Proceedings of the 22nd International Conference on Principles of Knowledge Representation and Reasoning
Main Track

On the Complexity of Global Necessary Reasons to Explain Classification

Marco Calautti[1] , **Enrico Malizia**[2] , **Cristian Molinaro**[3]
[1]University of Milano
[2]University of Bologna
[3]University of Calabria
marco.calautti@unimi.it, enrico.malizia@unibo.it, c.molinaro@dimes.unical.it

Abstract

Explainable AI has garnered considerable attention in recent years, as understanding the reasons behind decisions made by AI systems is crucial for their successful adoption. Explaining classifiers' behavior is one prominent problem. Work in this area has proposed notions of both *local* and *global* explanations, where the former are concerned with explaining a classifier's behavior for a specific instance, while the latter are concerned with explaining the overall classifier's behavior regardless of any specific instance. In this paper, we focus on global explanations, and explain classification in terms of "minimal" necessary conditions for the classifier to assign a specific class to a generic instance. We carry out a thorough complexity analysis of the problem for natural minimality criteria and important families of classifiers considered in the literature.

1 Introduction

Explainable AI (XAI) has become a very active research area in the latest years. Being able to explain AI systems' behavior is crucial for their successful adoption, and this becomes even more important in critical domains, such as healthcare and finance, where AI decisions impact people's life. Among different interesting problems in XAI, explaining classifiers' decisions has attracted significant attention (Baehrens et al. 2010; Marques-Silva and Ignatiev 2022; Marques-Silva 2024).

Work in this area has proposed notions to explain classifiers' behavior *on a specific feature vector (instance)*, providing so-called *local* explanations, as well as notions to explain the overall classifier behavior *regardless of any particular instance*, providing so-called *global* explanations. For both notions, a key issue is to analyze the computational complexity of the problems at hand, to understand how to approach the development of algorithmic solutions and their inherent limits. In XAI, there has been an extensive body of work addressing complexity issues (Bassan, Amir, and Katz 2024; Ordyniak, Paesani, and Szeider 2023; Arenas et al. 2023; Cooper and Marques-Silva 2023; Cooper and Amgoud 2023; Carbonnel, Cooper, and Marques-Silva 2023; Huang et al. 2022; Audemard et al. 2022b; de Colnet and Marquis 2022; Barceló et al. 2020a; Marques-Silva et al. 2020).

This paper falls within this ongoing research stream. Specifically, we deal with global explanations, and focus on so-called *global necessary reasons*, defined as conditions that instances must satisfy in order to be classified with a class of interest.

This notion has been considered by Ignatiev, Narodytska, and Marques-Silva (2019b), where an interesting relationship with another kind of global explanation is shown. However, this is all we know about global necessary reasons, from a technical point of view. [1]

On the other hand, global necessary reasons offer critical insights into classifiers' behavior for diverse purposes. When the class of interest is a desired outcome, a global necessary reason identifies conditions that must be necessarily met by any instance to achieve the desired prediction. Conversely, if the class is undesirable, a global necessary reason indicates how to avoid that class, as violating the condition provided by the global necessary reason *always* leads to a different classification. Global necessary reasons also help discover biases in the classifier, e.g., a global necessary reason stating that a person must be male to obtain a loan unveils a bias.

Thus, the goal of this paper is to deepen the study of global necessary reasons, making several steps forward. Specifically, we start by introducing a logic-based language to express global necessary reasons, as logic offers formal guarantees of rigor and has proven to be well-suited for explainability purposes—see, e.g., (Marques-Silva 2022; Marques-Silva and Ignatiev 2022; Darwiche 2023; Marques-Silva 2024).

As different global necessary reasons may convey different amount of information, we consider a notion of "minimality" to identify the most informative global necessary reasons.

We then provide a systematic complexity analysis of key problems related to global necessary reasons. In particular, given a classifier $\mathcal{M}$, a class c of interest, and a logical expression ϕ, we study the problems of checking whether ϕ is an arbitrary (resp., minimal) global necessary reason for why $\mathcal{M}$ classifies instances with c. We analyze the complexity of such problems for important families of classifiers, namely, binary decision diagrams (BDDs) and their subclass of decision trees (DTs), perceptrons, and multilayer perceptrons (MLPs), (see, e.g., Barceló et al. 2020a), and common min-

[1]We point out that another notion of "global necessary reason" has been considered by Bassan, Amir, and Katz (2024), who have also provided a complexity analysis. However, their notion is fundamentally different from the one we consider, as we thoroughly discuss in Section 5.

imality criteria, namely, cardinality and set-inclusion (see, e.g., Cooper and Marques-Silva 2023). The classifiers above are frequently mentioned in the literature as being at the extremities of the interpretability spectrum, and form the basis of more advanced classifiers. Indeed, these classifiers have been the focus of other foundational works—e.g., see (Bassan, Amir, and Katz 2024), (Arenas et al. 2021), and (Barceló et al. 2020a).

The complexity results we derive provide several interesting insights into (minimal) global necessary reasons. Somewhat surprisingly, the two minimality criteria (namely, cardinality and set-inclusion) turned out to lead to the same family of explanations, regardless of the considered classifiers. As a consequence, the complexity does not change across the two minimality criteria for all classifier families here considered. Moreover, the complexity does not increase when minimality is taken into account for perceptrons, DTs, and BDDs, while minimality increases the complexity for MLPs. More precisely, the problems we consider are in L (i.e., solvable in logarithmic space) for perceptrons and DTs, and NL-complete for BDDs. On the other hand, for MLPs, we show co-NP-completeness and D^{P}-completeness for arbitrary (i.e., not necessarily minimal) and minimal global necessary reasons, respectively.

Besides being interesting in their own right, the above complexity results also allow us to draw key insights on the complexity of *computing* minimal global necessary reasons. In particular, we show that (1) computing minimal global necessary reasons is at least as hard as the decision problem for minimal global necessary reasons, and (2) minimal global necessary reasons can be computed efficiently, i.e., in polynomial time, given access to a subroutine (a.k.a. oracle) solving the decision problem for arbitrary global necessary reasons. Such results are significant in that they imply minimal global necessary reasons can be computed very efficiently for perceptrons, DTs, and BDDs, since the decision problem for arbitrary global necessary reasons is in L for perceptrons and DTs, and NL-complete for BDDs, and thus can be solved on highly-parallel machines—see, e.g., (Arora and Barak 2009; Greenlaw, Hoover, and Ruzzo 1995). Furthermore, in the case of MLPs, our complexity results imply that minimal global necessary reasons can be computed by resorting to SAT solvers, which have proven to be very efficient at solving computationally hard problems (even co-NP-complete and D^{P}-complete ones) concerning the computation of classifiers' explanations (Marques-Silva 2022). Moreover, since computing minimal global necessary reasons is at least as hard as the decision version of the problem that we consider, which is D^{P}-hard for MLPs, minimal global necessary reasons cannot be computed more efficiently in polynomial time (unless PTIME = NP).

Because of the implications discussed above, we believe this paper is a first foundational step towards a full understanding and the adoption of global necessary reasons to explain classification.

2 Preliminaries

Classification In this paper, n denotes the number of features of the instance domain, hence n is assumed to be a (strictly) positive integer. An *n-instance* is an n-dimensional binary vector $\mathbf{x} = (x_1, \ldots, x_n) \in \{0,1\}^n$. We denote by $\mathbf{x}[i]$ the value x_i from $\mathbf{x}$, for $1 \leq i \leq n$. An *n-feature (binary) classifier* $\mathcal{M}$ can be modeled by a function $\mathcal{M}\colon \{0,1\}^n \to \{0,1\}$ mapping n-instances $\mathbf{x}$ to the binary class $\mathcal{M}(\mathbf{x})$. Restricting the analysis to binary classifiers makes our framework cleaner, while still covering several relevant practical scenarios. While our hardness results immediately apply to more general settings, such as those involving instances over the real numbers, a precise characterization of the associated complexity needs a dedicated analysis, which we leave as a interesting direction for future work.

We study the following key families of classifiers.

Binary Decision Diagram (BDD) and Decision Tree (DT) A *(free) n-feature binary decision diagram*, or n-BDD for short, is defined by a rooted directed acyclic graph (DAG) $\mathcal{G} = (V, E, \lambda, \eta)$, where λ and η are node- and edge-labeling functions, respectively. We recall that in a rooted DAG, there is a *single* node without incoming edges, which is the root, and the nodes without outgoing edges are called sinks. The n-BDD $\mathcal{G}$ is such that:

- each sink of $\mathcal{G}$ is labeled with either 1 or 0;
- each internal node (i.e., a node that is not a sink) is labeled with an element from $\{1, \ldots, n\}$;
- each internal node has two outgoing edges, one labeled with 1 and the other labeled with 0;
- no two internal nodes on a path of $\mathcal{G}$ originating from the root have the same label.

$\mathcal{G}$ classifies an n-instance $\mathbf{x}$ as c, denoted $\mathcal{G}(\mathbf{x}) = c$, iff there is a path $u_1, \ldots, u_m$ from the root of $\mathcal{G}$ to a sink of $\mathcal{G}$ such that u_m is labeled with c, and, for each i with $1 \leq i \leq m-1$, if u_i is labeled with j, then the edge (u_i, u_{i+1}) of $\mathcal{G}$ is labeled with $\mathbf{x}[j]$; note that by definition, if $\mathcal{G}(\mathbf{x}) = c$ there always exists exactly one path witnessing this. We use BDD to denote the family of all n-BDDs, for all $n > 0$.

An n-feature decision tree, or n-DT for short, is a special case of an n-BDD where the underlying DAG is a tree. We use DT to denote the family of all n-DTs, for all $n > 0$.

Perceptron An *n-feature perceptron* $\mathcal{S}$, or n-perceptron for short, is defined by a pair $\mathcal{S} = (\mathbf{w}, b)$, where $\mathbf{w} = (w_1, \ldots, w_n) \in \mathbb{Q}^n$ and $b \in \mathbb{Q}$ are the perceptron's weights and bias, respectively. Then, $\mathcal{S}$ classifies an n-instance $\mathbf{x}$ as '1', denoted by $\mathcal{S}(\mathbf{x}) = 1$, iff $\mathbf{x} \cdot \mathbf{w} + b \geq 0$; otherwise $\mathcal{S}$ classifies $\mathbf{x}$ by '0', which is denoted as $\mathcal{S}(\mathbf{x}) = 0$. Equivalently, $\mathcal{S}$ can be seen as an object which receives as input the values $(x_1, \ldots, x_n)$, weighed via the weights $(w_1, \ldots, w_n)$, and outputs the value $\mathcal{S}(\mathbf{x}) = \mathit{step}(\mathbf{x} \cdot \mathbf{w} + b)$, where $\mathit{step}(\cdot)$ is the Heaviside step function, which is defined as $\mathit{step}(x) \coloneqq 0$ if $x < 0$, and $\mathit{step}(x) \coloneqq 1$ if $x \geq 0$. We use PRC to denote the family of all n-perceptrons, for all $n > 0$.

Multilayer Perceptron (MLP) An *n-feature multilayer perceptron* $\mathcal{N}$, or n-MLP for short, is defined by a tuple $\mathcal{N} = (\mathbf{W}^1, \ldots, \mathbf{W}^k, \mathbf{b}^1, \ldots, \mathbf{b}^k, f^1, \ldots, f^k)$, where $k > 0$ is the number of layers of $\mathcal{N}$. Moreover, for each layer i with $1 \leq i \leq k$, assuming d_i denotes the number of neurons on the i-th layer, and $d_0 = n$ is the size of the input of $\mathcal{N}$:

- $\mathbf{W}^i \in \mathbb{Q}^{d_{i-1} \times d_i}$ is the i-th layer weight matrix of $\mathcal{N}$;

- $\mathbf{b}^i \in \mathbb{Q}^{d_i}$ is the i-th layer bias vector of $\mathcal{N}$;
- $f^i \colon \mathbb{Q}^{d_i} \to \mathbb{Q}^{d_i}$ is the i-th layer (d_i-dimensional) activation function of $\mathcal{N}$.

Since we deal with *binary* classifiers, here $d_k = 1$. We assume that the activation function of the non-output neurons (i.e., the function f^i, with $1 \leq i \leq k-1$) is the ReLU function[2] $relu(x) \coloneqq max(0, x)$, whereas the activation function f^k of the single output neuron is the Heaviside step function (see, e.g., Barceló et al.; Marques-Silva 2020b; 2022).

Given an n-instance $\mathbf{x}$, we inductively define

$$\mathbf{h}^i \coloneqq f^i(\mathbf{h}^{i-1}\mathbf{W}^i + \mathbf{b}^i), \quad \text{for each } i, \text{ with } 1 \leq i \leq k,$$

where $\mathbf{h}^0 \coloneqq \mathbf{x}$. Then, $\mathcal{N}$ classifies an n-instance $\mathbf{x}$ as c, denoted by $\mathcal{N}(\mathbf{x}) = c$, iff $\mathbf{h}^k = c$. We use MLP to denote the family of all n-MLPs, for all $n > 0$.

We point out that, in this paper, we do not deal with the task of training classifiers. Instead, we are interested in explaining the behavior of (already learned) classifiers—explanations in this setting are often called "post-hoc" explanations. For this reason, the classifiers will be assumed to be given as input with all the (already trained) parameters characterizing them.

Computational Complexity We briefly recall the complexity classes that we encounter. PTIME and L are the classes of all decision problems that can be decided in polynomial time and logarithmic space, respectively, by a deterministic Turing machine (the space constraint is over the work tape). NP and NL are the classes of all decision problems that can be decided in polynomial time and logarithmic space, respectively, by a nondeterministic Turing machine. co-NP is the complement class of NP, where 'yes' and 'no' answers are interchanged. We recall that L, NL, and PTIME are closed under complement. The class $\mathrm{D^P} = \mathrm{NP} \wedge \text{co-NP}$ is the class of all decision problems that are the intersection of a problem in NP and a problem in co-NP. The inclusion relationships (which are all currently believed to be strict) between the above complexity classes are: $\mathrm{L} \subseteq \mathrm{NL} \subseteq \mathrm{PTIME} \subseteq \mathrm{NP}, \text{co-NP} \subseteq \mathrm{D^P}$.

We refer the reader to any textbook on the topic, such as (Arora and Barak 2009), for a broader introduction.

3 Global Necessary Reasons

In this section, we consider the notion of a *global necessary reason* as a way to explain classifiers' behavior. Intuitively, for a classifier and a class of interest, the idea is to provide a condition that must be necessarily satisfied (by any instance) for the classifier to assign the class. Additionally, in order to identify the most informative global necessary reasons, we will be interested in "minimal" ones, as defined later on. This section also introduces the (decision) problems whose complexity will be analyzed in the rest of the paper.

To express global necessary reasons, we use the logical language defined below. Let $\mathcal{L}[n]$ denote the set of all expressions, called *conditions*, of the form $\bigwedge_{i=1}^{m} \ell_i$, where $m \geq 0$, each ℓ_i is a *literal* of the form $t \lozenge t'$, with $\lozenge \in \{=, \neq\}$, and t, t' are *terms* from the set $\{0, 1\} \cup \{v_i \mid 1 \leq i \leq n\}$, where each v_i is a Boolean variable, i.e., over the values $\{0, 1\}$, associated to the i-th feature. Notice that a condition can be empty (i.e., $m = 0$); we use $\top$ to denote such a condition.

Thus, conditions are conjunctions of (in)equalities. As customary when designing a formal language, we strove for a balance between expressiveness and complexity, and our choice is based on the following considerations.

Conjunctions are widely deemed easy to interpret—indeed, most related work employs (even simpler forms of) conjunctions to express explanations (cf. Section 5). However, slight extensions to the language can make it much less interpretable. Arguably, a natural extension is to allow disjunctions, leading to a more expressive language which even enables to precisely characterize a classifier's behavior. That is, for an n-feature classifier $\mathcal{M}$ and a class $c \in \{0, 1\}$ of interest, if $\mathbf{x}_1, \ldots, \mathbf{x}_m$ are all the n-instances classified with c by $\mathcal{M}$, one can always devise the Boolean expression of the form $\phi_1 \vee \cdots \vee \phi_m$, where each ϕ_i is a condition encoding $\mathbf{x}_i$. Clearly, an n-instance $\mathbf{x}$ is classified with c by $\mathcal{M}$ iff $\mathbf{x}$ satisfies the above expression. It is clear that this high level of expressiveness comes at the cost of providing complex expressions (e.g., unavoidably exponentially large) which would be hard to understand for a user, and at the same time it poses computational challenges.

Hence, we designed our language so that it goes beyond simple conjunctions of feature-value pairs adopted by Ignatiev, Narodytska, and Marques-Silva (2019b)[3], but at the same time supports a richer set of logical constructs that help better explain classifiers' behaviour without hindering interpretability and computational complexity. In fact, our language can express *relationships* among features by means of (in)equalities between feature variables, which is usually deemed as an important feature for explaining classifiers. From the technical point of view, (in)equalities between variables allow for a controlled form of disjunction, such as specifying alternatives between values assigned to features—e.g., $v_1 = v_2$ expresses the two alternatives $v_1 = v_2 = 0$ and $v_1 = v_2 = 1$. Computationally, as already mentioned in the introduction, the complexity results we derive justify the choice of the language also in terms of practical applicability. From a syntactical perspective, standard propositional logic might be used to express conditions. However, we chose to employ a dedicated syntax, as this allows us to express formulas that are more concise and easier to interpret.

We now proceed by introducing some notions related to our language, and then define global necessary reasons. Given an n-instance $\mathbf{x}$ and a condition $\phi \in \mathcal{L}[n]$, let $\phi[\mathbf{x}]$ denote the condition obtained from ϕ by replacing every v_i in ϕ with $\mathbf{x}[i]$. We say that $\mathbf{x}$ *satisfies* ϕ, denoted by $\mathbf{x} \models \phi$, iff $\phi[\mathbf{x}]$ is true under the usual semantics of comparison operators and Boolean logical connectives—in such a case, we also say that $\mathbf{x}$ is a *model* of ϕ. When $\phi = \top$, every n-instance trivially satisfies ϕ. We further define $[\![\phi]\!] = \{\mathbf{x} \in \{0, 1\}^n \mid \mathbf{x} \models \phi\}$, i.e., the set of all n-instances satisfying ϕ. Moreover, for two conditions $\phi, \psi \in \mathcal{L}[n]$, we say that ϕ *entails* ψ, denoted as

[2] As for the impact of this choice on our results, our upper bounds hold as far as evaluating the MLP over an instance is feasible in polynomial time, while the lower bounds immediately hold for many other activation functions that simply generalize ReLUs.

[3] To the best of our knowledge, this is the only work considering the same notion of explanation we consider.

$\phi \models \psi$, iff $[\![\phi]\!] \subseteq [\![\psi]\!]$.

As mentioned, for a classifier $\mathcal{M}$, and a class $c \in \{0,1\}$, a global necessary reason must provide a condition that is satisfied by every instance that is classified with c by $\mathcal{M}$.

Definition 1 (Global necessary reasons)**.** Let $\mathcal{M}$ be an n-feature classifier, and let $c \in \{0,1\}$ be a class. A *global necessary reason* for c w.r.t. $\mathcal{M}$ is a condition $\phi \in \mathcal{L}[n]$ where

$$\forall \mathbf{x} \in \{0,1\}^n,\ (\mathcal{M}(\mathbf{x}) = c) \rightarrow (\mathbf{x} \models \phi)\,.$$

For an n-feature classifier $\mathcal{M}$ and a class $c \in \{0,1\}$, with an abuse of notation, we let $[\![\mathcal{M}, c]\!]$ be the set of all n-instances $\mathbf{x}$ such that $\mathcal{M}(\mathbf{x}) = c$. Clearly, a condition ϕ is a global necessary reason for c w.r.t. $\mathcal{M}$ iff $[\![\mathcal{M}, c]\!] \subseteq [\![\phi]\!]$. The latter implies that a conjunction of global necessary reasons for c w.r.t. $\mathcal{M}$ is a global necessary reason for c w.r.t. $\mathcal{M}$.

Different global necessary reasons might convey different amounts of information. As an example, $\top$ is always a global necessary reason, but it does not provide useful information. In this regard, we point out that every global necessary reason ϕ for a class c w.r.t. a classifier $\mathcal{M}$ "over-approximates" the assignment of c by $\mathcal{M}$ in that $[\![\mathcal{M}, c]\!] \subseteq [\![\phi]\!]$. Thus, a criterion to identify the most informative global necessary reasons should select the ones for which such over-approximation is as small as possible. Such most informative global necessary reasons are the "minimal" ϕ such that $[\![\mathcal{M}, c]\!] \subseteq [\![\phi]\!]$. Formally, minimality is defined w.r.t. an arbitrary preorder $\preccurlyeq$, i.e., a reflexive and transitive binary relation, over $\mathcal{L}[n]$; $\phi \prec \phi'$ denotes that $\phi \preccurlyeq \phi'$ and $\phi' \not\preccurlyeq \phi$. Then, minimal global necessary reasons are naturally defined as follows.

Definition 2 (Minimal global necessary reason)**.** Let $\mathcal{M}$ be an n-feature classifier, and let $c \in \{0,1\}$ be a class. A global necessary reason $\phi \in \mathcal{L}[n]$ for c w.r.t. $\mathcal{M}$ is $\preccurlyeq$-*minimal* iff there is no global necessary reason $\phi' \in \mathcal{L}[n]$ for c w.r.t. $\mathcal{M}$ such that $\phi' \prec \phi$.

We consider two concrete common preorders (see, e.g., Cooper and Marques-Silva 2023), which we use to compare conditions w.r.t. their models:

- Model cardinality ($\leq$): Given two conditions ϕ and ϕ', we write $\phi \leq \phi'$ iff $|[\![\phi]\!]| \leq |[\![\phi']\!]|$.
- Model subset ($\subseteq$): Given two conditions ϕ and ϕ', we write $\phi \subseteq \phi'$ iff $[\![\phi]\!] \subseteq [\![\phi']\!]$ (or, equivalently, $\phi \models \phi'$).

As a simple example, for the two conditions $\phi := (v_1 = 1 \wedge v_2 = v_3)$ and $\phi' := (v_1 = 1)$, we have that both $\phi \leq \phi'$ and $\phi \subseteq \phi'$ hold. Obviously, ϕ makes a more specific statement than ϕ' by additionally requiring v_2 and v_3 to assume the same value, and we consider such statements more informative (as long as they are global necessary reasons).

An interesting property is that, for each $\preccurlyeq \in \{\leq, \subseteq\}$, any two $\preccurlyeq$-minimal global necessary reasons ϕ and ϕ' for a class c w.r.t. a classifier $\mathcal{M}$ have the same set of models, i.e., $[\![\phi]\!] = [\![\phi']\!]$. If this were not the case, then the conjunction $\phi \wedge \phi'$ would also be a global necessary reason preceding both ϕ and ϕ' w.r.t. $\preccurlyeq$, i.e., both $\phi \wedge \phi' \prec \phi$ and $\phi \wedge \phi' \prec \phi'$ would hold. Thus, any two $\preccurlyeq$-minimal global necessary reasons for a class c w.r.t. a classifier $\mathcal{M}$ have the same set of models, but can be (syntactically) different.

IsNecessary				IsMinNecessary				
PRC	DT	BDD	MLP	PRC	DT	BDD	MLP	
in L	in L	NL	co-NP	in L in L	in L in L	NL NL	D^P D^P	$\leq$ $\subseteq$

Table 1: Summary of the complexity of IsNecessary[C] and of IsMinNecessary[C, $\preccurlyeq$], for each family of classifiers $\mathsf{C} \in \{\mathsf{PRC}, \mathsf{DT}, \mathsf{BDD}, \mathsf{MLP}\}$, and preorder $\preccurlyeq \in \{\leq, \subseteq\}$. All non-"in" entries are completeness results.

As customary in complexity analysis, we focus on decision problems—then, in Section 4.4, we will show how our results provide insights into the complexity of the problem of *computing* minimal global necessary reasons. In particular, for a family $\mathsf{C} \in \{\mathsf{PRC}, \mathsf{DT}, \mathsf{BDD}, \mathsf{MLP}\}$ of classifiers, and a preorder $\preccurlyeq \in \{\leq, \subseteq\}$, we study the following problem:

Problem :	IsMinNecessary[C, $\preccurlyeq$]
Input :	An n-feature classifier $\mathcal{M} \in \mathsf{C}$, a class $c \in \{0,1\}$, and a condition $\phi \in \mathcal{L}[n]$.
Question :	Is ϕ a $\preccurlyeq$-minimal global necessary reason for c w.r.t. $\mathcal{M}$?

As we will see in the following, to study the complexity of the problem above, we will also need to focus on the complexity of deciding whether a condition is a global necessary reason (i.e., not necessarily minimal) for a class c w.r.t. a classifier $\mathcal{M}$. We hence define the following problem (notice how this problem is not parametric in the preorder):

Problem :	IsNecessary[C]
Input :	An n-feature classifier $\mathcal{M} \in \mathsf{C}$, a class $c \in \{0,1\}$, and a condition $\phi \in \mathcal{L}[n]$.
Question :	Is ϕ a global necessary reason for c w.r.t. $\mathcal{M}$?

Remarks: In the rest of the paper, to streamline the presentation, we implicitly assume, unless specified otherwise, that classifiers are over $n > 0$ features, instances and classes are from $\{0,1\}^n$ and $\{0,1\}$, respectively, and conditions and literals are from $\mathcal{L}[n]$. Finally, if ϕ is a global necessary reason for a class c w.r.t. to a classifier $\mathcal{M}$, and $\mathcal{M}$ and c are clear from the context, we may refer to ϕ simply as a global necessary reason, without mentioning $\mathcal{M}$ and c. Finally, as customary, we assume that rationals are encoded as pairs of coprime integers.

4 Complexity of Global Necessary Reasons

In this section, we study the complexity of IsNecessary[C] and IsMinNecessary[C, $\preccurlyeq$], for each family of classifiers $\mathsf{C} \in \{\mathsf{PRC}, \mathsf{DT}, \mathsf{BDD}, \mathsf{MLP}\}$, and preorder $\preccurlyeq \in \{\leq, \subseteq\}$. A summary of the complexity results obtained in this paper is reported in Table 1.

We start by carrying out some observations regarding the problem IsMinNecessary[C, $\preccurlyeq$]. Given a classifier $\mathcal{M}$, a class c, and a condition ϕ, an obvious procedure deciding

whether ϕ is a $\preccurlyeq$-minimal global necessary reason is to first check that ϕ is a global *necessary* reason, and then verify that ϕ is $\preccurlyeq$*-minimal*. The latter can naively be checked by iterating over every possible condition ψ and checking, whenever $\psi \prec \phi$, that ψ is *not* a global necessary reason.

The main issue with the above procedure is that in order to verify that ϕ is $\preccurlyeq$-minimal, the procedure iterates over *all* the possible conditions ψ, which are exponentially many in n. In addition, for each such condition ψ, the procedure must check whether $\psi \prec \phi$, which in turn may require to iterate over the (possibly) exponentially many instances $\mathbf{x}$ such that $\mathbf{x} \models \psi$. We are however able to provide the next characterization, enabling us to greatly simplify the process of checking whether a global necessary reason ϕ is $\preccurlyeq$-minimal; this also enables us to pinpoint the exact complexity of IsMinNecessary$[\mathsf{C}, \preccurlyeq]$ by deriving tight upper bounds.

Lemma 3. *Let $\mathcal{M}$ be a classifier, let c be a class, and let ϕ be a global necessary reason. Then, for each preorder $\preccurlyeq \in \{\leq, \subseteq\}$, ϕ is* not *$\preccurlyeq$-minimal iff there exists a literal ℓ such that $\phi \not\models \ell$ and ℓ is a global necessary reason.*

Proof. ($\Rightarrow$). We show that if ϕ is not $\preccurlyeq$-minimal, then there is a literal ℓ with $\phi \not\models \ell$ and ℓ is a global necessary reason.

Assume that ψ is a global necessary reason for c w.r.t. $\mathcal{M}$ such that $\psi \prec \phi$, and consider the condition $\Gamma = \phi \wedge \psi$.

We start by proving that: (i) Γ is also a global necessary reason for c w.r.t. $\mathcal{M}$, and (ii) $[\![\Gamma]\!] \subsetneq [\![\phi]\!]$. Regarding (i), since both ϕ and ψ are global necessary reasons for c w.r.t. $\mathcal{M}$, $[\![\mathcal{M}, c]\!] \subseteq [\![\phi]\!]$ and $[\![\mathcal{M}, c]\!] \subseteq [\![\psi]\!]$. By this, we have that $[\![\mathcal{M}, c]\!] \subseteq [\![\phi]\!] \cap [\![\psi]\!]$. Since $[\![\Gamma]\!] = [\![\phi \wedge \psi]\!] = [\![\phi]\!] \cap [\![\psi]\!]$, we can conclude that Γ is a global necessary reason for c w.r.t. $\mathcal{M}$. Regarding (ii), since $\Gamma = \phi \wedge \psi$, we have that $[\![\Gamma]\!] \subseteq [\![\phi]\!]$ and $[\![\Gamma]\!] \subseteq [\![\psi]\!]$. Moreover, regardless of $\prec$ actually being $<$ or $\subsetneq$, from $\psi \prec \phi$ it follows that $|[\![\psi]\!]| < |[\![\phi]\!]|$. The latter, together with $[\![\Gamma]\!] \subseteq [\![\psi]\!]$ and $[\![\Gamma]\!] \subseteq [\![\phi]\!]$, imply that the inclusion $[\![\Gamma]\!] \subseteq [\![\phi]\!]$ is actually strict, i.e., $[\![\Gamma]\!] \subsetneq [\![\phi]\!]$.

We now claim that there must be a literal ℓ in ψ such that $\phi \not\models \ell$. Indeed, if this were not the case, i.e., if every literal ℓ in ψ were such that $\phi \models \ell$, then it would mean that $[\![\Gamma]\!] = [\![\phi \wedge \psi]\!] = [\![\phi]\!]$, which cannot be the case, since we have proved that $[\![\Gamma]\!] \subsetneq [\![\phi]\!]$ (see above).

Consider now such a literal ℓ. Since ℓ belongs to ψ, and since $\Gamma = \phi \wedge \psi$, we have that $[\![\Gamma]\!] \subseteq [\![\ell]\!]$. Since we proved that Γ is a global necessary reason for c w.r.t. $\mathcal{M}$, then ℓ is also a global necessary reason for c w.r.t. $\mathcal{M}$. Thus, ℓ is a literal such that $\phi \not\models \ell$ and ℓ is a global necessary reason.

($\Leftarrow$). We show that if there is a literal ℓ such that $\phi \not\models \ell$ and ℓ is a global necessary reason, then ϕ is not $\preccurlyeq$-minimal.

Since ϕ and ℓ are both global necessary reasons for c w.r.t. $\mathcal{M}$, we have that $[\![\mathcal{M}, c]\!] \subseteq [\![\phi]\!]$ and $[\![\mathcal{M}, c]\!] \subseteq [\![\ell]\!]$. Because $[\![\phi \wedge \ell]\!] = [\![\phi]\!] \cap [\![\ell]\!]$, it must be the case that $[\![\mathcal{M}, c]\!] \subseteq [\![\phi \wedge \ell]\!]$, which implies that $\phi \wedge \ell$ is a global necessary reason for c w.r.t. $\mathcal{M}$. Moreover, since $\phi \not\models \ell$, there is an instance in $[\![\phi]\!]$ that does not belong to $[\![\ell]\!]$. The latter, together with the facts that $[\![\phi \wedge \ell]\!] \subseteq [\![\phi]\!]$ and $[\![\phi \wedge \ell]\!] = [\![\phi]\!] \cap [\![\ell]\!]$, imply that $[\![\phi \wedge \ell]\!] \subsetneq [\![\phi]\!]$. Hence, we conclude that $\psi = \phi \wedge \ell$ is a global necessary reason for c w.r.t. $\mathcal{M}$ such that $[\![\psi]\!] \subsetneq [\![\phi]\!]$, by which ϕ is *not* $\subseteq$-minimal. Observe that $[\![\psi]\!] \subsetneq [\![\phi]\!]$ implies $|[\![\psi]\!]| < |[\![\phi]\!]|$, and hence ϕ is also *not* $\leq$-minimal. □

Algorithm 1: A generic algorithm deciding whether a condition is a $\leq$-/$\subseteq$-minimal global necessary reason

Input: An n-feature classifier $\mathcal{M}$, a class $c \in \{0, 1\}$, and a condition $\phi \in \mathcal{L}[n]$
Output: accept, if ϕ is a $\preccurlyeq$-minimal global necessary reason for c w.r.t. $\mathcal{M}$; reject, otherwise

Procedure `MinNecessary`$(\mathcal{M}, c, \phi)$ **:**
1 **if** *ϕ is not a global necessary reason for c w.r.t. $\mathcal{M}$* **then return** reject
2 **foreach** *literal* $\ell \in \mathcal{L}[n]$ **do**
3 **if** $\phi \not\models \ell$ **then**
4 **if** *ℓ is a global necessary reason for c w.r.t. $\mathcal{M}$* **then return** reject
5 **return** accept

Very interestingly, Lemma 3 implies that $\leq$- and $\subseteq$-minimal global necessary reasons are actually equivalent notions—notice that the property stated in the lemma does not distinguish between the two minimality criteria. Clearly, every $\leq$-minimal global necessary reason is also a $\subseteq$-minimal global necessary reason. The more interesting question is why every $\subseteq$-minimal global necessary reason ϕ is also a $\leq$-minimal global necessary reason. Assume, towards a contradiction, that this is not the case. Then there would be a global necessary reason ψ with strictly fewer models than ϕ. Notice that $\psi \wedge \phi$ (i) is a global necessary reason, (ii) has strictly fewer models than ϕ, (iii) its models are a subset of those of ϕ. This means that $\psi \wedge \phi$ is a global necessary reason and its models are a proper subset of those of ϕ (this is implied by (ii) and (iii)), contradicting that ϕ was a $\subseteq$-minimal global necessary reason. Note that (ii) and (iii) hold for any logical language encoding conditions, while (i) holds as long as the language we employ to encode conditions is closed under logical conjunction, i.e., given two conditions ψ and ϕ, then $\psi \wedge \phi$ is still a condition. Since our language $\mathcal{L}[n]$ is closed under logical conjunction, we obtain the equivalence. Hence, closure under logical conjunction is the property that guarantees equivalence of the two notions.

As already observed, a conjunction of global necessary reasons is a global necessary reason. Thus, the conjunction ϕ of *all* literals that are individually global necessary reasons is a global necessary reason. A consequence of Lemma 3 is that ϕ is indeed a $\preccurlyeq$-minimal global necessary reason. The above observations provide useful insights on how to compute $\preccurlyeq$-minimal global necessary reasons, as we will discuss in Section 4.4.

Furthermore, Lemma 3 suggests a simple procedure to check whether a condition ϕ is a $\preccurlyeq$-minimal global necessary reason (see Algorithm 1): first, we check that ϕ is a global necessary reason, and then we check that ϕ is $\preccurlyeq$-minimal by verifying that there is *no* literal ℓ for which $\phi \not\models \ell$ and such that ℓ is a global necessary reason; by what we observed above, notice how the algorithm does *not* depend on the preorder $\leq$ or $\subseteq$.

Algorithm 1 provides a generic framework to analyze the complexity of IsMinNecessary$[\mathsf{C}, \preccurlyeq]$, for $\mathsf{C} \in \{\mathsf{PRC},$

$\mathsf{DT}, \mathsf{BDD}, \mathsf{MLP}\}$ and $\preccurlyeq \in \{\leq, \subseteq\}$. Observe that, with n features, there are only $O(n^2)$ literals to consider at line 2. Thus, the algorithm's complexity is essentially related to the complexity of checking whether a given condition or literal is or is not a global necessary reason (lines 1 and 4), and checking, for a given condition ϕ and a literal ℓ, whether $\phi \not\models \ell$ (line 3).

Notice that the complexity of deciding whether ϕ or ℓ are global necessary reasons depends on the specific family of classifiers considered, whereas the complexity of deciding $\phi \not\models \ell$ does not. We hence focus first on the latter problem, and we will study the former in the next sections, where we will consider each family of classifiers in turn. We now show that deciding $\phi \not\models \ell$ is an easy task, i.e., feasible in logspace.

Theorem 4. *Let ϕ be a condition, and let ℓ be a literal. Deciding whether $\phi \models \ell$ (or $\phi \not\models \ell$) is in* L.

The above complexity result is obtained by reducing in logspace the problem of deciding whether $\phi \models \ell$ to the problem of deciding the satisfiability of 2CNF formulas of a restricted form. This simpler form, which guarantees that 2CNF formulas have a certain symmetry property, allows us to adapt the existing satisfiability algorithm for arbitrary 2CNF formulas and obtain an algorithm executing in logspace.

With the above result in place, in the following we study the complexity of deciding whether a condition is a global necessary reason, and show how this analysis, together with Theorem 4, allows us to obtain the results in Table 1.

4.1 The Case of Perceptrons

We start considering the family of classifiers based on perceptrons. As already discussed in the previous section, we first need to understand the complexity of IsNecessary[PRC].

Theorem 5. IsNecessary[PRC] *is in* L.

As the complexity class L is closed under complement, we obtain the result above by showing membership in L of the complement problem IsNotNecessary[PRC]: for a perceptron $\mathcal{S}$, a class c, and condition ϕ, decide whether ϕ is *not* a global necessary reason. The condition ϕ can be shown not being a global necessary reason by finding an instance $\mathbf{x}$ such that $\mathcal{S}(\mathbf{x}) = c$ and $\mathbf{x} \not\models \phi$. Intuitively, we show that the latter can be achieved by encoding within $\mathcal{S}$ the *opposite* of a literal from ϕ, and by then showing that for such a modified perceptron there exists an instance classified as c.

Remember now that the generic Algorithm 1 decides IsMinNecessary[C, $\preccurlyeq$] also for C = PRC, and for each $\preccurlyeq \in \{\leq, \subseteq\}$. Since by Theorem 5 and Theorem 4, lines 1, 3 and 4 of Algorithm 1 are feasible in logspace, and each literal ℓ considered in each iteration can be stored in logarithmic space, the entire procedure can be carried out in logspace, when considering perceptrons. The next result follows.

Theorem 6. IsMinNecessary[PRC, $\preccurlyeq$] *is in* L*, for each* $\preccurlyeq \in \{\leq, \subseteq\}$.

4.2 The Case of BDDs and DTs

In this section we first consider the family of classifiers based on BDDs, and then focus on its subclass consisting of DTs.

The Case of BDDs As already done for perceptrons, we first analyze the complexity of checking whether a given condition is a global necessary reason. We focus on the complement problem, which we call IsNotNecessary[BDD], as the complexity result pertains NL, which is closed under complement.

More specifically, we pinpoint an interesting characterization for the conditions from $\mathcal{L}[n]$ that are *not* global necessary reasons, when focusing on BDDs. This property allows to devise a nondeterministic procedure with low space usage, i.e., logarithmic, which we report as Algorithm 2, and that decides whether a condition is *not* a global necessary reason for a BDD. We discuss this next.

Let $\mathcal{G} = (V, E, \lambda, \eta)$ be a BDD, and let $\Pi(\mathcal{G}, c)$ denote the set of all paths from the root of $\mathcal{G}$ to a sink of $\mathcal{G}$ labeled with the class c. For a path $\pi = u_1, \ldots, u_m \in \Pi(\mathcal{G}, c)$, intuitively we define ϕ_π as the condition assigning to each feature f_i, labeling a node u_i of π, the value a_i that the path π assigns to f_i—remember that this value a_i is the label of the edge connecting u_i to u_{i+1} in π. More formally,

$$\phi_\pi = \bigwedge_{i=1}^{m-1} (v_{f_i} = a_i),$$

where, for each i with $1 \leq i \leq m-1$, $f_i = \lambda(u_i)$, and $a_i = \eta((u_i, u_{i+1}))$. Our characterization follows.

Lemma 7. *A condition ϕ is* not *a global necessary reason for a class c w.r.t. a BDD $\mathcal{G}$ iff there exists a path $\pi \in \Pi(\mathcal{G}, c)$ and a literal ℓ such that $\phi_\pi \not\models \ell$ and $\phi \models \ell$.*

Proof. Recall that $[\![\mathcal{G}, c]\!]$ is the set of all instances $\mathbf{x}$ such that $\mathcal{G}(\mathbf{x}) = c$. By definition of BDDs, we have that the set of all instances that $\mathcal{G}$ classifies as c coincides with the set of all instances that agree with any of the paths in $\mathcal{G}$ leading to a node labeled with c; by this, $[\![\mathcal{G}, c]\!] = \bigcup_{\pi \in \Pi(\mathcal{G}, c)} [\![\phi_\pi]\!]$.

Hence, ϕ is *not* a global necessary reason for c w.r.t. $\mathcal{G}$ iff there exists a path $\pi \in \Pi(\mathcal{G}, c)$ such that $[\![\phi_\pi]\!] \not\subseteq [\![\phi]\!]$, which is $\phi_\pi \not\models \phi$. Observe that the latter is equivalent to $\mathsf{cl}(\phi_\pi) \not\supseteq \mathsf{cl}(\phi)$, where, for a condition ψ, $\mathsf{cl}(\psi)$ is the set of all literals ℓ such that $\psi \models \ell$. Hence, ϕ is *not* a global necessary reason for c w.r.t. $\mathcal{G}$ iff there exists a path $\pi \in \Pi(\mathcal{G}, c)$ and a literal ℓ such that $\phi_\pi \not\models \ell$ and $\phi \models \ell$, as needed. □

With the above characterization in place, we are now ready to discuss how IsNotNecessary[BDD] is decided by the non-deterministic procedure Algorithm 2. In what follows, a literal is said to be *trivially true* if it is of the form $(0 = 0)$, $(1 = 1)$, $(1 \neq 0)$, $(0 \neq 1)$, or $(v_i = v_i)$, for some $1 \leq i \leq n$. Moreover, since Algorithm 2 is nondeterministic, when we say the procedure accepts its input we mean that there is a way for the procedure to carry out the guesses such that "accept" is returned. We now argue that the procedure is correct; its space complexity will be discussed afterwards.

Correctness. Consider an n-feature BDD $\mathcal{G} = (V, E, \lambda, \eta)$, and a class $c \in \{0, 1\}$. We observe that for any path $\pi \in \Pi(\mathcal{G}, c)$, the condition ϕ_π contains *only* literals of the form $(v_i = a)$, with $a \in \{0, 1\}$, and where each Boolean variable appears at most once. Thus, there is a straightforward approach to test whether $\phi_\pi \not\models \ell$, for some arbitrary

Algorithm 2: A *nondeterminisic* algorithm deciding whether a condition is *not* a global necessary reason for a BDD

Input: An n-feature classifier $\mathcal{G} \in \mathsf{BDD}$, a class $c \in \{0,1\}$, and a condition $\phi \in \mathcal{L}[n]$
Output: accept, if ϕ is *not* a global necessary reason for c w.r.t. $\mathcal{G}$; reject, otherwise

Procedure `NotNecessaryBDD`$(\mathcal{G}, c, \phi)$**:**

1 $\ell \leftarrow$ **guess** *a literal from* $\mathcal{L}[n]$
2 **if** $\phi \not\models \ell$ **then return** reject
3 $u \leftarrow$ the root of $\mathcal{G}$
4 **while** u *is not a sink of* $\mathcal{G}$ **do**
5 $\quad e \leftarrow$ **guess** *an edge* (u, u') *in* $\mathcal{G}$
6 $\quad f \leftarrow \lambda(u)$
7 $\quad a \leftarrow \eta(e)$
8 $\quad$ Replace each occurrence of v_f in ℓ, if any, with a
9 $\quad u \leftarrow u'$
10 **if** $\lambda(u) = c$ *and* ℓ *is not trivially true* **then**
11 $\quad$ **return** accept
12 **else return** reject

literal $\ell = (t \Diamond t')$ having $t, t' \in \{v_i \mid 1 \leq i \leq n\} \cup \{0, 1\}$ and $\Diamond \in \{=, \neq\}$. In fact, it suffices to replace within ℓ the term t (resp., t') with the value a if the literal $(t = a)$ (resp., $(t' = a)$) appears in ϕ_π. If the resulting literal is *not* a trivially true literal, then $\phi_\pi \not\models \ell$.

The algorithm's correctness follows from the fact that it essentially implements the characterization of Lemma 7, where the check $\phi_\pi \not\models \ell$ is carried out as discussed above. In particular, the algorithm non-deterministically searches for a literal ℓ such that $\phi \models \ell$ (lines 1 and 2) and then non-deterministically searches for a path π in $\mathcal{G}$ such that $\phi_\pi \not\models \ell$. More specifically, the path π is non-deterministically guessed one node at the time in the while loop (in order not to use more than logarithmic space). Each time an edge is traversed, the feature identifier f is read out from the edge's source node label (line 6), and f's value a is read out from the edge label (line 7). These two together form one literal $v_f = a$ of the condition ϕ_π corresponding to the path being traversed. Then, in ℓ every occurrence of v_f is replaced with a.

When the whole path has been traversed, by Lemma 7 and the above discussion, it should now be clear that ϕ is *not* a global necessary reason for c w.r.t. $\mathcal{G}$ iff the last visited node is labeled with c and the literal ℓ is not trivially true, which is what the procedure checks in line 10. Observe moreover that each nondeterministic branch of the algorithm's execution terminates, because $\mathcal{G}$ is a DAG, and hence it cannot be the case that an execution branch gets stuck in a loop of $\mathcal{G}$, i.e., each execution branch reaches a sink of $\mathcal{G}$ at some point.

Space Usage. Regarding the algorithm's space complexity, the procedure only needs to keep in memory, overall, the literal ℓ, the currently visited node u, its label f, the guessed edge $e = (u, u')$, and its label a. All these elements can be encoded in binary, and thus requiring logarithmically many bits in the input size. Finally, the procedure needs to check whether a condition entails a literal (line 2) which, by Theorem 4, can be done in deterministic logspace.

With the above analysis, we can prove the following result.

Theorem 8. IsNecessary[BDD] *is* NL-complete*, and the hardness holds even when the condition is a single literal.*

The upper bound of Theorem 8 follows from the fact that IsNotNecessary[BDD] is in NL, as shown above, and the fact that NL is closed under complement.

The lower bound of Theorem 8 is shown via a reduction from the following intermediate NL-hard problem, which we define next. We say that a directed graph G is *uniform* if each node has either 0 or exactly 2 outgoing edges.

The UniformRootedAcyclicReach problem is defined as follows: given a uniform, rooted directed acyclic graph (RDAG) $G = (V, E)$, a sink of G, and an outgoing edge e of the root of G, decide whether there exists a path in G from its root to the given sink that traverses e.

We can show that UniformRootedAcyclicReach is NL-hard. Finally, we reduce such problem to IsNotNecessary[BDD] in logspace. This implies that also the problem IsNecessary[BDD] is NL-hard, since NL is closed under complement.

The complexity of IsMinNecessary[BDD, $\preccurlyeq$], for each $\preccurlyeq \in \{\leq, \subseteq\}$, can now be shown.

Theorem 9. IsMinNecessary[BDD, $\preccurlyeq$] *is* NL-complete*, for each* $\preccurlyeq \in \{\leq, \subseteq\}$.

The upper bound of Theorem 9 is obtained by exhibiting a nondeterministic machine N deciding this problem and working in logspace. This machine N is designed so to execute the procedure `MinNecessary`$(\mathcal{M}, c, \phi)$ of Algorithm 1 by suitably integrating two nondeterministic logspace machines capable of deciding whether a condition, or a literal, is, or is not, a global necessary reason for the BDD as input; the existence of these two machines is guaranteed by the fact that IsNecessary[BDD] is in NL (see Theorem 8), and so its complement (by NL being closed under complement).

The lower bound of Theorem 9 is shown by proving that the complement of IsMinNecessary[BDD, $\preccurlyeq$] is NL-hard, which in turn is shown via a non-trivial adaptation of the reduction from UniformRootedAcyclicReach to the complement of IsNecessary[BDD] shown for Theorem 8. Here the challenge is to guarantee that when there is no path from the root of the uniform RDAG to a sink going via a given edge, then not only the constructed condition ϕ is a global necessary reason, but also that *no literal* ℓ such that $\phi \not\models \ell$ becomes a global necessary reason.

The Case of DTs We can exploit Lemma 7 again to obtain complexity results for the family of DTs as well. In particular, we prove that the problems IsNecessary[DT] and IsMinNecessary[DT, $\preccurlyeq$] are in L, for each $\preccurlyeq \in \{\leq, \subseteq\}$.

Theorem 10. IsNecessary[DT] *is in* L.

Proof. We show that the complement of IsNecessary[DT] is in L, which implies that IsNecessary[DT] is in L. By Lemma 7, to check that a condition ϕ is *not* a global necessary reason for a class c w.r.t. a DT $\mathcal{T}$, it suffices to try all pairs of a literal ℓ with $\phi \models \ell$ and of a path π from the root of $\mathcal{T}$ to a leaf of $\mathcal{T}$ labeled with c, and check whether $\phi_\pi \not\models \ell$. We can easily iterate over all literals ℓ with $\phi \models \ell$ in logarithmic

space, since for each literal we only need to store at most two variables (using two numbers $i, j \in [n]$ encoded in binary) and we can reuse the space at each iteration, while $\phi \models \ell$ can be checked in logarithmic space by Theorem 4.

The crucial part is to iterate over each path π from the root of $\mathcal{T}$ to a leaf of $\mathcal{T}$ labeled with c, and to verify that $\phi_\pi \not\models \ell$, all without using more than logarithmic space. This can be achieved by iterating over each leaf v of $\mathcal{T}$, and if v is labeled with c, then traversing the (unique) path π connecting the root of $\mathcal{T}$ to v, by iteratively following the parents backwards. While traversing the path π, the literal ℓ is modified as done for BDDs in Algorithm 2. When the root of $\mathcal{T}$ is reached, the shape of ℓ, i.e., whether it is not trivially true, will determine if $\phi_\pi \not\models \ell$, again as done for BDDs. We conclude that the complement of IsNecessary[DT] is in L, which implies IsNecessary[DT] is in L as well. □

For each preorder $\preccurlyeq \in \{\leq, \subseteq\}$, we can show that IsMinNecessary[DT, $\preccurlyeq$] is in L by using the same argument used for showing that IsMinNecessary[PRC, $\preccurlyeq$] is in L (cf. the discussion before Theorem 6).

Theorem 11. IsMinNecessary[DT, $\preccurlyeq$] *is in* L*, for each* $\preccurlyeq \in \{\leq, \subseteq\}$.

4.3 The Case of MLPs

In this section we consider the family of classifiers based on MLPs. As usual, we first analyze the complexity of checking whether a given condition is a global necessary reason.

Theorem 12. IsNecessary[MLP] *is* co-NP-complete*, and the hardness holds even when the condition is a single literal.*

Proof. (Membership). We show that the problem is in co-NP by means of a simple polynomial-time guess and check procedure deciding the *complement* of IsNecessary[MLP]. Given an MLP $\mathcal{N}$, a class c, and a condition ϕ, we can decide whether ϕ is *not* a global necessary reason by guessing an instance $\mathbf{x}$, which is of polynomial size in n, and by then checking that $\mathcal{N}(\mathbf{x}) = c$ and $\mathbf{x} \not\models \phi$. Checking $\mathcal{N}(\mathbf{x}) = c$ can be carried out in polynomial time in the size of $\mathbf{x}$ and $\mathcal{N}$, as it suffices to compute the result of each layer of $\mathcal{N}$ via matrix multiplications. Finally, checking $\mathbf{x} \not\models \phi$ requires computing $\phi[\mathbf{x}]$ and verifying that the latter evaluates to false.

(Hardness). We show the co-NP-hardness of the problem via a polynomial-time reduction from the UnSat problem: given a 3CNF Boolean formula ψ, decide whether ψ is *un*satisfiable. The reduction constructs an MLP $\mathcal{N}_\psi$ starting from ψ by exploiting a result by Barceló et al. [2020b, Lemma 13] showing that any Boolean formula can be encoded as an MLP, which can be obtained in polynomial time in the size of the formula. Together with $\mathcal{N}_\psi$, the reduction constructs the class $c = 1$ and the condition $\phi = (1 = 0)$.

Now, if ψ is unsatisfiable, by Barceló et al.'s result, for every instance $\mathbf{x}$, $\mathcal{N}_\psi(\mathbf{x}) = 0 \neq c$, hence $[\![\mathcal{N}_\psi, c]\!] = \emptyset \subseteq [\![\phi]\!]$, and thus ϕ is a global necessary reason for c w.r.t. $\mathcal{N}_\psi$. If ψ is satisfiable, by Barceló et al.'s result, there exists an instance $\tilde{\mathbf{x}}$ with $\mathcal{N}_\psi(\tilde{\mathbf{x}}) = 1 = c$, and thus $[\![\mathcal{N}_\psi, c]\!] \neq \emptyset$, while $[\![\phi]\!] = \emptyset$. Hence, ϕ is not a global necessary reason for c w.r.t. $\mathcal{N}_\psi$. □

We can now exploit Theorem 12 to prove the complexity of IsMinNecessary[MLP, $\preccurlyeq$], for each $\preccurlyeq \in \{\leq, \subseteq\}$.

Theorem 13. IsMinNecessary[MLP, $\preccurlyeq$] *is* D^P-complete*, for each* $\preccurlyeq \in \{\leq, \subseteq\}$*, and the hardness holds even when the condition is a single literal.*

Proof. (Membership). Consider again the generic procedure `MinNecessary`$(\mathcal{M}, c, \phi)$ reported as Algorithm 1. Assume the input classifier $\mathcal{M}$ to the procedure is an MLP $\mathcal{N}$.

The procedure `MinNecessary`$(\mathcal{M}, c, \phi)$ is characterized by two distinct phases, where the second is executed only if the first succeeds, and both phases need to succeed in order to accept the input. The first phase (line 1) succeeds iff ϕ is a global necessary reason for c w.r.t. $\mathcal{M}$. The second phase (from line 2 to line 4) succeeds iff ϕ is $\preccurlyeq$-minimal.

By Theorem 12, the computation carried out to successfully complete the first phase is in co-NP. To prove the D^P upper bound, we need to show that the computation carried to successfully complete the second phase is in NP.

Remember that, by Lemma 3, the $\preccurlyeq$-minimality of ϕ can be tested by checking, for every literal $\ell \in \mathcal{L}[n]$ (line 2) for which $\phi \not\models \ell$ (line 3), that ℓ is *not* a global necessary reason for c w.r.t. $\mathcal{M}$ (line 4). Observe now that the number of distinct literals $\ell \in \mathcal{L}[n]$ explored at line 2 is $O(n^2)$, and the test at line 3 can be carried out in logspace (see Theorem 4), and hence in polynomial time. Focus now on line 4. In order to accept at line 5, the entire second phase needs to complete successfully. This requires that all tests at line 4 have to fail, i.e., every literal ℓ for which $\phi \not\models \ell$ must *not* be a global necessary reason: checking this is feasible in NP (see Theorem 12). Therefore, the overall computation in the second phase is in NP.

(Hardness (sketch)). Hardness is shown via a polynomial-time reduction from the D^P-hard problem Sat-UnSat: given a pair (γ, δ) of 3CNF Boolean formulas, decide whether γ is satisfiable *and* δ is unsatisfiable. We point out that such a reduction, which, given (γ, δ), builds an MLP $\mathcal{N}$, a class c, and a condition ϕ, is more complex than the reduction to show that IsNecessary[MLP] is co-NP-hard, because the two formulas γ and δ must be encoded *together* into a *single* MLP $\mathcal{N}$ that needs to enjoy two properties at the same time: the condition ϕ is a global necessary reason for c w.r.t. $\mathcal{N}$, *and* ϕ is $\preccurlyeq$-minimal. □

4.4 Computing Minimal Global Necessary Reasons

We discuss how the complexity results for IsNecessary and IsMinNecessary allow us to study the complexity of *computing* a minimal global necessary reason.

For a family C of classifiers, and a preorder $\preccurlyeq \in \{\leq, \subseteq\}$, FindMinNecessary[C, $\preccurlyeq$] denotes the problem of *computing*, given a classifier $\mathcal{M} \in \mathsf{C}$ and a class $c \in \{0, 1\}$, a $\preccurlyeq$-minimal global necessary reason ϕ for c w.r.t. $\mathcal{M}$.

Lower Bounds We observe that the complexity of FindMinNecessary[C, $\preccurlyeq$] is no lower than that of IsMinNecessary[C, $\preccurlyeq$], which is shown by reducing the latter to the former in polynomial time. That is, we show that IsMinNecessary[C, $\preccurlyeq$] can be solved in polynomial time

using an oracle for the problem FINDMINNECESSARY$[\mathsf{C}, \preccurlyeq]$. Recall that any two $\preccurlyeq$-minimal global necessary reasons have the same models. Consider a classifier $\mathcal{M} \in \mathsf{C}$, a class $c \in \{0, 1\}$, and a condition ϕ. To decide whether ϕ is a $\preccurlyeq$-minimal global necessary reason for c w.r.t. $\mathcal{M}$, it is enough to compute a $\preccurlyeq$-minimal global necessary reason ψ using the oracle for FINDMINNECESSARY$[\mathsf{C}, \preccurlyeq]$, and then verify that ϕ and ψ have the same models by checking that they entail exactly the same set of literals. The latter can be done in polynomial time since there are $O(n^2)$ literals to check, and by the fact that entailment of a literal can be checked in logarithmic space, by Theorem 4.

Upper Bounds We now show that the problem FINDMINNECESSARY$[\mathsf{C}, \preccurlyeq]$ can be solved in polynomial time using an oracle for the problem ISNECESSARY$[\mathsf{C}]$. Consider a classifier $\mathcal{M} \in \mathsf{C}$, and a class $c \in \{0, 1\}$. Recall that the conjunction of *all* literals that are individually global necessary reasons is a $\preccurlyeq$-minimal global necessary reason. Thus, to compute a $\preccurlyeq$-minimal global necessary reason, we can simply iterate over all literals, identify those that are global necessary reasons, and construct their conjunction. Checking whether a literal is a global necessary reason can be accomplished via the oracle for ISNECESSARY$[\mathsf{C}]$, while the number of oracle calls is polynomial since there are $O(n^2)$ literals to consider.

From the above discussions we conclude that for each preorder $\preccurlyeq \in \{\leq, \subseteq\}$:

- FINDMINNECESSARY$[\mathsf{C}, \preccurlyeq]$, with $\mathsf{C} \in \{\mathsf{PRC}, \mathsf{DT}, \mathsf{BDD}\}$, can be solved efficiently, i.e., in polynomial time, since ISNECESSARY$[\mathsf{PRC}]$ and ISNECESSARY$[\mathsf{DT}]$ are in L, and ISNECESSARY$[\mathsf{BDD}]$ is in NL (cf. Theorem 5, Theorem 10, and Theorem 8).
- FINDMINNECESSARY$[\mathsf{MLP}, \preccurlyeq]$ can be solved in polynomial time with a polynomial number of calls to a co-NP oracle, since ISNECESSARY$[\mathsf{MLP}]$ is in co-NP (cf. Theorem 13). Moreover, the above upper bound cannot be significantly improved (i.e., reduced to polynomial time), unless PTIME = NP, since the problem FINDMINNECESSARY$[\mathsf{MLP}, \preccurlyeq]$ is at least as hard as ISMINNECESSARY$[\mathsf{MLP}, \preccurlyeq]$, which is D^{P}-hard (cf. Theorem 13), and thus FINDMINNECESSARY$[\mathsf{MLP}, \preccurlyeq]$ is D^{P}-hard as well.

5 Related Work

Explaining *global* classifiers' decisions has been considered in previous work in different forms.

Ignatiev, Narodytska, and Marques-Silva (2019b) proposed two notions of global explanations: an *absolute explanation* (resp., *counterexample*) for a class c w.r.t. a classifier $\mathcal{M}$ is a subset-minimal set $\mathcal{E}$ of feature-value pairs (f_i, c_i), where no feature occurs twice in $\mathcal{E}$, such that every instance $\mathbf{x}$ matching $\mathcal{E}$ (i.e., the instance has value c_i on feature f_i, for every feature f_i in $\mathcal{E}$) is such that $\mathcal{M}(\mathbf{x}) = c$ (resp., $\mathcal{M}(\mathbf{x}) \neq c$). Thus, the "negation" of a counterexample can be seen as a global necessary reason for c, whose form is a disjunction of literals. Our explanations are expressed in terms of conjunctions, which is a widely adopted form—most of the work discussed in this section employs this form. Furthermore, our language goes beyond simple conjunctions of feature-value pairs, as already discussed in Section 3. Importantly, we deepen the study of global necessary reasons by providing a complexity analysis for concrete families of classifiers and consider different minimality criteria.

Izza, Ignatiev, and Marques-Silva (2022) generalize two local notions of explanations, namely *weak abductive explanations* (Cooper and Marques-Silva 2023; Ignatiev, Narodytska, and Marques-Silva 2019a)[4] and *weak contrastive explanations* (Miller 2019; Ignatiev et al. 2020). The generalization is done using so-called *generalized explanation functions*. More specifically, a generalized explanation function takes as input an instance and returns either 0 or 1, can be parameterized on a selected set of features $\mathcal{Z} \subseteq \mathcal{F}$ (where $\mathcal{F}$ is the set of all features), as well as other parameters, and is denoted as $\xi(\mathbf{x}, \mathcal{Z}, \dots)$. Then, a weak abductive explanation is a set of features $\mathcal{Z} \subseteq \mathcal{F}$ such that $\forall \mathbf{x} \in \{0, 1\}^n,\ (\xi(\mathbf{x}, \mathcal{Z}, \dots) = 1) \rightarrow (\mathcal{M}(\mathbf{x}) = c)$. A weak contrastive explanation is a set of features $\mathcal{Z} \subseteq \mathcal{F}$ such that $\exists \mathbf{x} \in \{0, 1\}^n,\ (\xi(\mathbf{x}, \mathcal{F} \setminus \mathcal{Z}, \dots) = 1) \land (\mathcal{M}(\mathbf{x}) \neq c)$. Such notions are different from ours. Also, Izza, Ignatiev, and Marques-Silva (2022) focus on DTs only, using generalized explanation functions that are conjunctions of conditions along a path in a decision tree.

Bassan, Amir, and Katz (2024) introduced a notion of "global necessary reason" where both concepts of necessity and globality have fundamentally different meanings from ours. In (Bassan, Amir, and Katz 2024), *necessity* applies to a *single* feature i of an instance $\mathbf{x}$, and means that i *must belong to all local sufficient* reasons for $\mathbf{x}$.[5] Equivalently, a feature i is *locally necessary* for $\mathbf{x}$ if changing the value of i in $\mathbf{x}$ changes the class that the classifiers assigns to $\mathbf{x}$. Similarly, in (Bassan, Amir, and Katz 2024), *globality* applies to a *single* feature i, and means that i is a local necessary reason for *all* instances. Thus, a global necessary reason, according to the definition proposed by (Bassan, Amir, and Katz 2024), focuses only on whether changing the value of a feature changes the class, and it does not take any specific class of interest into account, while doing so. In contrast, our notion of "necessity" is rooted in logic: in the implication $A \rightarrow B$, B is necessary for A to hold, because if B does not hold, A cannot hold either. Accordingly, for us, a formula ϕ is a necessary reason for a class c w.r.t. a classifier $\mathcal{M}$ if, for all instances $\mathbf{x}$, $(\mathcal{M}(\mathbf{x}) = c) \rightarrow (\mathbf{x} \models \phi)$, that is, if $\mathbf{x}$ does not satisfy ϕ, then $\mathcal{M}$ does not classify $\mathbf{x}$ with c. Finally, in our case, by "globality" we mean the ability of an explanation (in our case, a condition) to *best describe the family of instances classified with a specific class of interest*, using a logic language in our case. Employing logic formulas to characterize conditions that are necessary (and/or sufficient) for classifiers' decisions has been adopted

[4] These are also referred to as sufficient reasons (Darwiche and Hirth 2020) or PI (prime implicant) explanations (Shih, Choi, and Darwiche 2018).

[5] A local sufficient reason for $\mathbf{x}$ is a set $S \subseteq \{1, \dots, n\}$ of features such that the class of *every* instance $\mathbf{y}$ having $\mathbf{y}[i] = \mathbf{x}[i]$, for $i \in S$, coincides with the class of $\mathbf{x}$.

by different works, although only for local explanations—e.g., see the recent tutorial by Darwiche (2023).

Notions of "necessary" explanations have been proposed by Darwiche and Hirth; Audemard et al.; Audemard et al. (2020; 2022a; 2021), but they are all "locally" defined w.r.t. a specific instance, and thus differ from ours.

The works discussed below focus on *sufficient* properties for a classifier to assign a certain class, and thus differ from our approach in that we adopt *necessary* properties. Furthermore, the works below do not provide a complexity analysis. Wang et al. (2017) learn a *rule set model*, which is a set of rules, i.e., conjunctions of conditions. Rule set models predict that an observation is in the positive class when at least one of the rules is satisfied. Setzu et al.; Setzu et al. (2019; 2021) proposed approaches to derive a global explanation from local ones, where both are expressed as decision rules. Rawal and Lakkaraju (2020) use general *recourse rules*, which are if-then statements saying which changes should be applied under certain conditions to obtain a prediction.

Global explanations with aims very different from ours have been considered to explain agent behavior (Huber et al. 2021) and graph neural networks (Huang et al. 2023), to extract surrogate decision trees exploiting ontologies (Confalonieri et al. 2021), to compute top-k words with the highest global impact in document data (Mor, Belinkov, and Kimelfeld 2024), and to transform a decision tree into if-then statements (Huang and Marques-Silva 2023).

While there has been an extensive body of work on *local* explanations, its goal is clearly different from ours, as local explanation tries to capture a property of a single instance (which is given) while a global explanation tries to capture a property shared among a set of instances (which are not explicitly given). Among other things, this difference may affect the complexity of reasoning tasks, such as deciding whether an expression is a local or a global explanation. In the former case, the instance for which the property needs to be verified is given; in the latter case, no instance is given, and this can be a source of complexity.

Although there is a substantial body of work on local explanations, their objective differs fundamentally from ours. Local explanations aim to characterize a property of a *single given instance*, whereas global explanations seek to identify a property common to a *set of instances, which are not explicitly provided*. This distinction has important implications—for example, it can influence the complexity of reasoning tasks, such as determining whether a given expression qualifies as a local or global explanation. In the local case, the relevant instance is available, simplifying verification. In contrast, the absence of such an instance in the global case can significantly increase complexity.

Among approaches on local explanations, Rudin and Shaposhnik; Geng, Schleich, and Suciu (2023; 2022) proposed notions with some sort of "global consistency", meaning that a local explanation must be coherent with the prediction of a restricted set of instances. Gorji and Rubin (2022) consider *local* and *sufficient* reasons in the presence of constraints that allow only certain instances to be valid. Their work and ours adopt two significantly different approaches to explain classifiers' decision in that they consider local and sufficient explanations while we consider global and necessary ones.

We conclude by mentioning that there has been work focusing on defining more abstract explainability frameworks which can be specialized to define concrete explanability notions. One prominent example is the work by Arenas et al. (2021), which defines a formal language, dubbed FOIL, to express different "explainability queries". FOIL is flexible enough to encode different notions of explanations (both local and global) from the literature, but cannot encode the global necessary reasons of our paper, because FOIL is not able to reason about individual features of the instances.

6 Future Work

A natural next step is the development of classifier-specific algorithms for computing minimal global necessary reasons, along with an experimental evaluation. Another avenue for future work is to carry out complexity analyses for other families of classifiers, as well as considering other important notions proposed in the literature where a systematic study is still lacking. For instance, it would be interesting to carry out a complexity analysis of *global sufficient reasons* along the lines of what we have done in this paper, that is, considering different "optimality" criteria and distinguishing different classifier families. Global sufficient reasons are the logical dual of global necessary reasons. In particular, a *global sufficient reason* for a class c w.r.t. a classifier $\mathcal{M}$ is a formula ϕ such that, for each instance $\mathbf{x}$, $(\mathbf{x} \models \phi) \rightarrow (\mathcal{M}(\mathbf{x}) = c)$. This notion then coincides with the absolute explanation by Ignatiev, Narodytska, and Marques-Silva (2019b) and the (generalized) weak abductive explanation by Izza, Ignatiev, and Marques-Silva (2022).

Acknowledgments

We thank the anonymous referees for their feedback.

This work was partially supported by project SERICS (PE00000014) under the MUR National Recovery and Resilience Plan funded by the European Union - NextGenerationEU.

Marco Calautti's and Enrico Malizia's work was supported by the European Union – NextGenerationEU programme, through the Italian Ministry of University and Research (MUR) PRIN 2022-PNRR grant P2022KHTX7 "DISTORT"–CUP: H53D23008170001 (Calautti) & J53D23015000001 (Malizia), under the Italian "National Recovery and Resilience Plan" (PNRR), Mission 4 Component 1.

Cristian Molinaro's work was supported by the Italian Ministry of University and Research (MUR) PRIN 2022 grant 2022XERWK9 "S-PIC4CHU - Semantics-based Provenance, Integrity, and Curation for Consistent, High-quality, and Unbiased data science"— CUP: H53C24000990006.

Cristian Molinaro acknowledges the support of the PNRR project FAIR - Future AI Research (PE00000013), Spoke 9 - Green-aware AI, under the NRRP MUR program funded by the NextGenerationEU.

References

Arenas, M.; Baez, D.; Barceló, P.; Pérez, J.; and Subercaseaux, B. 2021. Foundations of symbolic languages for model interpretability. In *Proc. NeurIPS*, 11690–11701.

Arenas, M.; Barceló, P.; Bertossi, L. E.; and Monet, M. 2023. On the complexity of shap-score-based explanations: Tractability via knowledge compilation and non-approximability results. *J. Mach. Learn. Res.* 24:63:1–63:58.

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

Audemard, G.; Bellart, S.; Bounia, L.; Koriche, F.; Lagniez, J.; and Marquis, P. 2021. On the computational intelligibility of boolean classifiers. In *Proc. KR*, 74–86.

Audemard, G.; Bellart, S.; Bounia, L.; Koriche, F.; Lagniez, J.; and Marquis, P. 2022a. On the explanatory power of boolean decision trees. *Data Knowl. Eng.* 142:102088.

Audemard, G.; Bellart, S.; Bounia, L.; Koriche, F.; Lagniez, J.; and Marquis, P. 2022b. Trading complexity for sparsity in random forest explanations. In *Proc. AAAI*, 5461–5469.

Baehrens, D.; Schroeter, T.; Harmeling, S.; Kawanabe, M.; Hansen, K.; and Müller, K. 2010. How to explain individual classification decisions. *J. Mach. Learn. Res.* 11:1803–1831.

Barceló, P.; Monet, M.; Pérez, J.; and Subercaseaux, B. 2020a. Model interpretability through the lens of computational complexity. In *Proc. NeurIPS*.

Barceló, P.; Monet, M.; Pérez, J.; and Subercaseaux, B. 2020b. Model interpretability through the lens of computational complexity. Technical Report arXiv:2010.12265, CoRR.

Bassan, S.; Amir, G.; and Katz, G. 2024. Local vs. global interpretability: A computational complexity perspective. In *Proc. ICML*.

Carbonnel, C.; Cooper, M. C.; and Marques-Silva, J. 2023. Tractable explaining of multivariate decision trees. In *Proc. KR*, 127–135.

Confalonieri, R.; Weyde, T.; Besold, T. R.; and del Prado Martín, F. M. 2021. Using ontologies to enhance human understandability of global post-hoc explanations of black-box models. *Artif. Intell.* 296:103471.

Cooper, M. C., and Amgoud, L. 2023. Abductive explanations of classifiers under constraints: Complexity and properties. In *Proc. ECAI*, 469–476.

Cooper, M. C., and Marques-Silva, J. 2023. Tractability of explaining classifier decisions. *Artif. Intell.* 316:103841.

Darwiche, A., and Hirth, A. 2020. On the reasons behind decisions. In *Proc. ECAI*, 712–720.

Darwiche, A. 2023. Logic for explainable AI. In *Proc. LICS*, 1–11.

de Colnet, A., and Marquis, P. 2022. On the complexity of enumerating prime implicants from decision-dnnf circuits. In *Proc. IJCAI*, 2583–2590.

Geng, Z.; Schleich, M.; and Suciu, D. 2022. Computing rule-based explanations by leveraging counterfactuals. *Proc. VLDB Endow.* 16(3):420–432.

Gorji, N., and Rubin, S. 2022. Sufficient reasons for classifier decisions in the presence of domain constraints. In *Proc. AAAI*, 5660–5667.

Greenlaw, R.; Hoover, H. J.; and Ruzzo, W. L. 1995. *Limits to Parallel Computation: P-completeness Theory*. New York, NY, USA: Oxford University Press.

Huang, X., and Marques-Silva, J. 2023. From decision trees to explained decision sets. In *Proc. ECAI*, volume 372 of *Frontiers in Artificial Intelligence and Applications*, 1100–1108.

Huang, X.; Izza, Y.; Ignatiev, A.; Cooper, M. C.; Asher, N.; and Marques-Silva, J. 2022. Tractable explanations for d-dnnf classifiers. In *Proc. AAAI*, 5719–5728.

Huang, Z.; Kosan, M.; Medya, S.; Ranu, S.; and Singh, A. K. 2023. Global counterfactual explainer for graph neural networks. In *Proc. WSDM*, 141–149.

Huber, T.; Weitz, K.; André, E.; and Amir, O. 2021. Local and global explanations of agent behavior: Integrating strategy summaries with saliency maps. *Artif. Intell.* 301:103571.

Ignatiev, A.; Narodytska, N.; Asher, N.; and Marques-Silva, J. 2020. From contrastive to abductive explanations and back again. In *Proc. AIxIA*, volume 12414, 335–355.

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

Ignatiev, A.; Narodytska, N.; and Marques-Silva, J. 2019b. On relating explanations and adversarial examples. In *Proc. NeurIPS*, 15857–15867.

Izza, Y.; Ignatiev, A.; and Marques-Silva, J. 2022. On tackling explanation redundancy in decision trees. *J. Artif. Intell. Res.* 75:261–321.

Marques-Silva, J., and Ignatiev, A. 2022. Delivering trustworthy AI through formal XAI. In *Proc. AAAI*, 12342–12350.

Marques-Silva, J.; Gerspacher, T.; Cooper, M. C.; Ignatiev, A.; and Narodytska, N. 2020. Explaining naive bayes and other linear classifiers with polynomial time and delay. In *Proc. NeurIPS*.

Marques-Silva, J. 2022. Logic-based explainability in machine learning. In *Proc. RW*, 24–104.

Marques-Silva, J. 2024. Logic-based explainability: Past, present & future. Technical Report arXiv:2406.11873, CoRR.

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

Mor, A.; Belinkov, Y.; and Kimelfeld, B. 2024. Accelerating the global aggregation of local explanations. In *Proc. AAAI*, 18807–18814.

Ordyniak, S.; Paesani, G.; and Szeider, S. 2023. The parameterized complexity of finding concise local explanations. In *Proc. IJCAI*, 3312–3320.

Rawal, K., and Lakkaraju, H. 2020. Beyond individualized recourse: Interpretable and interactive summaries of actionable recourses. In *Proc. NeurIPS*.

Rudin, C., and Shaposhnik, Y. 2023. Globally-consistent rule-based summary-explanations for machine learning models:

Application to credit-risk evaluation. *J. Mach. Learn. Res.* 24:16:1–16:44.

Setzu, M.; Guidotti, R.; Monreale, A.; and Turini, F. 2019. Global explanations with local scoring. In *Proc. ECML PKDD Workshops*, 159–171.

Setzu, M.; Guidotti, R.; Monreale, A.; Turini, F.; Pedreschi, D.; and Giannotti, F. 2021. GLocalX – From local to global explanations of black box AI models. *Artif. Intell.* 294.

Shih, A.; Choi, A.; and Darwiche, A. 2018. A symbolic approach to explaining bayesian network classifiers. In Lang, J., ed., *Proc. IJCAI*, 5103–5111.

Wang, T.; Rudin, C.; Doshi-Velez, F.; Liu, Y.; Klampfl, E.; and MacNeille, P. 2017. A Bayesian framework for learning rule sets for interpretable classification. *J. Mach. Learn. Res.* 18:70:1–70:37.