ. For example, the size of $\neg\neg(p \wedge p)$ is 5 as it has one $\wedge$ and two occurrences of both $\neg$ and p. Note that this means that $\mathit{size}(\top) = \mathit{size}(\bot) = 0$.

A formula $\varphi \in \mathrm{PL}(\Phi)$ is in **disjunctive normal form**, or DNF, if φ is a disjunction, where each disjunct is a conjunction of literals. For $\Pi \subseteq \Phi$, the formula φ is in Π-**full DNF** if additionally each disjunct is a maximal conjunction w.r.t. Π. We also define **conjunctive normal form** CNF and Π-**full CNF** in a dual fashion, switching the roles of conjunction and disjunction. Note that all four normal forms presented here are expressively complete, meaning that any formula φ can be transformed to an equivalent formula in any of these four forms. To give an example, if $\varphi \in \mathrm{PL}(\Phi)$, then an equivalent Φ-full CNF is given by

$$\bigwedge_{\substack{s \text{ is a } \Phi\text{-assignment} \\ s \nvDash \neg\varphi}} \Big(\bigvee_{s(p)=0} p \vee \bigvee_{s(p)=1} \neg p \Big).$$

3 Notions of Explanation and Explainability

In this section we introduce four natural problems concerning the global and local perspectives to explainability. The global problems deal with the question of explaining the entire behaviour of a classifier, whereas the local ones attempt to explicate why a single input instance was classified in a given way. We give very general definitions of these problems, and for that we will devise a very general definition of the notion of a logic. Our definition of a logic covers various kinds of classifiers in addition to standard formal logics, including logic programs, Turing machines, neural network models, automata, and the like.

Definition 1. We define that a **logic** is a tuple $(\mathscr{M}, \mathscr{F}, v, m)$ where $\mathscr{M}$ and $\mathscr{F}$ are sets; $v : \mathscr{M} \times \mathscr{F} \to V$ is a function mapping to some set V; and $m : \mathscr{M} \times \mathscr{F} \to \mathbb{N}$ is a function. Emphasizing the set V, we can also call $(\mathscr{M}, \mathscr{F}, v, m)$ a V-**valued logic**.

The set $\mathscr{F}$ could consist of, for example, formulas in propositional logic or neural networks. The set $\mathscr{M}$ would then contain assignments or input instances, respectively. We refer to the elements of $\mathscr{M}$ as models and the elements of $\mathscr{F}$ as formulas, adhering to common terminology in mathematical logic. The function $v : \mathscr{M} \times \mathscr{F} \to V$

gives the semantics of the logic, with $v(\mathfrak{M}, \varphi)$ being the truth value of a formula φ in $\mathfrak{M}$, e.g., 0 or 1 in the case of classical logics. We call v a **valuation**. The function m gives a complexity measure for the formulas in $\mathscr{F}$. This measure can depend on a model $\mathfrak{M}$, such as the time required to check that $\mathfrak{M}$ satisfies $\mathscr{F}$, or be independent of models, such as formula size. If $m(\mathfrak{M}, \varphi)$ does not depend on $\mathfrak{M}$, we instead write $m(\varphi)$. In computational problems, where members of V are inputs, we of course assume that V is somehow representable.

Example 2. Propositional logic PL over a set Φ of propositional symbols can be defined as a tuple $(\mathscr{M}, \mathscr{F}, v, m)$, where $\mathscr{M}$ is the set of Φ-assignments; $\mathscr{F}$ the set $\mathrm{PL}(\Phi)$ of formulas; $v : \mathscr{M} \times \mathscr{F} \to \{0, 1\}$ is the standard valuation v_Φ; and $m(\varphi) = \mathit{size}(\varphi)$ for all $\varphi \in \mathscr{F}$. Note that here we do not utilize the possibility of having m depend on the model $\mathfrak{M}$.

Now, the following example demonstrates that we can consider much more general scenarios than ones involving the standard formal logics.

Example 3. Let $\mathscr{M}$ be a set of data and $\mathscr{F}$ a set of programs for classifying the data, that is, programs that take elements of $\mathscr{M}$ as inputs and output a value in some set V of suitable outputs. Now $v : \mathscr{M} \times \mathscr{F} \to V$ is just the function such that $v(D, P)$ is the output of $P \in \mathscr{F}$ on the input $D \in \mathscr{M}$. The function m can quite naturally give the program size for each $P \in \mathscr{F}$. We can also let $m(D, P)$ be for example the running time of the program P on the input D, or the length of the computation (or derivation) table.

Given a logic, we define the equivalence relation $\equiv \subseteq \mathscr{F} \times \mathscr{F}$ such that $(\varphi, \psi) \in \equiv$ if and only if $v(M, \varphi) = v(M, \psi)$ for all $M \in \mathscr{M}$. We shall now define four formal problems relating to explainability. The problems do work especially well for finite V, but this is not required as long as the elements of V are representable in the sense that they can be used as inputs to computational problems.

Definition 4. Let $L = (\mathscr{M}, \mathscr{F}, v, m)$ be a logic. We define the following four problems for L.

Global explanation problem
Input: $\varphi \in \mathscr{F}$, *Output*: $\psi \in \mathscr{F}$
Description: Find $\psi \in \mathscr{F}$ with $\psi \equiv \varphi$ and minimal $m(\psi)$.

Local explanation problem
Input: $(\mathfrak{M}, \varphi, b)$ where $\mathfrak{M} \in \mathscr{M}$, $\varphi \in \mathscr{F}$ and $b \in V$, *Output*: $\psi \in \mathscr{F}$ or `error`
Description: If $v(\mathfrak{M}, \varphi) \neq b$, output `error`. Else find $\psi \in \mathscr{F}$ with minimal $m(\mathfrak{M}, \psi)$ such that the following two conditions hold:
(1) $v(\mathfrak{M}, \psi) = b$ and
(2) For all $\mathfrak{M}' \in \mathscr{M}$, $v(\mathfrak{M}', \psi) = b \;\Rightarrow\; v(\mathfrak{M}', \varphi) = b$.

Global explainability problem
Input: (φ, k), where $\varphi \in \mathscr{F}$ and $k \in \mathbb{N}$, *Output*: Yes or no
Description: If there is $\psi \in \mathscr{F}$ with $\psi \equiv \varphi$ and $m(\psi) \leq k$, output yes. Otherwise output no.

Local explainability problem
Input: $(\mathfrak{M}, \varphi, b, k)$ where $\mathfrak{M} \in \mathscr{M}$, $\varphi \in \mathscr{F}$, $b \in V$ and $k \in \mathbb{N}$, *Output*: Yes or no
Description: Output "yes" if and only if there exists some $\psi \in \mathscr{F}$ with $m(\mathfrak{M}, \psi) \leq k$ such that the conditions (1) and (2) of the local explanation problem hold.

Informally speaking, if we view the formula φ as a classifier, then the global explanation problem provides an answer to the question "How does φ work?" while the local explanation problem answers the question "Why does φ classify $\mathfrak{M}$ as b?".

The above definitions are quite flexible. Notice, for example, that while the set $\mathscr{M}$ may typically be considered a set of models, or pieces of data, there are many further natural possibilities. For instance, $\mathscr{M}$ can be a set of

formulas. This nicely covers, e.g., model-free settings based on proof systems. However, the definitions above generalize quite naturally to a yet more comprehensive setting, as we will next demonstrate. The main reason that we have first given the above definitions is that in this article we shall concentrate mostly on the case of propositional logic, and the definitions in their above form suffice for that purpose. Secondly, the more general definitions below are easier to digest when reflected and compared to the above simpler scheme. In the following subsection we develop the more general definitions. As already noted, we shall stick to the above, less general definitions in the remaining parts of the article, starting from Section 4.

3.1 The Explanation and Explainability Problems More Generally

We first generalize local explainability by replacing the single truth value b in V with a set of truth values from V. Suppose $\mathscr{V}$ to be a finite or countably infinite set of symbols, we call a function $w : \mathscr{V} \to \mathscr{P}(V)$ a **representation over** $\mathscr{P}(V)$. Here, $\mathscr{P}(V)$ denotes the power set of V. For $b \in V$ and $B \in \mathscr{V}$, we write $b \in B$ to mean that $b \in w(B)$. The local explanation and local explainability problems are then defined as follows.

Definition 5. Let $L = (\mathscr{M}, \mathscr{F}, v, m)$ be a logic. We define the following two problems for L.
Local explanation problem
Input: $(\mathfrak{M}, \varphi, B)$ where $\mathfrak{M} \in \mathscr{M}$, $\varphi \in \mathscr{F}$ and $B \in \mathscr{V}$, *Output*: $\psi \in \mathscr{F}$ or `error`
Description: If $v(\mathfrak{M}, \varphi) \notin B$, output `error`. Else find $\psi \in \mathscr{F}$ with minimal $m(\mathfrak{M}, \psi)$ such that the following two conditions hold:
(1) $v(\mathfrak{M}, \psi) \in B$ and
(2) For all $\mathfrak{M}' \in \mathscr{M}$, $v(\mathfrak{M}', \psi) \in B \Rightarrow v(\mathfrak{M}', \varphi) \in B$.

Local explainability problem
Input: $(\mathfrak{M}, \varphi, B, k)$ where $\mathfrak{M} \in \mathscr{M}$, $\varphi \in \mathscr{F}$, $B \in \mathscr{V}$ and $k \in \mathbb{N}$, *Output*: Yes or no
Description: Output "yes" if and only if there exists some $\psi \in \mathscr{F}$ with $m(\mathfrak{M}, \psi) \leq k$ such that the conditions (1) and (2) of the local explanation problem hold.

Example 6. Let $\mathscr{M} = V = \mathbb{Z}$ and let $\mathscr{F}$ be the set of polynomials in the variable x with integer coefficients. The function $v : \mathscr{M} \times \mathscr{F} \to V$ evaluates a polynomial $p \in \mathscr{F}$ at a point $a \in \mathscr{M}$. Let $m : \mathscr{F} \to \mathbb{N}$ be the complexity measure which counts the number of occurrences of x, operations (addition and multiplication) and constants.

Consider the local explanation problem with the input

$$(4, p = x^5 - 2x^4 - x^3 - 5x^2 - 2x - 3, \mathbb{Z}_{>0})$$

where $\mathbb{Z}_{>0}$ is a symbol representing the set of positive integers. Intuitively, we are asking for an explanation for the fact that this polynomial evaluated at 4 gets a positive value. In this case the problem would output $x - 3$, a very simple explanation. We see that $4 - 3 = 1$ is positive and since $p = (x - 3)(x^2 + 1)(x^2 + x + 1)$, we also see that whenever $x - 3$ is positive, p is also positive. Finally, $m(x - 3) = 3$ so $x - 3$ is clearly the minimal explanation.

An interesting property of the previous example is that we can provide an explanation for the explanation. Indeed, the factorization of p explains why $x > 3$ implies that $p(x) > 0$. We leave it for future work to incorporate the explainability of explanations into our definitions.

We generalize the problems further to allow for uncertainty or approximation. To this end, we need the following notions related to metric spaces.

Definition 7. Let A be a set. A function $d : A \times A \to [0, \infty)$ is a **pseudometric over** A, if the following conditions hold for all $a, b, c \in A$:
(1) $d(a, a) = 0$,
(2) $d(a, c) \leq d(a, b) + d(b, c)$,
(3) $d(a, b) = d(b, a)$.

If additionally $d(a,b) = 0$ implies $a = b$, then d is a **metric over** A. In this case the pair $\mathscr{T} = (A, d)$ is called a **metric space**.

We consider metric spaces over the sets V of "truth values" in V-valued logics. To enable real numbers in inputs to computational problems, we also consider representable sets of reals: if $\mathscr{R}$ is a finite or countably infinite set of symbols, then a function $r : \mathscr{R} \to \mathbb{R}$ is called a **representation of reals**. If $\varepsilon \in \mathscr{R}$, we let $\mathfrak{B}(b, \varepsilon)$ denote the closed ball that contains all those points of V whose distance is at most $r(\varepsilon)$ from the point $b \in V$.

For the global explanation problem we additionally consider a pseudometric $d : \mathscr{F} \times \mathscr{F} \to \mathbb{R}_{\geq 0}$ over $\mathscr{F}$. The pseudometric d corresponds to the degree of equivalence between two formulas. Since two syntactically different formulas can nevertheless be equivalent, the use of a pseudometric instead of just a metric is well-justified.

The below definition extends Definitions 4 and 5.

Definition 8. Let $L = (\mathscr{M}, \mathscr{F}, v, m)$ be a V-valued logic as given in Definition 1. Let $\mathscr{T}$ be a metric space over V and $r : \mathscr{R} \to \mathbb{R}$ a representation of reals. Let $w : \mathscr{V} \to \mathscr{P}(V)$ be a representation over $\mathscr{P}(V)$. Let $d : \mathscr{F} \times \mathscr{F} \to \mathbb{R}_{\geq 0}$ be a pseudometric over $\mathscr{F}$. We define the following four problems for L.

Global explanation problem
Input: (φ, ε), where $\varphi \in \mathscr{F}$, $\varepsilon \in \mathscr{R}$ *Output*: $\psi \in \mathscr{F}$
Description: Find $\psi \in \mathscr{F}$ with $d(\varphi, \psi) \leq r(\varepsilon)$ and minimal $m(\psi)$.

Local explanation problem
Input: $(\mathfrak{M}, \varphi, B, \varepsilon)$ where $\mathfrak{M} \in \mathscr{M}$, $\varphi \in \mathscr{F}$, $B \in \mathscr{V}$ and $\varepsilon \in \mathscr{R}$
Output: $\psi \in \mathscr{F}$ or `error`
Description: If $v(\mathfrak{M}, \varphi) \notin B$, output `error`. Else find $\psi \in \mathscr{F}$ with minimal $m(\mathfrak{M}, \psi)$ such that the following two conditions hold:

(1) $v(\mathfrak{M}, \psi) \in \bigcup_{b \in B} \mathfrak{B}(b, \varepsilon)$ and
(2) For all $\mathfrak{M}' \in \mathscr{M}$, we have

$$v(\mathfrak{M}', \psi) \in \bigcup_{b \in B} \mathfrak{B}(b, \varepsilon) \;\Rightarrow\; v(\mathfrak{M}', \varphi) \in \bigcup_{b \in B} \mathfrak{B}(b, \varepsilon).$$

Global explainability problem
Input: $(\varphi, \varepsilon, k)$, where $\varphi \in \mathscr{F}$, $\varepsilon \in \mathscr{R}$ and $k \in \mathbb{N}$, *Output*: Yes or no
Description: If there is $\psi \in \mathscr{F}$ with $d(\varphi, \psi) \leq r(\varepsilon)$ and $m(\psi) \leq k$, output yes. Otherwise output no.

Local explainability problem
Input: $(\mathfrak{M}, \varphi, B, \varepsilon, k)$ where $\mathfrak{M} \in \mathscr{M}$, $\varphi \in \mathscr{F}$, $B \in \mathscr{V}$, $\varepsilon \in \mathscr{R}$ and $k \in \mathbb{N}$,
Output: Yes or no
Description: Output "yes" if and only if there exists some $\psi \in \mathscr{F}$ with $m(\mathfrak{M}, \psi) \leq k$ such that the conditions (1) and (2) of the local explanation problem hold.

Note that if we restrict the sets B in the above definition to singletons, we obtain an important, specific generalization of the local explanation and local explainability problems of Definition 4. In this version, there is only one truth value to be explained but we still have a metric space over the truth values.

Example 9. Let $\mathscr{M} = V = \mathbb{Q}$ and let $\mathscr{F}$ be the set of polynomials in the variable x with rational coefficients. As the metric of V we simply take $d(q, q') := |q - q'|$. The function $v : \mathscr{M} \times \mathscr{F} \to \mathbb{Q}$ evaluates a polynomial $p \in \mathscr{F}$ at a point $q \in \mathscr{M}$. Let $m : \mathscr{F} \to \mathbb{N}$ be the complexity measure that gives the degree of the polynomial, e.g., $m(x^2 + 1) = 2$.

Consider the local explanation problem with the input

$$(2, p = x^3 - x^2 + x - 5, B = \{1\}, \varepsilon = 0.01).$$

Intuitively, we ask for an approximate explanation for the fact that this polynomial evaluated at 2 gets the value 1. Since p is differentiable, we can approximate it near the point 2 using the following linear approximation

$$h(x) := p(2) + p'(2)(x - 2) = 1 + 9(x - 2) = 9x - 17.$$

We claim that h is a possible output. First, we have that $m(h) = 1$, which is optimal because no polynomial of degree 0 (a constant polynomial) can satisfy condition (2). Furthermore $h(2) = 1$, so it satisfies condition (1). Finally, a simple calculation shows that if $d(h(x), 1) < 0.01$ then also $d(p(x), 1) < 0.01$, so h also satisfies condition (2).

Concerning the global explanation problem, in Section 5.3 we define one natural possibility for the pseudometric d based on the number of models, where φ and ψ disagree.

3.2 Explanations via a Different Logic

All of the above definitions assume that the explanation is a formula of the same logic as the formula to be explained. In practice this is not always the case. One might, for example, wish to explain the behavior of a neural network using a simpler logic of some kind. Below we outline how any of the above definitions can be adapted for two different logics.

Let L_1 be the logic used for explanations and L_2 the logic to be explained. All sets and functions related to these logics will be denoted with the respective subscripts. The set of models $\mathscr{M}$ is used for both logics.

We start with global explainability. Here we only need to define the notion of equivalence or approximate equivalence over both logics L_1 and L_2. As an example, we give a version of the global explanation problem in Definition 8 for two different logics. We assume that d is a pseudometric over the set $\mathscr{F}_1 \cup \mathscr{F}_2$, that is, over all formulas of both logics. Even though the below definition is quite general, we can still conceive ways to push it even further, but this is left for future work.

Global explanation problem
Input: (φ, ε), where $\varphi \in \mathscr{F}_2$, $\varepsilon \in \mathscr{R}$ *Output*: $\psi \in \mathscr{F}_1$
Description: Find $\psi \in \mathscr{F}_1$ with $d(\varphi, \psi) \leq r(\varepsilon)$ and minimal $m_1(\psi)$.

For local explainability, the adaptation for different logics is also quite simple. We essentially just use different logics and truth value sets for the left and right sides of the implication in condition (2). This version of the local explanation problem as given in Definition 5 is defined as follows.

Local B_1-explanation problem
Input: $(\mathfrak{M}, \varphi, B_2)$ where $\mathfrak{M} \in \mathscr{M}$, $\varphi \in \mathscr{F}_2$ and $B_2 \in \mathscr{V}_2$, *Output*: $\psi \in \mathscr{F}_1$ or `error`
Description: If $v_2(\mathfrak{M}, \varphi) \notin B_2$, output `error`. Else find $\psi \in \mathscr{F}_1$ with minimal $m_1(\mathfrak{M}, \psi)$ such that the following two conditions hold:
(1) $v_1(\mathfrak{M}, \psi) \in B_1$ and
(2) For all $\mathfrak{M}' \in \mathscr{M}$, $v_1(\mathfrak{M}', \psi) \in B_1 \;\Rightarrow\; v_2(\mathfrak{M}', \varphi) \in B_2$.

The global explanation problem for different logics can be seen as formalizing "knowledge distillation", where the idea is to find a simpler machine learning model, such as a decision tree, which approximates the behavior of a more complex machine learning model, such as a neural network, see e.g., [8, 19]. The use of the local explanation problem for different logics with different sets of truth values is demonstrated in the experiments in Subsection 6.2, where we use (two-valued) propositional logic to explain a four-valued classifier.

3.3 Counterfactual Explainability

Counterfactual explainability concerns itself with scenarios where we are asking for the smallest change to the current input that leads to a desired truth value. This is common in applications. Consider, for example, a customer asking for the smallest change needed to get a rejected application accepted. Here we give two formalisations of the related tasks, again separating explanation and explainability problems from each other. For the definitions, we use the conventions from above. Furthermore, we use a metric $d_M \subseteq \mathscr{M} \times \mathscr{M}$ to measure distances between input models.

The counterfactual explanation problem is specified as follows.

Counterfactual explanation problem
Input: $(\mathfrak{M}, \varphi, b)$ where $\mathfrak{M} \in \mathscr{M}$, $\varphi \in \mathscr{F}$ and $b \in V$
Output: $\mathfrak{N} \in \mathscr{M}$ or `error`
Description: Output a model $\mathfrak{N}$ with minimum $d_M(\mathfrak{M}, \mathfrak{N})$ such that $v(\mathfrak{N}, \varphi) = b$. If there is no model $\mathfrak{N}$ such that $v(\mathfrak{N}, \varphi) = b$, output `error`.

The corresponding explainability problem is defined as follows.

Counterfactual explainability problem
Input: $(\mathfrak{M}, \varphi, b, \varepsilon)$ where $\mathfrak{M} \in \mathscr{M}$, $\varphi \in \mathscr{F}$ and $b \in V$, and $\varepsilon \in \mathscr{R}$
Output: Yes or no
Description: Output "yes" if there exist a model $\mathfrak{N} \in \mathscr{M}$ such that $d_M(\mathfrak{M}, \mathfrak{N}) \leq r(\varepsilon)$ and $v(\mathfrak{N}, \varphi) = b$. Otherwise output "no".

As the other classes of problems, also the counterfactual explanation and explainability problems have immediate natural generalizations. We here mention the variants where instead of $b \in V$, we specify a range of target truth values via an input $B \in \mathscr{V}$.

We further investigate the counterfactual explanation problem for propositional logic. To this end, we define $d_M(s, s')$ between two assignments s and s' to be the *Hamming distance* between the assignments, i.e., the number of truth value changes (of propositional symbols) required to make one of the assignments identical to the other. Note here that s and s' must have the same domain. Formally, the counterfactual explainability problem for propositional logic is specified as follows.

Counterfactual explainability problem for PL
Input: (s, φ, b, k) where s is an assignment interpreting at least all the propositional symbols in the formula φ of PL, while $b \in \{0, 1\}$ and $k \in \mathbb{N}$.
Output: Yes or no
Description: Output "yes" if there exists an assignment s' such that $d_M(s, s') \leq k$ and $v(s, \varphi) = b$. Otherwise output "no".

The following is now very easy to prove. As far as we know, this result has not been stated explicitly in the literature. However, as stated in the introduction, [1] studies the complexity of similar problems, but not for PL specifically.

Proposition 10. *The counterfactual explainability problem for* PL *is* NP*-complete.*

Proof. For the upper bound, to check an input (s, φ, b, k), guess an assignment t with $d_M(s, t) \leq k$ and check in polynomial time whether $v(t, \varphi) = b$.

The lower bound is a simple reduction from Boolean SAT. For a Boolean formula ψ, let $p(\psi)$ denote the set of propositional symbols in ψ, and let $s_\psi : p(\psi) \to \{0,1\}$ be, say, the constant assignment mapping every proposition to 0. Now the satisfiability of a Boolean formula φ is equivalent to $(s_\varphi, \varphi, 1, |p(\varphi)|)$. □

In fact, the above argument directly gives NP-completeness for any reasonable fragment of PL which has NP-hard satisfiability problem. Furthermore, Proposition 10 can be used to establish the NP-completeness of the counterfactual explainability problem for many extensions of PL for which the model checking problem is solvable in polynomial time.

4 Local Explainability for PL

Let (φ, ψ, b) be an input to the local explanation problem of propositional logic, where φ is a maximal conjunction w.r.t. some (any) finite Φ (thus encoding a Φ-assignment), ψ a Φ-formula and $b \in \{0,1\}$. The local explanation problem can be reformulated equivalently in the following way.

(1) Suppose $b = 1$. If $\varphi \models \psi$, find a minimal interpolant between φ and ψ. Else output `error`.
(2) Suppose $b = 0$. If $\varphi \models \neg\psi$, find a minimal interpolant between ψ and $\neg\varphi$. Else output `error`.

Let $\varphi \in \mathrm{PL}(\Phi)$ be a conjunction of literals. Let $P(\varphi)$ and $N(\varphi)$ be the sets of positive and negative literals in φ, respectively. We denote the De Morgan transformations of φ and $\neg\varphi$ by

$$\mathrm{DM}(\varphi) := \bigwedge_{p \in P(\varphi)} p \wedge \neg\Big(\bigvee_{\neg q \in N(\varphi)} q\Big) \quad \text{and} \quad \mathrm{DM}(\neg\varphi) := \neg\Big(\bigwedge_{p \in P(\varphi)} p\Big) \vee \bigvee_{\neg q \in N(\varphi)} q.$$

We additionally denote $\mathrm{DM}(\neg\top) = \bot$ and $\mathrm{DM}(\neg\bot) = \top$.

We begin with a simple lemma.

Lemma 11. *Let Φ be a finite set of propositional symbols and let $\theta \in \mathrm{PL}(\Phi)$ be a formula with no negations and no occurrences of $\top$ or $\bot$. Let $\Pi \subseteq \Phi$ with $\Pi \neq \emptyset$.*

(1) Let χ be a maximal conjunction w.r.t. Π. Let χ' be the subconjunction of χ with only the positive literals. If $\chi \models \theta$, then $\chi' \neq \top$ and $\chi' \models \theta$.
(2) Let χ be a maximal disjunction w.r.t. Π. Let χ' be the subdisjunction of χ with only the positive literals. If $\theta \models \chi$, then $\chi' \neq \bot$ and $\theta \models \chi'$.

Proof. We prove the first claim by induction on the structure of the negation-free formula θ.

Assume first that $\theta = p$ for some $p \in \Phi$. Now if $\chi \models \theta$, then p is a conjunct of χ. Thus p is also a conjunct of χ' so $\chi' \neq \top$ and $\chi' \models \theta$.

Next, assume $\theta = \varphi \wedge \psi$ for some negation-free $\varphi, \psi \in \mathrm{PL}(\Phi)$ with no occurrences of $\top$ or $\bot$. Now if $\chi \models \theta$, then $\chi \models \varphi$ and $\chi \models \psi$. By the induction hypothesis, $\chi' \neq \top$ and we obtain $\chi' \models \varphi$ and $\chi' \models \psi$. Thus $\chi' \models \theta$.

Finally, assume $\theta = \varphi \vee \psi$ for some negation-free $\varphi, \psi \in \mathrm{PL}(\Phi)$ with no occurrences of $\top$ or $\bot$. If $\chi \models \theta$, we may assume by symmetry that $\chi \models \varphi$. By the induction hypothesis, $\chi' \neq \top$ and $\chi' \models \varphi$. Thus $\chi' \models \theta$.

The second claim is proved in a dual fashion. □

We proceed with a theorem on interpolation in propositional logic.

Theorem 12. *Let Φ be a finite set of propositional symbols, let φ be a maximal conjunction w.r.t. $\Pi \subseteq \Phi$ and let $\psi \in \mathrm{PL}(\Phi)$.*

(1) If $\varphi \models \psi$, then there is a subconjunction χ of φ such that $\mathrm{DM}(\chi)$ is a minimal interpolant between φ and ψ.
(2) If $\varphi \models \neg\psi$, then there is a subconjunction χ of φ such that $\mathrm{DM}(\neg\chi)$ is a minimal interpolant between ψ and $\neg\varphi$.

Proof. Assume that $\varphi \models \psi$. Clearly at least one minimal interpolant exists, as for example φ itself is an interpolant. Since φ is a maximal conjunction, we have $\varphi \not\models \bot$ so $\bot$ is not an interpolant. If $\top$ is an interpolant,

then we set $\chi = \top$ as $\top$ is a subconjunction of φ and $size(\top) = 0$ so $\top$ is clearly minimal. Below we assume that the minimal interpolant contains at least one propositional symbol.

Let θ be a minimal interpolant. Let $\Phi(\theta) \neq \emptyset$ be the set of propositional symbols occurring in θ. We transform θ into an equivalent formula θ' in $\Phi(\theta)$-full disjunctive normal form where each disjunct is a maximal conjunction w.r.t. $\Phi(\theta)$.

Now, as φ is a maximal conjunction, exactly one disjunct of θ' is a subconjunction of φ. Let χ denote this disjunct. Now since χ is a subconjunction of φ, we have $\varphi \models \chi$. As χ is also a disjunct of θ', we have $\chi \models \theta'$. Thus $\varphi \models \chi \models \theta' \models \psi$ so χ is an interpolant between φ and ψ.

We next show that $size(\mathrm{DM}(\chi)) \leq size(\theta)$. Each proposition in $\Phi(\theta)$ occurs in χ and thus also in $\mathrm{DM}(\chi)$ exactly once, so $\mathrm{DM}(\chi)$ has at most the same number of occurrences of propositional symbols and binary connectives as θ. Furthermore, $\mathrm{DM}(\chi)$ has at most one negation. Thus, if θ has at least one negation, then $\mathrm{DM}(\chi)$ has at most the same number of negations as θ. If θ has no negations, then we claim $\mathrm{DM}(\chi)$ also has none. To see this, first note that θ' is equivalent to θ, so $\chi \models \theta$. Now θ is negation free and since θ is formula-size minimal, θ has no occurrences of $\top$ or $\bot$. Thus by Lemma 11, we obtain $\chi' \models \theta$, where χ' is the subconjunction of χ with only the positive literals. If χ had any negative literals, then χ' would be a smaller interpolant than the minimal θ. Thus, we have $\chi' = \chi$ so χ and $\mathrm{DM}(\chi)$ are negation free. We have shown that $size(\mathrm{DM}(\chi)) \leq size(\theta)$, so $\mathrm{DM}(\chi)$ is a minimal interpolant.

Suppose then that $\varphi \models \neg\psi$. Since φ is a partial conjunction, we have $\top \not\models \neg\varphi$ so $\top$ is not an interpolant between ψ and $\neg\varphi$. If $\bot$ is an interpolant, then we set $\chi = \top$, since $\mathrm{DM}(\neg\top) = \bot$ and $\bot$ is size minimal. Below we assume that the minimal interpolant contains at least one propositional symbol.

Let θ be a minimal interpolant between ψ and $\neg\varphi$. We proceed in a dual fashion compared to the positive case above.

We transform θ into an equivalent formula θ' in $\Phi(\theta)$-full conjunctive normal form where each conjunct is a disjunction with exactly one of the disjuncts p and $\neg p$ for each $p \in \Phi(\theta)$. Additionally let φ' be the negation normal form of $\neg\varphi$. Now φ' is a disjunction of literals and exactly one conjunct of θ' is a subdisjunction of φ'. Let χ' denote this conjunct and let χ denote the negation normal form of $\neg\chi'$. Now χ is a subconjunction of φ and thus $\neg\chi \models \neg\varphi$. On the other hand, $\neg\chi$ is equivalent to χ', which is a conjunct of θ', so $\theta' \models \neg\chi$. We obtain $\psi \models \theta' \models \neg\chi \models \neg\varphi$ so $\neg\chi$ is an interpolant between ψ and $\neg\varphi$.

As in the positive case, $\mathrm{DM}(\neg\chi)$ has at most the same number of propositional symbols and binary connectives as the minimal interpolant θ. The formula $\mathrm{DM}(\neg\chi)$ again has at most one negation so we only check the case where θ has no negations. Recall that $\neg\chi$ is equivalent to χ', which in turn is a conjunct of θ'. Thus $\theta \models \chi'$. As θ has no negations and χ' is a maximal disjunction w.r.t. $\Phi(\theta)$, by Lemma 11, we have $\theta \models \chi''$, where χ'' is the subdisjunction of χ' with only the positive literals. By the minimality of θ we obtain $\chi'' = \chi'$ so χ' has no negations. Thus also $\mathrm{DM}(\neg\chi)$ is negation-free and is a minimal interpolant. □

The above theorem implies that for propositional logic, it suffices to consider subconjunctions of the input in the local explanation and explainability problems. This will be very useful both in the theoretical considerations below and in implementations. The result also shows that our formulation of the local explainability problem for PL turns out to be essentially equivalent with the minimum sufficient reason problem of [1]. A differing detail arises from the fact that we count negations into the length of formulas. Indeed, consider two subconjunctions χ and χ' with equally many literals but such that χ has only positive literals whereas χ' has at least one negative literal. Our problem prefers χ' because $\mathrm{DM}(\chi)$ has no negations whereas $\mathrm{DM}(\chi')$ has exactly one negation. The result also reveals a connection to the prime implicant explanations of [21], with the difference being subset minimal versus globally minimal explanations. Note, however, that this is only a specific instantiation of the general definitions of local explanation and explainability in Definition 4. The fact that the problem for PL coincides with those in the literature is evidence in favor of the general problem being canonical.

We next prove Σ_2^p-completeness of the local explainability problem for PL. As mentioned in the introduction, it was proved (though not explicitly stated) already in [1, Lemma 21] that the minimum sufficient reason problem is Σ_2^p-complete for PL. The proof there utilizes a reduction from the shortest implicant problem for DNF-formulas, which was proved by Umans to be Σ_2^p-complete [24]. In contrast to this, our proof uses a direct reduction from Σ_2SAT, which is well-known to be Σ_2^p-complete. The input of the problem is a quantified Boolean formula φ of the form

$$\exists p_1 \dots \exists p_n \, \forall q_1 \dots \forall q_m \, \theta(p_1, \dots, p_n, q_1, \dots, q_m).$$

The output is yes iff φ is true.

Theorem 13. *The local explainability problem for* PL *is Σ_2^p-complete.*

Proof. The upper bound is clear. For the lower bound, we will give a polynomial time (Karp-) reduction from Σ_2SAT. Consider an instance

$$\exists p_1 \dots \exists p_n \forall q_1 \dots \forall q_m \theta(p_1, \dots, p_n, q_1, \dots, q_m)$$

of Σ_2SAT. We start by introducing, for every existentially quantified Boolean variable p_i, a new propositional symbol $\overline{p}_i$ (not to be confused with the negation of p_i). Denoting $\theta(p_1, \dots, p_n, q_1, \dots, q_m)$ simply by θ, we define

$$\psi \; := \; \bigwedge_{i=1}^{n} (p_i \vee \overline{p}_i) \wedge \left(\theta \vee \bigvee_{i=1}^{n} (p_i \wedge \overline{p}_i) \right).$$

We let v be the valuation mapping all propositional symbols to 1, i.e., the assignment corresponding to the maximal conjunction

$$\varphi_v := \bigwedge_{i=1}^{n} (p_i \wedge \overline{p}_i) \wedge \bigwedge_{j=1}^{m} q_j$$

w.r.t. the set of propositional symbols in ψ. Clearly $\varphi_v \models \psi$. We now claim that there exists an interpolant of size at most $2n-1$ between φ_v and ψ iff the original instance of Σ_2SAT is true.

Suppose first that the original instance of Σ_2SAT is true. Thus there exists a tuple $(u_1, \dots, u_n) \in \{0,1\}^n$ such that $\forall q_1 \dots \forall q_m \theta(u_1, \dots, u_n, q_1, \dots, q_m)$ is true. Consider now the subconjunction $\chi := \bigwedge_{u_i=1} p_i \wedge \bigwedge_{u_i=0} \overline{p}_i$ of φ_v. Clearly χ is of size $2n-1$, since for every $i \in \{1, \dots, n\}$ either p_i or $\overline{p}_i$ occurs in it. Because $\forall q_1 \dots \forall q_m \theta(u_1, \dots, u_n, q_1, \dots, q_m)$ is true, χ is also an interpolant between φ_v and ψ.

Suppose then that χ is an interpolant of size at most $2n-1$ between φ_v and ψ. Using Theorem 12, we can assume that χ is a subconjunction of φ_v. Since χ has size at most $2n-1$, it can contain at most n propositional symbols. Furthermore, χ must contain, for every $i \in \{1, \dots, n\}$, either p_i or $\overline{p}_i$, since otherwise χ would not entail $\bigwedge_{i=1}^{n} (p_i \vee \overline{p}_i)$. Thus χ contains precisely n propositional symbols. More specifically, χ contains, for every $i \in \{1, \dots, n\}$, either p_i or $\overline{p}_i$. Now, we define a tuple $(u_1, \dots, u_n) \in \{0,1\}^n$ by setting $u_i = 1$ if χ contains p_i and $u_i = 0$ if χ contains $\overline{p}_i$. It is easy to see that $\forall q_1 \dots \forall q_m \theta(u_1, \dots, u_n, q_1, \dots, q_m)$ is true. □

The above theorem immediately implies a wide range of corollaries. Recall that S5 is the system of modal logic where the accessibility relations are equivalence relations. Given a pointed S5-model $(\mathfrak{M}, w)$ and a formula φ of S5, we write $v((\mathfrak{M}, w), \varphi) = 1$ if $(\mathfrak{M}, w) \models \varphi$ and otherwise $v((\mathfrak{M}, w), \varphi) = 0$. Now, the local explainability problem for S5 has as input a pointed S5-model $(\mathfrak{M}, w)$, an S5-formula ψ, $b \in \{0,1\}$ and $k \in \mathbb{N}$. The goal is to determine whether there exists a S5-formula φ of size at most k which satisfies the following conditions:

(1) $v((\mathfrak{M}, w), \varphi) = b$ and
(2) for all S5-models $(\mathfrak{M}', w')$ we have that

$$v((\mathfrak{M}', w'), \varphi) = b \Rightarrow v((\mathfrak{M}', w'), \psi) = b.$$

The following result is easy to establish using Theorem 13.

Corollary 14. *The local explainability problem for* S5 *is* Σ_2^p*-complete.*

Proof. The lower bound follows immediately from Theorem 13, while the upper bound follows from the well-known fact that the validity problem for S5 is coNP-complete [3]. □

Note indeed that the Σ_2^p lower bound for propositional logic is a rather useful result, implying Σ_2^p-completeness of local explainability for various logics with an NP-complete satisfiability problem.

4.1 Local Explainability for DNF-formulas

In this section we show that the local explainability problem is already Σ_2^p-hard for DNF-formulas. As DNF-formulas have a simple structure, this result facilitates reductions to prove hardness results for other problems more easily. Below in Subsection 4.2 we will use it to prove that the local explainability problem for tree ensembles is also Σ_2^p-complete. We find it interesting that explaining DNF-formulas, which correspond to simple rule-based classifiers, is as hard as explaining complicated classifiers such as random forests.

We note that the Σ_2^p-hardness for DNF-formulas does not follow from the proof given in [1, Lemma 21] for the Σ_2^p-hardness of the minimum sufficient reason problem of PL. Indeed, the reduction used there produces formulas of propositional logic which are neither DNF-formulas nor can they be easily transformed into equivalent DNF-formulas. For the same reasons, these results do not follow from the proof of Theorem 13.

To prove the Σ_2^p-hardness result for DNF-formulas, we will give a polynomial time reduction from the local explainability problem for arbitrary PL-formulas to the local explainability problem for DNF-formulas. We start with the following lemma.

Lemma 15. *Let* ψ *be a formula of* $\mathrm{PL}(\Phi)$*. One can construct in polynomial time w.r.t. the size of* ψ *a* 3DNF*-formula* $\psi' \in \mathrm{PL}(\Phi')$*, where* $\Phi \subseteq \Phi'$*, such that for every conjunction* φ *of* Φ*-literals we have that* $\varphi \models \psi$ *iff* $\varphi \models \psi'$*.*

Proof. Consider the formula $\neg\psi$. By applying the well-known Tseytin transformation on ψ, we can construct in polynomial time w.r.t. $|\psi|$ a 3CNF-formula ψ'' over a possibly larger vocabulary $\Phi' = \Phi \cup \{q_1, \dots, q_n\}$, such that for every Φ-assignment s we have that $s \models \neg\psi$ iff $s \models \exists q_1 \dots \exists q_n \psi''$. Hence we have that $s \models \psi$ iff $s \models \forall q_1 \dots \forall q_n \neg\psi''$, for every Φ-assignment s. Now $\neg\psi''$ is clearly equivalent to a 3DNF-formula, which is also the desired formula. □

This lemma shows that the local explainability problem for PL can be reduced to a variant of the local explainability problem for 3DNF-formulas, where we consider conjunctions that are not necessarily maximal. That is, we consider partial assignments instead of full assignments. The main technical task is to now reduce this problem to the local explainability problem for DNF-formulas.

In [24] a technique called *parity-substitution* was used to increase the "cost" of some of the propositional symbols that occurred in a given formula. We will use a similar technique that we call *conjunction-substitution*. The idea is that by increasing the "cost" of certain propositional symbols, we can control which propositional symbols occur in the optimal solution to the local explainability problem.

Fix a formula ψ of $\mathrm{PL}(\Phi)$, $\Pi \subseteq \Phi$ and $k \in \mathbb{Z}_+$. For each $q \in \Pi$ we introduce $k+1$ fresh propositional symbols $\{q_1, \dots, q_{k+1}\}$. Define

$$\Pi_{k+1} := \{q_\ell \mid q \in \Pi \text{ and } 1 \leq \ell \leq k+1\}.$$

Let ψ_{k+1} denote the formula obtained from ψ by replacing each $q \in \Pi$ with the conjunction

$$q_1 \wedge \dots \wedge q_{k+1}.$$

Note that $size(\psi_{k+1}) \leq (2(k+1)-1)size(\psi)$.

Lemma 16. *Let* s *and* s' *be* $(\Phi \backslash \Pi) \cup \Pi_{k+1}$*-assignments such that*
(1) $s(p) = s'(p)$*, for every* $p \in (\Phi \backslash \Pi)$ *and*

(2) for every $q \in \Pi$ we have that

$$s \vDash \bigwedge_{\ell=1}^{k+1} q_\ell \text{ iff } s' \vDash \bigwedge_{\ell=1}^{k+1} q_\ell.$$

Then $s \vDash \psi_{k+1}$ iff $s' \vDash \psi_{k+1}$.

Proof. Trivial. □

Lemma 17. *Let $\varphi \in \mathrm{PL}(\Phi \backslash \Pi)$ be a conjunction of literals.*

(1) For every subconjunction φ' of φ we have that $\varphi' \vDash \psi$ iff $\varphi' \vDash \psi_{k+1}$.

(2) Suppose that there exists a subconjunction of

$$\varphi_{k+1} := \varphi \wedge \bigwedge_{q \in \Pi} (q_1 \wedge \ldots \wedge q_{k+1})$$

with at most k instances of literals and which entails ψ_{k+1}. Then there also exists one which does not contain any propositional symbols from Π_{k+1}.

Proof. The first claim is clear and hence we will focus on the second claim. Suppose that χ is a subconjunction of φ_{k+1} which contains at most k literals and which entails ψ_{k+1}. If χ does not contain any propositional symbols from Π_{k+1}, then we are done. Suppose then that χ contains some $q_\ell \in \Pi_{k+1}$. Let χ' denote the subconjunction of χ obtained by removing q_ℓ from χ (in the extreme case where $\chi = q_\ell$ we set $\chi' := \top$). We claim that also χ' is an interpolant between φ_{k+1} and ψ_{k+1}.

Aiming for a contradiction, suppose that there exists a $(\Phi \backslash \Pi) \cup \Pi_{k+1}$-assignment s such that $s \vDash \chi' \wedge \neg\psi_{k+1}$. If $s(q_\ell) = 1$, then $s \vDash \chi$ and thus $s \vDash \psi_{k+1}$, which is a contradiction. Hence $s(q_\ell) = 0$, which implies that $s \vDash \neg \bigwedge_{s=1}^{k+1} q_s$. Consider then the assignment $s(q_\ell/1)$. Since $s(q_\ell/1) \vDash \chi$, we have that $s(q_\ell/1) \vDash \psi_{k+1}$. If $s(q_\ell/1) \vDash \neg \bigwedge_{s=1}^{k+1} q_s$, then by applying Lemma 16 to s and $s(q_\ell/1)$ we would get that $s(q_\ell/1) \vDash \neg\psi_{k+1}$, a contradiction. Hence $s(q_\ell/1) \vDash \bigwedge_{s=1}^{k+1} q_s$. We can pick some $1 \leq \ell' \leq k+1$ such that $q_{\ell'}$ does not occur in χ, since χ contains at most k literals. Then $s(q_\ell/1)(q_{\ell'}/0) \vDash \chi$ and hence $s(q_\ell/1)(q_{\ell'}/0) \vDash \psi_{k+1}$. Since $s(q_\ell/1)(q_{\ell'}/0) \vDash \neg \bigwedge_{s=1}^{k+1} q_s$, by applying Lemma 16 to s and $s(q_\ell/1)(q_{\ell'}/0)$ we get that $s(q_\ell/1)(q_{\ell'}/0) \vDash \psi_{k+1}$, a contradiction. □

Theorem 18. *For* DNF*-formulas, the local explainability problem is Σ_2^p-complete.*

Proof. We will give a reduction from the local explainability problem for PL. The proof of Theorem 13 shows that this problem is Σ_2^p-hard already for instances of the form $(\varphi, \psi, 1, 2k-1)$, where φ is a maximal conjunction of propositional symbols (which are not negated). Let $(\varphi, \psi, 1, 2k-1)$ be such an instance.

Now, let ψ' denote a 3DNF-formula obtained by applying Lemma 15 to ψ. Let Π denote all the propositional symbols in ψ' that do not occur in φ. Let ψ_{k+1} denote the formula obtained by using conjunction substitution with ψ', Π and k. Note that ψ_{k+1} is not a DNF-formula, since some of its "disjuncts" contain subformulas of the form

$$\neg \bigwedge_{\ell=1}^{k+1} q_\ell.$$

However, since ψ' was a 3DNF-formula, we can rewrite these "disjuncts" into disjunctions of conjunctions of literals with at most a polynomial (more precisely cubic) blow-up in the size of the formula. For example, the following "disjunct"

$$\neg \bigwedge_{\ell=1}^{k+1} q_\ell \wedge \neg \bigwedge_{\ell=1}^{k+1} q'_\ell \wedge \neg \bigwedge_{\ell=1}^{k+1} q''_\ell$$

is equivalent to the disjunction

$$\bigvee_{\ell_1=1}^{k+1}\bigvee_{\ell_2=1}^{k+1}\bigvee_{\ell_3=1}^{k+1}(\neg q_{\ell_1}\wedge\neg q'_{\ell_2}\wedge\neg q''_{\ell_3})$$

Hence, we can assume that ψ_{k+1} is a DNF-formula.

Let

$$\varphi_{k+1} := \varphi\wedge\bigwedge_{q\in\Pi}(q_1\wedge\ldots\wedge q_{k+1}).$$

Notice that φ_{k+1} is a maximal conjunction. By Lemma 17 we have that there exists a subconjunction φ' of φ with at most k literals such that $\varphi'\models\psi$ iff there exists a subconjunction φ' of φ_{k+1} with at most k literals such that $\varphi'\models\psi_{k+1}$. Since φ and φ_{k+1} do not contain negation, this is the same as saying that there exists a subconjunction φ' of φ with size at most $2k-1$ such that $\varphi'\models\psi$ iff there exists a subconjunction φ' of φ_{k+1} with size at most $2k-1$ such that $\varphi'\models\psi_{k+1}$. Hence we have managed to give a polynomial time reduction from the local explainability problem for PL to that of DNF-formulas. □

By simply negating formulas, we obtain that the local explainability problem is also Σ_2^p-hard for CNF-formulas.

Corollary 19. *For* CNF*-formulas, the local explainability problem is* Σ_2^p*-complete.*

We next show that if the formula ψ in the local explainability problem is restricted to CNF-formulas and we consider only the case $b=1$, then the problem is only NP-complete (as opposed to remaining Σ_2^p-complete). As an immediate corollary of this result, we also get NP-completeness for DNF-formulas in restriction to the case $b=0$. These results imply, e.g., that in applications where we are only interested in explaining why an input was not accepted by the classifier, DNF-formulas will be easier to work with than CNF-formulas.

To prove NP-hardness for explaining CNF-formulas, we will give a reduction from the dominating set problem. For a graph $G=(V,E)$, a **dominating set** $D\subseteq V$ is a set of vertices such that every vertex not in D is adjacent to a vertex in D. The input of the dominating set problem is a graph and a natural number k. The output is yes, if the graph has a dominating set of at most k vertices.

Theorem 20. *For* CNF*-formulas, the local explainability problem with* $b=1$ *is NP-complete. The lower bound holds even if we restrict our attention to formulas without negations.*

Proof. For the upper bound, let $\psi\in\mathrm{PL}(\Phi)$ be a CNF-formula and let φ be a maximal conjunction w.r.t. Φ. We want to determine whether there is an interpolant of size at most k. Using Theorem 12, it suffices to determine whether there is a subconjunction χ of φ such that $\mathit{size}(\mathrm{DM}(\chi))\leq k$.

Our nondeterministic procedure will start by guessing a subconjunction χ of φ. If $\mathit{size}(\mathrm{DM}(\chi))>k$, then it rejects. Otherwise, we replace the formula ψ with the formula ψ' which is obtained from ψ by replacing each propositional symbol p that occurs in χ with either $\top$ or $\bot$, depending on whether p occurs positively or negatively in χ. Now, if ψ' is valid, then our procedure accepts, and if it is not, then it rejects. Since the validity of CNF-formulas can be decided in polynomial time, our procedure runs in polynomial time as well.

For the lower bound we will give a reduction from the dominating set problem. Consider a graph $G=(V,E)$ and a parameter k. Let

$$\psi := \bigwedge_{v\in V}\Big(p_v\vee\bigvee_{(v,u)\in E}p_u\Big) \tag{1}$$

and $\varphi := \bigwedge_{v\in V}p_v$. Now $\varphi\models\psi$ and ψ is a CNF-formula. It is easy to verify that there exists an interpolant θ of size at most $2k-1$ if and only if G has a dominating set of size at most k. □

Corollary 21. *For* DNF*-formulas, the local explainability problem with* $b=0$ *is NP-complete. The lower bound holds even if we restrict our attention to formulas in which no propositional symbol occurs positively.*

4.2 Application: Explaining Tree Ensemble Models Using PL

As a concrete example of how Theorem 18 can be applied, we show that explaining *ensembles of decision trees* using formulas of propositional logic is Σ_2^p-hard. We start with some definitions. Let Φ be a set of propositional symbols. A **decision tree over** Φ is a pair $\mathcal{T} = (G, \ell)$, such that the following conditions hold.

- $G = (V, E)$ is a rooted directed binary tree.
- $\ell : V \cup E \to \Phi \cup \{0, 1\}$ is a function which places labels on nodes and edges, and which satisfies the following conditions.
  - Every internal node is labelled with a symbol from Φ while every leaf is labelled with either 0 or 1.
  - For every path from the root to a leaf, no two nodes on that path have the same label.
  - Every internal node has one outgoing edge labeled with 0 and one labeled with 1.

Let $\mathcal{T}$ be a decision tree over Φ and $s : \Phi \to \{0, 1\}$ an assignment. The assignment s defines a unique path $\pi_s = (v_1, \ldots, v_n)$ from the root of $\mathcal{T}$ to one of its leaves as follows: for every $1 \leq i < n$ we have that the edge (v_i, v_{i+1}) is labeled with $s(\ell(v_i))$. We say that $\mathcal{T}$ **accepts** s, if the final node in the path π_s is labelled with 1.

An ensemble of decision trees over Φ is simply a multiset of decision trees over Φ. Given an assignment $s : \Phi \to \{0, 1\}$ and an ensemble $\mathscr{E}$ of decision trees over Φ, we define that

$$\mathscr{E}(s) := \begin{cases} 1, & \text{if the majority of the trees in } \mathscr{E} \text{ accepts } s, \text{ and} \\ 0, & \text{otherwise.} \end{cases}$$

In other words, $\mathscr{E}$ **accepts** s iff the majority of the trees in $\mathscr{E}$ accept it. Many state-of-the-art machine learning algorithms for tabular datasets, such as random forest [5] and XGBoost [7], output ensembles of decision trees as classifiers [11, 22]. As these ensembles are often quite large, they are not typically intrinsically interpretable. Hence, it is natural to try to at least explain particular decisions made by such ensembles.

The local explainability problem for ensembles of decision trees has as input an assignment $s : \Phi \to \{0, 1\}$, an ensemble $\mathscr{E}$ of decision trees over Φ, $b \in \{0, 1\}$ and $k \in \mathbb{N}$. The goal is to determine whether there exists a propositional formula φ of size at most k which satisfies the following conditions.

(1) $\nu(s, \varphi) = b$.
(2) For all Φ-assignments s' we have that if $\nu(s', \varphi) = b$, then $\mathscr{E}(s') = b$.

The following result is now easy to establish using Theorem 18.

Theorem 22. *The local explainability problem for ensembles of decision trees is Σ_2^p-complete.*

Proof. The upper bound is clear. For the lower bound we will give a reduction from the local explainability problem for DNF-formulas. Clearly it suffices to give a polynomial time translation which translates DNF-formulas to equivalent ensembles. Let

$$\varphi := t_1 \vee \ldots \vee t_m$$

be a DNF-formula. Thus each formula t_i is a conjunction of literals. Now, it is clear that for each such conjunction t_i there exists a decision tree $\mathcal{T}_i$ which is equivalent with t_i and furthermore $\mathcal{T}_i$ has linear size w.r.t. $size(t_i)$. Let $\mathcal{T}_\top$ denote a decision tree which consists of a single node labeled with one. That is, $\mathcal{T}_\top$ accepts every assignment. Consider now the following ensemble:

$$\mathscr{E} := \{\mathcal{T}_\top, \ldots, \mathcal{T}_\top, \mathcal{T}_1, \ldots, \mathcal{T}_m\},$$

where $\mathcal{T}_\top$ occurs m times. It is clear that $\mathscr{E}(s) = 1$ iff at least one of the decision trees $\mathcal{T}_1, \ldots, \mathcal{T}_m$ accepts s. Thus $\mathscr{E}$ is equivalent with φ. □

Listing 1. Domain Predicates for Clauses and Atoms

```
clause(C) :- rule(C,clause).
atom(P) :- pcond(C,P).
atom(N) :- ncond(C,N).
```

Remark 23. The proof of Theorem 22 shows that already for the case $b = 1$ the local explainability problem is Σ_2^p-hard. A very similar proof can be used to show that also the case $b = 0$ is Σ_2^p-hard by using CNF-formulas instead of DNF-formulas.

The reduction given in the proof of Theorem 22 is very simple, because DNF-formulas are a very simple class of formulas. Equally simple reductions can also be used to derive similar Σ_2^p-hardness results for other machine learning models, such as decision lists and neural networks.

In [1, Proposition 6] it was proved that the minimum sufficient reason problem for decision trees is NP-complete. As discussed above, it follows from Theorem 12 that the minimum sufficient reason problem is essentially equivalent with the variant of the local explanation problem where we use formulas of PL as explanations. Thus we get that the local explainability problem for decision trees is NP-complete. This means that at least from the point of view of computational complexity, decision trees are easier to explain than ensembles of them. This is, of course, what one would intuitively expect.

4.3 On Global Explainability for PL and Beyond

The global explainability problem for propositional logic has been discussed in the literature under motivations unrelated to explainability. The **minimum equivalent expression problem** (MEE) asks, given a formula φ and an integer k, if there exists a formula equivalent to φ and of size at most k. This problem has been shown in [6] to be Σ_2^p-complete under Turing reductions, with formula size defined as the number of occurrences of propositional symbols and with formulas in negation normal form. The case of standard reductions is still open. For DNF-formulas the Σ_2^p-completeness under standard reductions was already proved in [24].

For logics beyond PL, the literature on the complexity of formula minimization is surprisingly scarce. The study of formula size in first-order and modal logics has mainly focused on particular properties that either can be expressed very succinctly or via very large formulas. This leads to relative succinctness results between logics. The related discussions, including links to local explainability, are left for the future.

5 Implementation

In this section, we devise a proof-of-concept implementation of the explainability problems defined above. The implementation exploits the ASP fragment supported by the *Clingo* system[1] that combines the *Gringo* grounder with the *Clasp* solver. As regards local explainability, we will present implementations specific to two use cases. First, in Section 5.1, we concentrate on propositional theories encoded as SAT problems in CNF. Second, we consider complete classifiers specified by assigning categories to all possible data points in Section 5.2. Finally, in Section 5.3, we sketch the case of global explainability covered by follow-up work elsewhere [14].

5.1 Local Explanations in the Boolean Case

Since *CNF*-formulas are dominant in the context of SAT checking, we devise our first implementation under an assumption that input formulas take this particular normal form. Thus, in the spirit of Theorem 12, φ is essentially a set L of literals and ψ is a set S of *clauses*, i.e., disjunctions of literals. To enable meta-level encodings

[1] https://potassco.org/clingo/

Listing 2. Checking Positive Precondition (Theorem 12)

```
1 :- clause(C), nlit(P): pcond(C,P); plit(N): ncond(C,N).
2 simp(C) :- plit(P), pcond(C,P).
3 simp(C) :- nlit(N), ncond(C,N).
4 simp(C) :- pcond(C,A), ncond(C,A).
5 :- clause(C), not simp(C), pcond(C,P), not plit(P), not nlit(P).
6 :- clause(C), not simp(C), ncond(C,N), not plit(N), not nlit(N).
```

Listing 3. Checking Negative Precondition (Theorem 12)

```
1 t(A) :- plit(A), atom(A).
2 { t(A) } :- atom(A), not plit(A), not nlit(A).
3 :- clause(C), not t(P): pcond(C,P); t(N): ncond(C,N).
```

in ASP, a *CNF*-formula in DIMACS format can be reified into a set of first-order (ground) facts using the *lpreify* tool[2] (option flag `-d`). Since *lpreify* treats clauses as special kinds of rules with *positive* and *negative* conditions expressed with predicates `pcond/2` and `ncond/2`, respectively, Listing 1 defines domain predicates `clause/1` and `atom/1` for identifying clauses and atoms that occur in the input.

The literals present in the set L are expressed by using domain predicates `plit/1` and `nlit/1` for positive and negative literals, respectively. We relax the requirement that the set of literals L is maximal, so that any three-valued interpretation of atoms can be represented. However, the precondition for the positive (resp. negative) explanation is essentially the same: the result $S|_L$ of *partially evaluating* S with respect to L must remain valid (resp. unsatisfiable) in accordance to Theorem 12. The positive check is formalized in Listing 2. The constraint in Line 1 excludes the possibility that L falsifies S directly. Lines 2–4 detect which clauses of S are immediately true given L and removed from $S|_L$ altogether. Rules in Lines 5 and 6 deny any clause containing yet open literals that could be used to falsify the clause in question. The net effect is that the encoding extended by facts describing L and S has an answer set iff $S|_L$ is valid. Since the scope of negation is restricted to domain predicates only, the check is effectively polytime.

The negative case can be handled by a single ASP program evaluating a coNP query, see Listing 3. We deploy stable-unstable semantics [4] and the modular approach of [15] for the representation of oracles, therefore hiding disjunctive and saturating rules [9] from our encodings altogether. The rule in Line 1 infers any positive literal in L to be true while the negative ones in L remain false *by default*. In Line 2, the truth values of atoms undefined in L are freely chosen. The constraint in Line 3 ensures that each clause in the input S must be satisfied. Thus $S|_L$ is unsatisfiable iff the encoding extended by facts describing L and S has no answer set. In general, this check is deemed worst-case exponential, but for maximal L, the task reduces to simple polytime propagation as no choices are active in Line 2.

Our more general goal is to find *minimum-size* explanations $L' \subseteq L$ possessing the identical property as required from L, i.e., the set of clauses $S|_{L'}$ is either valid or unsatisfiable. In the negative case (the second item of Theorem 12), the check for unsatisfiablity is formalized by Listing 4. While the (fixed) set L is expressed by the predicates `plit/1` and `nlit/1` as before, its subset L' is determined by choosing L-compatible truth values for atoms in Lines 1 and 2. The rule in Line 3 detects when L' consists of positive literals only. As regards the

[2]https://github.com/asptools/software

Listing 4. Finding Minimum/Bounded-Size Explanations

```
1 { t(A) } :- atom(A), plit(A).
2 { f(A) } :- atom(A), nlit(A).
3 positive :- not f(A): nlit(A).
4
5 #const k=0.
6 #minimize { 2,A: t(A), k=0; 2,A: f(A), k=0;
7              -1: t(A), positive, k=0 }.
8 :- #sum { 2,A: t(A); 2,A: f(A); -1: positive } > k, k>0.
```

Listing 5. Oracle for the Negative Case

```
1 { t(A) } :- plit(A).
2 { f(A) } :- nlit(A).
3 et(A) :- t(A).
4 { et(A) }:- not t(A), not f(A), atom(A).
5 :- clause(C), not et(P): pcond(C,P); et(N): ncond(C,N).
```

respective minimum-size formula $\mathrm{DM}(L') = \mathrm{DM}(\bigwedge_{l \in L'} l)$, its size can be computed as

$$size(\mathrm{DM}(L')) = \begin{cases} 2 \times |L'| - 1, & \text{if } L' \neq \emptyset \text{ contains only positive literals and} \\ 2 \times |L'|, & \text{otherwise.} \end{cases} \tag{2}$$

In the first case of (2), $|L'| - 1$ conjunction signs are needed to connect $|L'|$ literals. In the latter case, $L' = \emptyset$ corresponds to an empty conjunction $\top$ with $size(\top) = 0$ as a corner case. But otherwise $L' \neq \emptyset$ has at least one negative literal and $\bigwedge_{l \in L'} l$ can be reorganized using de Morgan laws such that negative literals are conjoined with a single negation sign and negated atoms are connected by disjunction signs. E.g., the effective length $8 = 2 \times 4$ of $\{a, b, \neg c, \neg d\}$ is measured as that of $a \wedge b \wedge \neg(c \vee d)$.

The size of the formula corresponding to L' is put subject to minimization in Lines 6 and 7. The objective function is activated if the parameter `k` equals to its default value `0` set in Line 5. The term $2 \times |L'|$ present in all cases of (2) is calculated in Line 6. The corrective term `-1` is taken into account only if there are no negative literals and at least one positive literal (i.e., $L' \neq \emptyset$). Note that `-1` is counted only once even if there were several positive literals because `-1` is not extended to a pair `-1,A` in contrast with the occurrences of `2` above. Any positive parameter values `k>0` set by the user activate the *local explainability* mode: the size of the formula corresponding to L' is at most `k` by the weight constraint in Line 8. Besides evaluating the objective function, we check that $L' \cup S$ is unsatisfiable by using an oracle encoded in Listing 5. The *input atoms* (cf. [15]) are declared in Lines 1 and 2. The predicate `et/1` captures a two-valued truth assignment compatible with L' as enforced by Lines 3 and 4. Moreover, the clauses of S are satisfied by constraints introduced in Line 5. Thus, the oracle has an answer set iff $L' \cup S$ is satisfiable. However, the stable-unstable semantics [4] and the translation *unsat2lp* from [15] produce the complementary effect, which amounts to the unsatisfiability of $S|_{L'}$. On the other hand, the positive case (the first item of Theorem 12) can be covered by extending the program of Listing 4 by further rules in Listing 6. The rules are analogous to those in Listing 2, but formulated in terms of predicates `t/1` and `f/1` rather than `plit/1` and `nlit/1`. Thus L' inherits the properties of L, i.e., the encoding based on Listings 4 and 6 extended by facts describing L and S has an answer set iff $S|_{L'}$ is valid for $L' \subseteq L$ corresponding to a minimum-size formula.

Listing 6. Extension for the Positive Case

```
simp(C) :- t(P), pcond(C,P).
simp(C) :- f(N), ncond(C,N).
simp(C) :- pcond(C,A), ncond(C,A).
:- clause(C), not simp(C), pcond(C,P), not t(P), not f(P).
:- clause(C), not simp(C), ncond(C,N), not t(N), not f(N).
```

Listing 7. Finding Minimum/Bounded-Size Explanations for a Given Classifier

```
1  attr(A) :- val(D,A,B).
2  target(M) :- M = #max{ A: attr(A) }.
3  exclude(A) :- target(A).
4
5  #const i=1.
6  1 { in(A): attr(A), not exclude(A) }.
7  diff(D) :- val(D,A,B), val(i,A,1-B), in(A).
8  :- val(D,T,B), val(i,T,1-B), target(T), not diff(D).
9
10 #const k=0.
11 positive :- val(i,A,1): in(A).
12 #minimize { 2,A: in(A), k=0; -1: val(i,A,1), in(A), positive, k=0 }.
13 :- #sum { 2,A: in(A); -1: positive } > k, k>0.
```

5.2 Explaining Classifiers

Given a certain classifier as a starting point, we can encode the respective *local explanation problem* in a more direct way. The values of the target attribute can be generated for each possible data point and represented in terms of a predicate `val(D,A,B)` where `D` is the data point identifier, `A` the name of an attribute, and `B` the value of the attribute `A` at `D`, i.e., either `0` or `1` for Boolean data. Also, if there is some attribute `A` not intended for explanations, this is flagged by a fact `exclude(A)` in the input. Due to exhaustive enumeration, we can forget about the syntactic representation of the classifier (if any) and concentrate on how values assigned to attributes affect the value of the target attribute when solving the local explanation problem with respect to a particular data point. An ASP encoding of the problem in this exhaustive setting is provided as Listing 7.

In Line 1, the identifiers of attributes are extracted from data and, by default, the one with a maximum identifier is recognized as the target attribute in Line 2. The target attribute is also excluded from explanations in Line 3.

The global parameter `i` set in Line 5 signifies the identifier of the *data point* for which we intend to find a local explanation. The choice rule in Line 6 selects one or more attributes whose values are understood to form the explanation in question. The respective assignment satisfies the condition (1) of the local explanation problem by construction, so we concentrate on verifying the condition (2) as follows. In Line 7, we detect data points `D` that differ from the data point `i` by the value of at least one chosen attribute. Such data points are irrelevant with respect to the condition (2). The other data points agree with the values of attributes involved in the candidate. The constraint in Line 8 ensures that the same holds for the value of the target attribute in accordance with the condition (2).

Finally, the maximum size of the formula is a global parameter `k` of the encoding in Line 10 as before. The default value `k=0` implies optimization over all possible values, and otherwise `k` is treated as an upper bound for the size of the local explanation. The remaining rules have been tailored from the rules in Listing 4.

Listing 8. Propositional Specification for the n-Queens problem

```
1 #const n=8.
2 pair(X1,X2) :- X1=1..n, X2=X1+1..n.
3 triple(X1,X2,Y1) :- pair(X1,X2), Y1=1..n-(X2-X1).
4 queen(X,Y): Y=1..n :- X=1..n.
5 -queen(X,Y1) | -queen(X,Y2) :- X=1..n, pair(Y1,Y2).
6 -queen(X1,Y) | -queen(X2,Y) :- pair(X1,X2), Y=1..n.
7 -queen(X1,Y1) | -queen(X2,Y1+X2-X1) :- triple(X1,X2,Y1). % X1+Y2=X2+Y1
8 -queen(X1,Y1+X2-X1) | -queen(X2,Y1) :- triple(X1,X2,Y1). % By symmetry
```

5.3 Global Explanations

For an example on global explainability, we consider explaining data sets with approximation as in Definition 8. A widely-used notion of error is the percentage of data points where the explaining formula disagrees with the explained formula. This error is clearly a pseudometric and thus fits the definition of Section 3.1 for global explainability. Intuitively, the error approximates how equivalent the two formulas are, with 0 meaning full equivalence. More formally, letting Δ denote the symmetric difference operator, we define

$$d_{\equiv}(\varphi, \psi) = \frac{|\mathrm{Mod}(\varphi)\Delta\mathrm{Mod}(\psi)|}{|\mathscr{M}|}$$

where $\mathrm{Mod}(\varphi) = \{\mathfrak{M} \in \mathscr{M} \mid v(\mathfrak{M}, \varphi) = 1\}$.

For increasing upper bounds ℓ on the size of formulas, we compute in Section 6.3 the formula with the smallest error among formulas of size at most ℓ. This is intended as a small example of global explanations with error, and therefore we do not go into implementation details, but rather direct the reader to [14] for more information.

6 Experiments

In this section we apply the implementations presented above to obtain local explanations for some Boolean benchmark problems and for a blackbox classifier with multiple classification categories. We additionally briefly discuss global explainability using a previously published implementation. We provide code for reproducing the experiments as online appendices. Related material can also be found at GitHub[3].

The primary goal of our experiments is twofold. First, we wanted to evaluate the scalability of our ASP-encodings for the explainability problems of PL. Secondly, we wanted to explore the types of explanations our definitions produce in practical scenarios.

6.1 Boolean Benchmarks

In what follows, we evaluate the computational performance of the *Clingo* system by using the encodings from Section 5 and two benchmark problems, viz. the famous n-queens (n-Qs) problem and the dominating set (DS) problem of undirected graphs—recall the proof of Theorem 20 in this respect. We study explainability in the context of these benchmark problems to be first encoded as SAT problems in *CNF* using the declarative approach from [10]: clauses involved in problem specifications are stated with rules in ASP style, but interpreted in *CNF* by using an adapter called *Satgrnd*[4]. Thus, the *Gringo* grounder of *Clingo* can be readily used for the instantiation of the respective propositional schemata, as given in Listings 8 and 9, for subsequent SAT solving.

Our n-Qs encoding in Listing 8 introduces a default value for the number of queens `n` in Line 1 and, based on `n`, the pairs and triples of numbers relevant for the construction of clauses are formed in Lines 2 and 3. Then, for the

[3]https://github.com/asptools/benchmarks/tree/main/explainability
[4]https://github.com/asptools/software

Listing 9. Propositional Specification for the Dominating Set Problem

```
1 vertex(X) :- edge(X,Y).     vertex(Y) :- edge(X,Y).
2 in(X) | in(Y): edge(X,Y) | in(Z): edge(Z,X) :- vertex(X).
```

queen in a column X, the length n clause in Line 4 chooses a row Y made unique for X by the clauses introduced in Line 5. Similarly, columns become unique by the clauses in Line 6. Finally, queens on the same diagonal are denied by clauses resulting from Lines 7 and 8. Turning our attention to the DS problem in Listing 9, vertices are extracted from edges in Line 1. The clauses generated in Line 2 essentially capture the disjunctions collected as parts of ψ in (1): our encoding assumes that the edges of the graph are provided as ordered pairs for the sake of space efficiency. Intuitively, given any vertex X in the graph, either it is *in* the (dominating) set or any of its neighboring vertices is.

In the experiments, we evaluate the performance of the *Clasp* solver (v. 3.3.5) when used to solve various explainability problems. All test runs are executed on a cluster of Linux machines with Intel Xeon 2.40 GHz CPUs, and a memory limit of 16 GB. We report only the running times of the solver, since the implementations of grounding and translation steps are suboptimal due to our meta-level approach and such computations could also be performed off-line in general, and it is worth emphasizing that our current method has been designed to work for any *CNF* and set of literals given as input. For n-Qs, we generate (i) *positive* instances by searching for random solutions to the problem with different values of n and (ii) *negative* instances by moving, in each solution found, one randomly selected queen to a wrong row. The respective truth assignments are converted into sets of literals L ready for explaining. For DS, we first generate random planar graphs of varying sizes and search for minimum-size dominating sets for them. Then, *negative* instances are obtained by moving one random vertex outside each optimal set and by describing the outcomes as sets of literals L. For *positive* instances, we include the positive literal in(X) in L for all vertices X, in analogy to the reduction deployed in Theorem 20.

The results of our experiments are collected in Figure 1. Since initial screening suggests that explanation under exact size bounds is computationally more difficult, we present only results obtained by minimizing the size of L using the *Clasp* solver in its *unsatisfiable core* (USC) mode. The first plot shows the performance of *Clasp* when searching for positive explanations for DS based on planar graphs from 100 up to 550 vertices. Explanations are minimum-size dominating sets. The performance obtained for the respective negative explanations is presented in the second plot of Figure 1. Here we can only cover smaller planar graphs with the number of vertices in the range $10 \ldots 50$. In this case, explanations consist of sets of vertices based on some vertex and its neighbors. The final plot in Figure 1 concerns n-Qs when $n = 20 \ldots 60$ and negative explanations are sought. Explanations obtained from the runs correspond to either (i) single (misplaced) queens or (ii) pairs of (threatening) queens. The respective positive instances simply reproduce solutions and are computationally easy. Therefore, they are uninteresting.

Some observations are in order. The instances obtained by increasing their size, i.e., either the number of vertices or queens, give rise to higher running times almost systematically. For each size, the time is computed as an average for running 10 instances of equal size. Due to logarithmic scale, running times tend to scale exponentially. Moreover, positive explanations for *CNF* formulas appear to be easier to find than negative ones in compliance with complexity results.

6.2 Explaining a Particular Classifier

We demonstrate the local explanation problem by explaining a non-Boolean black-box classifier via propositional logic as in Section 3.2. We use a data set from the UCI Machine Learning Repository. The *Car Evaluation*[5] data set

[5]https://archive.ics.uci.edu/ml/datasets/Car+Evaluation

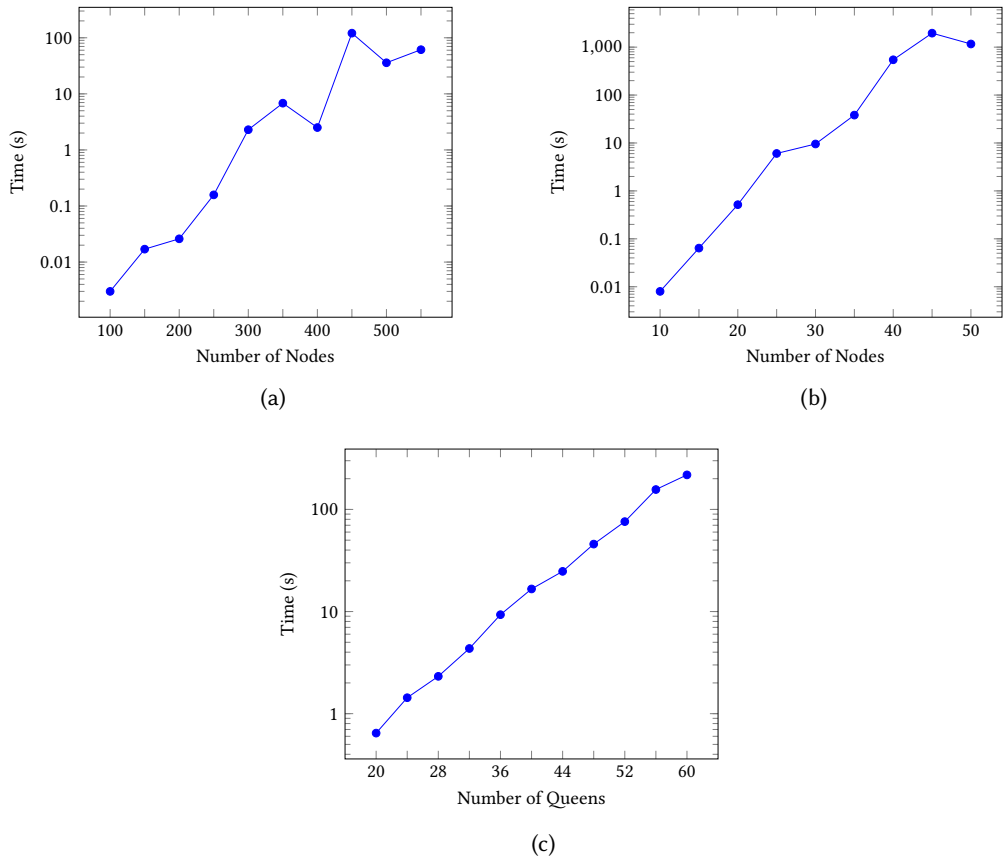

Fig. 1. Experimental Results on Explaining DS positively (a), negatively (b), and n-Qs negatively (c).

Table 1. Each cell contains the number of Car Evaluation inputs for which the shortest explanation for the given set of categories is the given length.

$B \downarrow$ Length $\rightarrow$	1	3	5	7	9	11
vgood	0	0	0	0	40	25
vgood, good	0	0	0	36	73	25
vgood, good, acc	0	0	144	280	89	5
good, acc, unacc	1632	26	5	0	0	0
acc, unacc	1440	112	35	6	1	0
unacc	960	144	75	21	9	1

lists the behavior of an expert system for classifying cars into four categories: unacceptable, acceptable, good and very good. We emphasize that the data set has exactly one row for each possible input to the classifier and thus it

is a black-box classifier rather than a true data set. We use local explanations in propositional logic to obtain information about the behavior of this classifier.

Car Evaluation has six categorical attributes with either three or four categories each, resulting in $4^3 \times 3^3 = 1728$ possible inputs. We Booleanize these (non-target) attributes by using a one-hot encoding, i.e., by giving each category of each original attribute its own propositional symbol. Thus after Booleanization each possible input has precisely six true propositions and the rest are false. Any local explanation of an input will only use some of the six true propositions since the explanations are formula length minimal and negations increase formula length. We further note that even though local explanations are based on only one input, they can provide very general information about the behavior of a classifier. For example, the single proposition $\texttt{safety(low)}$ suffices to place a car into the unacceptable category.

Since the target attribute is multi-valued, we consider different non-empty subsets $B \subset \{\mathsf{acc}, \mathsf{good}, \mathsf{unacc}, \mathsf{vgood}\}$ of its values as the basis for Booleanization. As an example, let us look at a car with a high buying price, medium maintenance price, three doors, four seats, a medium sized luggage boot and a medium safety rating. This car is classified as unacceptable. We first explain why the car is classified in the value set $B_1 = \{\mathsf{unacc}, \mathsf{acc}, \mathsf{good}\}$, hence receiving the single proposition explanation $\texttt{safety(low)}$. For the smaller set $B_2 = \{\mathsf{unacc}, \mathsf{acc}\}$ we obtain the explanation $\texttt{buying(med)} \wedge \texttt{maint(high)}$ of length three. Finally, for $B_3 = \{\mathsf{unacc}\}$, the shortest explanation is the conjunction of all six positive literals corresponding to the values of attributes in that instance.

We see that even though the instance is classified into only one class, we can vary the set of classes to be explained and receive different information. A larger set B gives more general information in the form of short explanations while a smaller B goes into the specifics and the explanations get longer. Even the longest explanations are conjunctions of six literals, which are easily readable. Any global explanation (with no error) of the Car Evaluation classifier would by necessity have to be quite complex, but via local explanations we can obtain easily readable partial information about the behavior of the classifier.

Table 1 reports for each set B of target values and each formula length ℓ from 1 to 11, how many inputs have an explanation of length ℓ for why they obtain a truth value in B. Note that only odd lengths are listed as the formulas contain no negations. We can see that in the Car Evaluation data, receiving a very good evaluation always requires many positive properties from the car, whereas lower evaluations can often be explained by just a single negative property. Indeed, the classifier does give most inputs a bad evaluation.

6.3 Global Explanations

We explain two data sets from the UCI Machine Learning Repository, Statlog (German Credit Data) and Breast Cancer Wisconsin (Original). We do not perform any kind of cross-validation here since the goal of this example is to explain the data itself, not produce a classifer for an underlying phenomenon. The paper [14] includes some different results including cross-validation via splitting the data. These results show that short explanations avoid overfitting to the training data in a very nice way.

The results discussed here are given in Figures 2a and 2b. By allowing the explanations to have some error, we can obtain very short explanations that are easily human readable. We can see from the figures that longer explanations allow for smaller errors. For the Breast Cancer data, already very short explanations give very small errors. In [14] we see also that the errors obtained in the German Credit data are comparable to those of other methods on the same data.

7 Conclusion

We have provided general, logic-based definitions of explainability and studied the particular case of propositional logic in detail. The related Σ_2^p-completeness result gives a useful, robust lower bound for a wide range of more expressive logics and future work. We have also shown NP-completeness of the explainability problems with

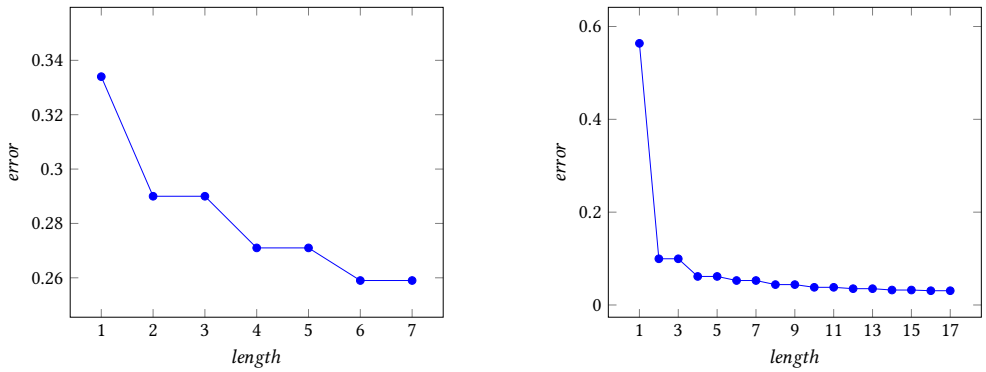

Fig. 2. Global Explainability Results for (a) the Bank Data Set and (b) the Breast Cancer Data Set

formulas in CNF and DNF when the input truth value is restricted. Moreover, we have presented a proof-of-concept implementation for the explanation of CNF-formulas (without truth value restrictions). Our experimental results confirm the expected worst-case exponential runtime behaviour of *Clasp*. Negative explanations have higher computational cost than positive explanations. The optimization variants of explanation problems seem interesting, because the USC strategy seems very effective and the users need not provide fixed bounds for queries in advance.

We note that short formulas or expressions can often be natural in explaining more complex ones. However, it is clear that in various settings, short expressions can be difficult to parse. Up to an extent, this phenomenon is taken into account already in our definition of a logic, where we let the complexity function m depend on $\mathscr{M}$ as well as $\mathscr{F}$. However, further important directions remain to be investigated. A central issue here is considering normal forms of logics, as such forms are often custom-made such that shorter formulas are *not* difficult to parse, but instead reveal the meaning of the formula directly.

Acknowledgments

Tomi Janhunen, Antti Kuusisto, Masood Feyzbakhsh Rankooh and Miikka Vilander were supported by the Academy of Finland consortium project *Explaining AI via Logic* (XAILOG), grant numbers 345633 (Janhunen) and 345612 (Kuusisto). Antti Kuusisto was also supported by the Academy of Finland project *Theory of computational logics*, grant numbers 324435, 328987 (to December 2021) and 352419, 352420 (from January 2022). The author names of this article have been ordered on the basis of alphabetic order.

References

[1] P. Barceló, M. Monet, J. Pérez, and B. Subercaseaux. 2020. Model interpretability through the lens of computational complexity. In *Advances in Neural Information Processing Systems*. H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, (Eds.) Vol. 33. Curran Associates, Inc., 15487–15498.

[2] S. Bassan and G. Katz. 2023. Towards formal XAI: formally approximate minimal explanations of neural networks. In *Tools and Algorithms for the Construction and Analysis of Systems*. S. Sankaranarayanan and N. Sharygina, (Eds.) Springer Nature Switzerland, Cham, 187–207. ISBN: 978-3-031-30823-9.

[3] P. Blackburn, M. de Rijke, and Y. Venema. 2001. *Modal Logic. Cambridge Tracts in Theoretical Computer Science*. Vol. 53. Cambridge University Press. ISBN: 978-1-10705088-4. DOI: 10.1017/CBO9781107050884.

[4] B. Bogaerts, T. Janhunen, and S. Tasharrofi. 2016. Stable-unstable semantics: beyond NP with normal logic programs. *Theory Pract. Log. Program.*, 16, 5-6, 570–586. DOI: 10.1017/S1471068416000387.

[5] L. Breiman. 2001. Random forests. *Machine Learning*, 45, 1, 5–32. DOI: 10.1023/A:1010933404324.
[6] D. Buchfuhrer and C. Umans. 2011. The complexity of Boolean formula minimization. *J. Comput. Syst. Sci.*, 77, 1, 142–153. DOI: 10.1016/j.jcss.2010.06.011.
[7] T. Chen and C. Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*. B. Krishnapuram, M. Shah, A. J. Smola, C. C. Aggarwal, D. Shen, and R. Rastogi, (Eds.) ACM, 785–794.
[8] M. W. Craven and J. W. Shavlik. 1995. Extracting tree-structured representations of trained networks. In *Advances in Neural Information Processing Systems 8, NIPS, Denver, CO, USA, November 27-30, 1995*. D. S. Touretzky, M. Mozer, and M. E. Hasselmo, (Eds.) MIT Press, 24–30.
[9] T. Eiter and G. Gottlob. 1995. On the computational cost of disjunctive logic programming: propositional case. *Ann. Math. Artif. Intell.*, 15, 3-4, 289–323. DOI: 10.1007/BF01536399.
[10] M. Gebser, T. Janhunen, R. Kaminski, T. Schaub, and S. Tasharrofi. 2016. Writing declarative specifications for clauses. In *JELIA*. L. Michael and A. C. Kakas, (Eds.), 256–271. DOI: 10.1007/978-3-319-48758-8_17.
[11] L. Grinsztajn, E. Oyallon, and G. Varoquaux. 2022. Why do tree-based models still outperform deep learning on typical tabular data? In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*. https://openreview.net/forum?id=Fp7__phQszn.
[12] R. Jaakkola, T. Janhunen, A. Kuusisto, M. F. Rankooh, and M. Vilander. 2022. Explainability via short formulas: the case of propositional logic with implementation. (2022). https://arxiv.org/abs/2209.01403v2.
[13] R. Jaakkola, T. Janhunen, A. Kuusisto, M. F. Rankooh, and M. Vilander. 2022. Explainability via short formulas: the case of propositional logic with implementation. In *RCRA 2022* (CEUR Workshop Proceedings). Vol. 3281. CEUR-WS.org, 64–77.
[14] R. Jaakkola, T. Janhunen, A. Kuusisto, M. F. Rankooh, and M. Vilander. 2023. Short Boolean formulas as explanations in practice. In *JELIA 2023*. Springer, 90–105.
[15] T. Janhunen. 2022. Implementing stable-unstable semantics with ASPTOOLS and Clingo. In *PADL*. J. Cheney and S. Perri, (Eds.), 135–153. DOI: 10.1007/978-3-030-94479-7_9.
[16] C. Ji and A. Darwiche. 2023. A new class of explanations for classifiers with non-binary features. In *Logics in Artificial Intelligence*. S. Gaggl, M. V. Martinez, and M. Ortiz, (Eds.) Springer Nature Switzerland, Cham, 106–122. ISBN: 978-3-031-43619-2.
[17] S. M. Lundberg and S.-I. Lee. 2017. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*. I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, (Eds.) Vol. 30. Curran Associates, Inc.
[18] J. Marques-Silva and A. Ignatiev. 2022. Delivering trustworthy AI through formal XAI. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36, 11, (June 2022), 12342–12350. DOI: 10.1609/aaai.v36i11.21499.
[19] A. Pluska, P. Welke, T. Gärtner, and S. Malhotra. 2024. Logical distillation of graph neural networks. In *Proceedings of the 21st International Conference on Principles of Knowledge Representation and Reasoning, KR 2024, Hanoi, Vietnam. November 2-8, 2024*. P. Marquis, M. Ortiz, and M. Pagnucco, (Eds.) DOI: 10.24963/KR.2024/86.
[20] M. T. Ribeiro, S. Singh, and C. Guestrin. 2016. "why should i trust you?": explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (KDD '16). Association for Computing Machinery, San Francisco, California, USA, 1135–1144. ISBN: 9781450342322. DOI: 10.1145/2939672.2939778.
[21] A. Shih, A. Choi, and A. Darwiche. 2018. A symbolic approach to explaining Bayesian network classifiers. In *IJCAI*. J. Lang, (Ed.), 5103–5111. DOI: 10.24963/ijcai.2018/708.
[22] R. Shwartz-Ziv and A. Armon. 2022. Tabular data: deep learning is not all you need. *Inf. Fusion*, 81, 84–90.
[23] P. Simons, I. Niemelä, and T. Soininen. 2002. Extending and implementing the stable model semantics. *Artif. Intell.*, 138, 1-2, 181–234. DOI: 10.1016/S0004-3702(02)00187-X.
[24] C. Umans. 2001. The minimum equivalent DNF problem and shortest implicants. *J. Comput. Syst. Sci.*, 63, 4, 597–611. DOI: 10.1006/jcss.2001.1775.

Received 17 October 2024; revised 2 May 2025; accepted 5 May 2025