# A logical investigation of explainable AI

Xinghan Liu


# THÈSE

**En vue de l'obtention du**

## DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

**Délivré par :** *l'Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)*

---

**Présentée et soutenue le** *9 octobre 2023* **par :**

**Xinghan LIU**

**Une investigation logique l'IA explicable**
**(A Logical Investigation of Explainable AI)**

---

**JURY**

| | | |
|---|---|---|
| Leila AMGOUD | IRIT | Président du Jury |
| John HORTY | University of Maryland | Rapporteur |
| Nicola OLIVETTI | Aix-Marseille Université | Rapporteur |
| Agata CIABATTONI | Technische Universität Wien | Examinatrice |
| Hans VAN DITMARSCH | IRIT | Examinateur |
| Emiliano LORINI | IRIT | Directeur de thèse |

---

**École doctorale et spécialité :**
*MITT : Informatique*
**Unité de Recherche :**
*Institut de Recherche en Informatique de Toulouse (IRIT)*
**Directeur de Thèse :**
*Emiliano LORINI*
**Rapporteurs :**
*John HORTY* et *Nicola OLIVETTI*

## Résumé

Expliquer pourquoi un classificateur classe une instance d'entrée donnée comme classification de sortie devient de plus en plus vital de nos jours, car l'intelligence artificielle (IA) continue d'évoluer rapidement et d'imprégner divers aspects de la vie quotidienne, tandis que les systèmes d'IA utilisés aujourd'hui manquent souvent de transparence.

L'approche symbolique de l'IA explicable (XAI) montre donc son importance, puisque les symboles et les règles qu'elle utilise sont intrinsèquement compréhensibles pour les humains. Dans cette thèse j'étudie la XAI avec différents outils logiques, y compris les logiques modales, les logiques épistémiques, les logiques conditionnelles et les logiques modales de produit.

Deux cadres logiques sont présentés pour modéliser les systèmes de classification. Le premier est appelé logique de classificateur à entrée binaire (BLC). Il modélise un classificateur à entrée binaire comme une partition d'un modèle de Kripke S5. En adoptant le point de vue de cette logique modale, de nombreuses notions d'explications pour les classificateurs booléens sont exprimables plutôt que définies dans un méta-langage, notamment l'explication abductive, l'explication contrastive, l'explication contrefactuelle et le biais de décision.

La seconde est appelée logique modale de produit pour les classificateurs (PLC) afin de représenter les classificateurs de boîte noire. L'idée clé est que la boîte noire est liée à l'incertitude d'un agent quant à savoir lequel est le vrai parmi de nombreux classificateurs possibles. Nous devons donc modéliser un classificateur de boîte noire comme un ensemble de classificateurs qui sont tous compatibles avec la connaissance de la boîte noire par l'agent. Il en résulte une logique modale avec deux dimensions pour les instances et les classificateurs respectivement. Par conséquent, les notions d'explication susmentionnées ont leurs correspondances subjectives naturelles.

Outre les cadres logiques eux-mêmes, d'autres questions connexes sont abordées dans la thèse. BCL fournit une nouvelle représentation du raisonnement basé sur les cas juridiques, de telle sorte qu'une base de cas est considérée comme un classificateur partiel. De cette manière, les notions d'explication dans XAI peuvent être appliquées au raisonnement basé sur les cas. L'explication par classificateur a une relation étroite avec le raisonnement contrefactuel. La distance de Hamming est une mesure largement utilisée dans le raisonnement contrefactuel. Dans BCL, un conditionnel contrefactuel est proposé pour l'explication du classificateur, et la mesure qu'il utilise est la distance de Hamming, une mesure de distance largement utilisée dans l'IA symbolique. La thèse démontre qu'avec un langage basique des contrefactuels et des propositions atomiques infinies, toute mesure de distance pour les classificateurs peut être réinterprétée comme une distance de Hamming via des variables cachées, sans perte de validité. Des aspects techniques tels que la complexité informatique, la complétude de l'axiomatique et l'extension de l'axiomatique avec une règle d'inférence infinie sont également étudiés.

**Mots-clés: logique modale, IA explicable, explication, classificateur booléen, contrefactuel, logique épistémique**

## Abstract

Explaining why a classifier classifies a given input instance as the output classification becomes increasingly vital nowadays, as artificial intelligence (AI) continues to evolve rapidly and permeate various aspects of everyday life, while the AI systems in use today often lack transparency.

The symbolic approach to explainable AI (XAI) therefore shows its significance, since the symbols and rules it uses are inherently understandable to humans. In the thesis I investigate XAI with various logic tools including modal logics, epistemic logics, conditional logics and product modal logics.

Two logical frameworks are presented to model classifier systems. The first one is called binary-input classifier logic (BLC). It models a binary-input classifier as a partition of an S5 Kripke model. By taking this modal logic viewpoint many notions of explanations for Boolean classifiers are expressible rather than defined in meta-language, including abductive explanation, contrastive explanation, counterfactual explanation and decision bias.

The second one is called product modal logic for classifiers (PLC) in order to represent black box classifiers. The key idea is that black box has to do with the uncertainty of an agent about which classifier is the real one among many possible classifiers. Therefore we need to model a black box classifier as a set of classifiers which are all compatible with the agent's knowledge of the black box. It results in a produce modal logic with two dimensions for instances and classifiers respectively. As a consequence, the notions of explanation aforementioned have their natural subjective correspondences.

Besides the logical frameworks themselves, there are related issues discussed in the thesis. BCL provides a new representation of legal case-based reasoning, such that a case base is viewed as a partial classifier. In this way, the notions of explanation in XAI can be applied to case-based reasoning. In BCL a counterfactual conditional is proposed for classifier explanation, and the measurement it uses is the Hamming distance, a widely used measure of distance in symbolic AI. A result of the thesis implies that given the basic language of counterfactuals with infinite atomic propositions, any distance measure for classifiers can be reinterpreted as Hamming distance by introducing hidden variables without loss of validity. Technical aspects such as computational complexity, completeness of axiomatics and extending the axiomatics with infinitary inference rule are also studied.

**Keywords: modal logic, explainable AI, explanation, Boolean classifier, counterfactual, epistemic logic**

## Acknowledgments

First of all, I would like to thank my advisor Emiliano Lorini with my deepest respect. Through the past three years he has always made himself available whenever I needed assistance and guidance. The fruitful collaboration demonstrates his outstanding supervisory ability. His insistence on rigor as well as his encouragement of work-life balance had a great impact on me. It is such a convincing example to see someone so productive in the professional life taking care of the family in such a responsive way at the same time. I can still recall the day of our online interview during the pandemic when he was talking on the balcony with a small lake in the background. I will always be grateful that he let me be his PhD student.

Besides Emiliano, many professors and senior researchers in and outside my lab have provided substantial assistance throughout my study. I am vastly indebted to Philippe Balbiani, Andreas Herzig, Giovanni Sartor and Antonino Rotolo.

I want to express my sincere thanks to the jury members of my defense. Besides Emiliano, they are John Horty and Nicola Olivetti ("rapporteurs"), Leila Amgoud (chair), Agata Ciabattoni and Hans van Ditmarsch (examiners). Their valuable comments and inspiring questions in the reviews and during the defense are of great significance in improving my thesis.

My heartfelt appreciation also goes to my PhD/post-doc colleagues in and out of my lab. They are Carlos Aguilera-Ventura, Zhenyu Bai, Jorge Fernandez-Davila, Cecilia Di Florio, Quentin Gougeon, Hao Hu, Xuanxiang Huang, Yacine Izza, Alexey Lazarev, Arnaud Lequen, Yaxin Li, Xiaolong Liu, Munique Mittelmann, Timothy Parker, Pengfei Song, Mohit Vaishnav and Jingling Zhang.... The list cannot be completed without being extended to a new page. I have collaborated with some of them, had academic discussion with some of them, had food, drinks and sports with some of them, and enjoyed the time I spent with each of them.

The support from the project ANITI (Artificial and Natural Intelligence Toulouse Institute) is gratefully acknowledged. Without its funding, my PhD position would not have been possible. Toulouse, "la ville rose", the center of Occitania, the headquarters of Airbus, is a livable and warmhearted place. It has left indelible marks on me. It is the only city that seeing a Beluga flying across the sky is not a big deal. The ritual of greeting the driver while getting on and off the bus even surprises the non-local French people. Its free tennis courts are no doubt also a bless for me.

My special thanks go to Yujing, my special one. We have managed the three years' long distance relationship. With seventeen times of meeting and parting, we have never been apart for more than three months. It amuses me every time when hearing "LH2222 to Toulouse" at the Munich airport.

At last, I would like to thank my parents with my deepest gratitude. They have little idea about my previous study in philosophy, and even less idea about the reason of my transition to logic. But they support me unconditionally no matter what. I was not able to visit them for almost four years due to the pandemic. As the thesis was completed and submitted, I finally went back home before the end of my PhD program. The thesis is dedicated to them.

# Contents

# List of Acronyms

| | |
|---|---|
| BCL | Binary-input Classifier Logic |
| CBR | Case-Based Reasoning |
| CM | Classifier Model |
| DM | Decision Model |
| MCM | Multi-Classifier Model |
| MDM | Multi-Decision Model |
| pBF | partial(ly-defined) Boolean Function |
| PLC | Product model modal Logic for binary-input Classifier |

# Notions and Notations

| | | |
|---|---|---|
| $Atm$ | the set of atomic propositions | Definition 2.1, Definition 3.2.1 |
| $Atm_0$ | the set of input variables | Definition 3.2.1 |
| $\mathsf{AXp}$ | abductive explanation | Definition 3.13 |
| $\mathsf{Bias}$ | decision bias | Definition 3.15 |
| $C$ | classifier model (CM) | Definition 3.2.1 |
| $c$ | classification / output value | p. 28 |
| $\mathfrak{c}$ | (legal) precedential case | Definition 4.1 |
| $CB$ | (legal) case base | p. 54 |
| $\mathsf{Compl}(X)$ | $X$-completeness | Definition 3.1 |
| $\mathsf{cn}_{X,Y}$ | conjunction of literals $\bigwedge_{p\in X} p \wedge \bigwedge_{p\in Y\setminus X} \neg p$ | p. 11 |
| $\mathsf{CXp}$ | contrastive explanation | Definition 3.14 |
| $Dec$ | the set of decision atoms | p. 28 |
| $\mathsf{Defin}(X)$ | $X$-definiteness | Definition 3.2 |
| $F_S$ | set of classifiers with common domain $S$ | Definition 6.1 |
| $\Gamma$ | multi-classifier model (MCM) | Definition 6.1 |
| $\lambda$ | conjunction of literals | p. 39 |
| $\mathsf{PImp}$ | prime implicant | Definition 3.12 |
| $S$ | set of states | Definition 2.30 |
| $s$ | state | Definition 10 |
| $\mathsf{t}(c)$ | decision atom | p. 28 |
| $\mathsf{t}(?)$ | undecided / indeterminate | p. 39, p. 53 |
| $Val$ | the set of classifications / output values | p. 28 |
| $X$ | finite set of atomic propositions | p. 11, Definition 3.2.1 |
| $\subseteq^{\text{fin}}$ | "being a finite subset of" | p. 11 |

Chapter 1

# Introduction

## 1.1 Why logic for XAI?

Given the increasing role that artificial intelligence (AI) plays in everyday life, there is a rapidly escalating demand in the transparency and trustworthiness of AI systems. Explainability is arguably the key prerequisite towards fair and trustworthy AI systems. Designers of AI systems are thus asked to provide controllability and explainability within automated decision-making processes as required, e.g., in EU by Art. 22 GDPR and by Art. 6 ECHR related to judicial decisions. Explainable AI (XAI) has therefore become a booming field, especially after the paper *Why should I trust you: explaining the predictions of any classifier* [Ribeiro *et al.* 2016]. For a systematic overview of the research in this area see, e.g., [Molnar 2023].

Although the most recent successes and concerns come from the field of subsymbolic AI, viz. deep neural networks, specifically transformers, symbolic AI still shows its relevance and significance, and logic lies in the core of symbolic AI.

The main goal of XAI is to make decision-making process of AI systems understandable to humans, so that when the decision / prediction /classification [1] is unfair or problematic, people know how to improve it. [2] That is to say, all XAI issues, no matter whether they are embedded in computer vision, natural language processing or other tasks, are eventually matters of classification.

Explanations provided by subsymbolic AI for classifications are often significantly different from the explanations people typically cope with in everyday life. For example in [Ribeiro *et al.* 2016] an explanation itself is a (simpler) classifier.[3] Another popular approach, [Lundberg & Lee 2017] has been widely taken in especially biology and medicine in recent years. They claim to provide the Shapley values of features, a notion from game theory, as explanations. However, it is also not satisfactory just to know a number indicating the importance of a feature in order to explain a patient's diagnosis. Moreover, those explanations are all probabilistic. These make the subsymbolic explanations hard to trust and understand.

[1] Throughout the thesis these three are interchangeable. After all, from a technical point of view they all refer to the output of some function differing only in context. We will do the same for atomic propositions / features / factors which all refer to input variables.

[2] For example the famous Google Gorilla issue. In 2015, the image recognition algorithm of Google misidentified two persons as gorillas. Google apologized for the mistake and promised to fix it. However, Google just removed the image-label gorilla at all and is still unable to find a way to fully resolve the problem as of this year. See https://petapixel.com/2023/05/22/googles-photos-app-is-still-unable-to-find-gorillas/.

[3] It makes more sense to say the explanation of the simpler classifier can serve as an approximate explanation of the original one.

On the other hand, the nature of logic lends itself well to explainability because it is both commensensical and rigorous. When people are seeking an explanation, they request a reason for the occurrence of a phenomenon; and the symbolic approach to XAI views explanations as propositions to answer why or why not questions in terms of logical validity [Darwiche & Hirth 2020, Ignatiev *et al.* 2020b]. In [Darwiche 2020] it is argued that logic plays three roles in contemporary AI research: for computation, for learning from a combination of data and knowledge, and for reasoning about the behavior of machine learning systems. Certainly in the symbolic XAI approach to classifier systems, whether they are decision trees, binary decision diagrams, Bayesian networks or neural networks, they are eventually represented as Boolean functions, so that explanations can be computed and reasoned. Hence it is also called *formal XAI* [Marques-Silva 2023]. Some typical notions of explanations, from both approaches, will be given in Chapter 2.2 as background.

## 1.2 Why modal logic for XAI?

Modal logics are extensions of propositional logic with modal operators expressing necessity, knowledge, belief etc. The most used ones in AI are (translated into) well-behaved fragments of first order logic which have relatively low computational complexities.

Traditionally a Boolean classifier is represented by, or even identical with, propositional logic. We argue that using some simple modal logics provide more natural representations of classifiers and their explanations.

**Syntactic vs. semantic representations** There are two ways to represent Boolean classifiers: syntactically as formulas, or semantically as models. It is not hard to view Boolean classifiers themselves as semantic models, where binary values naturally correspond to truth values. In the approach of propositional logic Boolean classifiers are represented as propositional formulas, e.g. canonical CNFs and DNFs (formal definitions are given in Chapter 2.1.2). The key point is that there is an isomorphism, as trivial as it may seem, between propositional formulas and Boolean classifiers, which we state below and also mention as Corollary 2.1 in Chapter 2.1.2.

**Fact 1.1.** *Every propositional formula expresses a unique Boolean function up to isomorphism.*

For more details how it works, given a binary sequence $s$, it can be represented as $\bigwedge_{\text{the i-th digit in } s \text{ is } 1} p_i \wedge \bigwedge_{\text{the j-th digit in } s \text{ is } 0} \neg p_j$ which we note tentatively as $\hat{s}$. So the classifier $f$ outputs 1 for the input $s$, if and only if $\hat{s} \rightarrow \varphi$ is a tautology, where $\varphi$ expresses $f$.

Nonetheless, we may have a model approach to Boolean classifiers. That means, a binary sequence is transformed into a state $s$, i.e. a set of variables / atomic propositions, s.t. the $i$-th digit in the sequence is 1 if and only if $p_i \in s$.[4] Thus, a

[4]It is intentional to use $s$ for both state and sequence here and in the last paragraph, for it is not hard to see that states, sequences and sets of literals are "the same thing" up to isomorphism.

Boolean classifier is represented as the set of all and only states denoting instances that are output 1 by the classifier. In symbols, $S = \{s : f(s) = 1\}$.

Since we do not have any special relation or property to constrain such a set of states, it is actually just an S5 model in modal logic (formal definitions are given in Chapter 2.3). Notice that a Boolean functions is defined to have finite arity, but by taking the model representation we can deal with any countable set of variables. Moreover, by introducing finitely many classifications as semantic entities, we no more restrict ourselves to the binary-output case by identifying, on one hand 1, $\top$ and *presence*, on the other hand 0, $\bot$ and *absence*. By this step, we can represent binary-input classifiers with possibly infinite arity, partially defined domain, and finitely many output values. While the technical details will be given in Chapter 3, we make the following statement as our slogan.

- The (simplest) model representation of a classifier is a *partition* of all the states in its domain.

We argue that this understanding fits better with the "standard intuition" of a classifier from the set-theoretic viewpoint, namely a classification function is nothing but a set of pairs of the form of $(s, c)$, with Boolean classifier as a special case. Therefore classifier as a partition of an S5 model is the guiding principle of Chapter 3 and subsequent chapters.

**Perturbation is modal** Perturbation is a basic operation for classifier explanation, whichever approach one takes. When perturbing some features (i.e. input variables) of an instance, we change their values in the current instance. In the binary-input case, it means just to "flip" their values.

Perturbation is required by all the notions of classifier explanation that we will discuss in this thesis. More specifically, they all lie in a pattern of perturbation-observation. For example, in counterfactual reasoning one perturbs some features and observes whether the output changes accordingly.

It is not hard to see that propositional logic cannot express perturbation in its language, but has to define it at the meta-language level. In contrast, we can come up with a modal formula, informally as

> *It is possible to find a state that shares the same values of variables in $X$ as the current state, but whose classification is $c'$*

which means that features in the complementary set can be perturbed and an instance whose classification is $c'$ is observable. By this interpretation perturbation is "reduced" to a modal statement. Based on this we can express the notions of classifier explanation in [Darwiche & Hirth 2020, Ignatiev *et al.* 2019, Ignatiev *et al.* 2020b], which is dealt in Chapter 3.

Along the way of developing modal logics for classifiers, many interesting applications and research questions have emerged. Two Chapters are therefore committed to exploring two topics respectively.

**Legal case bases as classifiers** The modal approach also helps bridge classifiers and legal case-based reasoning (CBR). A legal case base is a set of precedential cases (precedents). By viewing a precedent as a pair of a set of facts found by the court and a decision of the court (possibly with more structure, details in Chapter 4), case bases can be understood as partial classification functions, since cases bases usually do not complete all the possibilities. As a result, notions of XAI for classifiers can be used in the factor-based models of CBR. Chapter 4 is dedicated to this topic.

**Counterfactuals and Hamming distance** Perturbation has a close relation to counterfactual reasoning. A counterfactual conditional is formed like

$$\varphi \mathrel{\Box\!\!\rightarrow} \psi$$

which reads as "if it were the case that $\varphi$, then it would be the case that $\psi$." Let $\varphi$ be a conjunction of literals and $\psi$ stand for some classification $c$, it is the counterfactual correspondence to say perturbing the variables in $X$ the output is $c$. Hamming distance measures the cardinality difference between two (equally long) binary sequences and is widely used in AI. A natural question is that, is Hamming distance too specific? The technical result in Chapter 5 implies that given infinite variables, any measure of distance between input instances (e.g. whatever feature weight is assigned) can be re-interpreted as Hamming distance. Hence we can represent the distance of instances of a classifier in XAI as Hammingian without loss of generality.

## 1.3 Black boxes in AI

The driven force of XAI is to explain black box classifier systems. It is worth reflecting what does a black box mean when we use this seemingly self-evident metaphor for machine learning AI systems.

A black box refers to a system where one can observe inputs and outputs without seeing its inner mechanism / internal working, i.e. one cannot "open up" the box and "look inside".

Black box as a metaphor was first used in electronic circuit theory in engineering. Later on, it has been also used as a metaphor for human brain, especially in the behaviorist tradition. Before the rise of machine learning, black box was mostly used to refer to human brain/mind.

To see why this seemingly self-evident metaphor deserves a closer look, let us first point out that not all AI researchers accept the metaphor. In the paper *The "black box" metaphor in machine learning*, Dallas Card [Card 2017] argues that the metaphor is "actually quite misleading in general". His main arguments are quoted as follows.

> Although deep learning models are certainly complex, they are not black boxes. In fact, it would be more accurate to refer to them as glass

> boxes, because *we can literally look inside and see what each component is doing.* ...
>
> Exactly why it (stochastic gradient descent, SDG) works as well as it does is still not well understood, but *the main thing to keep in mind is that it, too, is transparent.*
>
> The actual computation performed by these models in making a prediction is typically quite straightforward; *where things get difficult is in the actual learning of the model parameters from data.*
>
> ...
>
> *The algorithm itself, however, is deterministic*, and if we used the same initialization and the same data, it would produce the same result. In other words, neither the model nor the algorithm is a black box. [Card 2017, emphasis and parentheses added]

Reasons that make Card concludes that "neither the model nor the algorithm is a black box" are that 1) since we can check the code and know that its inner mechanism uses stochastic gradient descent, the algorithm is transparent; 2) even the (learned) model's parameters is hard to get, the algorithm and the model are deterministic.

I disagree with Card's arguments for two reasons.

**What black is the classifier model, not the meta-algorithm** It is crucial to highlight that there are two objects involved as Card rightly puts, namely in his words the algorithm and the model. The whole point of black box stems from the difficult accessibility to parameters of the latter, and has nothing to do with the former.

The best expression on this issue, to my knowledge, comes from the book *The ethical algorithm: The science of socially aware algorithm design* [Kearns & Roth 2019]. I will quote their words instead of making my own. Notice, however, that they call meta-algorithm for what Card calls algorithm.[5]

> In traditional algorithm design, while the output might be useful,..., that output is not itself another algorithm that can be directly applied to further data. In contrast, in machine learning, that's the entire point. ...Rather than trying to directly specify an algorithm for making these predictions – which could be quite difficult and subtle – we write a meta-algorithm that uses the historical data to *derive* our model or prediction algorithm. Machine learning is sometimes considered a form of "self-programming", since it's primarily the data that determines the detailed form of the learned model. [Kearns & Roth 2019, p. 6]

[5] Their use of word meta-algorithm may be contentious, but we adopt it here since the algorithm itself is not our focus, but its output, the black box.

Therefore the black box is not the algorithm itself – surely its inner mechanism is fully transparent, even if not to us all, but to its designer(s). And any one who accesses the algorithm can of course look inside it. The black box is, rather, the *output of the algorithm* (and the training set feeding it).

> So when people talk about the complexity and opaqueness of machine learning, they really don't (or at least shouldn't) mean the actual optimazation algorithms, such as backgropagation. These are the algorithms designed by human beings. But the *models* they produce – the outputs of such algorithms – can be complicated and inscrutable, ... And this is why the human being deploying the model won't fully understand it. [Kearns & Roth 2019, p. 10]

In a word to summarize what designer understands (and needs no explanation) and what not (and needs).

> The designer may have had a good understanding of the algorithm that was used to *find* the decision-making model, but not of the model itself. [Kearns & Roth 2019, p. 11]

This leads us to my second reason.

**Black box is epistemic** Recall that one of Card's argument is that the (learned prediction) model is deterministic. This is true, but does not support his conclusion, because black box is epistemic rather than ontological.

By ontological I mean, without much (philo)sophic consideration, what it *really is*. Card is right to point out that the model is deterministic, given the machine learning algorithm and a fixed training set. Nevertheless, black box is not only on what it is, but what it is known.

Things are therefore different with the ontological case. In the latter, an explanation is enough to be causal which states a fact of the world.[6] We will formalize and prove a *valid* formula in Chapter 3 which, informally speaking,

$$\Box(\textit{there is a sufficient reason for its classification})$$

where $\Box$ means for all instances, says that for every instance, the classifier has a sufficient reason to explain its classification.

However, that reason may not be known when we introduce the epistemic dimension into account. We will use another box symbol $\blacksquare$ for that. That is to say a *satisfiable* formula says that

$$\neg\blacksquare\Box(\textit{there is a sufficient reason for its classification})$$

where $\blacksquare$ is interpreted here as knowing. The formula will be formalized and discussed in Chapter 6. Therefore, in the black box case, explanation is not merely

[6] Or in other words, the epistemic dimension is hence "curled up".

causal, but needs to be epistemic. To mark the difference between explanations in white and black boxes, we will also use terms objective and subjective explanations.

**Black box classifier as a set of classifiers** After reflecting why and how the black box metaphor makes sense for classifier systems, we are able to represent black box classifiers. Since the epistemic dimension is introduced, it is natural to think of not just mono-dimensional, but bi-dimensional modal logic. In a word, we will represent a black box classifier as a *set* of (white box) classifiers which are all *compatible* with the agent's (current, partial) knowledge about the black box. Two modal operators $\Box, \blacksquare$ will range over instances and classifiers respectively.

The agent's knowledge about the black box consists of specific information about some instances classifications (which may come from knowledge of data set or observation of perturbation), and general information about the global constraints of the model (for example, the non-bias constraints implemented in the meta-algorithm as discussed in [Kearns & Roth 2019]). As a result, though in the black box there is one and only one real classifier, the agent has *uncertainty* about which one among many compatible classifiers is the one. This is the leading thought of Chapter 6.

## 1.4 Structure and sources of chapters

The rest of the thesis will investigate XAI for classifier systems according to the themes aforementioned. Most chapters are based on published papers. Some of them are enriched by the targeted addition of new content. The details are as follows.

- Chapter 2 provides preliminaries of the study. Many fundamental definitions of propositional logic, Boolean functions and modal logic will be presented. In particular, I will use a *state semantics* to unify the three, so that through the thesis we can talk about them without unnecessarily switching between different semantics. Besides, the most popular notions and methods of XAI in both symbolic and subsymbolic approaches will be introduced serving as essential background information.

- Chapter 3 is based on a paper of conference CLAR 2021 [Liu & Lorini 2021] and a journal paper [Liu & Lorini 2023] published in *Journal of Logic and Computation*.

  Xinghan Liu and Emiliano Lorini. A logic for binary classifiers and their explanation. In *Logic and Argumentation - 4th International Conference, CLAR 2021, Hangzhou, China, 2021, Proceedings,* Lecture Notes in Computer Science, pages 302-321. Springer, 2021.

  Xinghan Liu and Emiliano Lorini. A unified logical framework for explanations in classifier systems. *Journal of Logic and Computation,* 33(2):485-515, 2023.

There we present the binary-classifier logic BCL, which is an extension of S5, for (single) classifiers and their explanations. The only addition to the published version is a completeness proof for the finite-variable case, which we promised in the published paper to accomplish in future work.

- Chapter 4 is based on a paper of conference JURIX 2022 [Liu *et al.* 2022]:

  Xinghan Liu, Emiliano Lorini, Antonino Rotolo and Giovanni Sartor. Modelling and explaining legal case-based reasoners through classifiers. In *Legal Knowledge and Information Systems,* pages 83-92. IOS Press, 2022.

  There we apply BCL to legal case-based reasoning, show that the latter can be studied in the context of classifier and therefore many notions of XAI can be transitioned to use. Sections 4.4 and 4.6 are new, while Section 4.5 is enriched.

- Chapter 5 is based on a paper forthcoming in conference KR 2023:

  Carlos Aguilera-Ventura, Andreas Herzig, Xinghan Liu, Emiliano Lorini. Counterfactual reasoning via grounded distance. Proceeding of *20th International Conference on Principles of Knowledge Representation and Reasoning, forthcoming.*

  We define a notion of counterfactual conditional by Hamming distance in Chapter 3. It abbrevaites from a modal formula when the variables in the language are finite. We show in this section that in the infinite-variable case Hamming distance cannot be axiomatized in the logical systems that we interest. It indicates that given countably infinite variables in the language, every notion of distance in counterfactual reasoning, including ones people use in XAI, can be reformulated in terms of Hamming distance.

- Chapter 6 is based on a paper of conference WOLLIC 2022 [Liu & Lorini 2022].

  – Xinghan Liu and Emiliano Lorini. A logic of "black box" classifier systems. In *Logic, Language, Information, and Computation: 28th International Workshop, WOLLIC 2022, Iasi, Romania, 2022, Proceedings,* pages 158-174. Springer 2022.

  There we present the product modal logic for multi-classifiers PLC and show how to extend some explanations for white boxes to black box classifiers. Its section of introduction is extended.

- Chapter 7 is new. Two topics are discussed. The first one involves a complexity study motivated by a conceptual question of how hard black box explanations generally are. The second one is driven by bridging the long-existing axiomatic divide in the logical frameworks that we will present.

Chapter 2

# Background

Contents



## 2.1 Propositional Logic & Boolean Functions

On one hand, propositional logic and Boolean functions can be seen as "the same thing", especially considering that Boole, the father of symbolic logic, himself studied logic from the binary function viewpoint. On the other hand, the two studies are now quite different in terms of different focuses, purposes and representations. In particular, the "standard semantics" for the former is truth assignment, while for the latter is Boolean algebra.

With this in mind, I will use a "state semantics" for propositional logic, where a state is nothing but the set of variables assigned as true. Then, after introducing its standard definition, I will represent Boolean function by states instead of binary sequences. In such a way we can represent the two in the same framework.

### 2.1.1 Propositional logic: semantics and syntax

**Definition 2.1** (Propositional language)**.** *Fix a countable set of atomic propositions* $Atm = \{p_1, p_2, \dots\}$*, the propositional language is defined recursively by the following BNF*

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi$$

*where $p$ ranges over Atm. Let $\varphi \vee \psi$ abbreviate $\neg(\varphi \wedge \psi)$, $\varphi \to \psi$ abbreviate $\neg\varphi \vee \psi$, $\varphi \leftrightarrow \psi$ abbreviate $(\varphi \to \psi) \wedge (\psi \to \varphi)$, $\bot$ abbreviate $p \wedge \neg p$, $\top$ abbreviate $\neg\bot$.*

#### 2.1.1.1 State semantics

**Definition 2.2** (State)**.** *Let $s \in 2^{Atm}$. We call it a* state *(*valuation, value assignment*). The class of all states is $2^{Atm}$.*

**Definition 2.3** (State semantics)**.** *Let $s \in 2^{Atm}$. The semantic interpretation of propositional formula $\varphi$ relative to $s$ is recursively defined as follows*

$$\begin{aligned} s \models p &\iff p \in s \\ s \models \neg\varphi &\iff s \not\models \varphi, \text{ i.e. it is not } s \models \varphi \\ s \models \varphi \wedge \psi &\iff s \models \varphi \text{ and } s \models \psi \end{aligned}$$

*We call a formula $\varphi$* satisfiable*, if $\exists s \in 2^{Atm}$ with $s \models \varphi$; call $\varphi$* valid *and write $\models \varphi$, if $\forall s \in 2^{Atm}$ $s \models \varphi$.*

Moreover, we write $\Phi \models \psi$ to mean that $\forall s \in 2^{Atm}$, if $s \models \varphi$ for all $\varphi \in \Phi$ then $s \models \psi$.

#### 2.1.1.2 Hilbert Axiomatics

Given a logic there are usually many proof-theoretic systems for it, e.g. natural deduction and sequent calculus. For the sake of our purpose, we introduce the Hilbert axiomatic system.

**Definition 2.4.** *The Hilbert axiomatics of propositional logic contains the following axioms and inference of rule in Table 2.1.*

$$\begin{array}{ll} \varphi \to (\psi \to \varphi) & \\ (\varphi \to (\psi \to \chi)) \to ((\varphi \to \psi) \to (\varphi \to \chi)) & \\ (\neg\psi \to \neg\varphi) \to (\varphi \to \psi) & \\ \dfrac{\varphi \quad \varphi \to \psi}{\psi} & \text{(Modus Ponens)} \end{array}$$

Table 2.1: Axioms and rule of inference of propositional logic

**Definition 2.5** (Syntactic consequence)**.** *Let $\Phi$ be a set of formulas and $\psi$ be a formula. We write $\Phi \vdash \psi$, if $\psi$ is derivable from $\Phi$ by the axiomatics of propositional logic. In particular, if $\Phi$ is a singleton $\{\varphi\}$ we write $\varphi \vdash \psi$; and if $\Phi$ is empty we write $\vdash \psi$.*

The following theorem is fundamental, which indicates that the connective $\rightarrow$ and the meta-symbol $\vdash$ "coincide".

**Fact 2.1** (Deduction theorem)**.** *$\Phi \cup \{\varphi\} \vdash \psi$ if and only if $\Phi \vdash \varphi \rightarrow \psi$.*

**Definition 2.6** (Consistency)**.** *Let $\Phi$ be a set of formulas. We call $\Phi$ inconsistent, if $\Phi \vdash \bot$; otherwise call it consistent.*

**Fact 2.2.** *Fix a countable set of atomic propositions Atm. Propositional logic is sound and (strongly) complete with respect to $2^{Atm}$. That is, for any set of formulas $\Phi$ and formula $\varphi$, $\Phi \vdash \varphi$ implies $\Phi \models \varphi$; and $\Phi \models \varphi$ implies that $\Phi \vdash \varphi$.*

#### 2.1.1.3 Literal and normal forms

**Definition 2.7** (Literal, term and clause)**.** *A* literal *is an atomic proposition $p$ or its negation $\neg p$. A* term *(elementary conjunction) is a conjunction of literals; a clause (elementary disjunction) is a disjunction of literals.*

**Definition 2.8** (Maximal consistent set of literals)**.** *Fix a countable set of atomic propositions Atm. A maximal consistent set of literals is a set of literals $\varsigma$ s.t. $\varsigma$ is consistent; and $\forall p_i \in Atm$, either $p_i \in \varsigma$ or $\neg p_i \in \varsigma$.*

Here we introduce an important pair of abbreviations. Let $X \subseteq Y \subseteq^{\text{fin}} Atm$, where $A \subseteq^{\text{fin}} B$ is defined as $A \subseteq B$ and $A$ is finite. We define

$$\mathsf{cn}_{X,Y} := \bigwedge_{p \in X} p \wedge \bigwedge_{q \in X \setminus Y} \neg q \tag{2.1}$$

$$\mathsf{ds}_{X,Y} := \bigvee_{p \in X} p \vee \bigvee_{q \in X \setminus Y} \neg q. \tag{2.2}$$

We say that $\mathsf{cn}_{X,Y}$ *absorbs* $\mathsf{cn}_{X',Y'}$, or abusing terminology that the former is a subset of the later, noted $\mathsf{cn}_{X,Y} \subseteq \mathsf{cn}_{X',Y'}$, if $X \subseteq X'$ and $Y \subseteq Y'$. Similarly $\mathsf{ds}_{X,Y}$ is a subset of $\mathsf{ds}_{X',Y'}$, if $X \subseteq X'$ and $Y \subseteq Y'$.

**Definition 2.9** (DNF and CNF)**.** *A* disjunction normal form *(DNF) is a disjunction of some terms; a* conjunction normal form *(CNF) is a conjunction of some clauses.*

### 2.1.2 Boolean functions and Boolean expressions

Traditionally, Boolean functions are defined in an algebraic way based on binary sequences (strings, vectors), see, e.g., definitions in [Crama & Hammer 2011]. Nonetheless, there is a straightforward isomorphism between binary sequences on a set of atomic propositions $A$ and subsets of $A$ by simply interpreting the binary value $1, 0$ as membership and non-membership. Therefore I present definitions of Boolean function and relevant notions based on propositional semantics instead of binary sequences. We always fix a countable set of atomic propositions $Atm$.

**Definition 2.10** (Boolean function, redefined)**.** *A Boolean function is* $f : 2^A \longrightarrow \{0,1\}$*, with* $A \subseteq^{\text{fin}} Atm$*. A point* $X \subseteq A$ *is a true point of* $f$ *if* $f(X) = 1$*;* $X$ *is a false point of* $f$ *if* $f(X) = 0$*. We denote* $\bot$ *the constant function of value* 0 *and* $\top$ *the constant function of value* 1*.*

Notice here that we use $A$ a finite subset of $Atm$, because Boolean function is by definition finitary, while $Atm$ can be countably infinite. However, we can extend this definition to the infinitary case naturally so that any state $s$ is either a true point or a false point of the $f$, as we will do from next chapter on.

The next basic notion is a Boolean expression, which is a syntactic representation of a Boolean function.

**Definition 2.11** (Boolean expression)**.** *Let* $A \subseteq^{\text{fin}} Atm$*. We say a propositional formula* $\varphi$ expresses *a Boolean function* $f : 2^A \to \{0,1\}$*, or* $\varphi$ *is a Boolean expression of* $f$*, if* $\forall X \in f^{-1}(1)$*,* $\mathsf{cn}_{X,A} \models \varphi$ *and* $\forall X \in f^{-1}(0), \mathsf{cn}_{X,A} \models \neg\varphi$*.*

We will use propositional formula, Boolean formula and Boolean expression interchangeably. Also we omit $A \subseteq^{\text{fin}} Atm$ for the rest of the section since the context is clear.

**Definition 2.12** (Canonical DNF and CNF[Crama & Hammer 2011, Definition 1.10])**.** *Let* $f : 2^A \longrightarrow \{0,1\}$ *be a Boolean function. Then the* canonical DNF *(minterm expression)*

$$\bigvee_{X \subseteq A, X \in f^{-1}(1)} \mathsf{cn}_{X,A},$$

*and the* canonical CNF *(maxterm expression) is*

$$\bigwedge_{Y \subseteq A, Y \in f^{-1}(0)} \mathsf{ds}_{A \setminus Y, A}.$$

The definition says that we can express $f$ as the disjunction of all its true points, or the conjunction of the negation of all its false points. Hence the following fact becomes obvious.

**Fact 2.3** ([Crama & Hammer 2011, Theorem 1.4])**.** *Every Boolean function* $f : 2^A \longrightarrow \{0,1\}$ *can be expressed by a DNF and a CNF.*

This allows us to state the following fact as a corollary.

**Corollary 2.1.** *Every propositional formula expresses a unique Boolean function up to isomorphism.*

Hence the statement below says nothing but $\varphi$ is satisfiable if the function it expresses has a true point, which is obvious enough to be rightly called a corollary. Notice that the $\bot$ denotes a function, i.e. the constant function of value 0.

**Corollary 2.2** (Classifier semantics of propositional logic)**.** *A propositional formula* $\varphi$ *is satisfiable, if and only if* $\bot \neq \varphi$*.*

### 2.1.3 Prime implicants and essential variables

The most important notion of Boolean function theory that we will use through the thesis is *prime implicant*, which, and its dual *prime implicate* are defined formally as follows.

**Definition 2.13** (Prime implicant and implicate)**.** *Let* $f : 2^A \longrightarrow \{0,1\}$ *be a Boolean function,* $\varphi$ *express* $f$ *and* $X \subseteq Y \subseteq A$*. A term* $\mathsf{cn}_{X,Y}$ *is a* implicant *of* $f$*, if* $\mathsf{cn}_{X,Y} \models \varphi$*; it is a* prime implicant *of* $f$*, if for any* $X' \subseteq Y' \subseteq A$*, if* $X \subset X', Y \subset Y'$ *then* $\mathsf{cn}_{X',Y'}$ *is not an implicant of* $f$*. A clause* $\mathsf{ds}_{X,Y}$ *is a* implicate *of* $f$*, if* $\mathsf{ds}_{X,Y} \models \varphi$*; it is a* prime implicate *of* $f$*, if for any* $X' \subseteq Y' \subseteq A$*, if* $X \subset X', Y \subset Y'$*, then* $\mathsf{ds}_{X',Y'}$ *is not an implicate of* $f$*.*

**Fact 2.4.** *Every Boolean function can be represented by the disjunction of all its prime implicants, and by the conjunction of all its prime implicates.*

**Definition 2.14** (Complete DNF and CNF)**.** *The* complete DNF *(*Blake canonical form*) of a Boolean function is the disjunction of all its prime implicants. The* complete CNF *of a Boolean function is the disjunction of all its prime implicates.*

**Fact 2.5** ([Crama & Hammer 2011, Theorem 1. 13])**.** *Every Boolean function can be expressed by the complete DNF. And two Boolean functions are equal if and only if they have the same set of prime implicants.*

**Definition 2.15** (Prime and irredundant DNF)**.** *A DNF* $\varphi := \bigvee_{i\in\{1,\dots,m\}} \mathsf{cn}_{X_i,A}$ *is said to be a* prime DNF *of a Boolean function* $f$*, if every term of it is a prime implicant of* $f$*. We say that* $\varphi$ *is an* irredundant DNF *of* $f$*, if there is no* $k \in \{1,\dots,m\}$ *s.t.* $\bigvee_{i\in\{1,\dots,m\}\setminus\{k\}} \mathsf{cn}_{X_i,A}$ *expresses* $f$*; otherwise* $f$ *is* redundant*.*

A notion closed related to prime implicant is essential variable. Plainly speaking, a variable is *inessential* for a function, if it does not occur in any of its prime implicant.

**Definition 2.16** (Essential variable)**.** *Let* $f : 2^A \longrightarrow \{0,1\}$ *be a Boolean function and* $p_k \in A$*. We say that the variable* $p_k$ *is* inessential *(*dummy *for* $f$*, or* $f$ *is* independent on $p_k$*), if* $\forall X \subseteq A \setminus \{p_k\}, f(X \cup \{p_k\}) = f(X)$*. Otherwise we say* $p_k$ *is* essential*.*

**Fact 2.6** ([Crama & Hammer 2011, Theorem 1.17])**.** *Let* $f : 2^A \longrightarrow \{0,1\}$ *be a Boolean function and* $p \in A$*. The followings are equivalent:*

1. *$p$ is inessential for $f$;*
2. *$p$ does not occur in any prime implicant of $f$;*
3. *$f$ has a DNF representation in which $p$ does not occur.*

### 2.1.4 Monotone variables and functions

The last topic of Boolean function theory that we will use later is on monotonicity. We give the key definitions here, and in Chapter 4 we will show how to generalize them to partial Boolean functions and use them in legal case-based reasoning.

**Definition 2.17** (Monotone variable)**.** *Let $f : 2^A \longrightarrow \{0,1\}$ and $p \in A$. We say that $f$ is* positive *(resp.* negative*) in variable $p$ if $\forall X \subseteq A, f(X) \leq f(X \cup \{p\})$ (resp. $f(X) \geq f(X \cup \{p\})$). We say that $f$ is* monotone *in $p$ if $f$ is either positive or negative in $p$.*

**Definition 2.18** (Monotone BF)**.** *A Boolean function is* positive *(resp.* negative*) if it is positive (resp. negative) in each of its variables. The function is* monotone *if it is either positive or negative. Moreover, we also say $\varphi$ is positive (resp. negative, monotone), if the Boolean function it expresses is so.*

**Fact 2.7** ([Crama & Hammer 2011, Theorem 1.21])**.** *Let $f : 2^A \longrightarrow \{0,1\}$ and $p \in A$. The folllowings are equivalent:*

1. *$f$ is positive in $p$;*
2. *$\neg p$ does not occur in any prime implicant of $f$;*
3. *$f$ has a DNF expression in which $\neg p$ does not occur.*

## 2.2 Explanations for Classifier Systems

The question of what is an explanation is an issue that has been long discussed in philosophy. Many types of explanation have been identified and developed, including deductive-nomological explanations [Hempel & Oppenheim 1948], statistical relevant explanations and pragmatic theories of explanations [Van Fraassen 1980] etc. While this is not the place for a detailed analysis of the different philosophical foundations within and between symbolic and subsymbolic XAI approaches, this section will introduce some basic notions, characterizations, and examples as background knowledge. (Conceptual discussions can be found at the beginning of Chapters 3 and 6.) The following notions are used in both approaches.

**Global and local explanations** By global explanation we mean an explanation applies to all input instances. Usually obtaining global explanation is possible when we have knowledge about how the classifier globally behaves. Typical methods are, e.g. partial dependence plot (PDP) and Global Surrogate.

In contrast, local explanation only works for the current/actual input instance. A typical example is counterfactual explanation, which is always indexed by the actual/factual state. Arguably XAI places more emphasis on local explanation, for the main concern is whether the classification/decision/prediction for some particular instance is fair or trustworthy.

**Explanandum and explanans** The pair of notions of explanation is created by Carl Hempel [Hempel & Oppenheim 1948]. Roughly speaking an *explanandum* is a proposition describing a phenomenon to be explained, and an *explanans* is a proposition that explains the former, i.e. is responsible for the occurrence of the phenomenon.

The explanandum of the local explanation, for all approaches, is certainly nothing but $f(s)$, namely the classification of $f$ for the current instance $s$. However, different approaches have different explanans. For example, as we will see the explanans can be a "part" of the input instance in the symbolic approach. In the subsymbolic approach it could be others, e.g. a simpler classifier or the importance of a feature.

**Perturbation-based explanations** As mentioned in the Introduction, plainly speaking, perturbation is nothing but changes the values of some features of the current input instance. In the binary context, perturbation is just to flip the value from zero to one or the other way around. Perturbation-based explanations refer to methods that to obtain knowledge of the target classifier system by perturbing some inputs. When the inner mechanism of the classifier is unknown (some researchers like [Ribeiro *et al.* 2016] call it *model-agnostic*), e.g. a black box classifier trained by machine learning that is practically impossible to open, perturbation-based explanations are the only ways to explain it.

### 2.2.1 Subsymbolic approach

The subsymbolic approach focuses on classifier systems trained by machine learning. Those systems are notoriously hard to explain and hence are called black boxes, i.e. unable to open. Since globally explain a classifier is almost impossible for these black boxes, the recent focus is on explaining the classification of a given instance, which was embarked by the LIME paper [Ribeiro *et al.* 2016]).

#### 2.2.1.1 LIME

LIME is short for "local interpretable model-agnostic explanation". Local means focusing on a given instance; model-agnostic means no unknown of the inner mechanism of the model; and explanation here means a classifier that simpler than the black box. The method LIME searches the object function which reaches the balance between precision and simplicity.

**Definition 2.19** (LIME in [Ribeiro *et al.* 2016])**.** *Fix a target, black box classifier $f$ and an input $s$, LIME is defined as*

$$\xi(s) = \arg\min_{g \in F_S} \mathbf{L}(f, g, \pi_{s'}) + \Omega(g) \tag{2.3}$$

*where $F_S$ is the space of functions; $\pi_{s'}$, plainly speaking, is the neighborhood of the*

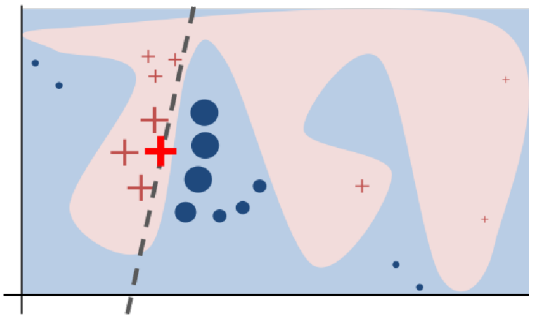


Figure 2.1: A toy example of LIME in [Ribeiro *et al.* 2016], where the black box classifier $f$ is represented by the blue-pink background. The red cross is the local instance to be explained. The dashed line is a LIME-classifier for $f$. The LIME-classifier makes no sense for the global explanation, but is both simple and relatively precise for the red cross.

*current input* $s$;[1] $\mathbf{L}(f, g, \pi_{s'})$ *is the loss function measuring the difference between* $f$ *and* $g$ *with respect to* $\pi_{s'}$*; and* $\Omega(g)$ *is a measurement of the complexity of* $g$.

Therefore, LIME gives the set of classifiers $g$s as explanations, which have the best belance between precision (on the neighbors of the local input) and simplicity.

#### 2.2.1.2 SHAP

LIME and some other local explanations belong to the so called additive feature attribution method. The key part is that an explanation model is a linear function of binary variables $g(z') = \phi_0 + \sum_{1 \le i \le m} \phi_i z'_i$, where $z' \in \{0,1\}^m$, with $m$ being the number of simplified input features, and $\phi_i \in \mathbb{R}$.

The key notion is the Shapley value in cooperative game theory. Players cooperate as coalitions and receive a certain profit therefrom. The Shapley value is a valuation of the contribution of a player with respect to all possible coalitions.

Features are viewed as players of the game, whose goal is to predict the output of a given instance in order to minimize the difference between the prediction and the average prediction of instances.

Suppose the set of all features are enumerated as $p_1, p_2, \ldots, p_m$, and let $J \subseteq \{1, \ldots m\}$ indicates a subset of all features. The Shapley value of $p_i$, viewed as the contribution of the $i$-th feature, is computed as following:

$$\phi_i(f) = \sum_{J \subseteq \{1,2,\ldots,m\} \setminus \{i\}} \frac{|J|!(m - |J| - 1)!}{m!} (f(J \cup \{i\}) - f(J)). \tag{2.4}$$

Though the thought of using Shapley value for local explanation has already thoroughly studied in [Strumbelj & Kononenko 2010], it is [Lundberg & Lee 2017]

[1] There is a $'$, because strictly speaking the input is not $s$ itself but some simplified version as an "interpretable input". Technical details can be found in their paper and [Lundberg & Lee 2017].

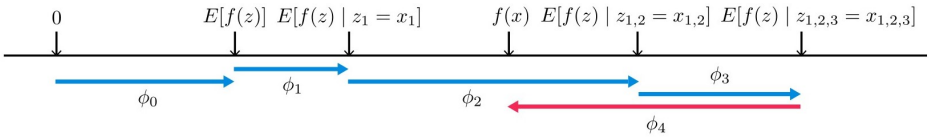



Figure 2.2: A toy example of SHAP in [Lundberg & Lee 2017] on a black box $f$ and an instance $x$. $E(f(z))$ represents the base value, i.e. $\phi_0$, given we know nothing about any feature's Shapley value. And $\phi_i$ is the Shapley value of the $i$-th feature. The ordering matters if the model is non-linear or features are dependent.

who gain much more attention with the explosive growth of explainable AI. The kernel SHAP presented by the latter is defined as follows.

**Definition 2.20** (Shapley kernel in [Lundberg & Lee 2017])**.** *The Shapley kernel for LIME is defined as*

1. $\Omega(g) = 0$
2. $\pi_{s'}(z') = \dfrac{(m-1)}{\binom{m}{|z'|}|z'|(m-|z'|)}$
3. $\mathbf{L}(f, g, \pi_{s'}) = \sum_{z' \in Z}(f(h_s(z') - g(z'))^2 \pi_{s'}(z')$.

Kernel SHAP is thus a particular algorithm of LIME, which does not take the complexity of the explanation model into account at all by letting $\Omega(g) = 0$. One can argue whether it is still in line with LIME since it overlooks the main consideration of LIME. Nevertheless, SHAP has been widely applied especially in medical diagnosis studies.

### 2.2.2 Symbolic approach

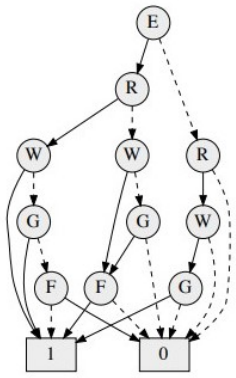
Many machine learning classifiers have been compiled as Boolean circuits, including Bayesian networks and some neural networks. Theoretically by binarization one can transform any features into zero-one vectors however large they may be. Figure 2.3 is a toy example of a Boolean classifier.

In the context of local explanation as [Ignatiev *et al.* 2020b] points out, we can explain the current classification directly by answering a "why" question; and also explain it indirectly by answering a "why not" question. They are stated as follows.

- Why does $f$ classify $s$ as $c$?
- Why not classify $s$ as non-$c$?

And a main driven force of XAI is to make the system trustworthy, fair and unbiased. Hence another question would be the follow.

- Is the classification $c$ of $f$ for $s$ biased/fair?




| Applicant | Scott | Robin | April |
|---|---|---|---|
| **E**ntrance Exam | ✓ | ✓ | ✓ |
| **F**irst Time Applicant | ✗ | ✓ | ✓ |
| **G**ood GPA | ✓ | ✓ | ✓ |
| **W**ork Experience | ✓ | ✓ | ✓ |
| **R**ich Hometown | ✓ | ✓ | ✗ |
| Decision | 1 | 1 | 1 |

Figure 2.3: An admission classifier in [Darwiche & Hirth 2020] represented by an OBDD (ordered binary decision diagram), where solid and dotted lines of a feature denote that the feature value of the current instance is truth and falsity respectively. It can be expressed by the propositional formula $E \wedge (F \wedge (G \vee W) \vee (\neg F \wedge R)) \vee (G \wedge R \wedge W)$.

**Prime implicant explanation** To answer the "why" question, it is natural to think of prime implicants as a cause of the classification. The prime implicant explanation works as follows. Let $f$ be a Boolean function and $s$ an instance. A prime implicant of $f$, $\mathsf{cn}_{X,Y}$ is an explanation of $s$, if $s \cap Y = X$. In other words, the prime implicant $\mathsf{cn}_{X,Y}$ is true at state $s$, namely $s$ has the property $\mathsf{cn}_{X,Y}$.

> A PI explanation is a *subset-minimal* part of the actual instance s.t. the classification keeps invariant under perturbing all *the other variables*.

Other names are *sufficient reason* [Darwiche & Hirth 2020] and *abductive explanation* [Ignatiev *et al.* 2019]. We adopt the latter due to its "duality" with contrastive explanation introduced below.

**Contrastive explanation and counterfactual explanation** To answer the "why not" question, we need counterfactual reasoning for the hypothesis "what if the actual input were perturbed in this way?".

In [Ignatiev *et al.* 2020b] contrastive explanation is formed as a counterpart of abductive explanation, which informally we can speak of as follows.

> A contrastive explanation is a *subset-minimal* part of the actual instance s.t. the classification changes under perturbing all the variables.

In both conceptual and formal senses, contrastive explanation can be seen as a special case of counterfactual explanation. Namely, a counterfactual condition whose antecedent consists only conjunction of literals.

**Bias and fairness** In order to address whether a classification is biased, we need to have the prior knowledge of protected features in our formal language. A protected feature is a feature that is deemed as causing discriminism such as race, gender and age. The idea of biased classification aligns with common sense, e.g. in [Darwiche & Hirth 2020] it is expressed as the following.

> A classification for the actual instance is biased, if the classification changes by only perturbing the protected features of the current instance.

## 2.3 Modal Logics

In this section we introduce the basic notions of modal logics. Most definitions can be found in most textbooks on the subject, e.g. [Blackburn *et al.* 2001].

### 2.3.1 Kripke semantics

**Definition 2.21** (Basic modal language)**.** *Fix a set of countable atomic propositions Atm, the language of (uni-)modal language is defined in the following BNF:*

$$\varphi \quad ::= \quad p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \Box\varphi,$$

*where $p$ ranges over Atm.*

We let $\Diamond$ abbreviate $\neg\Box\neg$, and read $\Box\varphi$ and $\Diamond\varphi$ "it is necessarily that $\varphi$" and "it is possibly that $\varphi$" respectively.

**Definition 2.22** (Frame)**.** *A (Kripke) modal frame is a pair $(W, R)$ where $W$ is a set of points so called* possible worlds*, and $R \subseteq W \times W$ is called* accessibility relation*.*

**Definition 2.23** (Kripke model)**.** *A Kripke model $M = (W, R, V)$ is a triple where $(W, R)$ is a frame and $V : W \longrightarrow 2^{Atm}$ is a valuation function. Let $w \in W$, and we call $(M, w)$ a pointed model.*

**Definition 2.24** (Satisfaction relation)**.** *Let $M = (W, R, V)$ be a Kripke model and $w \in W$. The satisfaction relation regarding the pointed model $(M, w)$ is recursively defined as follows:*

$$\begin{aligned}
(M, w) \models p &\iff p \in V(w) \\
(M, w) \models \neg\varphi &\iff (M, w) \not\models \varphi \\
(M, w) \models \varphi \wedge \psi &\iff (M, w) \models \varphi \text{ and } (M, w) \models \psi \\
(M, w) \models \Box\varphi &\iff \forall v \in R(w), (M, w) \models \varphi.
\end{aligned}$$

*We say $\varphi$ is* satisfied *(*locally true*) in $(M, w)$ if $(M, w) \models \varphi$; $\varphi$ is* globally true *in $M$ if $\forall v \in W, (M, v) \models \varphi$, and write $M \models \varphi$; $\varphi$ is* valid *if for any of its model $M$, $M \models \varphi$, and write $\models \varphi$.*

| Acronyms | semantic constraints | characteristic axioms |
|---|---|---|
| $T$ | $w \in R(w)$ | $\Box\varphi \to \varphi$ |
| $D$ | $R(w) \neq \emptyset$ | $\Box\varphi \to \Diamond\varphi$ |
| $B$ | $wRv$ implies $vRw$ | $\varphi \to \Box\Diamond\varphi$ |
| 4 | $wRv\&vRu$ implies $wRu$ | $\Box\varphi \to \Box\Box\varphi$ |
| 5 | $wRv\&wRu$ implies $vRu$ | $\Diamond\varphi \to \Box\Diamond\varphi$ |

Table 2.2: Characteristic axioms

**Definition 2.25** (Some relations)**.** *Let $(W, R)$ be a modal frame, it is*

- reflexive*, if $\forall w \in W, wRw$;*
- shift-reflexive*, if $\forall w, v \in W, wRv$ implies $vRv$*
- symmetric*, if $\forall w, v \in W, wRv$ and $vRw$;*
- serial*, if $\forall w \in W, \exists v \in W, wRv$;*
- transitive*, if $\forall w, v, u \in W$, $wRv$ and $vRu$ implies $wRu$;*
- euclidean*, if $\forall w, v, u \in W, wRv$ and $wRu$ implies $vRu$;*
- dense*, if $\forall w, v \in W$, $wRv$ implies $\exists u \in W, wRu$ and $uRv$;*
- convergent*, if $\forall w, v, u, wRv$ and $wRu$ implies $\exists x \in W, vRx$ and $uRx$.*

The final definition of this subsection is bisimulation, which plays a fundamental role in many semantic equivalence results.

**Definition 2.26** (Bisimulation)**.** *Let $M = (W, R, V)$ and $M' = (W', R', V')$ be two models. We say that they are bisimilar, if there exists a relation $Z \subseteq W \times W'$ s.t. $\forall w \in W, w' \in W'$ with $wZw'$, the following conditions are satisfied:*

1. *(Atom) $\forall p \in Atm, p \in V(w) \iff p \in V'(w')$;*
2. *(Zig) $\forall v \in W$, if $wRv$ then $\exists v' \in W', w'R'v'$ and $vZv'$;*
3. *(Zag) $\forall v' \in W$, if $w'R'v'$ then $\exists v \in W, wRv$ and $vZv'$.*

### 2.3.2 Axiomatics and completeness

Like propositional logic, we use Hilbert axiomatization for modal logics.

**Definition 2.27** ($\mathsf{K}$)**.** *The normal modal logic $\mathsf{K}$ results from extending propositional logic with the following axiom and rule of inference*

$$\Box(\varphi \to \psi) \to (\Box\varphi \to \Box\psi) \qquad (K)$$

$$\frac{\varphi}{\Box\varphi} \qquad (Necessitation)$$

Notice that as a rule of inference, Necessitation differs Modus Ponens in a subtle way: the former requires that f $\varphi$ *derives* from $\mathtt{K}$, then $\Box\varphi$ *derives* from $\mathtt{K}$. (Hence it should be $\vdash \varphi$ and $\vdash \Box\varphi$, but convention we omit them.) As a result, the latter preserves both validity, global truth and local truth (i.e. satisfaction), while the former only preserve the first two [Blackburn *et al.* 2001, p. 35]. To apply Modus Ponens, $\varphi$ and $\varphi \to \psi$ do not need be tautological but suffice if they are true at the given state/world. In contrast, to apply Necessitation $\varphi$ must be tautological.

To see that, consider a model $M = (W, R, V)$ where $W = \{w, v\}$, $R = \{(w, v)\}$, $V(w) = \{p\}$ and $V(v) = \emptyset$. We have $p$ locally true at $w$, i.e. $(M, w) \models p$, but we have no $(M, w) \models \Box p$.

**Definition 2.28** (More logic and characteristic axioms)**.** *Some basic extensions of* $\mathsf{K}$ *are listed below, where* $+$ *means adding the latter (some characteristic axiom) into the former (some logic system).*

$$\begin{aligned} \mathsf{T} &= \mathsf{K} + T \\ \mathsf{D} &= \mathsf{K} + D \\ \mathsf{B} &= \mathsf{K} + B \\ \mathsf{S4} &= \mathsf{T} + 4 \\ \mathsf{KD45} &= \mathsf{K} + D + 4 + 5 \\ \mathsf{S5} &= \mathsf{S4} + B = \mathsf{T} + 5 \end{aligned}$$

**Definition 2.29** (Soundness, completeness and strong completeness in modal logic)**.** *Fix a modal logic, its syntactic consequence* $\vdash$ *and semantic consequence* $\models$*. We say that it is sound, if for any set of formulas* $\Phi$ *and formula* $\varphi$ *(in its language),* $\Phi \vdash \varphi$ *implies* $\Phi \models \varphi$*; it is (weakly) complete, if* $\models \varphi$ *implies* $\vdash \varphi$*; it is strongly complete, if for any set of formulas* $\Phi$*,* $\Phi \vdash \varphi$ *implies that for all pointed model* $(M, w)$ *of the logic, if* $\forall \psi \in \Phi, (M, w) \models \psi$*, then* $(M, w) \models \varphi$*.*

**Fact 2.8.** *All the modal logics listed above are both sound and strongly complete.*

### 2.3.3 State semantics for S5

It is now time to give the answer for a possible question raised by the interested reader: why we call $s$ a state? It can be traced back to Carnap [Carnap 1967] who defined the *state description* as, technically, nothing but a set of atomic propositions. We can use states instead of worlds as elements in models of modal logic. In other words, let $w$ be identified with its valuation $V(w)$ in the Kripke model. With this move it is possible to provide a semantics for the modal logic S5, as a natural extension of the state semantics we gave for propositional logic.

**Definition 2.30.** *An S5 (state) model is nothing but a set of states* $S$*, where* $\forall s \in S, s \in 2^{Atm}$*. A pointed S5 model is* $(S, s)$ *with* $s \in S$*. Let* $\varphi$ *be a formula,* $(S, s)$ *a*

*pointed model, the satisfaction relation is defined as follows*

$$\begin{aligned}
(S,s) \models p &\iff p \in s \\
(S,s) \models \neg\varphi &\iff (S,s) \not\models \varphi \\
(S,s) \models \varphi \wedge \psi &\iff (S,s) \models \varphi \text{ and } (S,s) \models \psi \\
(S,s) \models \Box\varphi &\iff \forall s' \in S, (S,s') \models \varphi.
\end{aligned}$$

*The class of all S5 models is noted* $\mathbf{S}$*, which equals* $2^{2^{Atm}}$*.*

Notions of global truth and validity are defined in the same way.

It is not hard to see a straightforward relation between Boolean classifiers and S5. We can simply say that an S5 model $S$ represents a Boolean function $f : 2^A \longrightarrow \{0,1\}$ with $A \subseteq^{\text{fin}} Atm_0$, if and only if $\{s \cap A : s \in S\} = f^{-1}(1)$. Namely any $X \subseteq A$ is a true point of $f$, if and only if there is some $s$ present in $S$ s.t. $s \cap A = X$. Nevertheless, for more generality we do not use the presence vs. absence to represent binary outputs. Instead, we will introduce a finite set of classifications to partition the states in $S$. In such a way we are allowed to represent classifiers with a partial domain and a finitary output, which is the topic of the next chapter.

Chapter 3

# A Logic of Binary-input Classifiers and Their Explanation

Recent years have witnessed a renewed interest in the explanation of classifier systems in the field of explainable AI (XAI). The standard approach is based on propositional logic. We present a modal language which supports reasoning about binary input classifiers and their properties. We study a family of classifier models, axiomatize it as two proof systems regarding the cardinality of the language and show completeness of our axiomatics. Moreover, we show that the satisfiability checking problem for our modal language is NEXPTIME-complete in the infinite-variable case, while it becomes polynomial in the finite-variable case. We moreover identify an interesting NP fragment of our language in the infinite-variable case. We leverage the language to formalize counterfactual conditional as well as a variety of notions of explanation including abductive, contrastive and counterfactual explanations, and biases. Finally, we present two extensions of our language: a dynamic extension by the notion of assignment enabling classifier change and an epistemic extension in which the classifier's uncertainty about the actual input can be represented.

## Contents

## 3.1 Introduction

The notions of explanation and explainability have been extensively investigated by philosophers [Hempel & Oppenheim 1948, Kment 2006, Woodward 2000] and are key aspects of AI-based systems given the importance of explaining the behavior and prediction of an artificial intelligent system. Classifier systems compute a given function in the context of a classification or prediction task. Artificial feed-forward neural networks are special kinds of classifier systems aimed at learning or, at least approximating, the function mapping instances of the input data to their corresponding outputs. Explaining why a system has classified a given instance in a certain way is crucial for making the system intelligible and for finding biases in the classification process. This is the main target of explainable AI (XAI). Thus, a variety of notions have been defined and used to explain classifiers including abductive, contrastive and counterfactual explanations [Biran & Cotton 2017, Wachter *et al.* 2017, Dhurandhar *et al.* 2018, Ignatiev *et al.* 2019, Mittelstadt *et al.* 2019, Miller 2019, Mothilal *et al.* 2020, Verma *et al.* 2020, Miller 2021, Mertes *et al.* 2022].

Inputs of a classifier are called instances, i.e., valuations of all its variables/features/factors, and outputs are called classifications/predictions/decisions.[1] When both input and output of the classifier are binary, it is just a Boolean function $f : \{0,1\}^n \longrightarrow \{0,1\}$, and furthermore can be expressed by a propositional formula. This isomorphism between Boolean functions and logic has been known ever since the seminal work of Boole. Recently there has been a renewed interest in Boolean functions in the area of logic-based approaches to XAI [Shih *et al.* 2018, Ignatiev *et al.* 2019, Darwiche & Hirth 2020, Ignatiev *et al.* 2020b, Shi *et al.* 2020, Audemard *et al.* 2021, Amgoud & Ben-Naim 2022]. They concentrate on *local* explanations, i.e., on explaining why an actual instance is classified in a certain way.

We argue that it is natural and fruitful to represent binary-input classifiers and their explanation with the help of a modal language. To that end let us first explain the conceptual foundation of explanation in the context of classifiers, which is largely ignored in the recent literature.

What is an explanation? Despite subtle philosophical debates,[2] by explanation people usually mean *causal explanation*, an answer to a "why" question in terms of "because". Then what is a causal explanation? Ever since the seminal deductive-nomological (D-N) model [Hempel & Oppenheim 1948], one can view it

[1] Recall that we use them as synonyms through the thesis. Another set of synonyms is perturbation/intervention/manipulation. The variety of terminology is unfortunate.

[2] E.g., whether all explanations are causal, whether metaphysical explanation/grounding should be distinguished from causal explanation.

as a logical relation between an explanandum (the proposition being explained) and an explanans (the proposition explaining), which is itself expressible by a logical formula. According to the D-N model, a causal explanation of a certain fact should include a reference to the *laws* that are used for deducing it from a set of premises.

More recently Woodward & Hitchcock [Woodward & Hitchcock 2003, p. 2, p. 17] (see also [Woodward 2003, Ch. 5 and 6]) proposed that causal explanations make reference to generalizations, or descriptions of dependency relations, which specify relationships between the explanans and explanandum variables. No need of being laws, such generalizations exhibit how the explanandum variable is counterfactually dependent on the explanans variables by relating changes in the value of the latter to changes in the value of the former.[3] According to Woodward & Hitchcock, a generalization used in a causal explanation is *invariant under intervention* insofar as it remains stable after changing the actual value of the variables cited in the explanation.[4]

We claim that existing notions of explanation leveraged in the XAI domain rest upon the idea of invariance under intervention. However, while Woodward & Hitchcock apply it to the notion of generalization, in the XAI domain it usually concerns the result of the classifier's decision to be explained. Another minor difference with Woodward & Hitchcock is terminological: when explaining the decision of a binary classifier system, the term 'perturbation' is commonly used instead of 'intervention'. But they both mean switching some features' values from the current ones to other ones. Let us outline it by introducing informally our running example.

**Example 3.1** (Applicant Alice, informal)**.** *Alice applies for a loan. She is not male, she is employed, and she rents an apartment in the city center, which we note* $\neg male \wedge employed \wedge \neg owner \wedge center$*. The classifier* $f$ *only accepts the application if the applicant is employed, and either is a male or owns a property. Hence, Alice's application is rejected.*

In the XAI literature $\neg male \wedge \neg owner$ is called an abductive explanation (AXp) [Ignatiev *et al.* 2019] or sufficient reason [Darwiche & Hirth 2020] of the actual decision of rejecting Alice's application, because perturbing the values of the other features ('employment' and 'address' in this setting), while keeping the values of 'gender' or 'ownership' fixed, will not change the decision. More generally, for a term (a conjunction of literals) to be an abductive explanation of the classifier's actual decision, the classifier's decision should be invariant under perturbation on the variables not appearing in the term.[5]

[3] Using the notion of counterfactual dependence for reasoning about natural laws and causality traces back to [Goodman 1955, Lewis 1979, Lewis 1995]. The focus nowadays, e.g. [Woodward 2000, Halpern 2016], is on the use of counterfactuals for modeling the notion of *actual* cause in order to test (rather than define) causality.

[4] Woodward & Hitchcock also discuss invariance with respect to the background conditions not figuring in the relationship between explanans and explanandum. Nonetheless, they consider this type of invariance less central to causal explanation.

[5] AXp satisfies an additional restriction of minimality that will be elucidated at a later stage: an AXp is a 'minimal' term for which the classifier's actual decision is invariant under perturbation.

On the contrary, $\neg male$ is called a contrastive explanation (CXp) [Ignatiev *et al.* 2020b],[6] because perturbing nothing but 'gender' will change the decision from rejecting the application to accepting it. Therefore, the "duality" between two notions rests on the fact that AXp answers a *why*-question by indicating that the classification would stay unchanged under intervention on variables other than 'gender' and 'ownership', whereas CXp answers a *why not*-question by indicating that the classification would change under intervention on 'gender'. More generally, for a term (a conjunction of literals) to be a contrastive explanation of the classifier's actual decision, the classifier's decision should be variant under perturbation on *all* variables appearing in the term, where 'variant' is assumed to be synonym of 'non-invariant'.

As Woodward [Woodward 2000, p. 225, footnote 5] clarifies:

> [I]nvariance is a *modal* notion – it has to do with whether a relationship would remain stable under various hypothetical changes.

Therefore, following Woodward, the most natural way of modeling invariance is by means of a modal language whereby the notions of necessity and possibility can be represented. This is the approach we take in this work.

In particular, in order to model explanations in classifier systems, we use a modal language with a *ceteris paribus* (other things being equal) flavor. Indeed, the notion of invariance under intervention we consider presupposes that one intervenes on specific input features of the classifier, while keeping the values of the other input features unchanged (i.e., the values of the other input features being equal). So, for Alice's example we expect two modal formulas saying:

a) 'gender' and 'ownership' keeping their actual values, changing other features' values *necessarily* does not affect the actual decision of rejecting Alice's application;

b) other features keeping their actual values, changing the value of 'gender' *necessarily* modifies the classifier's decision of rejecting Alice's application.

Specifically, we will extend the *ceteris paribus* modal logic introduced in [Grossi *et al.* 2015] by a finite set of atoms representing possible decisions/classifications of a classifier and axioms regarding them. The resulting logic is called BCL which stands for Binary input Classifier Logic, since the input variables of a classifier are assumed to be binary. One may roughly thinks of its models as S5 models supplemented with a classification function which allows us to fully represent a classifier system. Each state in the model corresponds to a possible input instance of the classifier. Moreover, the classification function induces a partition of the set of instances, where each part corresponds to a set of input instances which are classified equally by the classifier. We call these models *classifier models*. BCL and its extensions open up new vistas including (i) defining counterfactual conditionals and studying their relationship with the notions of abductive and contrastive explanation, (ii) modeling

[6] We prefer the notation AXp used by Ignatiev et al.[Ignatiev *et al.* 2019, Ignatiev *et al.* 2020b] for its connection with CXp.

classifier dynamics through the use of formal semantics for logics of communication and change [Van Benthem *et al.* 2006, van Ditmarsch *et al.* 2007], and (iii) viewing a classifier as an agent and representing its uncertainty about the actual instance to be classified through the use of epistemic logic [Fagin *et al.* 1995].

Before concluding this introduction, it is worth noting that a classifier system is a simple form of causal system whose only dependency relations are between the input variables and the single output variable. Unlike Bayesian networks or artificial neural networks, a classifier system does not include 'intermediate' endogenous variables that, at the same time, depend on the input variables and causally influence the output variable(s). Therefore, many distinctions and disputations addressed in the theory of causality and causal explanation do not emerge in our work. For example, the vital distinction between correlation and causality [Pearl 2009], the criticism of *ceteris paribus* as natural law [Woodward 2000], and whether a causal explanation requires providing information about a causal history or causal chain of events [Lewis 1986]. All these subtleties only show up when the causal structure is complex, and hence collapse in a classifier system, which has only two layers (input-output).

The chapter is structured as follows. In Section 2 we introduce our modal language as well as its formal semantics using the notion of classifier model. In Section 3 two proof systems are given, BCL and 'weak' BCL (WBCL). We show they are sound and complete relative to the classifier system semantics with, respectively, finite-input and infinite-input variables. Section 4 presents a family of counterfactual conditional operators and elucidates their relevance for understanding the behavior of a classifier system. Section 5 is devoted to classifier explanation. We extend the existing notions of explanation for Boolean classifiers to binary input classifiers. The notions include AXp, CXp and bias in the field of XAI. We will see that in the binary input classifier setting their behaviors are subtler. Besides, their connection with counterfactual is studied. Finally, in Section 6 we present two extensions of our language: (i) a dynamic extension by the notion of assignment enabling classifier change and (ii) an epistemic extension in which the classifier's uncertainty about the actual input can be represented. Further possible researches are discussed in the conclusion. Main results are either proven in the appendix or pointed out as corollaries.

## 3.2 A Language for Binary-input Classifiers

In this section we introduce a language for modeling binary-input classifiers and its semantics. The language has a *ceteris paribus* nature that comes from the *ceteris paribus* operators of the form $[X]$ it contains. They were first introduced in [Grossi *et al.* 2015].[7]

[7] More recently, similar operators have been used in the context of the logic of functional dependence by Baltag & van Benthem [Baltag & van Benthem 2021].

### 3.2.1 Basic Language and Classifier Model

Let $Atm_0$ be a countable set of atomic propositions with elements noted $p, q, \ldots$ which are used to represent the value taken by an input variable (or feature). When referring to input variables/features we sometimes use the notation '$p$' to distinguish it from the symbol $p$ for atomic proposition. In this sense, the atomic proposition $p$ should be read "the Boolean input variable '$p$' takes value 1", while its negation $\neg p$ should be read "the Boolean input variable '$p$' takes value 0".

We introduce a finite set $Val$ to denote the *output values* (classifications, decisions) of the classifier. Elements of $Val$ are also called *classes* in the jargon of classifiers. For this reason, we note them $c, c', \ldots$ For any $c \in Val$, we call $\mathtt{t}(c)$ a decision atom, to be read as "the actual decision (or output) takes value $c$", and have $Dec = \{\mathtt{t}(c) : c \in Val\}$. Finally, let $Atm = Atm_0 \cup Dec$ be the set of atomic formulas. Notice symbols $c$ and $p$ have different statuses: $p$ is an atomic proposition representing an atomic fact, while $c$ is not. This explains why $c$ (an output value) and $\mathtt{t}(c)$ (an atomic formula representing the fact that the actual output has a certain value) are distinguished.

The modal language $\mathcal{L}(Atm)$ is hence defined by the following grammar:

$$\varphi \quad ::= \quad p \mid \mathtt{t}(c) \mid \neg\varphi \mid \varphi \wedge \varphi \mid [X]\varphi,$$

where $p$ ranges over $Atm_0$, $c$ ranges over $Val$, and $X$ is a finite subset of $Atm_0$ which we note $X \subseteq^{\text{fin}} Atm_0$. As we will justify below, we often write $\Box$ instead of $[\emptyset]$.

The set of atomic formulas occurring in a formula $\varphi$ is noted $Atm(\varphi)$.

The formula $[X]\varphi$ has to be read "$\varphi$ is necessary all features in $X$ being equal" or "$\varphi$ is necessary regardless of the truth or falsity of the atoms in $Atm_0 \setminus X$". Operator $\langle X\rangle$ is the dual of $[X]$ and is defined as usual: $\langle X\rangle\varphi =_{def} \neg[X]\neg\varphi$.

The language $\mathcal{L}(Atm)$ is interpreted relative to classifier models whose class is defined as follows.

**Definition 3.1** (Classifier model)**.** *A classifier model (CM) is a tuple $C = (S, f)$ where:*

- *$S \subseteq 2^{Atm_0}$ is a set of states or input instances, and*
- *$f : S \longrightarrow Val$ is a decision (or classification) function.*

*The class of classifier models is noted* **CM**.

A pointed classifier model is a pair $(C, s)$ with $C = (S, f)$ a classifier model and $s \in S$. Formulas in $\mathcal{L}(Atm)$ are interpreted relative to a pointed classifier model, as follows.

**Definition 3.2** (Satisfaction relation)**.** *Let $(C, s)$ be a pointed classifier model with*

*$C = (S, f)$ and $s \in S$. Then:*

$$\begin{aligned}
(C, s) \models p &\iff p \in s, \\
(C, s) \models \mathsf{t}(c) &\iff f(s) = c, \\
(C, s) \models \neg\varphi &\iff (C, s) \not\models \varphi, \\
(C, s) \models \varphi \wedge \psi &\iff (C, s) \models \varphi \textit{ and } (C, s) \models \psi, \\
(C, s) \models [X]\varphi &\iff \forall s' \in S, \textit{ if } (s \cap X) = (s' \cap X) \textit{then } (C, s') \models \varphi.
\end{aligned}$$

We can think of a pointed model $(C, s)$ as a pair $(s, c)$ of $f$ with $f(s) = c$. Thus, $c$ is the output of the input instance $s$ according to $f$. The condition $(s \cap X) = (s' \cap X)$, which induces an equivalence relation modulo $X$, indeed stipulates that $s$ and $s'$ are indistinguishable regarding the atoms (the features) in $X$. The formula $[X]\varphi$ is true at a state $s$ if $\varphi$ is true at all states that are modulo-$X$ equivalent to state $s$. It has the *selectis paribus* (SP) (selected things being equal) interpretation "features in $X$ being equal, necessarily $\varphi$ holds (under possible perturbation on the other features)". When $Atm_0$ is finite, $[Atm_0 \setminus X]\varphi$ has the standard *ceteris paribus* (CP) interpretation "features other than $X$ being equal, necessarily $\varphi$ holds (under possible perturbation of the features in $X$)".[8] When $X = \emptyset$, $[\emptyset]$ coincides with the S5 universal modality since every state is modulo-$\emptyset$ equivalent to all states. Hence instead of $[\emptyset]$ we often write $\Box$.

A formula $\varphi$ of $\mathcal{L}(Atm)$ is said to be satisfiable relative to the class **CM** if there exists a pointed classifier model $(C, s)$ with $C \in \mathbf{CM}$ such that $(C, s) \models \varphi$. It is said to be valid relative to **CM**, noted $\models_{\mathbf{CM}} \varphi$, if $\neg\varphi$ is not satisfiable relative to **CM**. Moreover, we say that that $\varphi$ is valid in the classifier model $C = (S, f)$, noted $C \models \varphi$, if $(C, s) \models \varphi$ for every $s \in S$.

It is worth noting that every modality $[X]$ can be defined by means of the universal modality $\Box$. To show this, let us introduce the following abbreviation for every $Y \subseteq X \subseteq^{\text{fin}} Atm_0$:

$$\mathsf{cn}_{Y,X} =_{def} \bigwedge_{p \in Y} p \wedge \bigwedge_{p \in X \setminus Y} \neg p.$$

$\mathsf{cn}_{Y,X}$ can be seen as the syntactic expression of a valuation on $X$, and therefore represents a set of states in a classifier model satisfying the valuation. We have the following validity for the class **CM**:

$$\models_{\mathbf{CM}} [X]\varphi \leftrightarrow \Big( \bigwedge_{Y \subseteq X} (\mathsf{cn}_{Y,X} \rightarrow \Box(\mathsf{cn}_{Y,X} \rightarrow \varphi)) \Big).$$

It means that $[X]\varphi$ is true at state $s$, if and only if, for whatever $Y \subseteq X$, if $s \cap X = Y$ then for any state $s'$ such that $s' \cap X = Y$, $\varphi$ is true at $s'$.

Let us close this section by formally introducing our running example.

[8] We thank Giovanni Sartor for drawing the distinction between CP and SP.

**Example 3.2** (Applicant Alice, formal)**.** *Let* $Atm = \{male, center, employed, owner\} \cup \{\mathtt{t}(1), \mathtt{t}(0)\}$*, where* 1 *and* 0 *stand for accepted and rejected respectively. Suppose* $C = (S, f)$ *is a CM such that* $S = 2^{Atm_0}$ *and*

$$C \models (\mathtt{t}(1) \leftrightarrow ((male \wedge employed) \vee (employed \wedge owner))).$$

*Consider the state* $s = \{center, employed\}$*. Then,* $s$ *stands for the instance Alice and* $f$ *for the classifier in Example 3.1 such that* $f(s) = 0$*.*

Now Alice is asking for explanations of the decision/classification, e.g., 1) which of her features (necessarily) lead to the current decision, 2) changing which features would make a difference, 3) perhaps most importantly, whether the decision for her is biased. In Section 3.5 we will show how to use the language $\mathcal{L}(Atm)$ and its semantics to answer these questions.

### 3.2.2 Discussion

In this subsection we discuss in more detail some subtleties of classifier models in relation with the modal language $\mathcal{L}(Atm)$ which is interpreted over them.

**$X$-Completeness** In the definition of classifier model (Definition 3.1) given above, we stipulated that the set of states $S$ does not necessarily include all possible input instances of a classifier. More generally, according to our definition, a classifier model could be incomplete with respect to a set of atoms $X$ from $Atm_0$, that is, there could be a truth assignment for the atoms in $X$ which is not represented in the model. Incompleteness of a classifier model is justified by the fact that in certain domains of application hard constraints exist which prevent for some input instance to occur. For example, a hard constraint may impose that a male cannot be pregnant (i.e., all states in which atoms *male* and *pregnant* are true should be excluded from the model).

However, it is interesting to see how completeness of a classifier with respect to a finite set of features can be represented in our semantics. This is what the following definition specifies.

**Definition 3.3** ($X$-completeness)**.** *Let* $C = (S, f)$ *be a classifier model and* $X \subseteq^{\text{fin}} Atm_0$*. Then,* $C$ *is said to be* $X$*-complete, if* $\forall X' \subseteq X, \exists s \in S$ *such that* $s \cap X = X'$*.*

In plain words, the definition means that any truth assignment for the atoms in $X$ is represented by some state of the model. As the following proposition indicates, the class of $X$-complete CMs can be syntactically represented. The proof is straightforward and omitted.

**Proposition 3.1.** *Let* $C = (S, f)$ *be a CM and* $X \subseteq^{\text{fin}} Atm_0$*.* $C$ *is* $X$*-complete if and only if* $\forall s \in S$*, we have* $(C, s) \models \mathsf{Comp}(X)$*, with*

$$\mathsf{Comp}(X) =_{def} \bigwedge_{X' \subseteq X} \Diamond \mathsf{cn}_{X',X}.$$

**$X$-Definiteness** In certain situations, there might be a portion of the feature space which is irrelevant for the classifier's decision. For example, in the Alice's example the fact of renting an apartment in the city center (the feature *center*) plays no role in the classification. In this case, we say that the classifier is definite with respect to the subset of features $\{male, employed, owner\}$.

More generally, a classifier is said to be definite with respect to a set of features $X$ if its decision is only determined by the variables in $X$, that is to say, the variables in the complementary set $Atm_0 \setminus X$ play no role in the classifier's decision. In other words, the classifier is said to be $X$-definite if its decision is independent of the variables in $Atm_0 \setminus X$.[9]

The following definition introduces the concept of $X$-definiteness formally.

**Definition 3.4** ($X$-definiteness)**.** *Let $C = (S, f)$ be a classifier model and $X \subseteq^{\text{fin}} Atm_0$. Then, $C$ is said to be $X$-definite, if $\forall s, s' \in S$, if $s \cap X = s' \cap X$ then $f(s) = f(s')$.*

$X$-definiteness is tightly related to the notion of dependence studied in (propositional) dependence logic [Yang & Väänänen 2016]. The latter focuses on so-called dependence atoms of the form $=(X, q)$ where $q$ is a propositional variable and $X$ is a finite set of propositional variables. The latter expresses the fact that the truth value of the propositional variable $q$ only depends on the truth values of the propositional variables in $X$. It turns out that dependence atoms can be expressed in our *ceteris paribus* modal language $\mathcal{L}(Atm)$ as abbreviations:

$$=(X, q) =_{def} \Box\big((q \to [X]q) \wedge (\neg q \to [X]\neg q)\big).$$

Interestingly, the notion of $X$-definiteness is expressible in our modal language by means of the dependence atoms. This is what the following proposition indicates.

**Proposition 3.2.** *Let $C = (S, f)$ be a CM and $X \subseteq^{\text{fin}} Atm_0$. $C$ is $X$-definite if and only if $\forall s \in S, (C, s) \models \mathsf{Defin}(X)$ with*

$$\mathsf{Defin}(X) =_{def} \bigwedge_{c \in Val} =(X, \mathsf{t}(c)).$$

We conclude this section by mentioning some remarkable properties of $X$-definiteness. The first fact to be noticed is that $X$-definiteness is upward closed.

**Fact 3.1.** *For every $C \in \mathbf{CM}$ and $X \subseteq Y \subseteq^{\text{fin}} Atm_0$, if $C$ is $X$-definite then $C$ is $Y$-definite too.*

Secondly, $X$-definiteness for some $X \subseteq^{\text{fin}} Atm_0$ is guaranteed in the finite-variable case.

**Fact 3.2.** *For every $C \in \mathbf{CM}$, if $Atm_0$ is finite then $C$ is $Atm_0$-definite.*

[9] Thus the relation between $X$-definiteness and essential variables in partial Boolean functions (as defined in 2.16, which drops the assumption that the function is total) is not hard to see: if $p \in Atm_0 \setminus X$ then $p$ is *inessential*.

This does not hold in the infinite case.

**Fact 3.3.** *If $Atm_0$ is countably infinite and $|Val| > 1$ then there exists $C \in \mathbf{CM}$ such that, for all $X \subseteq^{\text{fin}} Atm_0$, $C$ is not $X$-definite.*

The previous fact is witnessed by any CM $C = (S, f)$ such that

- $S = 2^{Atm_0}$,
- $f(Atm_0) = 1$,
- $\forall s \in S$, if $|Atm_0 \triangle s| = 1$ then $f(s) = 0$,

where $Dec = \{0, 1\}$ and $\triangle$ denotes symmetric difference, viz., $s \triangle s' = (s \setminus s') \cup (s' \setminus s)$. It is easy to show that a CM so defined is not $X$-definite for any $X \subseteq^{\text{fin}} Atm_0$.

## 3.3 Axiomatization and Complexity

In this section, we provide axiomatics for our logical setting. We distinguish the finite-variable from the infinite-variable case. We moreover prove complexity results for satisfiability checking for both cases. But before, we will first introduce an alternative Kripke semantics for the interpretation of the language $\mathcal{L}(Atm)$. It will allow us to use the standard canonical model technique for proving completeness. Indeed, this technique cannot be directly applied to CMs in the infinite-variable case since our modal language is not expressive enough to capture the "functionality" property of CMs when $Atm_0$ is infinite. An alternative proof applying the canonical model method directly to CMs in the finite-variable case is given in Section A.

### 3.3.1 Alternative Kripke Semantics

In our alternative semantics the concept of classifier model is replaced by the following concept of decision model. It is a multi-relational Kripke structure with one accessibility relation per finite set of atoms *plus* a number of constraints over the accessibility relations and the valuation function for the atomic propositions.

**Definition 3.5** (Decision model)**.** *A decision model (DM) is a tuple $M = (W, (\equiv_X)_{X \subseteq^{\text{fin}} Atm_0}, V)$ such that $W$ is a set of possible worlds, $V : W \longrightarrow 2^{Atm}$ is a valuation function for atomic formulas, and $\forall w, v \in W, c, c' \in Val$ the following constraints are satisfied:*

**(C1)** *$w \equiv_X v$ iff $V_X(w) = V_X(v)$,*

**(C2)** *$V_{Dec}(w) \neq \emptyset$,*

**(C3)** *if $\mathtt{t}(c), \mathtt{t}(c') \in V(w)$ then $c = c'$,*

**(C4)** *if $V_{Atm_0}(w) = V_{Atm_0}(v)$ then $V_{Dec}(w) = V_{Dec}(v)$;*

*where $V_X(w)$ abbreviates $V(w) \cap X$. The class of DMs is noted* **DM**.

A DM $(W, (\equiv_X)_{X \subseteq^{\text{fin}} Atm_0}, V)$ is called finite if $W$ is finite. The class of finite-DM is noted **finite-DM**.

The interpretation of formulas in $\mathcal{L}(Atm)$ relative to a pointed DM goes as follows.

**Definition 3.6** (Satisfaction relation)**.** *Let $(W, (\equiv_X)_{X \subseteq^{\text{fin}} Atm_0}, V)$ be a DM and let $w \in W$. Then,*

$$\begin{aligned}
(M, w) \models p &\iff p \in V(w), \\
(M, w) \models \mathsf{t}(c) &\iff \mathsf{t}(c) \in V(w), \\
(M, w) \models \neg\varphi &\iff (M, w) \not\models \varphi, \\
(M, w) \models \varphi \wedge \psi &\iff (M, w) \models \varphi \text{ and } (M, w) \models \psi, \\
(M, w) \models [X]\varphi &\iff \forall v \in W, \text{ if } w \equiv_X v \text{ then } v \models \varphi.
\end{aligned}$$

Validity and satisfiability of formulas in $\mathcal{L}(Atm)$ relative to class **DM** (resp. **finite-DM**) is defined in the usual way.

The following theorem appears obvious, since it only has to do with the matter whether the decision function (classifier) $f$ is given as a constituent of the model or induced from the model. Notice that it holds regardless of $Atm_0$ being finite or countably infinite.

**Theorem 3.1.** *Let $\varphi \in \mathcal{L}(Atm)$. Then, $\varphi$ is satisfiable relative to the class* **CM** *if and only if it is satisfiable relative to the class* **DM***.*

### 3.3.2 Axiomatization: Finite-Variable Case

In this section we provide a sound and complete axiomatics for the language $\mathcal{L}(Atm)$ relative to the formal semantics defined above under the assumption that the set of atomic propositions $Atm_0$ is finite.

**Definition 3.7** (Logic BCL)**.** *We define* BCL *(Binary-input Classifier Logic) to be the extension of classical propositional logic given by the following axioms and rule of inference:*

$$\big(\Box\varphi \wedge \Box(\varphi \to \psi)\big) \to \Box\psi \qquad (\mathbf{K}_\Box)$$

$$\Box\varphi \to \varphi \qquad (\mathbf{T}_\Box)$$

$$\Box\varphi \to \Box\Box\varphi \qquad (\mathbf{4}_\Box)$$

$$\varphi \to \Box\Box\varphi \qquad (\mathbf{B}_\Box)$$

$$[X]\varphi \leftrightarrow \bigwedge_{Y \subseteq X} \big(\mathsf{cn}_{Y,X} \to \Box(\mathsf{cn}_{Y,X} \to \varphi)\big) \qquad (\mathbf{Red}_{[X]})$$

$$\bigvee_{c \in Val} \mathsf{t}(c) \qquad (\mathbf{AtLeast})$$

$$\mathsf{t}(c) \to \neg\mathsf{t}(c') \text{ if } c \neq c' \qquad (\mathbf{AtMost})$$

$$\bigwedge_{Y \subseteq Atm_0} \Big(\big(\mathsf{cn}_{Y,Atm_0} \wedge \mathsf{t}(c)\big) \to \Box\big(\mathsf{cn}_{Y,Atm_0} \to \mathsf{t}(c)\big)\Big) \qquad (\mathbf{Funct})$$

$$\frac{\varphi}{\Box\varphi} \qquad (\mathbf{Nec}_\Box)$$

As the semantics indicates, $\Box$ is an S5 style modal operator; $\mathbf{Red}_{[X]}$ reduces any $[X]$ to $\Box$. **AtLeast**, **AtMost**, **Funct** represent the classification function syntactically such that every expression $\mathsf{cn}_{Y,Atm_0}$ maps to some unique $\mathsf{t}(c)$.

A decision model can contain two copies of the same input instance, while a classifier model cannot. Thus, Theorem 3.1 above shows that our modal language is not powerful enough to capture this difference between CMs and DMs. Axiom **Funct** intervenes in the finite-variable case to guarantee that two copies of the same input instance (that may exist in a DM) have the same output value. The expression $\mathsf{cn}_{Y,Atm_0}$ used in the axiom is an instance of the abbreviation we defined in Section 3.2.1. It represents a specific input instance. Notice that this abbreviation is only legal when $Atm_0$ is finite. Otherwise it would be the abbreviation of an infinite conjunction which is not a well-formed formula in our language.

The proof of the following theorem is entirely standard and based on a canonical model argument.

**Theorem 3.2.** *Let $Atm_0$ be finite. Then, the logic* BCL *is sound and complete relative to the class* **DM**.

The main result of this subsection is now a corollary of Theorems 3.1 and 3.2.

**Corollary 3.1.** *Let $Atm_0$ be finite. Then, the logic* BCL *is sound and complete relative to the class* **CM**.

### 3.3.3 Axiomatization: Infinite-Variable Case

In Section 3.3.2, we have assumed that the set of atomic propositions $Atm_0$ representing input variables is finite. In this section, we assume (countably) infinite variables and prove completeness of the resulting logic.

An essential feature of the logic BCL is the "functionality" Axiom **Funct**. Such an axiom cannot be represented in a finitary way when assuming that the set $Atm_0$ is countably infinite. For this reason, it has to be dismissed and the axiomatics becomes weaker.

**Definition 3.8** (Logic WBCL)**.** *The logic* WBCL *(Weak* BCL*) is defined by all principles of logic* BCL *given in Definition 3.7 except Axiom* ***Funct****.*

In order to obtain the completeness of WBCL relative to the class **CM**, besides decision models (DMs), we need additionally quasi-decision models (QDMs).

**Definition 3.9** (Quasi-DM)**.** *A quasi-DM is a tuple* $M = (W, (\equiv_X)_{X \subseteq^{\text{fin}} Atm_0}, V)$ *where* $W$, $(\equiv_X)_{X \subseteq^{\text{fin}} Atm_0}$ *and* $V$ *are defined as in Definition 3.5 and which satisfies all constraints of Definition 3.5 except* **C4***. The class of quasi-DMs is noted* ***QDM****.*

A quasi-DM $(W, (\equiv_X)_{X \subseteq^{\text{fin}} Atm_0}, V)$ is said to be finite if $W$ is finite. The class of finite quasi-DMs is noted **finite-QDM**.

Semantic interpretation of formulas in $\mathcal{L}(Atm)$ relative to quasi-DMs is analogous to semantic interpretation relative to DMs given in Definition 3.6. Moreover, validity and satisfiability of formulas in $\mathcal{L}(Atm)$ relative to class **QDM** (resp. **finite-QDM**) is again defined in the usual way.

We are going to show the equivalence between **QDM** and **CM** step by step. The following theorem is proven by filtration.

**Theorem 3.3.** *Let* $Atm_0$ *be countably infinite and* $\varphi \in \mathcal{L}(Atm)$*. Then,* $\varphi$ *is satisfiable relative to the class* ***QDM*** *if and only if* $\varphi$ *is satisfiable relative to the class* ***finite-QDM****.*

Then, let us establish the crucial fact that, in the infinite-variable case, the language $\mathcal{L}(Atm)$ cannot distinguish finite-DMs from finite-QDMs. We are going to prove that any formula $\varphi$ satisfiable in a finite-QDM $M$ is also satisfiable in some finite-DM $M'$. Since the only condition to worry is **C4**, we just need to transform the valuation function of $M$ to guarantee that **C4** holds while still satisfying $\varphi$.

**Theorem 3.4.** *Let* $Atm_0$ *be countably infinite and* $\varphi \in \mathcal{L}(Atm)$*. Then,* $\varphi$ *is satisfiable relative to the class* ***finite-QDM*** *if and only if* $\varphi$ *is satisfiable relative to the class* ***finite-DM****.*

Recall Theorem 3.1 shows that $\mathcal{L}(Atm)$ can not distinguish between CMs and DMs regardless of $Atm_0$ being finite or infinite. Thus, we obtain the desired equivalence between model classes **QDM** and **CM** in the infinite-variable case, as a corollary of Theorems 3.1, 3.3 and 3.4. This fact is highlighted by Figure 3.1. More generally, Figure 3.1 shows that when $Atm_0$ is countably infinite the five semantics for the modal language $\mathcal{L}(Atm)$ are all equivalent, since from every node in the graph we can reach all other nodes.

**Theorem 3.5.** *Let* $Atm_0$ *be countably infinite and* $\varphi \in \mathcal{L}(Atm)$*. Then,* $\varphi$ *is satisfiable relative to the class* **QDM** *if and only if* $\varphi$ *is satisfiable relative to the class* **CM***.*

As a consequence, we are in position of proving that the logic WBCL is also sound and complete for the corresponding classifier model semantics, under the infinite-variable assumption. The only missing block is the following completeness theorem. The proof is similar to the proof of Theorem 3.2 (with the only difference that the canonical QDM does not need to satisfy **C4**), and omitted.

**Theorem 3.6.** *Let $Atm_0$ be countably infinite. Then, the logic* WBCL *is sound and complete relative to the class* **QDM**.

The main result of this subsection turns out to be a direct corollary of Theorems 3.5 and 3.6.

**Corollary 3.2.** *Let $Atm_0$ be countably infinite. Then, the logic* WBCL *is sound and complete relative to the class* **CM**.

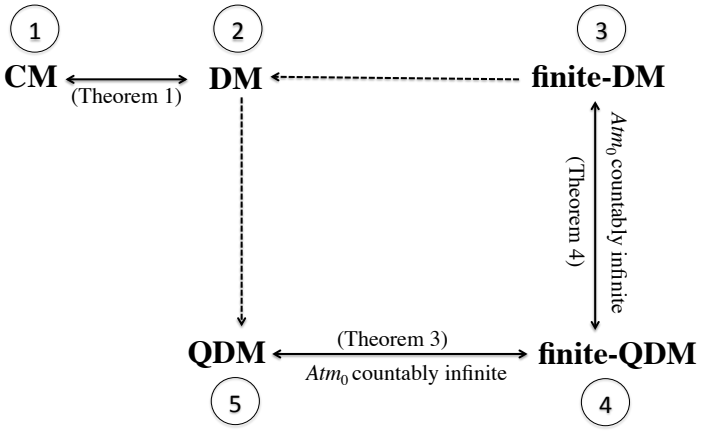



Figure 3.1: Relations between semantics for the modal language $\mathcal{L}(Atm)$. An arrow means that satisfiability relative to the first class of structures implies satisfiability relative to the second class of structures. Full arrows correspond to the results stated in Theorems 3.1, 3.3 and 3.4. Dotted arrows denote relations that follow straightforwardly given the inclusion between classes of structures. The bidirectional arrows connecting node 3 with node 4 and node 4 with node 5 only apply to the infinite-variable case.

### 3.3.4 Complexity Results

Let us now move from axiomatics to complexity issues. Our first result is about the complexity of checking satisfiability for formulas in $\mathcal{L}(Atm)$ relative to the class **CM** when $Atm_0$ is finite and fixed. It is in line with the satisfiability checking problem of the modal logic S5 which is known to be polynomial in the finite-variable case [Halpern 1995].

**Theorem 3.7.** *Let $Atm_0$ be finite and fixed. Then, checking satisfiability of formulas in $\mathcal{L}(Atm)$ relative to* **CM** *can be done in polynomial time.*

As the following theorem indicates, the satisfiability checking problem becomes intractable when dropping the finite-variable assumption.

**Theorem 3.8.** *Let $Atm_0$ be countably infinite. Then, checking satisfiability of formulas in $\mathcal{L}(Atm)$ relative to* **CM** *is NEXPTIME-complete.*

Let us consider the following fragment $\mathcal{L}^{\{\Box\}}(Atm)$ of the language $\mathcal{L}(Atm)$ where only the universal modality $\Box$ is allowed:

$$\varphi \quad ::= \quad p \mid \mathtt{t}(c) \mid \neg\varphi \mid \varphi \wedge \varphi \mid \Box\varphi.$$

Clearly, satisfiability checking for formulas in $\mathcal{L}^{\{\Box\}}(Atm)$ remains polynomial when there are only finitely many primitive propositions. As the following theorem indicates, complexity decreases from NEXPTIME to NP when restricting to the fragment $\mathcal{L}^{\{\Box\}}(Atm)$ under the infinite-variable assumption.

**Theorem 3.9.** *Let $Atm_0$ be countably infinite. Checking satisfiability of formulas in $\mathcal{L}^{\{\Box\}}(Atm)$ relative to* **CM** *is NP-complete.*

The complexity results of this section are summarized in Table 3.1.

| | **Fixed finite variables** | **Infinite variables** |
|---|---|---|
| **Fragment** $\mathcal{L}^{\{\Box\}}(Atm)$ | Polynomial | NP-complete |
| **Full language** $\mathcal{L}(Atm)$ | Polynomial | NEXPTIME-complete |

Table 3.1: Summary of complexity results

## 3.4 Counterfactual Conditional

In this section we investigate a simple notion of counterfactual conditional for binary classifiers, inspired from Lewis' notion [Lewis 1973]. In Section 3.5, we will elucidate its connection with the notion of explanation.

We start our analysis by defining the following notion of similarity between states in a classifier model relative to a finite set of features $X$.

**Definition 3.10** (Similarity between states)**.** *Let $C = (S, f)$ be a classifier model, $s, s' \in S$ and $X \subseteq^{\text{fin}} Atm_0$. The similarity between $s$ and $s'$ in $S$ relative to the set of features $X$, noted $sim_C(s,s',X)$, is defined as follows:*

$$sim_C(s,s',X) = |\{p \in X : (C,s) \models p \text{ iff } (C,s') \models p\}|.$$

A dual notion of distance between worlds can defined from the previous notion of similarity:

$$dist_C(s,s',X) = |X| - sim_C(s,s',X).$$

This notion of distance is in accordance with [Dalal 1988] in knowledge revision.[10]

The following definition introduces the notion of counterfactual conditional as an abbreviation. It is a form of relativized conditional, i.e., a conditional with respect to a finite set of features.[11]

**Definition 3.11** (Counterfactual conditional)**.** *We write $\varphi \Rightarrow_X \psi$ to mean that "if $\varphi$ were true then $\psi$ would be true, relative to the set of features $X$" and define it as follows:*

$$\varphi \Rightarrow_X \psi =_{def} \bigwedge_{0\leq k\leq |X|} (\mathsf{maxSim}(\varphi,X,k) \rightarrow \bigwedge_{Y\subseteq X:|Y|=k} [Y](\varphi \rightarrow \psi)),$$

*with*

$$\mathsf{maxSim}(\varphi,X,k) =_{def} \bigvee_{Y\subseteq X:|Y|=k} \langle Y\rangle\varphi \wedge \bigwedge_{Y\subseteq X:k<|Y|} [Y]\neg\varphi.$$

As the following proposition highlights, the previous definition of counterfactual conditional is in line with Lewis' view: the conditional holds if all closest worlds to the actual world in which the antecedent is true satisfy the consequent of the conditional.

**Proposition 3.3.** *Let $C = (S, f)$ be a classifier model, $s \in S$ and $X \subseteq^{\text{fin}} Atm_0$. Then, $(C, s) \models \varphi \Rightarrow_X \psi$ if and only if $closest_C(s,\varphi,X) \subseteq ||\psi||_C$, where*

$$closest_C(s,\varphi,X) = \arg\max s' \in ||\varphi||_C \; sim_C(s,s',X),$$

*and for every $\varphi \in \mathcal{L}(Atm)$:*

$$||\varphi||_C = \{s \in S : (C, s) \models \varphi\}.$$

For notational convenience, we simply write $\varphi \Rightarrow \psi$ instead of $\varphi \Rightarrow_{Atm_0} \psi$, when $Atm_0$ is finite. Formula $\varphi \Rightarrow \psi$ captures the standard notion of conditional of conditional logic. One can show that $\Rightarrow$ satisfies all semantic conditions of Lewis' logic VC.[12] However, when $Atm_0$ is infinite, $\varphi \Rightarrow \psi$ is not a well-formed formula since it ranges over an infinite set of atoms. In that case $\varphi \Rightarrow_X \psi$ has to be always indexed by some finite $X$.

[10] There are other options besides measuring distance by cardinality, e.g., distance in sense of subset relation as [Borgida 1985]. We will consider them in further research.

[11] A similar approach to conditional is presented in [Girard & Triplett 2016]. They also refine Lewis' semantics for counterfactuals by selecting the closest worlds according to not only the actual world and antecedent, but also a set of formulas noted $\Gamma$. The main technical difference is that they allow any counterfactual-free formula as a member of $\Gamma$, while in our setting $X$ only contains atomic formulas.

[12] A remarkable fact is that not all $\Rightarrow_X$ satisfy the *strong centering* condition, which says that the actual world is the only closest world when the antecedent is already true there. To see it, consider a toy classifier model $(C, s)$ such that $S = \{s, s', s'', s'''\}$ with $s = \{p, q\}$, $s' = \{p\}$, $s'' = \{q\}$, $s''' = \emptyset$. We have $closest_C(s,p,\{p\}) = \{s, s'\}$, rather than $closest_C(s,p,\{p\}) = \{s\}$.

The interesting aspect of the previous notion of counterfactual conditional is that it can be used to represent a binary classifier's approximate decision for a given instance. Let us suppose the set of decision values *Val* includes a special symbol ? meaning that the classifier has no sufficient information enabling it to classify an instance in a precise way. More compactly, ? *is interpreted as* that the classifier abstains from making a precise decision. In this situation, the classifier can try to make an approximate decision: it considers the closest instances to the actual instance for which it has sufficient information to make a decision and checks whether the decision is uniform among all such instances. In other words, $c$ is the classifier's approximate classification of (or decision for) the actual instance relative to the set of features $X$, noted $\mathsf{apprDec}(X,c)$, if and only if "if a precise decision were made relative to the set of features $X$, then this decision would be $c$". Formally:

$$\mathsf{apprDec}(X,c) =_{def} \Big(\bigvee_{c' \in Val: c' \neq ?} \mathsf{t}(c')\Big) \Rightarrow_X \mathsf{t}(c).$$

The following proposition provides two interesting validities.

**Proposition 3.4.** *Let $Atm_0$ be finite, $c, c' \in Val \setminus \{?\}$. Then,*

$$\models_{\mathbf{CM}} \mathsf{apprDec}(X,c) \rightarrow \neg \mathsf{apprDec}(X,c') \text{ if } c \neq c',$$
$$\models_{\mathbf{CM}} \mathsf{t}(c) \rightarrow \mathsf{apprDec}(Atm_0, c).$$

According to the first validity, a classifier cannot make two different approximate decisions relative to a fixed set of features $X$.

According to the second validity, if the classifier is able to make a precise decision for a given instance, then its approximate decision coincides with it. This second validity works since the actual state/instance is the only closest state/instance to itself. Therefore, it the actual state/instance has a precise classification $c$, all its closest states/instances also have it.

It is worth noting that the following formula is not valid relative to the class **CM**:

$$\bigvee_{c \in Val \setminus \{?\}} \mathsf{apprDec}(X,c).$$

This means that a classifier may be unable to approximately classify the actual instance. The reason is that there could be different closest states to the actual one with different classifications.

## 3.5 Explanations and Biases

In this section, we are going to formalize some existing notions of explanation of classifiers in our logic, and deepen the current study from a (finitely) Boolean setting to a multi-valued output, partial domain and possibly infinite-variable setting. For this purpose it is necessary to introduce the following notations.

Let $\lambda$ denote a conjunction of finitely many literals, where a literal is an atom $p$ or its negation $\neg p$. We write $\lambda \subseteq \lambda'$, call $\lambda$ a part (subset) of $\lambda'$, if all literals in $\lambda$ also occur in $\lambda'$; and $\lambda \subset \lambda'$ if $\lambda \subseteq \lambda'$ but not $\lambda' \subseteq \lambda$. By convention $\top$ is a term of zero conjuncts. In particular, suppose $\lambda$ is $\mathsf{cn}_{X,Y}$ for some $X \subseteq Y \subseteq^{\text{fin}} Atm_0$, then $\overline{\lambda}$ will denote the conjunction resulting from "flipping" (or "perturbing") all literals of $\lambda$, i.e., $\mathsf{cn}_{Y \setminus X,Y}$.

In the glossary of Boolean classifiers, $\lambda$ is called a *term* or *property* (of an instance). The set of terms is noted *Term*. We use $Term(X)$ to denote all terms whose atoms are in $X$. Additionally, to define the notion of bias we distinguish the set of protected features $\mathsf{PF} \subseteq Atm_0$, like 'gender' and 'race', and the set of non-protected features $\mathsf{NF} = Atm_0 \setminus \mathsf{PF}$.

Notice that in this section the cardinality of $Atm_0$ matters. Notions and results in Section 3.5.1 (without special instruction) apply to both $Atm_0$ finite and $Atm_0$ countably infinite. On the contrary, in Sections 3.5.2 and 3.5.3, we restrict to the case $Atm_0$ finite, which is due to the use of formulas $[Atm_0 \setminus X]\varphi, [\mathsf{NF}]\varphi$ and $[\mathsf{PF}]\varphi$ there. We clarify it here instead of clarifying it below repeatedly.

### 3.5.1 Prime Implicant and Abductive Explanation

We are in position to formalize the notion of *prime implicant*, which plays a fundamental role in the theory of Boolean functions since [Quine 1955].

**Definition 3.12** (Prime implicant ($\mathsf{PImp}$))**.** *We write* $\mathsf{PImp}(\lambda, c)$ *to mean that* $\lambda$ *is a* prime implicant *for* $c$ *and define it as follows:*

$$\mathsf{PImp}(\lambda, c) =_{def} \Box\Big(\lambda \rightarrow (\mathsf{t}(c) \wedge \bigwedge_{p \in Atm(\lambda)} \langle Atm(\lambda) \setminus \{p\}\rangle \neg \mathsf{t}(c))\Big).$$

It is a proper extension of the definition of prime implicant in the Boolean setting since it is a term $\lambda$ such that 1) it necessarily implies the actual classification (why it is called an *implicant*); 2) any of its proper subsets fails to necessarily imply the actual classification (why it is called *prime*). Notice that being a prime implicant is a global property of the classifier, though we formalize it by means of a pointed model. The syntactic abbreviation for prime implicant can be better understood by observing that for a given CM $C = (S, f)$ and $s \in S$, we have:

$$\begin{aligned}(C, s) \models \mathsf{PImp}(\lambda, c) \text{ iff } &(i)\ \forall s' \in S,\ \text{if } (C, s') \models \lambda \text{ then } (C, s') \models \mathsf{t}(c);\ \text{and} \\ &(ii)\ \forall \lambda' \subset \lambda, \exists s' \in S \text{ such that } (C, s') \models \lambda' \wedge \neg \mathsf{t}(c).\end{aligned}$$

To explain the actual classification of a given input, some XAI researchers consider a prime implicant which is actually true. We use the terminology by [Ignatiev *et al.* 2019] and call it an abductive explanation (AXp).[13]

[13]There is a weaker version of the notion, called the weak AXp [Huang *et al.* 2022], which requires only an implicant rather than a prime implicant. We will not formally define it until Chapter 4 where it will be needed.

**Definition 3.13** (Abductive explanation ($\mathsf{AXp}$))**.** *We write* $\mathsf{AXp}(\lambda, c)$ *to mean that* $\lambda$ abductively explains *the decision* $c$ *and define it as follows:*

$$\mathsf{AXp}(\lambda, c) =_{def} \lambda \wedge \mathsf{PImp}(\lambda, c).$$

AXp is a local explanation, because $\lambda$ is not only a prime implicant for the classification, but also a property of the actual instance to be classified. AXp can be expanded to highlight its connection with the notion of variance/invariance.

**Proposition 3.5.** *Let* $\lambda \in \mathit{Term}$ *and* $c \in \mathit{Val}$*. Then, we have the following validity:*

$$\models_{\mathbf{CM}} \mathsf{AXp}(\lambda, c) \leftrightarrow (\lambda \wedge [\mathit{Atm}(\lambda)]\mathsf{t}(c) \wedge \bigwedge_{p \in \mathit{Atm}(\lambda)} \langle \mathit{Atm}(\lambda) \setminus \{p\}\rangle \neg \mathsf{t}(c)).$$

The formula $[\mathit{Atm}(\lambda)]\mathsf{t}(c)$ expresses the idea of invariance under intervention (perturbation): as long as the explanans variables are kept fixed, namely the variables in $\lambda$, any perturbation on the other variables does not change the explanandum, namely classification $c$.

Many names besides AXp are found in literature, e.g., *PI-explanation* [Shih *et al.* 2018] and *sufficient reason* [Darwiche & Hirth 2020]. In [Darwiche & Hirth 2020] it was proved that any decision has a sufficient reason in the Boolean setting. The result is not a surprise, for a Boolean function always has a prime implicant, since by definition the arity of a Boolean function is always finite. However, since we allow functions with infinitely many variables, AXps are not guaranteed to exist in general.

**Fact 3.4.** *Let* $\mathit{Atm}_0$ *be countably infinite and* $|\mathit{Val}| > 1$*. Then, there exists some* $C = (S, f)$*,* $s \in S$*, such that* $\exists c \in \mathit{Val}, \forall \lambda \in \mathit{Term}$*,* $(C, s) \models \neg \mathsf{AXp}(\lambda, c)$*.*

The statement can be proved by exhibiting the same infinite countermodel as in Fact 3.3 in Section 3.2.2. However, if a CM is $X$-definite for some $X \subseteq^{\text{fin}} \mathit{Atm}_0$, then every state has an AXp, even when the CM is infinite.

**Proposition 3.6.** *Let* $C = (S, f) \in \mathbf{CM}$ *and* $X \subseteq^{\text{fin}} \mathit{Atm}_0$*. If* $C$ *is* $X$*-definite then* $\forall s \in S, \exists \lambda \in \mathit{Term}$ *such that* $(C, s) \models \mathsf{AXp}(\lambda, f(s))$*.*

Lastly, let us continue with the Alice example.

**Example 3.3.** *Recall the state of Alice* $s = \{\mathit{center}, \mathit{employed}\}$*. We have* $(C, s) \models \mathsf{AXp}(\neg \mathit{male} \wedge \neg \mathit{owner}, 0)$*, namely that Alice's being female and not owning a property abductively explains the rejection of her application.*

### 3.5.2 Contrastive Explanation

AXp is a minimal part of the actual instance guaranteeing the current decision. A natural counterpart of AXp is contrastive explanation (CXp, named in [Ignatiev *et al.* 2020b]).

**Definition 3.14** (Contrastive explanation (CXp))**.** *We write* $\mathsf{CXp}(\lambda, c)$ *to mean that* $\lambda$ contrastively explains *the decision* $c$ *and define it as follows:*

$$\mathsf{CXp}(\lambda, c) =_{def} \lambda \wedge \langle Atm_0 \setminus Atm(\lambda)\rangle \neg \mathsf{t}(c) \wedge \bigwedge_{p \in Atm(\lambda)} [(Atm_0 \setminus Atm(\lambda)) \cup \{p\}]\mathsf{t}(c).$$

The definition says nothing but 1) $\lambda$ is part of the actual input instance; 2) if the values of all variables in $\lambda$ are changed while the values of the other variables are kept fixed, then the actual classification may change; 3) the classification will not change, if the variables outside $\lambda$ and at least one variable in $\lambda$ keep their actual values. The latter captures a form of necessity: when the values of the variables outside $\lambda$ are kept fixed, all variables in $\lambda$ should be *necessarily* perturbed to change the actual classification.

The syntactic abbreviation for contrastive explanation can be better understood by observing that for a given CM $C = (S, f)$ and $s \in S$, we have:

$$\begin{aligned}(C, s) \models \mathsf{CXp}(\lambda, c) \text{ iff } &(i)\ (C, s) \models \lambda;\\ &(ii)\ \exists s' \in S \text{ s.t. } s \triangle s' = Atm(\lambda) \text{ and } (C, s') \models \neg \mathsf{t}(c);\ \text{and}\\ &(iii)\ \forall s' \in S,\ \text{if } s \triangle s' \subset Atm(\lambda) \text{ then } (C, s') \models \mathsf{t}(c).\end{aligned}$$

CXp has a counterfactual flavor since it answers to question: would the classification differ from the actual one, if the values of all variables in the explanans were different? So, there seems to be a connection with the notion of counterfactual conditional we introduced in Section 3.4. Actually in XAI, many researchers consider contrastive explanation and counterfactual explanation either closely related [Verma *et al.* 2020] or even interchangeable [Sokol & Flach 2019]. The following proposition sheds light on this point.

**Proposition 3.7.** *Let* $\lambda$ *be a term and let* $l$ *be a literal. Then, we have the following two validities:*

$$\models_{\mathbf{CM}} \mathsf{CXp}(\lambda, c) \rightarrow \Big(\mathsf{t}(c) \wedge (\overline{\lambda} \Rightarrow \neg \mathsf{t}(c))\Big),$$
$$\models_{\mathbf{CM}} \mathsf{Comp}(Atm_0) \rightarrow \Big(\mathsf{CXp}(l, c) \leftrightarrow \Big(\mathsf{t}(c) \wedge (\neg l \Rightarrow \neg \mathsf{t}(c))\Big)\Big).$$

According to the first validity, in the general case contrastive explanation implies counterfactual explanation. According to the second validity, when the explanans is a literal (a single-conjunct term), contrastive explanation coincides with counterfactual explanation given $Atm_0$-completeness. Particularly, literal $l$ contrastively explains the decision $c$ if and only if (i) the actual decision is $c$ and (ii) if literal $l$ were perturbed, the decision would be different from $c$. In other words, in the "atomic" case under the completeness assumption, CXp is the same as counterfactual explanation.

Note that the right-to-left direction of the first validity does not necessarily

hold. To see this, it is sufficient to suppose that $Atm_0 = \{p, q\}$ and $Dec = \{0, 1\}$ and to consider the CM $(S, f)$ such that $S = 2^{Atm_0}$ with $f(\{p, q\}) = 0$ and $f(\{p\}) = f(\{q\}) = f(\emptyset) = 1$. It is easy to check that in the model so defined we have

$$(C, \{p, q\}) \models \mathsf{t}(0) \wedge (\overline{p \wedge q} \Rightarrow \neg \mathsf{t}(0)),$$

but at the same time,

$$(C, \{p, q\}) \models \neg \mathsf{CXp}(p \wedge q, 0).$$

The problem is that the model fails to satisfy the necessity condition of contrastive explanation: it is not necessary to perturb both literals in $p \wedge q$ to change the actual decision from 0 to 1, it is sufficient to perturb one of them. We can conclude that CXp is a special kind of counterfactual explanation with the additional requirement of necessity for the explanans.

**Example 3.4.** *In Alice's case, we have* $(C, s) \models \mathsf{CXp}(\neg male, 0) \wedge \mathsf{CXp}(\neg owner, 0)$. *This means that both Alice's being female and not owning property contrastively explain the rejection. Moreover, we have* $(C, s) \models (male \vee owner) \Rightarrow \mathsf{t}(1)$, *namely if Alice were a male or an owner (of an immobile property), then her application would have been accepted.*

Moreover, since the feature 'gender' is hard to change, owing a property is the (relatively) *actionable* explanation for Alice,[14] if she intends to comply with the classifier's decision. But surely Alice has another option, i.e., alleging the classifier as biased. As we will see in the next subsection, an application of CXp is to detect decision biases in a classifier.

### 3.5.3 Decision Bias

A primary goal of XAI is to detect and avoid biases. Bias is understood as making decision with respect to some protected features, e.g., 'race', 'gender' and 'age'.

There is a widely accepted notion of decision bias in the setting of Boolean functions which can be represented in our Example 3.2 (see [Darwiche & Hirth 2020, Ignatiev *et al.* 2020a]). Intuitively, the rejection for Alice is biased if there is another applicant, say Bob, who only differs from Alice on the protected feature 'gender', but gets accepted.

**Definition 3.15** (Decision bias)**.** *We write* $\mathsf{Bias}(c)$ *to mean that the decision* $c$ is biased *and define it as follows:*

$$\mathsf{Bias}(c) =_{def} \mathsf{t}(c) \wedge \langle \mathsf{NF} \rangle \neg \mathsf{t}(c).$$

The definition says that the decision $c$ is biased at a given state $s$, if (i) $f(s) = c$, and (ii) $\exists s' \in S$ such that $s \triangle s' \subseteq \mathsf{PF}$ and $f(s') \neq c$. The latter, in plain words,

[14] For the significance of actionability in XAI, see e.g. [Sokol & Flach 2019].

requires another instance $s'$, which only differs from $s$ on some protected features, but obtains a different classification.

As we stated, CXp can be used to detect decision biases. The following result makes the statement precise.

**Proposition 3.8.** *We have the following validity:*

$$\models_{\mathbf{CM}} \mathsf{Bias}(c) \leftrightarrow \bigvee_{Atm(\lambda)\subseteq \mathsf{PF}} \mathsf{CXp}(\lambda, c).$$

Let us end up the whole section by answering the last question regarding Alice raised at the end of Section 3.2.1.

**Example 3.5.** *Split $Atm_0$ in Example 3.2 into $\mathsf{PF} = \{male, center\}$ and $\mathsf{NF} = \{employed, owner\}$. We then have $(C, s) \models \mathsf{Bias}(0) \wedge \mathsf{CXp}(male, 0) \wedge (\neg male \Rightarrow \mathsf{t}(1))$. The decision for Alice is biased since 'gender' is the protected feature which contrastively explains the rejection, and if Alice were male, her application would have been accepted.*

## 3.6 Extensions

In this section, we briefly discuss two interesting extensions of our logical framework and analysis of binary classifiers. Their full development is left for future work.

### 3.6.1 Dynamic Extension

The first extension we want to discuss consists in adding to the language $\mathcal{L}(Atm)$ dynamic operators of the form $[c{:=}\varphi]$ with $c \in Val$, where $c{:=}\varphi$ is a kind of assignment in the sense of [Van Benthem *et al.* 2006, van Ditmarsch *et al.* 2005] and the formula $[c{:=}\varphi]\psi$ has to be read "$\psi$ holds after every decision is set to $c$ in context $\varphi$". The resulting language, noted $\mathcal{L}^{dyn}(Atm)$, is defined by the following grammar:

$$\varphi \quad ::= \quad p \mid \mathsf{t}(c) \mid \neg\varphi \mid \varphi \wedge \varphi \mid [X]\varphi \mid [c{:=}\varphi]\psi,$$

where $p$ ranges over $Atm_0$, $c$ ranges over $Val$, and $X \subseteq^{\text{fin}} Atm_0$. The interpretation of formula $[c{:=}\varphi]\psi$ relative to a pointed classifier model $(C, s)$ with $C = (S, f)$ goes as follows:

$$(C, s) \models [c{:=}\varphi]\psi \iff (C^{c:=\varphi}, s) \models \psi,$$

where $C^{c:=\varphi} = (S, f^{c:=\varphi})$ is the updated classifier model where, for every $s' \in S$:

$$f^{c:=\varphi}(s') = \begin{cases} c \text{ if } (C, s') \models \varphi, \\ f(s') \text{ otherwise.} \end{cases}$$

Intuitively, the operation $c := \varphi$ consists in globally classifying all instances satisfying $\varphi$ with value $c$.

Dynamic operators $[c{:=}\varphi]$ are useful for modeling a classifier's revision. Specifically, new knowledge can be injected into the classifier thereby leading to a change in its classification. For example, the classifier could learn that if an object is a furniture, has one or more legs and has a flat top, then it is a table. This is captured by the following assignment:

$$\mathtt{table}{:=}\, \mathit{objIsFurniture} \wedge \mathit{objHasLegs} \wedge \mathit{objHasFlatTop}.$$

An application of dynamic change is to model the training process of a classifier, together with counterfactual conditionals with "?" in Section 3.4. Suppose at the beginning we have a CM $C = (S, f)$ which is totally ignorant, i.e., $\forall s \in S, f(s) =?$. We then prepare to train the classifier. The training set consists of pairs $(s_1, x_1), (s_2, x_2) \dots (s_n, x_n)$ where $\forall i \in \{1, \dots, n\}$, $s_i \in S, x_i \in (\mathit{Val} \setminus \{?\})$ and $\forall j \in \{1, \dots, n\}$, $i \neq j$ implies $s_i \neq s_j$. We train the classifier by revising it with $[x_1 := \widehat{s}_1] \dots [x_n := \widehat{s}_n]$ one by one. Obviously the order does not matter here. In other words, we re-classify some states. With a bit abuse of notation, let $C^{train} = (S, f^{train})$ denote the model resulting from the series of revisions. We finish training by inducing the final model $C^{\dagger} = (S, f^{\dagger})$ from $C^{train}$, where $\forall s \in S, f^{\dagger}(s) = c$, if $(C^{train}, s) \models \mathsf{apprDec}(\mathit{Atm}_0{,}c)$, otherwise $f^{\dagger}(s) = f^{train}(s)$. This is an example of modeling a special case of the so-called *k-nearest neighbour (KNN) classification* in machine learning [Cunningham & Delany 2022], where the distance is measured by cardinality. If a new case/instance has to be classified, we see how the most similar cases to the new case were classified. If all of them ($k$ of them in the case of KNN) were classified using the same category, we put the new case into that category.

The logics BCL−DC and WBCL−DC (BCL and WBCL with Decision Change) extend the logic BCL and WBCL by the dynamic operators $[c{:=}\varphi]$. They are defined as follows.

**Definition 3.16** (Logics BCL−DC and WBCL−DC)**.** *We define* BCL−DC *(resp.* WBCL−DC*) to be the extension of* BCL *(resp.* WBCL*) of Definition 3.7 (resp. Definition 3.8) generated by the following reduction axioms for the dynamic operators* $[c{:=}\varphi]$*:*

$$\begin{aligned}
[c{:=}\varphi]\mathsf{t}(c) &\leftrightarrow (\varphi \vee \mathsf{t}(c)) \\
[c{:=}\varphi]\mathsf{t}(c') &\leftrightarrow (\neg\varphi \wedge \mathsf{t}(c')) \ \textit{ if } c \neq c' \\
[c{:=}\varphi]p &\leftrightarrow p \\
[c{:=}\varphi]\neg\psi &\leftrightarrow \neg[c{:=}\varphi]\psi \\
[c{:=}\varphi](\psi_1 \wedge \psi_2) &\leftrightarrow ([c{:=}\varphi]\psi_1 \wedge [c{:=}\varphi]\psi_2) \\
[c{:=}\varphi][X]\psi &\leftrightarrow [X][c{:=}\varphi]\psi
\end{aligned}$$

*and the following rule of inference:*

$$\frac{\varphi_1 \leftrightarrow \varphi_2}{\psi \leftrightarrow \psi[\varphi_1/\varphi_2]} \qquad (\mathbf{RE})$$

It is routine exercise to verify that the equivalences in Definition 3.16 are valid for the class **CM** and that the rule of replacement of equivalents (**RE**) preserves validity. The completeness of BCL−DC (resp. WBCL−DC) for this class of models under the finite-variable assumptions (resp. infinite-variable assumption) follows from Corollary 3.1 (resp. Corollary 3.2), in view of the fact that the reduction axioms and the rule of replacement of proved equivalents can be used to find, for any $\mathcal{L}^{dyn}$-formula, a provably equivalent $\mathcal{L}$-formula.

**Theorem 3.10.** *Let $Atm_0$ be finite. Then, the logic* BCL−DC *is sound and complete relative to the class* **CM**.

**Theorem 3.11.** *Let $Atm_0$ be countably infinite. Then, the logic* WBCL−DC *is sound and complete relative to the class* **CM**.

The following complexity results are consequences of Theorems 3.7 and 3.8 and the fact that via the reduction axioms in Definition 3.8 we can find a polynomial reduction of satisfiability checking for formulas in $\mathcal{L}^{dyn}$ to satisfiability checking for formulas in $\mathcal{L}$.

**Theorem 3.12.** *Let $Atm_0$ be finite and fixed. Then, checking satisfiability of formulas in $\mathcal{L}^{dyn}(Atm)$ relative to* **CM** *can be done in polynomial time.*

**Theorem 3.13.** *Let $Atm_0$ be countably infinite. Then, checking satisfiability of formulas in $\mathcal{L}^{dyn}(Atm)$ relative to* **CM** *is NEXPTIME-complete.*

### 3.6.2 Epistemic Extension

In the second extension we suppose that a classifier is an agent which has to classify what it perceives. The agent could have uncertainty about the actual instance to be classified since it cannot see all its input features.

In order to represent the agent's epistemic state and uncertainty, we introduce an epistemic modality of the form $\mathsf{K}$ which is used to represent what the agent knows in the light of what it sees. Similar notions of visibility-based knowledge can be found in [Charrier *et al.* 2016, Van Der Hoek *et al.* 2011, Herzig *et al.* 2015, van der Hoek *et al.* 2012].

The language for our epistemic extension is noted $\mathcal{L}^{epi}(Atm)$ and defined by the following grammar:

$$\varphi \quad ::= \quad p \mid \mathsf{t}(c) \mid \neg\varphi \mid \varphi \wedge \varphi \mid [X]\varphi \mid \mathsf{K}\varphi,$$

where $p$ ranges over $Atm_0$, $c$ ranges over $Val$, and $X \subseteq^{\text{fin}} Atm_0$.

In order to interpret the new modality $\mathsf{K}$, we have to enrich classifier models with an epistemic component.

**Definition 3.17** (Epistemic classifier model)**.** *An epistemic classifier model (ECM) is a tuple $E = (S, f, Obs)$ where $C = (S, f)$ is a classifier model and $Obs \subseteq Atm_0$ is the set of atomic propositions that are visible to the agent. The class of ECMs is noted* **ECM**.

Given an ECM $E = (S, f, Obs)$, we can define an epistemic indistinguishability relation which represents the agent's uncertainty about the actual input instance.

**Definition 3.18** (Epistemic indistinguishability relation)**.** *Let $E = (S, f, Obs)$ be an ECM. Then, $\sim$ is the binary relation on $S$ such that, for all $s, s' \in S$:*

$$s \sim s' \text{ if and only if } (s \cap Obs) = (s' \cap Obs).$$

Clearly, the relation $\sim$ so defined is an equivalence relation. According to the previous definition, the agent cannot distinguish between two states $s$ and $s'$, noted $s \sim s'$, if and only if the truth values of the visible variables are the same at $s$ and $s'$.

The interpretation for formulas in $\mathcal{L}^{epi}(Atm)$ extends the interpretation for formulas in $\mathcal{L}(Atm)$ given in Definition 3.2 by the following condition for the epistemic operator:

$$(E, s) \models \mathsf{K}\varphi \iff \forall s' \in S : \text{ if } s \sim s' \text{ then } (E, s') \models \varphi.$$

As the following theorem indicates, the complexity result of Section 3.3.2 for the finite-variable case generalizes to the epistemic extension.

**Theorem 3.14.** *Let $Atm_0$ be finite. Then, checking satisfiability of formulas in $\mathcal{L}^{epi}(Atm)$ relative to* **ECM** *can be done in polynomial time.*

In order to illustrate the intuition behind the epistemic modality $\mathsf{K}$ we go back to the example of the application for a loan to a bank.

**Example 3.6.** *Suppose the application is submitted through an online system which has to automatically decide whether it is acceptable or not. In his/her application, an applicant has to specify a value for each feature. Moreover, suppose the system receives an incomplete application: the applicant has only indicated that she is female, owns an apartment and lives in the city center, but she has forgotten to specify whether she has an employment or not. In this case, the value of the employement variable is not "visible" to the system. In formal terms, we extend the CM given in Example 3.2 by the visibility set $Obs = \{male, center, owner\}$ to obtain a ECM $E = (S, f, Obs)$. It is easy to check that the following holds:*

$$(E, \{center, employed, owner\}) \models \neg\mathsf{K}\ \mathsf{t}(0) \wedge \neg\mathsf{K}\ \mathsf{t}(1).$$

*This means that, on the basis of its partial knowledge of the applicant's identity, the system does not know what to decide.*

*However, the system knows that if turns out that the applicant is employed then its application should be accepted:*

$$(E, \{center, employed, owner\}) \models \mathsf{K}(employed \rightarrow \mathsf{t}(1)).$$

*Finally, the classifier knows that if turns out that the applicant is employed, then the fact that she is employed and that she owns a property will abductively explain the decision to accept her application:*

$$(E, \{center, employed, owner\}) \models \mathsf{K}(employed \rightarrow \mathsf{AXp}(employed \wedge owner, 1)).$$

## 3.7 Conclusion

We have introduced a modal language and a formal semantics that enable us to capture the *ceteris paribus* nature of binary classifiers. We have formalized in the language a variety of notions which are relevant for understanding a classifier's behavior including counterfactual conditional, abductive and contrastive explanation, bias. We have provided two extensions that support reasoning about classifier change and a classifier's uncertainty about the actual instance to be classified. We have also offered axiomatics and complexity results for our logical setting.

We believe that the complexity results presented here are exploitable in practice. We have shown that satisfiability checking in the basic setting and in its dynamic and epistemic extension is polynomial when finitely many variables are assumed. In the infinite-variable setting, it becomes NEXPTIME-complete and NP-complete when restricting to the language in which the only primitive modal operator is the universal modality $[\emptyset]$. In future work, we plan (i) to find a number of satisfiability preserving translations from our modal languages to the modal logic S5 and then from S5 to propositional logic using existing techniques [Caridroit *et al.* 2017], and (ii) to exploit SAT solvers for automated verification and generation of explanations and biases in binary classifiers.

Another direction of future research is the generalization of the epistemic extension given in Section 3.6.2 to the multi-agent case. The idea is to conceive classifiers as agents and to be able to represent both the agents' uncertainty about the instance to be classified and their knowledge and uncertainty about other agents' knowledge and uncertainty (i.e., higher-order knowledge and uncertainty). Similarly, we plan to investigate more in depth classifier dynamics we briefly discussed in Section 3.6.1. The idea is to see them as learning dynamics. Based on this idea, we plan to study the problem of finding a sequence of update operations guaranteeing that the classifier will be able to make approximate decisions for a given set of instances.

Finally, all classifiers we handle in this paper do not represent "black box" classifiers, in the sense that we have perfect knowledge of them, so that we can compute their explanations. However, black box classifiers are the most interesting ones to XAI. As mentioned, in Chapter 6 we will conceive a black box classifier as

an agent's uncertainty among many possible classifiers. All notions of explanation we defined in this chapter can be generalized to the black box setting. However, there are some important differences between the two settings. For instance, in a black box classifier AXp does not always exist, as we will showed in Chapter 6, which contradicts Proposition 3.6.

Chapter 4

# Application to Legal Case-based Reasoning

This chapter counts as an application of BCL in the last chapter. It offers a novel framework for factor-based models of legal case-based reasoning (CBR). The underlying intuition is simple: a case base is just a partial Boolean function (pBF), which is in turn representable by our classifier model.

Case bases in the legal theory of precedent usually have more intricate structures than mere pBFs largely due to the constraint of a fortiori reasoning. Horty has developed factor-based models of CBR in relation to the theory of precedent. Our aim is to associate case bases consistent in Horty's sense with a subclass of our classifier models. We will examine their relationship particularly with respect to monotone pBFs. Furthermore, by introducing the perspective of the classifier, we can analyze case bases using the notions of classifier explanation presented in the previous chapter.

**Contents**



## 4.1 Introduction

This chapter brings together two lines of research: factor-based models of case-based reasoning (CBR) and the logical specification of classifiers.

As discussed in the previous chapter, logical approaches to classifiers capture the connection between features and outcomes in classifier systems. They are well-suited for modeling and computing a large variety of explanations of a classifier's

decisions, e.g., prime implicants, abductive, contrastive and counterfactual explanations. They can thus contribute to provide controllability and explainability over automated decision-making (as required, e.g., by Art. 22 GDPR and by Art. 6 ECHR relative to judicial decisions).

Factor-based reasoning [Ashley 1990, Aleven 2003] is a popular approach to precedential reasoning in AI&law research. The key idea is that a case can be represented as a set of factors, where a factor is a legally relevant aspect. Factors are assumed to have a direction, i.e., to favor certain outcomes. Usually both factors and outcomes are assumed to be binary, so that each factor can be labelled with the outcome it favors (usually denoted as $\pi$, the outcome requested by the plaintiff, and $\delta$, the outcome requested by the defendant). The party which is interested in a certain outcome in a new case can support her request by citing a past case that has the same outcome, and shares with the new case some factors supporting that outcome. The party that is interested in countering that outcome can respond with a distinction, i.e., can argue that some factors which supported that outcome in the precedent are missing in the new case or that some additional factors against that outcome are present in the new case. Horty [Horty 2004, Horty & Bench-Capon 2012] has developed the factor-based models of precedent into a theory of precedential constraints, i.e., of how a new case must be decided, in order to preserve consistency in the case law. In [Horty 2011, Horty 2017], he takes into account the fact that judges may also provide explicit reasons for their choice of a certain outcome. This leads to the distinction between the result and the reason model of precedents. In the first model, the message conveyed by the case is only that all factors supporting the case-outcome (pro-factors) outweigh all factors against that outcome (con-factors). In the second, the message is that the factors for the case outcome indicated by the judge (a subset of all pro-factors) outweigh all con-factors.

In this chapter we shall combine the modal logic approach to classifiers and their explanations in the previous chapter with the CBR introduced above. The combination is based on the fact that both a classifier and CBR map sets of features to decisions or classifications. In this way, our contribution is at least twofold.

First, we explore the relation between two apparently unrelated reasoning systems. While the connection between CBR and reasoning about classifier systems is of interest in itself, we believe that, through this relation, new research perspectives can be offered, since we could in the future investigate CBR by exploiting several techniques and results from modal logic. We will see that the challenge of this chapter is to adapt the formal representation of a classifier to the bidirectionality of factors in the HYPO model. Once this is solved, we can provide a logical model and a semantics for factor-based CBR.

Second, we investigate the idea of normative explanation: while the literature on the concept of explanation is immense, the AI community is now paying attention to it due to the development of explainable AI (XAI) [Miller *et al.* 2022, Atkinson *et al.* 2020]. Our paper, by connecting CBR and reasoning about classifier systems, explores different notions of explanation in law, such as abductive and contrastive explanations for the outcome suggested by the case-based reasoner.

Our model allows for building explainable case-based reasoners, which could also be deployed to reproduce and analyze the functioning of opaque predictors of the outcome of cases. We import notions such as prime implicant and contrastive explanation in the domain of XAI for classifiers to showcase how to analyze CBR in the field of XAI.

The chapter is organized as follows. Section 4.2 presents Horty's models of CBR. In Section 4.3 we explore the connection between CBR and classifier models. Section 4.4 deepens the connection by establishing a one-one relation between consistent case bases and certain classifier models. Section 4.5 shows that notions for classifier explanation in XAI help study case bases. Section 4.6 reveals the relation between Horty's models of case bases and monotone pBFs. Finally, Section 4.7 discusses related work and concludes. Proofs are in Appendix B.

## 4.2 Horty's Two Models of Case-Based Reasoning

In this section we introduce the two models of precedential constraint of case-based reasoning proposed by Horty. We will use the term *result model* as shorthand for "the factor-based result model of precedential constraint", and *reason model* for "the factor-based reason model of precedential constraint". Since the result model can be viewed as a special kind of reason model, we will also use *Horty case bases* and *Horty's models* as umbrella terms for these cases.

Let $Atm_0 = Plt \cup Dfd$, where $Plt$ and $Dfd$ are disjoint sets of finitely many factors favoring the plaintiff and defendant respectively. In addition, let $Val = \{1, 0, ?\}$ where elements stand for *plaintiff wins*, *defendant wins* and *indeterminacy* respectively. Let $Dec = \{\mathtt{t}(c) : c \in Val\}$ and read $\mathtt{t}(c)$ as "the actual decision/outcome (of the judge/classifier) takes value $c$". An outcome $\mathtt{t}(1)$ or $\mathtt{t}(0)$ means that, the judge is predicted to decide for the plaintiff or for the defendant (the classifies "forces" one of the two outcomes). The outcome $\mathtt{t}(?)$ means either outcome would be consistent: the judge may develop the law in one direction or the other. This reflects the incompleteness nature of CBR. We use $Atm$ to denote $Atm_0 \cup Dec$.

We call $s \subseteq Atm_0$ a *fact situation*. A set of atoms $X$ is called a *reason* for an outcome (decision) $c$ if it a set of factors all favoring the same outcome: $X \subseteq Plt$ is a reason for 1 and $X \subseteq Dfd$ is a reason for 0. A (defeasible) *rule* consist of a reason and the corresponding outcome: $X \mapsto c$ is rule, if $X \subseteq Plt$ and $c = 1$, or $X \subseteq Dfd$ and $c = 0$.[1] For readability, we make a convention that, for $c \in \{0, 1\}$, let $\overline{c} = 1 - c$ and $\overline{\overline{c}} = c$. Moreover, let $Atm_0^c = Plt$ if $c = 1$, and $Atm_0^c = Dfd$ if $c = 0$.

In the reason model, a *precedent case* (precedent) is a triple $\mathfrak{c} = (s, X, c)$, where $s \subseteq Atm_0$, $X \subseteq Atm_0^c, c \in \{0, 1\}$. In plain words, $s \cap Atm_0^c$ contains all *pro-factors* in $s$ for $c$, while $s \cap Atm_0^{\overline{c}}$ all *con-factors* in $s$ for $c$. $X$ is the *reason of the case*, namely a subset of the pro-factors which the judge considers sufficient to support that outcome, relative to all con-factors in the case.

[1] Unlike [Horty 2011], we will follow [Prakken 2021] in using *reason* instead of *rule* when defining a case as a triple.

A *case base CB* (for reason model) is a set of precedential cases. When the reason contains all pro-factors within the situation (i.e., when $\mathfrak{c} = (s, s \cap Atm_0^c, c)$) all such factors are considered equally decisive. If a case base only contains cases of this type, we obtain what Horty calls "the result model", and note such a case base $CB^{res}$.[2] The class of all *CBs* and $CB^{\text{res}}s$ are noted **CB** and $\mathbf{CB}^{\text{res}}$ respectively.

**Example 4.1** (Running example)**.** *In the chapter we refer to the following running example taken from [Prakken 2021]. Let us assume the following six factors, each of which either favors the outcome 'misuse of trade secrets' ('the plaintiff wins') or rather favors the outcome 'no misuse of trade secrets' ('the defendant wins'). Factors pro the plaintiff are*

1. *the defendant has obtained the secret by deceiving the plaintiff (`deceive`),*
2. *the defendant has bribed an employee of the plaintiff (`bribe`),*
3. *the plaintiff had taken security measures to keep the secret (`security`).*

*Factors pro the defendant are*

1. *the information is obtainable elsewhere (`else-obtain`),*
2. *the product is reverse-engineerable (`re-engine`),*
3. *the plaintiff had voluntarily disclosed the secret to outsiders (`voluntary`).*

*Hence in our running example* $Atm = \{$`deceive`, `bribe`, `security`, `else-obtain`, `re-engine`, `voluntary`, $\mathsf{t}(0), \mathsf{t}(1), \mathsf{t}(?)\}$ *Let us consider a case base* $CB^{ex} = \{\mathfrak{c}_1, \mathfrak{c}_2\}$ *with*

- $\mathfrak{c}_1 = (\{$`deceive`, `security`, `else-obtain`, `voluntary`$\}, \{$`deceive`$\}, 1)$;
- $\mathfrak{c}_2 = (\{$`bribe`, `else-obtain`, `voluntary`$\}, \{$`voluntary`$\}, 0)$

*That is to say, in precedent 1 the judge classified the plaintiff won, because of (regarding its fact situation) the deceive; in precedent 2 the judge classified the defendant won, because of (regarding its fact situation) the voluntary disclose.*

A case base can be inconsistent when two precedents map the same fact situation to different outcomes. Another scenario is that a consistent case base becomes inconsistent after *update*, namely after expanding it with some new case. Hence maintaining consistency is the crucial concern of case-based reasoning. But first of all, one need to define these notions. The following definitions, except symbolic difference, are based on [Horty 2011, Prakken 2021].

**Definition 4.1** (Preference relation derived from a case)**.** *Let* $\mathfrak{c} = (s, X, c)$ *be a case. Then the preference relation* $<_{\mathfrak{c}}$ *derived from* $\mathfrak{c}$ *is s.t. for any two reasons* $Y$*,* $Y'$ *favoring* $c$ *and* $\overline{c}$ *respectively,* $Y' <_{\mathfrak{c}} Y$ *iff* $Y' \subseteq s \cap Atm_0^{\overline{c}}$ *and* $X \subseteq Y$*.*

[2] So we view a result model as a special kind of reason model, as [Horty 2011, p. 25] also mentioned.

**Definition 4.2** (Preference relation derived from a case base)**.** *Let CB be a case base. Then the preference relation* $<_{CB}$ *derived from CB is s.t. for any two reasons* $Y, Y'$ *favoring* $c$ *and* $\bar{c}$ *respectively,* $Y' <_{CB} Y$ *iff* $\exists \mathfrak{c} \in CB$ *s.t.* $Y' <_{\mathfrak{c}} Y$*.*

**Definition 4.3** ((In)consistency)**.** *A case base CB is inconsistent, if there are two reasons* $Y, Y'$ *s.t.* $Y' <_{CB} Y$ *and* $Y <_{CB} Y'$*. CB is consistent if it is not inconsistent.*

**Definition 4.4** (Precedential constraint)**.** *Let CB be a consistent case base, X is a reason for c in CB and applicable in a new fact situation* $s'$*, i.e.* $X \subseteq s'$*. Updating CB with the new case* $(s', X, c)$ *meets the precedential constraint, iff* $CB \cup \{(s', X, c)\}$ *is still consistent.*

There is more than one way to satisfy the precedential constraint, depending on how the precedents in *CB* interact with the new case. The requirement of consistency dictates the outcome when the so-called *a fortiori constraint* applies: if reason $X$ for $c$ outweighs (i.e., is stronger that) reason $s \cap Atm_0^{\bar{c}}$, "a fortiori" any superset of $X$ outweighs any subset of $s \cap Atm_0^{\bar{c}}$, so that only by deciding for $c$ rather than for $\bar{c}$ consistency is maintained.[3] In this way the doctrine of the theory of precedent, *stare decisis* (to stand by things decided), is upheld.

**Example 4.2** (Running example)**.** *Let us consider two fact situations according to case base* $CB^{ex}$ *running example.*

- *In* $s_3 = \{\texttt{deceive}, \texttt{else-obtain}, \texttt{voluntary}\}$*, only a decision for* 1 *in* $s_3$ *is consistent with* $CB^{ex}$*, since a decision for* 0 *would entail that* $\{\texttt{else-obtain}\} >_{CB^{ex}} \{\texttt{deceive}\}$*, contrary to the preference* $\{\texttt{deceive}\} >_{CB^{ex}} \{\texttt{else-obtain}\}$*, which is derivable from* $\mathfrak{c}_1$*.*
- *In* $s_4 = \{\texttt{bribe}, \texttt{re-engine}\}$ *both* $(s_4, \{\texttt{bribe}\}, 1)$ *and* $(s_4, \{\texttt{re-engine}\}, 0)$ *are consistent with* $CB^{ex}$*, since we have neither* $\{\texttt{bribe}\} >_{CB^{ex}} \{\texttt{re-engine}\}$ *nor* $\{\texttt{re-engine}\} >_{CB^{ex}} \{\texttt{bribe}\}$*.*

## 4.3 A Representation Theorem

In this section we shall show that the language of case bases can be translated into the language $\mathcal{L}(Atm)$; hence case bases can be studied by classifier models. Recall $Atm = Atm_0 \cup Dec$, and we require $Atm_0$ to be finite and $Dec = \{\mathsf{t}(1), \mathsf{t}(0), \mathsf{t}(?)\}$. Definitions of classifier models and their semantics can be found in Chapter 3. Recall an important abbreviation that we use: for any $X \subseteq Y \subseteq Atm_0$,

$$\mathsf{cn}_{X,Y} =_{def} \bigwedge_{p \in X} p \wedge \bigwedge_{p \in Y \setminus X} \neg p.$$

[3] We generalize a fortiori constraint from only acting on result models in [Horty 2011] to also on reason models in the same manner as viewing a result model as a special reason model, whose reason contains all pro-factors.

It will be shown that a case base is consistent in Horty's sense, if and only if its translation, together with the following two formulas that we abbreviate as `Compl` and `2Mon`, is satisfiable in **CM** the class of classifier models:

$$\texttt{Compl} =_{def} \bigwedge_{X \subseteq Atm_0} \Diamond \mathsf{cn}_{X,Atm_0}$$

$$\texttt{2Mon} =_{def} \bigwedge_{c \in \{0,1\}, X \subseteq Atm_0^c, Y \subseteq Atm_0^{\overline{c}}} \Big( \Diamond(\mathsf{cn}_{X \cup Y, Atm_0} \wedge \mathsf{t}(c)) \rightarrow \bigwedge_{Atm_0^c \supseteq X' \supseteq X, Y' \subseteq Y} \Box(\mathsf{cn}_{X' \cup Y', Atm_0} \rightarrow \mathsf{t}(c)) \Big)$$

According to `Compl`, *every possible* situation description must be satisfied in the classifier, where a situation description is a conjunction of factors (those being present in $X$) and negations of factors (those being absent from $X$).

`2Mon` introduces a *two-way monotonicity*, which is meant to implement the *a fortiori* constraint: if the classifier associates a situation $s$ to an outcome $c$, then it must assign the same outcome to every situation $s'$ such that both (a) $s'$ includes all factors for $c$ that are in $s$ and (b) $s'$ does *not include* factors for $\overline{c}$ that are *outside of* $s$. This formula is meant to maintain consistency with respect to the preference relation, as Definition 4.1 indicates: if a case including reason $X$ for $c$ and factors $Y$ for $\overline{c}$, has outcome $c$, it means that $X > Y$. Thus it cannot be that outcome $\overline{c}$ is assigned to a situation $s'$ including both a superset $X' \supseteq X$ of factors for $c$ and only a subset $Y' \subseteq Y$ of factors for $\overline{c}$. In fact, if $X > Y$, then is must be the case that also $X' > Y'$, while a decision for $\overline{c}$ would entail that $X' < Y'$.

Let $\mathbf{CM}^{prec} = \{C = (S, f) \in \mathbf{CM} : \forall s \in S, (C, s) \models \texttt{Compl} \wedge \texttt{2Mon}\}$, where $\mathbf{CM}^{prec}$ means the class of CMs for precedent theory. Satisfiability and validity relative to $\mathbf{CM}^{prec}$ are defined in an analogous way as **CM**.[4]

To translate a case base for result model $CB^{\text{res}}$ into a classifier model $(S, f)$, we need to ensure that all the precedents in the case base are "verified" in the classifier model. That means, $\forall(s, s \cap Atm_0^c, c) \in CB^{\text{res}}$, $f(s) = c$. This can be accomplished directly through the following definition.

**Definition 4.5** (Translation of case base for result model)**.** *The translation function $tr_1$ maps each case from a case base $CB^{\text{res}}$ to a corresponding formula in the language $\mathcal{L}(Atm)$. It is defined as follows:*

$$tr_1(s, s \cap Atm_0^c, c) =_{def} \Diamond(\mathsf{cn}_{s,Atm_0} \wedge \mathsf{t}(c)).$$

*We generalize it to the entire case base $CB^{\text{res}}$ as follows:*

$$tr_1(CB^{\text{res}}) =_{def} \bigwedge_{(s, s \cap Atm_0^c, c) \in CB} tr_1(s, s \cap Atm_0^c, c).$$

[4]Since `Compl` makes it mandatory that $\forall(S, f) \in \mathbf{CM}^{prec}, S = 2^{Atm_0}$, we will also sometimes simply write $(2^{Atm_0}, f)$.

**Example 4.3.** *The precedent* $(\{\texttt{else-obtain}, \texttt{re-engine}, \texttt{deceive}\}, \{\texttt{else-obtain}, \texttt{re-engine}\}, 1\})$ *in a case base for result model is translated as* $\Diamond(\texttt{else-obtain} \wedge \texttt{re-engine} \wedge \texttt{deceive} \wedge \neg\texttt{voluntary} \wedge \neg\texttt{bribe} \wedge \neg\texttt{security} \wedge \mathsf{t}(1))$*, which means that* $f(\{\texttt{else-obtain}, \texttt{re-engine}, \texttt{deceive}\}) = 1$

In the translation for the reason model we need to capture the role of reasons. This is obtained by ensuring that for every precedent $(s, X, c)$, not the fact situation $s$ directly, but the one consisting only of reason $X$ and all $\overline{c}$-factors in $s$ (i.e. $s \cap Atm_0^{\overline{c}}$) is classified as $c$. It reflects the fact that the precedent finds pro-factors outside of $X$ dispensable for the outcome.

**Definition 4.6** (Translation of case base for reason model)**.** *The translation function* $tr_2$ *maps each case from a case base CB to a corresponding formula in the language* $\mathcal{L}(Atm)$*. It is defined as follows:*

$$tr_2(s, X, c) =_{def} \Diamond(\mathsf{cn}_{X \cup (s \cap Atm_0^{\overline{c}}), Atm_0} \wedge \mathsf{t}(c)).$$

*We generalize it to the entire case base CB as follows:*

$$tr_2(CB) =_{def} \bigwedge_{(s,X,c) \in CB} tr_2(s, X, c).$$

Notice that the function $tr_1$ for the result model is a special case of the function $tr_2$ for the reason model, since $((s \cap Atm_0^c) \cup (s \cap Atm_0^{\overline{c}}) = s$.

**Fact 4.1.** $tr_1(s, s \cap Atm_0^c, c) = tr_2(s, s \cap Atm_0^c, c)$.

Let us clarify the meaning of $tr_2$, which may seem less obvious than $tr_1$. The goal is to find a CM $C$ from $\mathbf{CM}^{prec}$ for any consistent case base $CB$, such that *all* the fact situations that can be forced to make decision $c$ via a fortiori reasoning by $CB$ are "correctly" classified as $c$ in the CM. In other words, $C$ encodes not only all *actual*, but also *potential* decisions that $CB$ would make. Therefore for reason models $tr_1$ is insufficient, since for any precedent $(s, X, c)$, even if $X \cup (s \cap Atm_0^{\overline{c}})$ is not the fact situation of any precedent in $CB$, we shall guarantee its classification as $c$. As a result we have $tr_2$, which enables us to obtain the following theorem.

**Theorem 4.1.** *Let* $CB \in \mathbf{CB}$ *be a case base. Then, CB is consistent iff* $tr_2(CB)$ *is satisfiable in* $\mathbf{CM}^{prec}$.

In light of the theorem and the fact above, the special case regarding result models turns to be a corollary.

**Corollary 4.1.** *Let* $CB^{\text{res}} \in \mathbf{CB}^{\text{res}}$ *be a case base for the result model. Then,* $CB^{\text{res}}$ *is consistent iff* $tr_1(CB^{\text{res}})$ *is satisfiable in* $\mathbf{CM}^{prec}$.

Similarly, the precedential constraint can also be represented as a corollary.

**Corollary 4.2.** *Let* $CB \in \mathbf{CB}$ *be a consistent case base and* $(s', X, c)$ *a case. Updating CB with* $(s', X, c)$ *meets the precedential constraint, iff* $tr_2(CB) \wedge tr_2(s', X, c)$ *is satisfiable in* $\mathbf{CM}^{prec}$.

**Example 4.4.** *Case* $\mathfrak{c}_3 = (\{\texttt{deceive}, \texttt{bribe}, \texttt{voluntary}\}, \{\texttt{voluntary}\}\ , 0)$ *is incompatible with the* $CB^{ex}$*. Otherwise according to* $tr_2(CB^{ex} \cup \{c_3\})$*,* `2Mon` *and* `Compl`*, for any classifier model representing the updated case base, the fact situation* $\{\texttt{deceive}, \texttt{bribe}, \texttt{else-obtain}, \texttt{voluntary}\}$ *should be classified both as* 1*, based on* $CB^{ex}$*, and* 0*, based on* $c_3$*.*

## 4.4 Genuine Classifier of Horty Case Base

The representation theorem above associates a Horty case base with a *set* of classifier models. In this section we refine the association by corresponding a Horty case base to a unique classifier model, in which sense we call it "genuine". In doing so, we can study Horty case bases with notions of explanation for classifiers, which is the topic of the next section.

The basic idea is that the genuine classifier of a case base outputs ? for *all and only* fact situations which cannot be forced to make a decision by the case base.[5]

**Definition 4.7** (Genuine classifier)**.** *Let CB be a case base. Then the genuine classifier of CB is defined as a function* $f : 2^{Atm_0} \longrightarrow \{0, 1, ?\}$*, s.t. for any fact situation* $s$*,*

$$f(s) = \begin{cases} c \text{ if } & \exists \mathfrak{c} = (s', X, c) \in CB \text{ s.t.} \\ & X \cap Atm_0^c \subseteq s \cap Atm_0^c, s' \cap Atm_0^{\bar{c}} \supseteq s \cap Atm_0^{\bar{c}}; \\ ? & \text{otherwise.} \end{cases}$$

*The genuine classifier model of CB is therefore* $C = (2^{Atm_0}, f)$*.*

That is to say, $f(s') =?$ if and only if a fortiori reasoning fails to force $s'$ to take either decision 0 or 1. We can also define the genuine CM for a case base in terms of a formula.

**Proposition 4.1.** *Let CB be a Horty case base. Then a classifier model* $C = (S, f)$ *is the genuine classifier of CB, if* $\forall s \in S$*,* $(C, s) \models \varphi_{CB}$ *where*

$$\varphi_{CB} =_{def} \bigwedge_{X \subseteq Atm_0} \Diamond \mathsf{cn}_{X, Atm_0} \wedge \bigwedge_{c \in \{0,1\}} \Box\Big(\mathsf{t}(c) \leftrightarrow \bigvee_{\mathfrak{c} = (s', X, c) \in CB} \mathsf{cn}_{X, X \cup (Atm_0^{\bar{c}} \setminus s')}\Big).$$

The "syntactic" representation $\varphi_{CB}$ is stronger than the formula $\texttt{Compl} \wedge \texttt{2Mon} \wedge tr_2(CB)$ in the previous section, due to the bi-conditional in $\varphi_{CB}$ for $\mathsf{t}(c)$ with $c \in \{0, 1\}$. Hence, fixing the language, $\varphi_{CB}$ is satisfied in exactly one CM.

**Proposition 4.2.** *Every Horty case base induces exactly one genuine classifier model.*

[5] It is therefore the "smallest" representation, if we think of every classifier as a partial function by naturally viewing ? as *undefined*.

The inverse, however, does not hold, for two possibilities. 1) There can be two case bases forcing exactly the same set of fact situations but they have some precedents differing in their fact situations and/or reasons. 2) More intriguing, two case bases are identical but in different *languages*, namely, their plaintiff-defendant partitions of sets of factors differ, but they still force the same set of fact situations. This is possible, because case bases can be incomplete and some factors are "dummy". The following examples instantiates both possibilities.

**Example 4.5.** *Let* $Plt = \{p_1\}$ *and* $Dfd = \{p_2\}$*. Then, the following two case bases have the same genuine classifier*

- $\{(\{p_1, p_2\}, \{p_1\}, 1)\}$
- $\{(\{p_1, p_2\}, \{p_1\}, 1), (\{p_1\}, \{p_1\}, 1)\}$

*namely an* $f : 2^{\{p_1,p_2\}} \longrightarrow \{0, 1, ?\}$ *s.t.* $f(s) = 1$ *if* $p_1 \in s$, $f(s) =?$ *otherwise. Moreover, if we set* $Plt' = \{p_1, p_2\}, Dfd' = \emptyset$*, the two case bases are still instances of the new language, and have the same genuine classifier.*

## 4.5 Explanations and Landmarks

The notion of genuine classifier of Horty case bases paves the way to providing classifier explanations for the outcomes of legal cases. For this purpose it is necessary to introduce the following notations. (Most of them are already introduced in Chapter 3 and for the sake of self-containment of the chapter we restate them here.)

Let $\lambda$ denote a conjunction of finitely many literals, where a literal is an atom $p$ (positive literal) or its negation $\neg p$ (negative literal). We write $\lambda \subseteq \lambda'$, call $\lambda$ a part (subset) of $\lambda'$, if all literals in $\lambda$ also occur in $\lambda'$; and $\lambda \subset \lambda'$ if $\lambda \subseteq \lambda'$ but not $\lambda' \subseteq \lambda$. We write $Lit(\lambda), Lit^+(\lambda), Lit^-(\lambda)$ to mean all literals, all positive literals and all negative literals in $\lambda$ respectively. By convention $\top$ is a term of zero conjuncts. In the glossary of Boolean classifier (function), $\lambda$ is called a *term* or *property* (of the instance $s$). The set of terms is noted *Term*. A key role in our analysis is played by the notion of a (prime) implicant, i.e., a (subset-minimal) term which makes a classification necessarily true.

### 4.5.1 Prime implicant and abductive explanation

**Definition 4.8** (Implicant (Imp) and prime implicant (PImp))**.** *We write* $\mathsf{Imp}(\lambda, c)$ *to mean that* $\lambda$ *is an* implicant *for* $c$ *and define it as* $\mathsf{Imp}(\lambda, c) =_{def} \Box(\lambda \rightarrow \mathsf{t}(c))$*. We write* $\mathsf{PImp}(\lambda, c)$ *to mean that* $\lambda$ *is a* prime implicant *for* $c$ *and define it as*

$$\mathsf{PImp}(\lambda, c) =_{def} \Box\Big(\lambda \rightarrow (\mathsf{t}(c) \wedge \bigwedge_{p \in Atm(\lambda)} \langle Atm(\lambda) \setminus \{p\}\rangle \neg \mathsf{t}(c))\Big).$$

According to the definition, $\lambda$ being an implicant for $c$ means that any state $s$ verifying $\lambda$ is necessarily classified as $c$ (necessity); and $\lambda$ being a prime implicant for $c$ means that any proper subset of $\lambda$ is not an implicant for $c$ (minimality).[6] Implicants explain the classifier in the sense that to know an implicant satisfied at a state is to know the classification of the state.

Our first result bridging classifier explanations and case bases is the following proposition. To put it simply, if a consistent case base $CB$ has a precedent $(s, X, c)$, then for any classifier model representing $CB$, $s$ must be incompatible with every prime implicant $\lambda$ for $\overline{c}$. Otherwise, by definition $s$ should be classified as $\overline{c}$ instead of $c$. This would imply that $CB$ is inconsistent thanks to the representation theorem. To ensure that $s$ is incompatible with every prime implicant $\lambda$ for $\overline{c}$, either $\lambda$ has some literal $\neg p$, where $p$ is in $X$ and hence is true at $s$; or $\lambda$ has some literal $p$, where $p \notin s \cap Atm_0^{\overline{c}}$ and hence is false at $s$.

**Proposition 4.3.** *Let $CB$ be a consistent case base and $(s, X, c) \in CB$, and $C \in \mathbf{CM}^{prec}$ s.t. $(C, s) \models tr_2(CB)$. Then, $\forall \lambda \in Term,$, if $(C, s) \models \mathsf{PImp}(\lambda, \overline{c})$, then either $X \cap Atm(Lit^-(\lambda)) \neq \emptyset$ or $s \cap Atm_0^{\overline{c}} \not\supseteq Atm(Lit^+(\lambda))$.*

**Example 4.6.** *Let $C = (S, f) \in \mathbf{CM}^{prec}$ be a model of $tr_2(CB^{ex})$. Obviously* `deceive` *cannot be PImp for* 0*, otherwise $f(s_1) = 0$, contrary to $\mathfrak{c}_1$. Also $\neg$* `re-engine` *$\wedge$* `bribe` *cannot be PImp for* 1*, otherwise $f(\{$*`bribe`*,* `else-obtain`*,* `voluntary`*$\}) = 1$, contrary to $\mathfrak{c}_2$.*

As mentioned, people in XAI focus more on local (prime) implicants, namely (prime) implicants *true at the current state*. This gives rise to definitions of abductive explanation and its weak version.

**Definition 4.9** (Abductive explanation (AXp) and weak abductive explanation (wAXp) )**.** *We write $\mathsf{AXp}(\lambda, c)$ to mean that $\lambda$* abductively explains *the decision $c$ and define it as $\mathsf{AXp}(\lambda, c) =_{def} \lambda \wedge \mathsf{PImp}(\lambda, c)$. We write $\mathsf{wAXp}\ (\lambda, c)$ to mean that $\lambda$* weak-abductively explains *the decision $c$ and define it as $\mathsf{wAXp}\ (\lambda, c) =_{def} \lambda \wedge \mathsf{Imp}(\lambda, c)$.*

The proposition below states that every reason (of a fact situation in a consistent case base) is a positive part of some weak AXp of that situation in any classifier representing the case base. In fact, for any precedent $(s, X, c)$ in a consistent case base, we can always identify one weak AXp in any of its CM representation, i.e., simply the conjunction of all factors in $X$ and negations of all $\overline{c}$-factors that are in $s$.

**Proposition 4.4.** *Let $CB$ be a consistent case base, $(s, X, c) \in CB$, and $C \in \mathbf{CM}^{prec}$ be a model of $tr_2(CB)$. Then $\exists \lambda \in Term$ s.t. $Atm(Lit^+(\lambda)) = X$ and $(C, s) \models \mathsf{wAXp}\ (\lambda, c)$.*

[6] Notice that we have not fully used the expressive power of $[X]\varphi$ and $\langle X\rangle\varphi$ until now for minimality. The intuitive meaning of $\langle Atm(\lambda) \setminus \{p\}\rangle\neg\mathsf{t}(c)$ in the formula is that even if we just perturb one variable $p$ in $\lambda$ from its actual value, the classification will no longer be $c$.

**Example 4.7.** *Let* $C \in \mathbf{CM}^{prec}$ *be a model of* $tr_2(CB^{ex})$*. Then we have* $(C, s_1) \models$ wAXp $(\texttt{deceive} \wedge \neg\texttt{re-engine}, 1)$ *and* $(C, s_2) \models$ wAXp $(\texttt{voluntary} \wedge \neg\texttt{deceive} \wedge \neg\texttt{security}, 0)$*. Notice that* $(C, s_2) \models \neg$wAXp $(\texttt{voluntary}, 0)$ *though* $\{\texttt{voluntary}\}$ *is the reason of* $\mathfrak{c}_2$*, because e.g.* $(C, s_1) \models \texttt{voluntary} \wedge \neg\mathsf{t}(0)$*. It exemplifies that both pro-factors in the reason and con-factors absent (i.e.* `deceive, security` *here) are decisive.*

The proposition above has a notable limitation: it bridges only the relation between reasons (of a consistent case base) and weak AXps (in the CMs that represent it), since reasons in general do not meet the minimality condition in the CM representations. This is due to the fact that according to Theorem 4.3, there are different CMs which can represent the same case base $CB$, as long as they satisfy $tr_2(CB) \wedge \texttt{2Mon} \wedge \texttt{Compl}$. They may well have different prime implicants. In light of this we should concentrate on genuine classifiers as defined in the last section, for we have "better control" over the genuine classifier of a case base compared to other classifiers representing it. Interestingly, in genuine classifiers the prime implicant is shown to be related to a specific notion that has been recently suggested in case-based reasoning.

### 4.5.2 Prime implicant and landmark case

In [Van Woerkom *et al.* 2022], a notion called *landmark case* was introduced. The notion relies on an observation, that some precedents are superfluous in a case base, since their outcomes are forced by other precedents. In contrast, a landmark case "represents new legal ground, and the decision maker has used its discretion, going beyond what is decided by other cases" [Van Woerkom *et al.* 2022].

The original context of landmark cases is in *dimensions* rather than factor-based models. However, we can easily define the landmark case in the latter context. We bridge landmark cases and prime implicants, in showing that in the factor-based models landmarks "are" prime implicants, because the former have a *subset-minimal pro-factors present*, and a *subset-minimal con-factors absent*, which correspond the positive literals and negative literals in the latter respectively.

**Definition 4.10** (Landmark case in factor-based case base)**.** *Let* $CB$ *be a Horty case base and* $\mathfrak{c} = (s, X, c) \in CB$*. We say that* $\mathfrak{c}$ *is* ordinary*, if* $\exists \mathfrak{c}' = (s', X', c) \in CB$ *s.t.* $s \cap Atm_0^{\overline{c}} <_{\mathfrak{c}'} X$ *and* $\mathfrak{c}' \neq \mathfrak{c}$*. Otherwise* $\mathfrak{c}$ *is called a* landmark *case.*

In plain words, $(s, X, c) \in CB$ is a landmark, if the decision $c$ for $s$ cannot be forced by any other precedent in $CB$ via a fortiori reasoning.

**Example 4.8.** *Take the case base* $CB^{ex}$*. Suppose it expands with a new case* $(\{\texttt{deceive, else-obtain, re-engine, voluntary}\}, \{\texttt{deceive}\}, 1)$*. The new case is a landmark and* $\mathfrak{c}_1 = (\{\texttt{deceive}, \texttt{security}, \texttt{else-obtain}, \texttt{voluntary}\}, \{\texttt{deceive}\}, 1)$ *becomes ordinary, since the latter is forced by the former.*

Indeed, the new precedent above is a "strongest" case, since it has only one *Plt*-factor present but beats all *Dfd*-factors (i.e. no *Dfd*-factor absent). We say "a" rather than "the", because it is subset-minimality rather than cardinal-minimality.

The result below reveals the aforementioned relation between landmarks in a case base and prime implicants in its genuine classifier.

**Proposition 4.5.** *Let CB be a consistent case base and* $C = (2^{Atm_0}, f)$ *its genuine CM. Then,* $\mathfrak{c} = (s, X, c)$ *is a landmark if and only if* $(C, s) \models \mathsf{PImp}(\bigwedge_{p \in s \cap X} p \wedge \bigwedge_{q \in Atm_0^{\bar{c}} \setminus s} \neg q, c)$.

The following fact is not hard to see in light of the fact that a classifier always has a prime implicant for each direction.

**Fact 4.2.** *Let* $f$ *be the genuine classifier of some consistent case base CB and* $s$ *a fact situation. Then,* $f(s) = c$ *with* $c \in \{0, 1\}$ *if and only if there is a landmark precedent for* $c$ *in CB.*

At the end of last section we mentioned that some case bases have the same genuine classifier. The following fact is not hard to see and gives the sufficient and necessary condition of having the same genuine classifier in terms of landmark cases.

**Fact 4.3.** *Two case bases (regardless of whether in the same language) force the same set of fact situations, if and only if they have the same landmark cases.*

### 4.5.3 Contrastive explanation

The idea of contrastive explanation is dual with abductive explanation, since it points to a minimal part of a situation whose change would falsify the current decision, and the duality between their weak versions is similar [Huang *et al.* 2022]. A conjunction of literals $\lambda$ is a contrastive explanation for outcome $c$ in situation $s$, if the following conditions are satisfied: (a) $\lambda$ is true at $s$, and $s$ has outcome $c$, (b) if all literals in $\lambda$ were false then the outcome would be different, (c) $\lambda$ is the subset-minimal literals satisfying (a) and (b). A weak contrastive explanation is only based on conditions (a) and (b).

**Definition 4.11** (Contrastive explanation (CXp) and weak CXP (wCXp))**.** *We write* $\mathsf{CXp}(\lambda, c)$ *to mean that* $\lambda$ constrastively explains *the decision* $c$ *and define*

$$\mathsf{CXp}(\lambda, c) =_{def} \lambda \wedge \langle Atm_0 \setminus Atm(\lambda)\rangle \neg \mathsf{t}(c) \wedge \bigwedge_{p \in Atm(\lambda)} [(Atm_0 \setminus Atm(\lambda)) \cup \{p\}]\mathsf{t}(c).$$

*We write* $\mathsf{wCXp}$ $(\lambda, c)$ *to mean that* $\lambda$ weak-contrastively explains *the decision* $c$ *and define it as* $\mathsf{wCXp}$ $(\lambda, c) =_{def} \lambda \wedge \mathsf{t}(c) \wedge \langle Atm_0 \setminus Atm(\lambda)\rangle \neg \mathsf{t}(c)$.

Intuitively speaking, we can test whether $\lambda$ is a wCXp of situation $s$ having outcome $c$ by "flipping" its positive literals to negative, and negative to positive, and observe if the resulting state is classified differently from $c$. CXps are the subset-minimal wCXps.

We can investigate the relations between precedents in a case base and CXps of its genuine classifier. [7] In plain words, given a genuine classifier of a case base, consider any CXp $\lambda$ of a fact situation $s$. According to the proposition below, there exists a precedent in the case base, such that by flipping values of factors in $\lambda$ one moves from $s$ to a fact situation which is classified differently by a fortiori reasoning with the precedent.

**Proposition 4.6.** *Let CB be a consistent case base, $C = (2^{Atm_0}, f)$ its genuine classifier. Then, for all $\lambda \in Term, s \in 2^{Atm_0}$ with $f(s) = c$, if $(C, s) \models \mathsf{CXp}(\lambda, c)$ then $\exists (s', X, \overline{c}) \in CB$ s.t. $Lit^-(\lambda) \subseteq X$ and $Lit^+(\lambda) \subseteq Atm_0^c \setminus s'$.*

Notice that the inverse of the proposition does not hold but has to be weakened from CXp to wCXp . The reason for that is, again, that a precedent in a case base may not be "minimal" in terms of a fortiori reasoning.

Let us end this section with a continuation of our running example.

**Example 4.9.** *Let $C = (S, f)$ be the genuine classifier of $CB^{ex}$. Recall that $\mathfrak{c}_2 = (\{\texttt{bribe}, \texttt{else-obtain}, \texttt{voluntary}\}, \{\texttt{voluntary}\}, 0)$ where $\{\texttt{bribe}, \texttt{else-obtain}, \texttt{voluntary}\}$ is simply noted $s_2$. We have $(C, s_2) \models \mathsf{CXp}(\texttt{bribe} \wedge \neg \texttt{deceive}, 0)$, for $\mathfrak{c}_1 = (\{\texttt{deceive}, \texttt{security}, \texttt{else-obtain}, \texttt{voluntary}\}, \{\texttt{deceive}\}, 1) \in CB^{ex}$, such that $Lit^-(\texttt{bribe} \wedge \neg \texttt{deceive}) = \{\texttt{deceive}\}$ and $Lit^+(\texttt{bribe} \wedge \neg \texttt{deceive}) = \{\texttt{bribe}\} \subseteq Dfd \setminus s_1$.*

## 4.6 Horty Case Bases and Monotone pBFs

We have been frequently spoken of monotonicity. For instance, the pivotal formula for the representation theorem is termed *two-way monotonicity*. The formula is designed to encode a fortiori reasoning, which follows the *superset of pro-factors plus subset of con-factors* pattern. Now we show that this naming is not arbitrary due to the relation between genuine classifiers and partial Boolean functions (pBFs).

Partial Boolean function (or *partially defined Boolean function*, as being used in [Crama & Hammer 2011]) is a generalization of Boolean function where the domain is partial. The definition of monotone Boolean function defined in Section 2.1.4 applies to partial Boolean functions with only a slight refinement, namely introducing a third output 0.5 standing for "undefined". Though the idea is natural and simple, to the best of my knowledge, monotone PBFs have not been studied in literature. Notice that in pBF there are also another two ways of defining monotone variables, see details in Appendix C.

**Definition 4.12** (Monotonicity in pBF)**.** *Let $f : 2^{Atm_0} \longrightarrow \{0, 1, 0.5\}$. We say that*

- *$f$ is* positive in *$p$, if $\forall s \in 2^{Atm_0}, f(s) \geq f(s \setminus \{p\})$;*

[7] Without defining genuine classifier we cannot use CXp, since for a case base, its CM representations may have different contrastive fact situations *minimally changed* from the current one.

- *$f$ is* negative in *$p$, if* $\forall s \in 2^{Atm_0}, f(s) \leq f(s \setminus \{p\})$.

*We say that $f$ is* monotone in *$p$, if $f$ is positive or negative in $p$. Moreover, $f$ is* monotone*, if $f$ is monotone in all its variables.*

We also sometimes say $p$ is positive/negative instead of $f$ being positive/negative in $p$ for short. Obviously the definition above is completely in line with us if we interpret ? as 0.5, and therefore below we will write ? instead of 0.5 for uniformity.

Equipped with the definitions of monotone variable and pBFs we can see their correspondence with Horty case bases. The main result is below.

**Proposition 4.7.** *Fix a language with $Atm_0 = Plt \cup Dfd$. Let $f$ be the genuine classifier of some consistent Horty case base. Then $\forall p \in Atm_0$,*

1. *if $p$ is non-negative, then $p \in Plt$; and if $p \in Plt$, then $p$ is positive;*
2. *$p$ is non-negative, if and only if*
   - *$p$ is present in the reason of some landmark precedent for* 1 or *$p$ is absent from some landmark precedent for* 0*;*
3. *if $p$ is non-positive, then $p \in Dfd$; and if $p \in Dfd$, then $p$ is negative;*
4. *$p$ is non-positive, if and only if*
   - *$p$ is present in the reason of some landmark precedent for* 0 or *$p$ is absent from some landmark precedent for* 1*.*

So we provide a necessary and a sufficient condition respectively for a variable being a member of *Plt* (or *Dfd*) by means of monotone variables. What is more, we show what role a variable plays in the case base if it is non-positive or non-negative. Namely, $p$ plays essential roles in landmarks – *either it is present in the reason in some landmark, or it is absent from the set of con-factors in some landmark.* The equal importance of being present and being absent is in line with the observations in the previous sections.

Notice that the inverses of the first and third statements in Proposition 4.7 do not hold. Namely, we cannot from e.g. $p$ being positive infer $p \in Plt$. The reason is that $p$ can be inessential when it is both positive and negative. The notion of (in)essential variable in Boolean functions is defined in Section 2.16, and naturally applies also to pBFs. Plainly speaking, it means $f(s) = f(s \setminus \{p\})$ for any $s$. The partition of $Plt \cup Dfd$ is a priori such that in general we cannot "revive" the partition through the monotonicity of variables in the classifier.

The observation therefore indicates that, comparing with $p$ being A for A $\in$ {positive, negative}, it is rather $p$ being non-A more informative. To be non-A one needs a "pair of witnesses" $(s, s \setminus \{p\})$ s.t. flipping the value of $p$ alone changes the output. This gives us extra information about the case base that the classifier genuinely represents.

| a variable can be | yes/no | when/because |
|---|---|---|
| both positive and negative | ✓ | when it is inessential |
| neither positive nor negative | × | because of a fortiori reasoning |
| both A and non-B | ✓ | when it plays a pivotal role landmarks |

Table 4.1: Monotonicity of variables in genuine classifiers, $A, B \in \{\text{negative, positive}\}$ and $A \neq B$

We can summarize the monotonicity of variables in genuine classifiers through Table 4.1, where ✓ means can be and × cannot be. We say "it plays a pivotal role in landmarks" to mean it appears in the reason of, or disappears from the con-factors of some landmark precedent, s.t. when its value flips in some case of the case base, the outcomes changes accordingly.

The final result comes as a corollary of Proposition 4.7, which establishes the expected relation between genuine classifiers and monotone pBFs.

**Corollary 4.3.** *Any genuine classifier of some consistent Horty case base is a monotone pBF.*

At last, we analyze what grounds the correspondence. Apparently a fortiori reasoning is a key property that leads to monotone pBFs. But it is not the only one responsible for that. There are another two properties, without which it is impossible to represent a case base as a pBF, "a fortiori" a monotone pBF.

1) The closed-world assumption. In database and AI, the closed-world assumption says that absence means falsity. Namely, if a variable is not found in the current database, then it is assumed to be false. Since we represent the absence of $p$ as $\neg p$, we adopt the closed-world assumption.

2) The finiteness of $|Atm_0|$. Recall that we fix our language as the union of finite set of variables/features/factors and the decision set $\{\mathtt{t}(0), \mathtt{t}(1), \mathtt{t}(?)\}$. This, together with the closed-world assumption, allows us to syntactically represent a fact situation as the conjunction of a maximal consistent set of literals. Recall that if $|Atm_0|$ is infinite, the conjunction of maximal consistent set of literals is syntactically illegal, i.e. not a well-formed formula.

## 4.7 Conclusion

In this chapter, we have demonstrated that through the concept of classifier a novel logical model of factor-based case-based reasoning can be provided, which allows for a rigorous analysis of case bases and of the inferences they support. In addition, it is revealed that under which conditions case bases under precedential constraint with a fortiori reasoning can be viewed as monotone partial Boolean functions.

As noted in the introduction, our work is based upon the case-based reasoning models of HYPO and CATO [Ashley 1990, Aleven 2003] and upon the analysis of

precedential constraint by Horty [Horty 2011, Horty & Bench-Capon 2012]. Further approaches exist that make use of logic in reasoning with cases. For instance, [Prakken & Sartor 1998] provided a factor-based model based on formal defeasible argumentation. More recently [Zheng *et al.* 2020a, Zheng *et al.* 2020b] represent precedents as propositional formulas and compare precedents by (propositional) logical entailment.

However, the propositional representation does not fully use the power of logic, in the sense that it does not provide a proof theory (axiomatics) for reasoning with precedents. By contrast, besides the semantic framework presented here, we can make syntactic derivations of properties of CBR using the axiomatics of BCL (an instance can be found at the end of Appendix B).

Moreover, our results allow for exploring different notions of explanation, such as abductive and contrastive explanations. We can accordingly explain why a case-based reasoning suggests a particular outcome (rather then a different one) in a new case. Thus, our model could be used to build explainable case-based reasoners, which could also be deployed to reproduce and analyze the functioning of opaque predictors of the outcome of cases. Thus, by bringing CBR into the broader context of classifier systems, we connect three lines of research: legal case-based reasoning, AI&Law approaches to explanation [Atkinson *et al.* 2020], techniques and results developed in the context of XAI.

In future work we will examine more deeply the relation between classifiers, explanations, and reasoning with legal precedents. Interesting developments pertain to addressing analogical reasoning beyond the a fortiori constraint considered here and to deploying ideas of explanation to extract knowledge out of cases (e.g., to determine the direction of factors and the way in which they interact).

Chapter 5

# Hamming Distance as Grounded Distance

In Chapter 3 we have introduced a family of counterfactual conditional operator $\Rightarrow_X$. The semantic interpretation is based on the Hamming distance between states/worlds. It was mentioned there when $Atm_0$ is finite, $\Rightarrow_{Atm_0}$ satisfies all semantic conditions of Lewis' logic VC. Moreover, since in classifiers we assume all inputs can be perturbed into any other input, the semantic constraint **U** (uniformity) is also satisfied, which makes the $\Rightarrow_{Atm_0}$ an operator for Lewis' VCU.

This chapter is dedicated to a follow-up, relatively independent, question: what if $Atm_0$ is infinite, and we make the counterfactual conditional operator as primitive instead of defining it from the modal operator (as we did for $\Rightarrow_X$)? That is to ask, what axiom besides the ones in VCU, if any, does one need in order to capture such a Hammingian semantics. As we will see, the answer is no, no extra axiom is needed. In other words, any abstract notion of minimal change for VC and VCU can be re-interpreted as the Hamming distance. It is in this sense we call Hamming distance "grounded".

## Contents

## 5.1 Introduction

Logics of counterfactual conditionals are widely studied and used in different areas including philosophy, linguistics and artificial intelligence. Among many logics of counterfactuals Lewis' VC, VCU and their relatives are arguably the most influential ones.[1] They correspond to, not only many other logics of counterfactuals, but also many popular theories in other fields e.g. AGM (belief revision) [Grove 1988], KM (belief update) [Grahne 1998], and KLM (preferential reasoning) [Kraus *et al.* 1990].

One reason for the correspondences is that they can all be subject to some semantics of *minimal change*, a term first used in [Gärdenfors 1984] and later becomes a standard umbrella term for the relative fields, e.g. [Katsuno & Mendelzon 1991, Aiguier *et al.* 2018]. That is to say, for instance, given a counterfactual conditional $\varphi \boxrightarrow \psi$, if $\varphi$ is not true at the actual world $w$, then one should check whether $\psi$ holds at the worlds "closest" to $w$ regarding $\varphi$. Closeness in the sense that the change from the actual one must be minimal.

But how is the distance for closeness defined? Most former systems in the literature are equipped with some abstract relation such as epistemic entrenchment, system of spheres, subformulas relations, faithful ordering etc, in order to obtain the additional information to construct the distance measure.

On the other hand, the *Hamming distance* is defined as the minimum number of substitutions required to change one string into another. In the context of possible world, it needs no more information than the number of atomic propositions that two worlds disagree on.

Now if we ask for some concrete definition of distance instead of those mysterious ones, the Hamming distance is a natural example, e.g. "one candidate of explication ... is the Hamming distance" [Dizadji-Bahmani & Bradley 2014], "the most commonly used is the Hamming distance" [Aiguier *et al.* 2018]. To our knowledge, this is the much preferred of the only two concrete definitions of distance.[2]

Hence it is not a surprise that a few systems directly define distance as Hammingian. The most famous one is the Dalal operator for belief revision [Dalal 1988], and follow-up works are e.g. [Pozos-Parra *et al.* 2013, Delgrande & Peppas 2015]. However, a shortcoming is that they all only consider a finite set of atoms/variables. To our limited knowledge, no much literature in AI studies/justifies using Hamming distance given (countably) infinite atoms, though there are some [Williamson 1988, Floridi 2010] in philosophy with different interests.

In explainable AI (XAI), the Hamming distance is the "right" distance measure for binary classifier explanation, because the input variables are mutually independent, and counterfactual reasoning is performed by perturbing some variables and

[1] Through this chapter, we use VC to denote both the logic and its axiomatics, VC is the name of the model of VC and **VC** its model class. We do the same for other Lewis' logics mentioned here.

[2] The other one not requiring additional information uses the subset relation: $v \preceq_w u$ iff $V(w) \triangle V(v) \subseteq V(w) \triangle V(u)$, $\triangle$ denoting symmetric difference.

observing the output. In Chapter 3 we came up with a conditional operator $\Rightarrow$ based on the Hamming distance and finite atoms in the language that essentially corresponds to a version of Lewis' VCU. Moreover, $\Rightarrow$ "is axiomatizable" by reducing to the S5 modal operator. But the case of infinite atoms was left unstudied. [3]

To address the undone work, we conjecture *the semantic constraint of distance being Hammingian is unaxiomatizable if the language has (countably) infinite atoms.* That indicates that the Hamming distance grounds VC and VCU, in the sense that their classes of models satisfy the same set of formulas as their subclasses with Hamming distance.

The rest of the chapter is structured as follows. Section 2 introduces Lewis' V models. In Section 3 we define Hammingian models and demonstrate some findings in their own right. The main result is in Section 4, where we show Hamming distance grounds VC and VCU. Section 5 concludes. [4]

## 5.2 Lewis' V Models

**Definition 5.1.** *The language for logics of counterfactual $\mathcal{L}(Atm)$ is defined as follows*

$$\varphi \quad ::= \quad p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \boxright \varphi,$$

*where $p$ ranges over $Atm$, a set of countable atomic propositions. Let $atm(\varphi)$ denote the atoms occurring in $\varphi$.*

Operators $\vee, \rightarrow, \leftrightarrow$ are defined as usual, and $\bot$ defined as $p \wedge \neg p$, $\top$ as $\neg\bot$, $\varphi \diamondright \psi$ as $\neg(\varphi \boxright \neg\psi)$, $\Box\varphi$ as $\neg\varphi \boxright \bot$ and $\Diamond\varphi$ as $\neg(\varphi \boxright \bot)$.

We then introduce the comparative similarity model of Lewis' V logics (V model in short).

**Definition 5.2** (V model)**.** *A tuple $M = (W, (W_w)_{w\in W}, (\preceq_w)_{w\in W}, V)$ is called a V model if $W$ is a non-empty set of worlds, $V : W \longrightarrow 2^{Atm}$ a valuation, and for all $w \in W$, $W_w \subseteq W$ and $\preceq_w$ is a partial order on $W_w$ for comparative similarity with the following constraint:*

- ***Connectedness**: $\forall v, u \in W_w$ either $v \preceq_w u$ or $u \preceq_w v$.*

*We have $v \prec_w u$ if $v \preceq_w u$ and $u \npreceq_w v$; $v \approx_w u$ if $v \preceq_w u$ and $u \preceq_w v$. We call $M$ finite if $W$ is finite. The class of V models is noted* $\mathbf{V}$*.* [5]

[3] We keep the notational difference of $\Rightarrow$ in Chapter 3 and $\boxright$ here. It helps remind the fact that $\Rightarrow$ is a shorthand for $\Rightarrow_{Atm_0}$ where $Atm_0$ is finite, while $\boxright$ has been discussed in a broader context where the cardinality of the language is restricted to finiteness. Also we do not introduce *Dec* into *Atm* as before since the axioms for decision atoms are not at stake.

[4] Since the main section of the chapter is a proof, we do not put proofs into the appendices as before.

[5] Unfortunately the $V$ for $V$ model coincides with the $V$ for valuation. While the reader can distinguish them from context, we mostly discuss V models with extra properties like VC and VCU models, so there should be minimal confusion. We do not simply say $\preceq_w$ is a total order to hint that there are weaker models than V models as investigated e.g. in [Burgess 1981].

We note that standard presentations of V models usually do not contain the family of world-indexed sets of accessible worlds $W_w$ but define it from $\preceq_w$ by $W_w =_{def} \{u : \exists v \in W, u \preceq_w v\}$. We prefer to make this component explicit because it will be useful in the rest of the chapter.

**Definition 5.3** (Satisfaction relation)**.** *Let $M = (W, (W_w)_{w\in W}, (\preceq_w)_{w\in W}, V)$ be a V model and $w \in W$:*

$$\begin{aligned}
(M,w) \models p &\iff p \in V(w);\\
(M,w) \models \neg\varphi &\iff \textit{it is not } (M,w) \models \varphi;\\
(M,w) \models \varphi \wedge \psi &\iff (M,w) \models \varphi \textit{ and } (M,w) \models \psi;\\
(M,w) \models \varphi \boxdot\!\!\rightarrow \psi &\iff \forall v \in W_w, \textit{ if } (M,v) \models \varphi \textit{ then}\\
&\quad \exists u \in W_w \textit{ s.t. } 1)\ u \preceq_w v,\\
&\quad 2)\ (M,u) \models \varphi \wedge \psi, 3)\ \nexists u' \in W_w,\\
&\quad u' \preceq_w u \textit{ and } (M,u') \models \varphi \wedge \neg\psi.
\end{aligned}$$

Satisfiability and validity are defined in the usual way. We write $\models_{\mathbf{V}} \varphi$ if $\varphi$ is valid relative to $\mathbf{V}$, that is, if $(M,w) \models \varphi$ for every $M \in \mathbf{V}$ and every $w$ of $M$.

The satisfaction relation for $\varphi \boxdot\!\!\rightarrow \psi$, complex as it seems, captures the idea of minimal change by means of $\preceq_w$. In virtue of connectedness, it can be simplified as $(\forall v \in W_v, (M,v) \models \neg\varphi)$ *or* $(\exists u \in W_v, (M,u) \models \varphi$ *and* $\forall u' \leq_w u, (M,u') \models \varphi \rightarrow \psi)$. Intuitively, $(M,w) \models \varphi \boxdot\!\!\rightarrow \psi$ means that all the closest $\varphi$-worlds to $w$ make $\psi$ true, with the vacuously true case when $\varphi$ is false everywhere in $W_w$. Actually, if the model satisfies the *limit assumption* (defined below), we have a simpler, equivalent satisfaction relation below in the light of Lewis' famous equivalence result.[6]

**Definition 5.4** (Selection function)**.** *Let $M = (W, (W_w)_{w\in W}, (\preceq_w)_{w\in W}, V)$ be a V model, $w \in W$ and $\varphi \in \mathcal{L}(Atm)$. We define $\sigma_w(\varphi)$, the* selection function *of $w$ regarding $\varphi$, as*

$$\begin{aligned}
\sigma_w(\varphi) =_{def} \{v \in W_w : (M,v) \models \varphi \ \&\ \forall u \in W_w,\ &\textit{if } u \neq v\\
&\textit{and } (M,u) \models \varphi,\ \textit{then } u \not\prec_w v\}.
\end{aligned}$$

**Definition 5.5** (Limit assumption)**.** *Let $M = (W, (W_w)_{w\in W}, (\preceq_w)_{w\in W}, V)$ be a V model. It satisfies the limit assumption, if for all $w$ and $\varphi$, if $\exists v \in W_w$ s.t. $(M,v) \models \varphi$ then $\exists u \in W_w$ s.t. $(M,u) \models \varphi$, and $\forall u' \in W_w$ either $u' \not\prec_w u$ or $(M,u') \models \neg\varphi$.*

**Fact 5.1.** *Let $M = (W, (W_w)_{w\in W}, (\preceq_w)_{w\in W}, V)$ be a V model sharing limit assumption and $w \in W$. Then $(M,w) \models \varphi \boxdot\!\!\rightarrow \psi$ if and only if $\forall v \in \sigma_w(\varphi), (M,v) \models \psi$.*

Since the limit assumption cannot be axiomatized, accepting it or not (Lewis rejected it, unlike most people) actually does not make a substantial difference. However, it echoes in the next section as an interlude.

[6] We ignore the technical issue that $\sigma_w$ shall take as input the semantic proposition $||\varphi||_M =_{def} \{v : (M,v) \models \varphi\}$ instead of the formula $\varphi$.

The V models are too weak to capture most intuitions about counterfactual reasoning. Hence many additional constraints have been considered in the literature. The following are commonly accepted, for every $w$ of the given model,

- **Normality (N)**: $W_w \neq \emptyset$;
- **Total reflexivity (T)**: $w \in W_w$;
- **Weak centering (W)**: $w \in W_w$ and $\forall v \in W_w$, $w \preceq_w v$;
- **Centering (C)**: $w \in W_w$ and $\forall v \in W$, if $v \preceq_w w$ then $v = w$;
- **Uniformity (U)**: $W_w = W$.

Most constraints are self-explained. **Weak centering** says that no other world is closer to the current world than itself (but can be equally close); while **Centering** says that the current world is closer to itself than any other world. The hierarchy of the first four is not hard to see, that each one is stronger than the one above it.

From the metaphysical viewpoint, **Centering** is an almost self-evident assumption. Therefore Lewis takes VC, the logic V in adddition with **Centering**, as his "official logic for counterfactuals".[7] **Uniformity** is an additional constraint desirable for VC "in order to forget the bothersome accessibility restrictions and identify the outer modalities with the logical modalities" [Lewis 1995, p. 130]. That means, $\Box\varphi$ expresses the universal S5 modality. Lewis names the resulting logic VCU.

By contrast, the following constraints are less desirable:

- **Stalnakerian (S) (Conditional excluded middle):** for each $w$ and $\varphi$, either $\sigma_\varphi(w) = \emptyset$ or $|\sigma_\varphi(w)| = 1$;
- **Absoluteness (A)**: $\forall w, v \in W$, $\preceq_w = \preceq_v$.

Lewis calls the first one "Stalnaker's assumption", for [Stalnaker 1968] assumes the selection function associates to every world at most one world (and not a set of worlds as the above functions $\sigma_w$). It is a bit arbitrary to rule out the possibility that two worlds are equally close to the current one, as illustrated by the famous "if Bizet and Verdi had been compatriots, they would be French" example of [Quine 1950].

As for the second one, it is assumed in some papers in the literature e.g. [Kraus *et al.* 1990, Goldszmidt & Pearl 1992] .[Friedman & Halpern 1994] proves that for the complexity of conditional logics "absoluteness makes the problem easier". However, it is such a strong constraint that it becomes unimportant which the indexical/actual world is. Hence [Lewis 1973, p. 131] already says that (to design a logic for counterfactuals) "we surely must reject absoluteness".

[7] Rigorously speaking, it is V plus the characteristic axiom of **Centering**, similar for other cases. We will see that the model and axiomatic characterizations not always coincide in the next section.

**Definition 5.6** (Semantics of subclasses of **V**)**.** *A VX model is a V model satisfying property(ies) X with* $\boldsymbol{X} \subseteq \{\boldsymbol{N}, \boldsymbol{T}, \boldsymbol{W}, \boldsymbol{C}, \boldsymbol{U}, \boldsymbol{S}, \boldsymbol{A}\}$*. The class of VX models is noted* **VX***. Satisfaction relation, satisfiability and validity in each* **VX** *are defined in the same way as in* **V***.*

All the model classes above can be axiomatized in a combinatorial way as the axiomatic of V plus characteristic axioms. But for our main interests we only introduce the axiomatics of VC and VCU.

**Definition 5.7** (Axiomatics of VC and VCU )**.** *The axiomatics of* VCU *is the extension of propositional logic with the following axioms and inference rule. A4 characterizes* ***Weak centering****, A5* ***Centering*** *and A6-7* ***Uniformity****. Hence the axiomatics of* VC *is* VCU *minus A6-7.*

$$\varphi \boxdot\!\!\rightarrow \varphi \tag{A1}$$
$$(\varphi \boxdot\!\!\rightarrow \neg\varphi) \rightarrow (\psi \boxdot\!\!\rightarrow \neg\varphi) \tag{A2}$$
$$\big((\varphi \boxdot\!\!\rightarrow \neg\psi) \vee ((\varphi \wedge \psi) \boxdot\!\!\rightarrow \chi)\big) \leftrightarrow (\varphi \boxdot\!\!\rightarrow (\psi \rightarrow \chi)) \tag{A3}$$
$$(\varphi \boxdot\!\!\rightarrow \psi) \rightarrow (\varphi \rightarrow \psi) \tag{A4}$$
$$(\varphi \wedge \psi) \rightarrow (\varphi \boxdot\!\!\rightarrow \psi) \tag{A5}$$
$$(\varphi \boxdot\!\!\rightarrow \bot) \rightarrow (\neg(\varphi \boxdot\!\!\rightarrow \bot) \boxdot\!\!\rightarrow \bot) \tag{A6}$$
$$\neg(\varphi \boxdot\!\!\rightarrow \bot) \rightarrow ((\varphi \boxdot\!\!\rightarrow \bot) \boxdot\!\!\rightarrow \bot) \tag{A7}$$
$$\frac{(\psi_1 \wedge \cdots \wedge \psi_n) \rightarrow \chi}{\big((\varphi \boxdot\!\!\rightarrow \psi_1) \wedge \ldots (\varphi \boxdot\!\!\rightarrow \psi_n)\big) \rightarrow (\varphi \boxdot\!\!\rightarrow \chi)} \tag{RCK}$$

Table 5.1: Axioms and rule of inference

The last notion to mention is *semantic strength*. Besides comparing two model classes by subset relation, we can say one class is *no weaker than* the other regarding their sets of satisfiable formulas.

**Definition 5.8** (Semantic strength)**.** *Let* $\mathbf{A}, \mathbf{B}$ *be two model classes on the same language. By* $\mathbf{A} \sqsubseteq \mathbf{B}$ *we denote for every* $\varphi$*, if* $\varphi$ *is satisfiable in* $\mathbf{A}$*, then* $\varphi$ *is satisfiable in* $\mathbf{B}$*; by* $\mathbf{A} \sqsubset \mathbf{B}$ *we denote* $\mathbf{A} \sqsubseteq \mathbf{B}$ *but not* $\mathbf{B} \sqsubseteq \mathbf{A}$*; by* $\mathbf{A} \equiv \mathbf{B}$ *we denote both* $\mathbf{A} \sqsubseteq \mathbf{B}$ *and* $\mathbf{B} \sqsubseteq \mathbf{A}$ *and call them* equivalence*.*

Notice that if $\mathbf{A} \subseteq \mathbf{B}$ then $\mathbf{B} \sqsubseteq \mathbf{A}$, but the inverse does not necessarily hold. In particular, possibly $\mathbf{A} \neq \mathbf{B}$ and $\mathbf{A} \equiv \mathbf{B}$.

## 5.3 Hammingian Models for Counterfactuals

### 5.3.1 Hammingian Lewis Models

In a certain sense, Lewis' models are Kripke models plus relations of comparative similarity. A natural question is: closeness (similarity) according to what measure?

As mentioned in literature, the most concrete and almost standard example in the literature is closeness in the sense of the *Hamming distance* between possible worlds.

**Definition 5.9** (Hamming distance between worlds)**.** *Let $W$ be a non-empty set of worlds and $V : W \longrightarrow 2^{Atm}$. For any $w, v \in W$, their Hamming distance under $V$ is defined as $\hbar_V(w, v) = |V(w) \triangle V(v)|$, where $\triangle$ denotes symmetric difference.*

**Definition 5.10** (Hammingian V model)**.** *A V model $M = (W, (W_w)_{w\in W}, (\preceq_w)_{w\in W}, V)$ is Hammingian if $\forall v, u \in W_w$, $v \preceq_w u$ iff $\hbar_V(w, v) \leq \hbar_V(w, u)$. The class of HV models is noted* **HV***. The subclasses of HV models are defined and noted in the similar way as V models.*

Interestingly, the disputation of accepting limit assuption or not does not bother us in **HV**.

**Fact 5.2.** *Every HV model satisfies the limit assumption.*

Although the limit assumption is closely related to well-foundedness, it is not the case that $\preceq_w$ is well-founded. Indeed, let $p_1, p_2, \ldots$ be some enumeration of the atoms of $Atm$ and let $M = (W, (W_w)_{w\in W}, (\preceq_w)_{w\in W}, V)$ be the HVU model where $W = Atm \cup \{p_1, \ldots, p_n : n \in \mathbb{N}\}$ and $V$ is identity. Then $\{p_1\} \succ_{Atm} \{p_1, p_2\} \succ_{Atm} \{p_1, p_2, p_3\} \succ_{Atm} \ldots$ is an infinite descending chain.

A special case are HV models containing all logically possible worlds, i.e., all elements of $2^{Atm}$. This corresponds to the semantics of [Dalal 1988] for database updates. For that semantics, $\Pi^p_2$ completeness of deciding whether $\varphi \rightarrow (\psi \boxdot\!\!\rightarrow \chi)$ was proved in [Eiter & Gottlob 1992], and the validities were axiomatized in [Herzig 1998].

### 5.3.2 Model (Sub)classes: a Comparison

Subset relations between V model subclasses are shown in [Lewis 1973, Figure 5, p. 131], where semantic strength relations are just inverses of the former. We will see that in **HV**, Hamming distance not only determines the comparative similarity, but also "perturbs" the constraints of V models. Consequently, more relations between subclasses of **HV** can be found. Particularly, subset and semantic strength relations no more just inverse. A summary is in Figure 5.1.

**Proposition 5.1.** **HVT** = **HVW**.

*Proof.* Inherited from the V models, **HVT** $\supseteq$ **HVW**. For the other direction, let $M = (W, (W_w)_{w\in W}, (\preceq_w)_{w\in W}, V)$ be an HVT model. Then by the Hamming distance obviously $\forall w \in W, \forall v \in W_w, w \preceq_w v$, i.e. $M$ is weakly centered. □

Similarly, the fact below is easy to see.

**Fact 5.3.** **HVU** $\subset$ **HVW**.

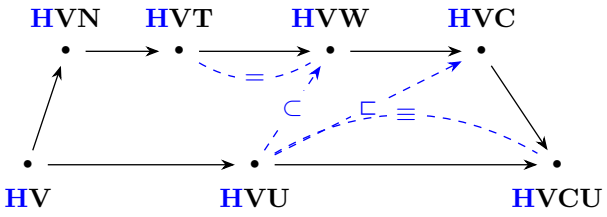



Figure 5.1: Model class relations. Black parts are results of Lewis where arrow means subset relation between model classes; blue parts are new findings of their Hammingian subclasses, where each dash line means the relation by its indicator in $\{=, \equiv, \subset, \sqsubset\}$.

**Fact 5.4** (Indiscernibility)**.** *Let* $M = (W, (W_w)_{w\in W}, (\preceq_w)_{w\in W}, V)$ *be an HVU model. Then* $\forall w, v \in W$*,* $V(w) = V(v)$*, if and only if* $v \preceq_w w$ *if and only if* $\preceq_w = \preceq_v$*.*

**Proposition 5.2.** $\mathbf{HVC} \sqsupset \mathbf{HVU}$*.*

*Proof.* Suppose $M \in \mathbf{HVU}$. First, observe that for every $w \in W$, the closest worlds around $w$ is all those worlds $v$ such that $v \preceq_w w$. Second, by Fact 5.4 $v \preceq_w w$ implies that $V(v) = V(w)$ and $\preceq_v = \preceq_w$. Third, we define a relation $Z \subseteq W \times W$ by: $wZv$ iff $V(w) = V(v)$. Thanks to the second observation, it is not hard to see that $Z$ is a bisimulation. It follows that all closest worlds around $w$ satisfy the same formulas. Hence the centering axiom is valid in **HVU** models. □

Noticing that $\mathbf{HVCU} \subset \mathbf{HVU}$, the following result becomes obvious.

**Fact 5.5.** $\mathbf{HVU} \equiv \mathbf{HVCU}$*.*

### 5.3.3 Hamming State Models

One can argue conceptually that Hamming distance commits us to identify a world with its valuation, or even stronger, that the real model shall be defined by valuations of variables rather than more abstract entities, namely worlds. Formally we can define the following.

**Definition 5.11** (Hamming state model)**.** *We call a model* $\mathcal{S} = (S, (S_s)_{s\in S})$ a Hamming state model with parameters[8] *if* $S \subseteq 2^{Atm}$ *and* $\forall s \in S, S_s \subseteq S$*. If* $S_s = S$ *for each* $s \in S$*, we just call it a* Hamming state model*.*

The use of the Hamming distance is justified by Leibniz's law in [Floridi 2010]. For any $w, v$ in a model, they are identical if $w = v$; equivalent if $V(w) = V(v)$; indiscernible if $w \approx_u v$ for every $u$ in the model. Fact 5.4 states the indiscernibility of equivalences in **HVU**. **Centering** states a stronger property: the identity of equivalences in **HVU** (but with a restriction to accessible worlds). Thus we can

[8] We call this one "Hamming", while the adjective "Hammingian" qualifies HV models.

examine the philosophy from a more logical viewpoint in terms of bisimilarity and isomorphism.

**Fact 5.6.** *Let $M = (W, (W_w)_{w\in W}, (\preceq_w)_{w\in W}, V)$ be an HVC model. Then $\forall v \in W_w$, $V(w) = V(v)$ if and only if $w = v$. Particularly, if $M$ is an HVCU model then $\forall w, v \in W, V(w) = V(v)$ if and only if $w = v$.*

**Proposition 5.3.** *Every HVU model is bisimilar to a Hamming state model with parameters; every HVCU model is isomorphic to a Hamming state model.*

*Proof.* Let $M = (W, (W_w)_{w\in W}, (\preceq_w)_{w\in W}, V)$ be an HVU model. For each $w \in W$, let $s_w$ denote $V(w)$. Consider $\mathcal{S} = (S, (S_s)_{s\in S})$ s.t. $S = \{s_w : w \in W\}$ and $\forall s_w \in S, s_v \in S_{s_w}$ iff $v \in W_w$. The bisimulation between the two models is obvious. The same construction applies to HVCU but the result is stronger because of Fact 5.6. □

The above result for HVCU does not hold for weaker logics: for every $\mathbf{X} \in \{\mathbf{N}, \mathbf{T}, \mathbf{W}, \mathbf{C}\}$ there is a model in **HVX** that is not bisimular to any Hamming state model with parameters. To see that, simply consider an HVX model with two worlds $w, v$, s.t. $V(w) = V(v)$, $W_w = \{w\}$ and $W_v = \{v\}$.

Despite the isomorphism between HVCU and state models, we keep using possible worlds semantics in line with other HV models until Section 6 when state models debut.

## 5.4 Equivalence Results Given Infinite Atoms

Now that the stage is set, let us raise the bold question: is the Hamming distance not just an example, but *the* grounded measure of distance for VC and VCU? Grounded in the intuitive sense that, given an infinite supply of atoms, we can transform any non-Hammingian model to a Hammingian one while preserving the truth of some formula. Formally the question is put as the following theses of equivalences.

**Thesis 1.** **VC** $\equiv$ **HVC**.

**Thesis 2.** **VCU** $\equiv$ **HVCU**.

We describe below the strategy of our proof, so that the basic line of thought is transparent from the beginning.

**Proof strategy** Not all $VC$ models can be Hammingized by simply manipulating their valuations (no need to say preserve the truth of some $\varphi$), but any $VC$ model which has some tree structure can. Moreover, [Friedman & Halpern 1994] offered a tree construction from some $VC$ model while preserving the truth of some formula $\varphi$. Hence, we aim to Hammingize the Friedman-Halpern tree $VC$ model while not affecting the truth of $\varphi$. To divide the proof into steps and conquer them separately, we will show that if $\varphi$ is satisfiable in **VC** then it is satisfied in a pointed VC model $(M, w_0)$, where $M$ is a Friedman-Halpern tree model, which fulfills the following missions:

1. HAMMINGIANIZATION: $M$ induces an HVC model $M'$;

2. TRUTH-PRESERVATION: $(M', w_0) \models \varphi$.

The strategy for $VCU$ is the same but need one further treatment.

### 5.4.1 A Failed Attempt

Let us start with an easy but failed attempt.

A simple thought for Mission 1, Hammingianization, is to keep the worlds and their similarity relations, and *only* manipulate the valuation on $Atm \setminus atm(\varphi)$, resulting in a new valuation $V'$ so that the Hamming distance by $V'$ is in accordance with the similarity relations in the original model. We may call such a VC model "substantially Hammingian".[9]

**Definition 5.12** (Substantial Hammingianness)**.** *Let $M = (W, (W_w)_{w\in W}, (\preceq_w)_{w\in W}, V)$ be a VC model. We call $M$* substantially Hammingian *if there is a valuation $V'$, s.t. $M' = (W, (W_w)_{w\in W}, (\preceq_w)_{w\in W}, V')$ is a Hammingian model.*

Naturally this leads us to the following question: Are all VC models substantially Hammingian? The answer is, however, negative, especially when a VC model has a vicious circle.

**Definition 5.13** (Vicious circle)**.** *Let $M = (W, (W_w)_{w\in W}, (\preceq_w)_{w\in W}, V)$ be a VC model. We say that $M$ has a vicious counterfactual circle, if $\exists w_0, w_1, \ldots, w_n \in W$ s.t. $w_0 \preceq_{w_1} w_2 \ldots$, and $w_{n-2} \preceq_{w_{n-1}} w_n$, and $w_{n-1} \preceq_{w_n} w_0$, but $w_n \prec_{w_0} w_1$.*

Then we have the following impossibility result.

**Proposition 5.4.** *Any VC model that has a vicious circle is not substantially Hammingian.*

*Proof.* We show the case when the circle consists of 3 worlds; the other cases are similar. Let $M = (W, (W_w)_{w\in W}, (\preceq_w)_{w\in W}, V)$ be a VC model and $w, v, u \in W$ form a vicious circle. For whatever $V'$, we should have $\hbar_{V'}(w, v) \leq \hbar_{V'}(v, u) \leq \hbar_{V'}(u, w) < \hbar_{V'}(w, v)$, by the definition of Hammingian model and using the condition of the circle. But $\hbar_{V'}(w, v) = \hbar_{V'}(v, w)$, a contradiction. □

The same definitions and same result, as its proof indicates, apply to VCU models as well.

**Example 5.1.** *Let $\varphi^\dagger$ be the formula $\neg p_1 \wedge (p_1 \boxdot\!\!\rightarrow (\neg p_2 \wedge (p_2 \boxdot\!\!\rightarrow (\neg p_3 \wedge (p_3 \boxdot\!\!\rightarrow p_1))))$. Let a VCU model $M = (W, (W_w)_{w\in W}, (\preceq_w)_{w\in W}, V)$ be s.t. $W = \{w, v, u\}$, $V(w) = \{p_2, p_3\}, V(v) = \{p_1, p_3\}, V(u) = \{p_1, p_2\}$, and $v \prec_w u, u \prec_v w, w \prec_u v$. This is depicted in Figure 5.2. Then $(M, w) \models \varphi^\dagger$, but $M$ is not substantially Hammingian.*

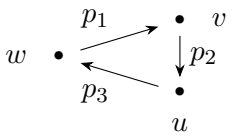



Figure 5.2: The vicious circle in Example 5.1. Arrows denote the relevant selection functions. E.g., $\sigma_w(p_1) = \{v\}$ due to the fact that $v \prec_w u$, though $p_3$ is in both $V(v)$ and $V(u)$.

Actually this is a difficulty not only to Hamming distance, but any *total order on pairs of worlds* which intends to extend the sets of *triple relations on worlds*. We can take advantage of the study in [Williamson 1988], which helps us prove the following proposition.

**Proposition 5.5.** *Let $M = (W, (W_w)_{w\in W}, (\preceq_w)_{w\in W}, V)$ be a VCU model. It has no vicious circle, if and only if there exists a total order $\leq$ on $W^2$ s.t. if $v \preceq_w u$ then $(v, w) \leq (w, u)$.*

*Proof sketch.* Necessity is shown by Example 5.1. For sufficiency, we need constitute a total ordering $\leq$ on $W^2$. First, notice that every $\preceq_w$ can be seen as a partial ordering on $W^2$ by stipulating $\forall v \in W_w, u \notin W_w, v \prec_w u$. Hence we union all of them to obtain a partial ordering on $W^2$, noted $\preceq$. No vicious circle ensures the union. The second step is taking transitive closure of it, noted $\preceq'$. The next step is taking the quotient w.r.t. equivalence relation of $\preceq'$, i.e. the relation on the equivalent classes of $W^2$ with respect to $\preceq'$, noted $\preceq''$. Last, we can use the famous theorem in order theory that every partial order can be extended to a total order. □

### 5.4.2 Weighted Tree is Hammingian

What can we learn from the former failure? The lesson is that without further constraint on the original VC or VCU model, the ternary relations may conflict with each other.

The last proposition indicates that the necessary condition of being a Hammingian model is no vicious circle. So the tree structure appears as a natural choice.

The intuition illustrated in Figure 5.3 is that if the model associates with a tree structure, and moreoever the tree is weighted, then it can be Hammingized by adding the weights of edges of the path between any two vertices (worlds). But before formalizing the intuition, let us make two remarks.

1) We beg patience at this stage about where the mysterious tree structure of a model comes from. It will be clear in the next steps.

[9]Let us distinguish "substantially" and "potentially" Hammingian. The former only needs to manipulate $V$ to become Hammingian; while the later may copy worlds to unravel the vicious circle, as we will do. Actually all VC models are potentially Hammingianizable by first transforming to substantially Hammingian ones.

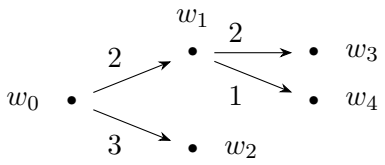



Figure 5.3: Given such a weighted tree, one can build an HV model as follows: let $W$ consist of the five vertices, and take a $V$ s.t. $\forall w_i, w_j, \hbar_V(w_i, w_j) = n$ if $\pi(w_i, w_j)$ is weighted $n$. Restricting $W_{w_i} = W$ for all $w_i$ the model is HVCU, otherwise HVC. But of course, we cannot guarantee any truth-preservation at this stage.

2) We recall the basic notions of graph theory. For any two points (vertices) $w, v$, $(w, v)$ denotes the undirected edge between $w$ and $v$. A path between $w$ and $v$ is a sequence of vertices $(w_1, \ldots, w_n)$ s.t. $w = w_1, v = w_n$ and $(w_i, w_{i+1})$ is an edge for $1 \leq i < n$. A tree is an undirected graph where each two vertices $w$ and $u$ have exactly one path, denoted by $\pi(w, v)$. We write $(w_i, w_j) \in \pi(w, v)$ if $(w_i, w_j)$ is a member of the sequence.

A weighted tree is a triple $G^{\#} = (W, E, \#)$ where $G = (W, E)$ is a tree (with $E \subseteq W \times W$) and $\# : E \longrightarrow \mathbb{N}$. The weight of a path $\pi(w, v)$ in $G^{\#}$ is $\#\pi(w, v) =_{def} \sum_{(w_i, w_j) \in \pi(w,v)} \#(w_i, w_j)$.

**Definition 5.14** (Weighted tree VC model)**.** *Let $M = (W, (W_w)_{w \in W}, (\preceq_w)_{w \in W}, V)$ be a VC model, for which there exists an associated weighted tree $G^{\#} = (W, E, \#)$, s.t. $\forall w \in W$, $\forall v, u \in W_w$, $v \preceq_w u \iff \#\pi(v, w) \leq \#\pi(w, u)$.*

**Lemma 5.1.** *Let $M = (W, (\preceq_w)_{w \in W}, V)$ be a finite VC model associated with a weighted tree $G^{\#}$. Then, there is an HVC model $M' = (W, (W_w)_{w \in W}, (\preceq_w)_{w \in W}, V')$, s.t. $\forall w, v \in W$, $\hbar_{V'}(w, v) = 2 \times \#\pi(w, v)$, and $\forall w \in W$, $|V(w)|$ is finite.*

*Proof.* Since no formula need be truth-preserved here, we construct $V'$ ignoring $V$. We take a series of disjoint unions $X_1 \cup Y_1 \cup X_2 \cup Y_2 \cup \ldots X_{|E|} \cup Y_{|E|} \subset Atm$ and enumerate $E$ as $e_1, e_2, \ldots, e_{|E|}$, s.t. $|X_i| = |Y_i| = \#(e_i)$ for all $1 \leq i \leq |E|$. For every $p$ that is not in those disjoint unions, let $p \notin V'(w)$ for all $w$. The construction of $V'$ says for all $e_i = (w_j, w_k) \in E$, if $\pi(w_0, w_j) \subset \pi(w_0, w_k)$, viz. $w_j$ is nearer to $w_0$ than $w_k$, then let $V'(w_j) \cap (X_i \cup Y_i) = X_i$ and $V'(w_k) \cap (X_i \cup Y_i) = Y_i$; for all $e_l$ not linking $w_j$, simply let $V(w_j) \cap (X_l \cup Y_l) = \emptyset$. Thus, for any $V'(w), V'(v)$ they differ on $2 \times \#\pi(w, v)$ many variables, which makes the desired $\hbar_{V'}(w, v) = 2 \times \#\pi(w, v)$. $|V(w)|$ is finite because $G^{\#}$ has finitely many edges with finite weights. □

The proposition below directly follows from the lemma.

**Proposition 5.6.** *All weighted tree VC models are substantially Hammingian.*

### 5.4.3 VC ≡ HVC

Before exhausting the reader's patience, we now reveal where the tree comes from: *it is constructed according to subformulas in the formula $\varphi$ of interest.* The tree

construction is described in [Friedman & Halpern 1994].

**Proposition 5.7** ([Friedman & Halpern 1994])**.** *If $\varphi$ is satisfiable in* **VC***, then $\varphi$ is satisfiable in some tree VC model.*

The proof relies on a series of lemmas to construct such a tree, which we shall call the *FH tree* after the authors. For the sake of both self-containedness and simplicity, we rephrase how the tree is constructed.

**Friedman-Halpern tree for VC model** The first key notion is $basic_i(\varphi) \subseteq atm(\varphi) \cup sub_{\boxrightarrow}(\varphi)$ where $sub_{\boxrightarrow}(\varphi)$ denotes the subformulas of $\varphi$ whose principal connective is $\boxrightarrow$. Intuitively, $basic_i(\varphi)$ is defined as the union of all atoms in $\varphi$ and counterfactuals in *exactly* the $i$-th level of the nesting of $\varphi$. A formal definition is:

$$
\begin{aligned}
basic_i(p) &= \begin{cases} \{p\} & \text{if } i = 0, \\ \emptyset & \text{otherwise;} \end{cases} \\
basic_i(\neg\varphi) &= basic_i(\varphi); \\
basic_i(\varphi \wedge \psi) &= basic_i(\varphi) \cup basic_i(\psi); \\
basic_i(\varphi \boxrightarrow \psi) &= \begin{cases} \{\varphi \boxrightarrow \psi\} & \text{if } i = 0, \\ basic_{i-1}(\varphi) \cup basic_{i-1}(\psi) & \text{otherwise.} \end{cases}
\end{aligned}
$$

Take $\varphi^\dagger = \neg p_1 \wedge (p_1 \boxrightarrow (\neg p_2 \wedge (p_2 \boxrightarrow (\neg p_3 \wedge (p_3 \boxrightarrow p_1)))))$ from Example 5.1, then $basic_0(\varphi^\dagger) = \{p_1, p_1 \boxrightarrow (\neg p_2 \wedge (p_2 \boxrightarrow (\neg p_3 \wedge (p_3 \boxrightarrow p_1))))\}$, $basic_1(\varphi^\dagger) = \{p_1, p_2, p_2 \boxrightarrow (\neg p_3 \wedge (p_3 \boxrightarrow p_1))\}$, $basic_2(\varphi^\dagger) = \{p_2, p_3, p_3 \boxrightarrow p_1\}$ and $basic_3(\varphi^\dagger) = \{p_1, p_3\}$.

We describe an FH tree given a finite VC model $M = (W, (W_w)_{w\in W}, (\preceq_w)_{w\in W}, V)$ and a formula $\varphi$ s.t. $(M, w_0) \models \varphi$. The tree iteratively "chooses" worlds in $W$ as vertices according to vertices and formulas at the previous level. The root is $w_0$. Since the function of choosing is not necessarily injective, for any vertex $v$ we write $v^{-1}$ for the chosen world in $M$. But we only write $w_0$ for simplicity. Level 0 has only the root $w_0$. At level 1, for any $\xi \boxrightarrow \theta \in basic_0(\varphi)$ there is a vertex named as $w_{0,\xi\boxrightarrow\theta}$. And $w^{-1}_{0,\xi\boxrightarrow\theta}$ was chosen from $M$ with the following constraints:

1. $w^{-1}_{0,\xi\boxrightarrow\theta} \in \sigma_{w_0}(\xi \boxrightarrow \theta)$, if $(M, w_0) \models \xi \boxrightarrow \theta$ and $\sigma_{w_0}(\xi \boxrightarrow \theta) \neq \emptyset$;
2. $w^{-1}_{0,\xi\boxrightarrow\theta}$ is $w_0$, if $\sigma_{w_0}(\xi \boxrightarrow \theta) = \emptyset$;
3. $w^{-1}_{0,\xi\boxrightarrow\theta} \in \sigma_{w_0}(\xi \boxrightarrow \theta)$ and $(M, v) \models \neg\theta$, if $(M, w_0) \models \neg(\xi \boxrightarrow \theta)$.

Notice that when $(M, w_0) \models \xi$, $w^{-1}_{0,\xi\boxrightarrow\theta}$ has to be $w_0$. Naturally, for every such vertex, we draw an edge between it and $w_0$ to obtain a (sub)tree. Then define a model $M^{w_0} = (W^{w_0}, (W^{w_0}_v)_{v\in W_0}, (\preceq^{w_0}_w)_{w\in W_{w_0}}, V^{w_0})$ s.t. $v \in W^{w_0}$ if $(w_0, v)$ is an edge; $V^{w_0}(v) = V(v^{-1})$. Then, we simply put $W^{w_0}_v = \emptyset$ if $v \neq w_0$. Let $\preceq^{w_0}_{w_0}$ be a total order on $W^{w_0}$ s.t. **Centering** is satisfied and $\forall w_{0,\xi\boxrightarrow\theta}, w_{0,\xi'\boxrightarrow\theta'} \in W^{w_0}$, $w_{0,\xi\boxrightarrow\theta} \preceq^{w_0}_{w_0} w_{0,\xi'\boxrightarrow\theta'}$ iff $(M, w_0) \models \xi \vee \xi' \boxrightarrow \xi$.

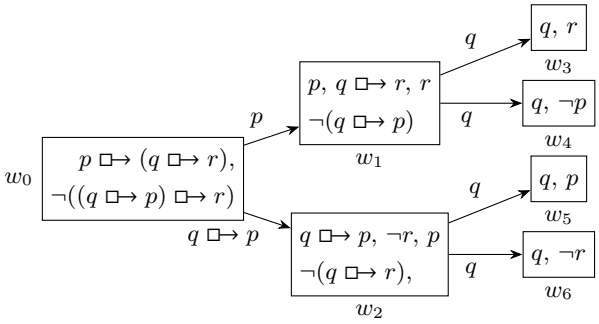



Figure 5.4: Example for $(p \boxdot\!\!\rightarrow (q \boxdot\!\!\rightarrow r) \wedge \neg((q \boxdot\!\!\rightarrow q) \boxdot\!\!\rightarrow r)$

Now let $v$ be a vertex at level 1. We define $\varphi_v := \bigwedge_{\psi \in basic_1(\varphi),(M,v^{-1}) \models \psi} \psi \wedge \bigwedge_{\psi \in basic_1(\varphi),(M,v^{-1}) \models \neg\psi} \neg\psi$. We recursively apply the procedure on $(M, v^{-1})$ and $\varphi_v$ to obtain a subtree and a submodel. Since $sub_{\boxdot\!\!\rightarrow}(\varphi_v) \subseteq sub_{\boxdot\!\!\rightarrow}(\varphi), basic_i(\varphi_v) \subseteq basic_{i+1}(\varphi)$, the construction terminates. Finally, we union all of them to obtain the FH tree and its associated VC model noted $M^t$. Figure 5.4 illustrates a tree truth-preserving $\varphi$ relative to $\mathsf{VC}$.[10]

**Remark & convention** For readability we save the recursively defined function for the "standard name" of worlds in $M^t$, which takes the form $w_{*,\xi \boxdot\!\!\rightarrow \theta}$ where $\xi \boxdot\!\!\rightarrow \theta \in basic_i(\varphi)$ for some $i$ and $w_*$ is the name of a world at level $i$. We may enumerate these names and hence $w_k = w_{j,\psi}$ if $w_k$ exists w.r.t. some world $w_j$ and formula $\xi \boxdot\!\!\rightarrow \theta$. Notice that while $w_j \neq w_k$ for $j \neq k$, it is possible that $w_j^{-1} = w_k^{-1}$.

**Example 5.2.** *Figure 5.4 is the graph $G$ of the tree model $M^t$ constructed from some finite VC model $M$ and $(p \boxdot\!\!\rightarrow (q \boxdot\!\!\rightarrow r) \wedge \neg((q \boxdot\!\!\rightarrow q) \boxdot\!\!\rightarrow r)$. The formula attached to every edge denotes the member of $basic_i(\varphi)$ making the target vertex exist. For example, the arrow with $p$ means that $w_1$ is chosen from $\sigma_{w_0}(p \boxdot\!\!\rightarrow (q \boxdot\!\!\rightarrow r))$ during the tree construction. Formulas in each world $v$ are the conjuncts of $\varphi_v$ as defined in the proof of Proposition 5.7. Notice, e.g. though $w_4$ exists for sake of $\neg(q \boxdot\!\!\rightarrow p)$, we also know $(M^t, w_4) \models r$, because $w_4$ is chosen from $M$ as $w_4^{-1} \in \sigma_{w_1^{-1}}(q)$. Since $(M, w_1^{-1}) \models q \boxdot\!\!\rightarrow r$, it must be $r \in V^t(w_3)$.*

**Lemma 5.2.** *Let $(M, w_0) \models \varphi$ where $M$ is a finite VC model. Then there is an FH tree VC model $M^t$ built from $(M, w_0)$ and $\varphi$, s.t. $(M^t, w_0) \models \varphi$.*

**Hammingize the tree VC model** Are we done? Almost yet not. We know that FH tree construction is truth-preserving, and a weighted tree is substantially Hammingian. It remains to Hammingize the weighted FH tree in order to turn Thesis 1 into a theorem.

[10] We defined our tree as undirected in accordance with the semantics of $\preceq_w$. In fact, the tree in the construction above is better understood as directed in accordance with the semantics of $\sigma_w$. However, since it causes minor problems of understanding, we do not add more definitions to increase the opaqueness.

**Theorem 5.1.** *Let Atm be infinite. Then* **VC** $\equiv$ **HVC**.

*Proof.* **HVC** $\subset$ **VC**, so only need prove the only-if-part. For any $\varphi$ satisfiable in **VC**, using the filtration result of [Segerberg 1989], there is a finite model $M^f = (W^f, (W^f_w)_{w\in W^f}, (\preceq^f_w)_{w\in W^f}, V^f)$ s.t. $(M^f, w_0) \models \varphi$, and particularly $\bigcup_{v\in W^f} V^f(v) \subseteq atm(\varphi)$. That is, no world in $W^f$ verifies any variable outside of $atm(\varphi)$. We build an FH tree model $M = (W, (W_w)_{w\in W}, (\preceq_w)_{w\in W}, V)$ from $(M^f, w_0)$ and $\varphi$. By Lemma 5.2, $(M, w_0) \models \varphi$. A weighted tree $G^{\#} = (W, E, \#)$ is defined s.t. $E = \{(w_j, w_{j,\xi\Box\to\theta}) : w_j, w_{j,\xi\Box\to\theta} \in W)\}$, and $\forall w, v, u \in W$, if $v \preceq_w u$ then $\#(v, w) \leq \#(w, u)$. We obtain $V'$ by assembling three valuations. The first one is some $V^h$ which enables Hammingization in Lemma 5.1, with $\bigcup_{v\in W} V^h(v) \cap atm(\varphi) = \emptyset$. The second one is $V$, because we want $V'(w) \cap atm(\varphi) = V(w) \cap atm(\varphi)$ for truth-preserving $\varphi$. But now the Hamming distance perturbs, which needs a third one $V^b$ to "counterbalance" it. Let $w^\circ, w_\circ$ be two worlds that differ at most on atoms in $atm(\varphi)$, say, $\hbar_V(w^\circ, w_\circ) = n$. Let $\bigcup_{v\in W} V^b(v) = X$ be disjoint from $atm(\varphi)$ and $\bigcup_{v\in W} V^h(v)$ with $|X| = n$. Then we enumerate $V(w^\circ)\triangle V(w_\circ)$ as $p_1, p_2, \ldots p_n$, and $X$ as $q_1, q_2, \ldots q_n$,

1. let $V^b(w^\circ) \cap X = X$ and $V^b(w_\circ) \cap X = \emptyset$;
2. $\forall v \in W$, $\forall p_i \in atm(\varphi)$, $p_i \in V(v) \cap V(w^\circ)$ if and only if $q_i \notin V^b(v)$.

This step guarantees that $\forall w, v \in W, |V^b(w) \cap (X \cup atm(\varphi))| = |V^b(v) \cap (X \cup atm(\varphi))|$. Namely $w$ and $v$ are "numerically equal" regarding $atm(\varphi) \cup X$, so that $V^h$ can do its job right. Finally we let $\forall w \in W, V'(w) = V(w) \cup V^b(w) \cup V^h(w)$. Obviously, $M'$ is Hammingian and still $(M', w_0) \models \varphi$. □

### 5.4.4 VCU $\equiv$ HVCU

We did not apply the method above directly to VCU, because of an apparent shortcoming and a potential danger. 1) The tree VC model is not uniform but "local": each $W_v$ contains only $v$ and its adjacents. We can of course extend $\preceq_v$ to obtain **Uniformity** by the information of the weighted FH tree. But then 2) one may suppose that $(M_v, v) \models \neg\xi \Box\!\!\to \bot$ holds vacuously, i.e. $\sigma^v_v(\neg\xi) = \emptyset$, but after the extending possibly in $M$, $\sigma_v(\neg\xi) \neq \emptyset$. Hence finally $(M, v) \models \neg(\neg\xi \Box\!\!\to \bot)$.[11]

We are going to show that refining the tree construction in a certain way, not only the shortage is overcome, the potential danger is actually no danger. The key fact is that, though every $M_{w_j}$ is constructed "shortsightedly", the original finite model this time satisfies **Uniformity**. Hence if some $\xi \Box\!\!\to \bot$ is vacuously true in the tree model, that must be already vacuously true in the original model.

Instead of first presenting the refined FH tree for a VCU model and then Hammingianize it, for simplicity we do the two steps simultaneously.

[11] [Friedman & Halpern 1994] mentioned a similar concern and hinted how to resolve it, and the solution will in general raise the complexity to EXPTIME. They claimed to give the full details in the full paper. According to personal communication, no full paper.

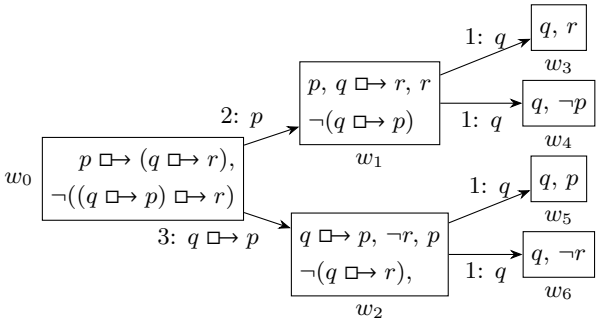



Figure 5.5: A weighted tree $G^{\#}$ for $G$ in Figure 5.4

**Forward-weighted tree for HVCU model** Let the FH tree construction remain the same, but the input model is VCU instead of VC. The output model is (still) a tree VC model. Now our goal is an HVCU model, where **Uniformity** is obtained naturally by generalizing all $W^t_w$ to the whole $W^t$; and the information of the weighted tree truth-preserves certain formulas.

To this end we need ensure the distance between worlds to go with the "direction" of the tree construction, so that for each $v$, its sets of closest worlds regarding $basic_0(\varphi_v)$ remain invariant. Thus we need the wanted weighted tree to have a particular global property defined as following.

**Definition 5.15** (Forward-weighted tree)**.** *Let $M$ be a tree VC model associated with a weighted tree $G^{\#} = (W, E, \#)$ and $w_0$ be the root. We call $G^{\#}$* forward-weighted*, if $\forall w, v, u$, if $\pi(w_0, w) \subset \pi(w_0, v) \subset \pi(w_0, u)$, then $\#\pi(w, v) > \#(v, u)$.*

In plain words, the farther we go from the root, the smaller weights we assign the edges. If in the graph $(w_0, w, v, u)$ forms a path, then $\#(w, v) > \#(v, u)$. So when we search the closest worlds of $v$ regarding some $\xi$, we will not "go back" to $w$. This is the intuition where the term comes from.

It is not hard to see, similar to what we did in the last subsection, that a forward-weighted tree associating a VCU model induces an HVCU model. In particular, we have the following lemma, which is proven similar to Theorem 5.1.

**Example 5.3.** *Figure 5.5 hammingizes the $G$ in Figure 5.4 as $G^{\#}$. For example, the edge labelled 2:$p$ is because we want $w_1 \in \sigma_{w_0}(p)$. Notice if the edge with $q \boxdot\!\!\rightarrow p$ weighted as 2, then we would also have $w_2 \in \sigma_{w_0}(p)$ since $(M, w_2) \models p$, which would make $(M, w_0) \not\models p \boxdot\!\!\rightarrow (q \boxdot\!\!\rightarrow r)$, since $(M, w_2) \models \neg(q \boxdot\!\!\rightarrow r)$. Also, if $(M, w_0) \models q \wedge \neg r$ and we weighted the edge of $r \boxdot\!\!\rightarrow q$ as 1, then we would have $(M, w_2) \not\models q \boxdot\!\!\rightarrow r$, since $w_0 \in \sigma_{w_2}(q)$, and we "go back". Finally, construct an HV model $M'$ via $G^{\#}$ instructed by Theorem 5.1, s.t. for its $V'$:*

- $\hbar_{V'}(i, j) = 2n$ *iff* $\#\pi(w_i, w_j) = n$;
- $\forall p_k \in \{p, q, r\}, p_k \in V'(w_i)$ *iff* $p_k \in V(w_i)$.

**Lemma 5.3.** *Let $M^t = (W^t, (W^t_w)_{w\in W^t}, (\preceq^t_w)_{w\in W}, V)$ be an FH tree VC model constructed from some finite VCU model and $\varphi$. Then there exists a forward-weighted tree $G^{\#}$ of $M$, which induces an HVCU model $M' = (W^t, (W'_w)_{w\in W}, (\preceq'_w)_{w\in W}, V')$, s.t. $\forall w \in W$, $W'_w = W^t$ and $V(w) \cap atm(\varphi) = V'(w) \cap atm(\varphi)$.*

**Key lemma** We have enabled FH tree VC model with **Uniformity** and Hammingized it to obtain an HVCU model. We aim to show a key property. But before, let $sub_{\Box}(\varphi)$ denote all subformulas of $\varphi$ of the form $\chi \Box\!\!\rightarrow \bot$. Clearly but crucially, $\forall\psi \in sub_{\Box}(\varphi), \psi \in basic_i(\varphi)$ for some $i$.

**Lemma 5.4.** *Let $M = (W, (W_w)_{w\in W}, (\preceq_w)_{w\in W}, V)$ be a finite VCU model, $w_0 \in W$, $\varphi \in \mathcal{L}(Atm)$ s.t. $(M, w_0) \models \varphi$. Let $M' = (W^t, (W^t_w)_{w\in W^t}, (\preceq'_w)_{w\in W}, V')$ be an HVCU model constructed through FH tree and Lemma 5.1. Then $(M, w_0) \models \varphi$ if and only if $(M', w_0) \models \varphi$.*

*Proof.* Inherited from the FH tree for VC models, we have $\forall w_j \in W^t$ chosen by the tree construction at level $i$, $\forall\psi \in basic_i(\varphi)$, we have $(M, w_j^{-1}) \models \psi$ if and only if $(M', w_j) \models \psi$.

So the only concern is when $\psi \in sub(\varphi)$ is some $\chi \Box\!\!\rightarrow \chi'$ such that either $\chi$ is vacuously true at $(M, w_j^{-1})$ but not vacuously true at $(M', w_j)$; or the other way around. Notice $\psi$ may not be at the same level as $w_j$ is chosen, but crucially it must be $\psi \in basic_k(\varphi)$ for some $k$. For convenience instead of saying vacuously true or $\sigma_{w_j^{-1}}(\chi) = \emptyset$ we write $(M, w_j^{-1}) \models \chi \Box\!\!\rightarrow \bot$. We do induction on the conditional degree of $\chi$.

The induction basis is $cd(\chi) = 0$, viz. $\chi$ is Boolean. If $(M, w_j^{-1}) \models \chi \Box\!\!\rightarrow \bot$, then $\forall v \in W^t$ we have $(M, v^{-1}) \models \neg\chi$. Since $\chi$ is Boolean we have $(M', v) \models \neg\chi$, hence $(M', w_j) \models \chi \Box\!\!\rightarrow \bot$. For the other direction let $(M', w_j) \models \chi \Box\!\!\rightarrow \bot$, and we need show $(M, w_j^{-1}) \models \chi \Box\!\!\rightarrow \bot$. Notice, crucially, that $\chi \Box\!\!\rightarrow \chi'$ must occur in $basic_k(\varphi)$ for some $k$. At level $k$ we must choose a world $w_l$ to decide whether $\chi \Box\!\!\rightarrow \chi'$ holds at $(M', w_l)$ according to what happens at $(M, w_l^{-1})$. But whatever $w_l^{-1} \in W$ is, it must be $(M, w_l^{-1}) \models \chi \Box\!\!\rightarrow \bot$, otherwise at level $k+1$ we would have chosen a $w_{l,\chi\Box\!\!\rightarrow\chi'}$ s.t. $(M', w_{l,\chi\Box\!\!\rightarrow\chi'}) \models \chi$ , which eventually made $(M', w_j) \models \neg(\chi \Box\!\!\rightarrow \bot)$, a contradiction. This indicates $\forall v \in W, (M, v) \models \neg\chi$. Since $\chi$ is Boolean, $\forall w \in W^t, (M, w) \models \neg\chi$, viz. $(M, w_j^{-1}) \models \chi \Box\!\!\rightarrow \bot$ as we want.

Now we run the induction. Suppose for any subformula of conditional degree $n$, it is true at $(M, v^{-1})$ if and only if true at $(M', v)$ for all $v \in W^t$. Now we consider $\chi$ with $cd(\chi) = n+1$ and show $(M, w_j^{-1}) \models \chi \Box\!\!\rightarrow \bot$ iff $(M', w_j) \models \chi \Box\!\!\rightarrow \bot$. It needs a further induction on the main connective of $\chi$.

1) The case of conjunction is straightforward. 2) If $\chi$ has the form $\xi \Box\!\!\rightarrow \theta$ and $(M, w_j^{-1}) \models (\xi \Box\!\!\rightarrow \theta) \Box\!\!\rightarrow \bot$, then suppose towards a contradiction that $\exists v \in W^t$ s.t. $(M', v) \models \xi \Box\!\!\rightarrow \theta$. By induction hypothesis we have $(M, v^{-1}) \models \xi \Box\!\!\rightarrow \theta$, which contradicts $(M, w_j^{-1}) \models (\xi \Box\!\!\rightarrow \theta) \Box\!\!\rightarrow \bot$. For the other direction suppose towards a contradiction that $(M', w_j) \models (\xi \Box\!\!\rightarrow \theta) \Box\!\!\rightarrow \bot$ but $(M, w_j^{-1}) \models \neg((\xi \Box\!\!\rightarrow \theta) \Box\!\!\rightarrow \bot)$. Notice, crucially, that $(\xi \Box\!\!\rightarrow \theta) \Box\!\!\rightarrow \chi'$ occurs in $basic_k(\varphi)$ for some

$k$. Thus at level $k$ of the tree construction there was a $w_l \in W^t$ which chose a $w_{l,(\xi \boxdot\!\!\rightarrow \theta) \boxdot\!\!\rightarrow \chi'}$ from $W$ for the level $k+1$ according to whether $(\xi \boxdot\!\!\rightarrow \theta) \boxdot\!\!\rightarrow \chi'$ holds at $(M, w_l^{-1})$. It must be $(M, w_l^{-1}) \models \neg((\xi \boxdot\!\!\rightarrow \theta) \boxdot\!\!\rightarrow \bot)$ because of the supposition $(M, w_j^{-1}) \models \neg((\xi \boxdot\!\!\rightarrow \theta) \boxdot\!\!\rightarrow \bot)$. Thus the tree construction chose a $w_{l,(\xi \boxdot\!\!\rightarrow \theta) \boxdot\!\!\rightarrow \chi'}$ s.t. $(M, w^{-1}_{l,(\xi \boxdot\!\!\rightarrow \theta) \boxdot\!\!\rightarrow \chi'}) \models \xi \boxdot\!\!\rightarrow \theta$. By induction hypothesis $(M', w_{l,(\xi \boxdot\!\!\rightarrow \theta) \boxdot\!\!\rightarrow \chi'}) \models \xi \boxdot\!\!\rightarrow \theta$, contradicting $(M', w_j) \models (\xi \boxdot\!\!\rightarrow \theta) \boxdot\!\!\rightarrow \bot$ as we want.

3) If $\chi$ has the form $\neg\zeta$, we need a further induction. But the only interesting case is when $\chi$ equals some $\neg(\xi \boxdot\!\!\rightarrow \theta)$. Assume $(M, w_j^{-1}) \models \neg(\xi \boxdot\!\!\rightarrow \theta) \boxdot\!\!\rightarrow \bot$, we need now show $(M', w_j) \models \neg(\xi \boxdot\!\!\rightarrow \theta) \boxdot\!\!\rightarrow \bot$. Suppose not towards a contradiction. Then $\exists v \in W^t, (M', v) \models \neg(\xi \boxdot\!\!\rightarrow \theta)$, viz. $\exists u \in \sigma^t_v(\xi), (M', u) \models \xi \wedge \neg\theta$. By induction hypothesis, $(M, u^{-1}) \models \xi \wedge \neg\theta$. By **Centering** we have $(M, u^{-1}) \models \neg(\xi \boxdot\!\!\rightarrow \theta)$, contradicting the assumption.

For the other direction, suppose towards a contradiction that $(M', w_j) \models \neg(\xi \boxdot\!\!\rightarrow \theta) \boxdot\!\!\rightarrow \bot$ but $(M, w_j^{-1}) \models \neg(\neg(\xi \boxdot\!\!\rightarrow \theta) \boxdot\!\!\rightarrow \bot)$. Now notice, crucially, that $\neg(\xi \boxdot\!\!\rightarrow \theta) \boxdot\!\!\rightarrow \chi'$ occurs in $basic_k(\varphi)$ for some $k$. Then at level $k$ there was a $w_l \in W^t$ which chose a $w_{l,\neg(\xi \boxdot\!\!\rightarrow \theta) \boxdot\!\!\rightarrow \chi'}$ from $W$ for the level $k+1$. Because of the eventual $(M', w_j^{-1}) \models \neg(\xi \boxdot\!\!\rightarrow \theta) \boxdot\!\!\rightarrow \bot$ it must be during the tree construction we had $(M', w_{l,\neg(\xi \boxdot\!\!\rightarrow \theta) \boxdot\!\!\rightarrow \chi'}) \models \neg(\xi \boxdot\!\!\rightarrow \theta)$. By induction hypothesis $(M, w^{-1}_{l,\neg(\xi \boxdot\!\!\rightarrow \theta) \boxdot\!\!\rightarrow \chi'}) \models \neg(\xi \boxdot\!\!\rightarrow \theta)$, a wanted contradiction. □

Now Thesis 2 becomes a theorem.

**Theorem 5.2.** *Let Atm be infinite. Then* **VCU** $\equiv$ **HVCU**.

*Proof.* Since **HVCU** $\subset$ **VCU**, we only need prove the rest direction. Similar to the proof of Theorem 5.1, for any $\varphi$ satisfiable in **VCU**, we start with a filtration model $M^f = (W^f, (W^f_w)_{w \in W^f}(\preceq^f_w)_{w \in W^f}, V^f)$ and a $w_0 \in W^f$ s.t. $(M^f, w_0) \models \varphi$. Then we build an FH tree model $M$ by Lemma 5.3. The next step is associating $M$ with a forward-weighted tree $G^{\#}$ and construct an HVCU model $M'$. Finally, with the help of Lemma 5.4, an induction on $\varphi$ can show that $(M', w_0) \models \varphi$, which is what we want. □

## 5.5 Conclusion

We studied the Hammingian V models, and in particular proved **VC** $\equiv$ **HVC** and **VCU** $\equiv$ **HVCU** given infinite variables in the language. Notice the precondition, which is because Hammingization relies on manipulating variables out of $atm(\varphi)$. This cannot happen without an *unbounded* number of fresh variables: it is known that when *Atm* is *finite* then Hamming distance can be axiomatized by, essentially, taking the conjunction of a maximal consistent set of literal to express a state syntactically.

The technical conclusion is that the property of being Hammingian is unaxiomatizable given the basic language of counterfactuals with infinite variables. The most straightforward philosophical interpretation is that any abstract notion

of distance, e.g. epistemic entrenchment, system of spheres, etc, can be reinterpreted/implemented by Hamming distance by means of "hidden variables". In this sense we call Hamming distance "grounded" for VC and VCU.

For future work, we conjecture the complexity of model checking for **VCU** is PSPACE-complete and will address it. Another intriguing topic is that given that Hamming distance grounds the comparative similarity when it is a total preorder, does another concrete definition of distance, subset relation of valuations, ground the partial order version?

Chapter 6

# A Logic of "Black Box" Classifier Systems

Boolean classifiers are traditionally studied by propositional logic. It expresses the inner mechanisms of the classifiers transparently as propositional formulas. Classifiers trained by machine learning usually have opaque inner mechanisms, and are therefore described as black boxes. In this chapter, we provide a product modal logic called PLC (Product modal Logic for binary input Classifier) in which the notion of "black box" is interpreted as the uncertainty over a set of classifiers. We give results about axiomatics and complexity of satisfiability checking for our logic. Moreover, we present a dynamic extension in which the process of acquiring new information about the actual classifier can be represented.

**Contents**



## 6.1 Introduction

In Chapter 3 we have introduced several logic-based approaches to XAI. They can compute e.g., abductive explanations of a given classification, detect biases in the classification process by means of counterfactual reasoning. But all these approaches do not apply to black box classifiers, for the classifiers they deal with are expressible by propositional formulas, and therefore not opaque. Nevertheless, as mentioned

in Chapter 1, the motivation and main driven force of XAI is to explain black box classifiers. Thus a natural topic is to investigate is whether the modal logic framework in Chapter 3 can be further developed to handle black box classifiers. However, before delving into technical details, a conceptual analysis is needed on what do we talk about when using the black box metaphor in AI.

**What is a black box classifier, comparing with the white box?** In AI the black box metaphor is almost exclusively used for classifier systems, which mathematically are nothing but classification functions. Let us first compare the black box classifier with its opponent, the white box classifier. We start by stating the followings.

- A white box is a function whose inner mechanism is transparent.
  A black box is a function whose inner mechanism is opaque.

It should be little controversial to understand white and black employed as metaphorical vehicles to convey the transparency/opacity of the model at stake. Actually, the concern of black box is also addressed in name of *model transparency*[1], where model here refers to the final classification function. It is also almost self-evident that the opacity of a model depends on the opacity of its inner mechanism / inside mechanism / inner working. When people talk about opening or without opening the black box, as Zednik describes "this metaphorical way of speaking is grounded in the intuition that a system's behavior can be explained by 'looking inside' " [Zednik 2021].

The next question is, what is the inner mechanism? Or more properly, what represents the inner mechanism? We argue that the inner mechanism of a classifier is represented by its mathematical expression/formula/algorithm. For example, a white box classifier is the classification function $f : \mathbb{N} \longrightarrow \{0, 1\}$ expressed by the formula

$$f(x) = \begin{cases} 1 & \text{if } (x+1) \ mod \ 4 = 0 \\ 0 & \text{otherwise.} \end{cases} \tag{6.1}$$

This expression instructs its mechanism of decision-making, namely it outputs 1 for odd numbers which modulo 4 is 3.

By contrast, a black box can be trained by a training set $T = \{(3, 1), (7, 1), (8, 0), (11, 1)\}$ with the following algorithm

$$\arg\min_{\theta} \frac{1}{2|T|} \sum_{(x_i, y_i) \in T} (f_\theta(x_i) - y_i)^2 \tag{6.2}$$

[1]For example, the Amazon Web Services' 2023 white paper https://docs.aws.amazon.com/pdfs/whitepapers/latest/model-explainability-aws-ai-ml/model-explainability-aws-ai-ml.pdf

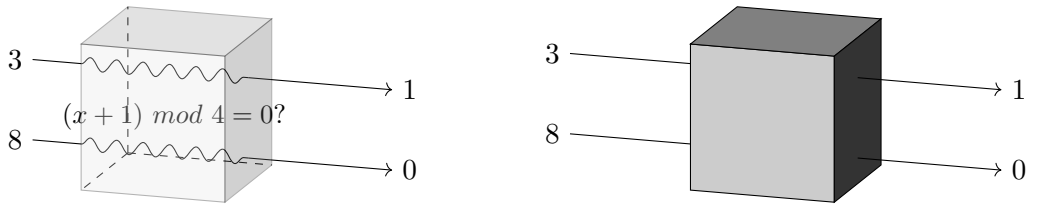



Figure 6.1: Example of two boxes

where $\theta$ denotes parameters of modulus function $f$, and the loss function is simply the average of squares of differences between the predicted $f_\theta(x_i)$ and the actual $y_i$.

Though this is a simple example and we do not train a machine learning algorithm to learn modulus functions, it shows the crucial difference between white and black boxes which is ascribed to the key property of machine learning, as Kearn and Roth in their book *The ethical algorithm* describes.

> Rather than trying to directly specify an algorithm for making these predictions – which could be quite difficult and subtle – we write a meta-algorithm that uses the historical data to *derive* our model or prediction algorithm. [Kearns & Roth 2019, p. 6]

Using their term, in machine learning there is a distinction between the *meta-algorithm* and *our model or prediction algorithm*. We, or at least the designer, have complete knowledge about the meta-algorithm and the training set that feeds into it. The knowledge may increase as we can observe its outputs given new inputs. What we do not know, or more properly speaking, fully know is the resulting model. "The designer may have had a good understanding of the algorithm that was used to *find* the decision-making model, but not the model itself."[Kearns & Roth 2019, p. 11]

Roughly we can view *the model itself* as *the real* resulting model doing the classification task. Since the expression/algorithm of the real model is unknown, with only partial knowledge there are many *compatible models* such that we have *uncertainty* which among them is *the real*.

Back to the example, the classifier expressed by Formula 6.1 is compatible with our knowledge of the black box. However, the real classifier learned may fairly be the function which classifies odd numbers and even numbers. We would never know which one it really is until inputting the new, unseen data which make differences, e.g. 1, 9. If the observation so far is credible (if not even too plain to say), we derive the following statement, which motivates our representation framework.

- A black box can be represented as a set of functions, which are all compatible with the observer's partial knowledge such that she is uncertain about the real classifier.

This intuition is illustrated by Figures 6.1 and 6.2.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | |
| $f_1$ | **1** | **1** | **1** | **0** | **1** | **1** | $\cdots$ |
| $f_0$ | **0** | **1** | **1** | **0** | **0** | **1** | $\cdots$ |
| | 1 | 3 | 7 | 8 | 9 | 11 | |

Figure 6.2: A multi-classifier model for the black box example Equation 6.2

**What are explanations for black boxes binary classifiers** The nouns and phrases "(partial) knowledge", "we know", "we do not know" are heavily used so far. They are more than rhetoric. Indeed, we have to introduce the epistemic viewpoint/dimension to talk about explanations for black boxes. But again, let us first compare it with the white box case, where the epistemic dimension is not needed.[2]

In literature it is common to treat prime implicant as explanations for the Boolean classifiers as we have introduced many times. A prime implicant, if it is locally true in an input, counts as an explanans (sentence that explains the explanandum) of the current input, for it necessarily leads to the classification.

This picture is meaningful in the sense of causal explanation, where agents' possibly limited knowledge of facts of the world are not involved. A locally true prime implicant can be seen as a cause of the phenomenon.[3] Nevertheless, in the black box contest the picture is no more proper, not because it is not wrong, but insufficient. It is not enough to state that a classification *has* a cause, but we need to *know* the cause.

Since we understand a black box as a set of classifiers compatible with the current knowledge, it is possible that our knowledge is limited that, though ontologically speaking every classification has a cause (which can be proven with ease, since the expression of a binary classifier, i.e. a propositional formula, is known), but no cause is known to us according to our current knowledge of the classifier.

Last, it is worth noting that being black is not the only obstacle to explanation. Some AI researchers argue the terminal difference between *explainable AI* and *interpretable AI*, as the title of the paper "Stop explaining black box machine learning models and use interpretable models instead" [Rudin 2019]. While the former is about black box, the latter is about obtaining white boxes, e.g. decision tree, decision set and binary decision digraph. But white boxes may still too complex to be intuitively understood, hence explanation work may still be

[2]There is a subtle distinction between 'white' and 'colorless' boxes. Precisely speaking, white box has the epistemic dimension, just it is fully known, while the examples we give below can be interpreted without knowledge-based perspective (viz. no color) at all. Therefore we use objective vs. subjective explanation to remark situations without or with epistemic viewpoint, instead of white box vs. black box explanation.

[3]We intend to use "a" instead of "the" to circumvent the philosophical discussion of overdeterminism – if both can cause the explanandum, they are both causes.

needed, as the subtitle "interpretable ML models must be explained" of the paper [Marques-Silva & Ignatiev 2023] argues. Also philosophically speaking it is far from clear the terminological difference between interpretable and explainable. Since our focus is on representing black boxes (and their explanations), we will not be back to this issue.

**Representing black boxes with product modal logic** We have shown the central idea to represent a black box classifier as a set of classifiers compatible with the agent's partial knowledge, also the necessity to introduce the epistemic dimension in explaining the black box. By using a two-dimensional logic both can be satisfied in the proper way. One dimension is that fixing a classifier, how all possible input instances are classified. As we have shown in Chapter 3, it is natural to think of a classifier with binary inputs as a partition of an S5 Kripke model, where each possible state stands for an input instance. Another, additional dimension is that fixing an input instance, how all possible classifiers classifies it, where possible is interpreted as compatible/admissible with respect to the agent's partial knowledge. Two dimensions indexed by modal operators $\Box, \blacksquare$ respectively. The agent knows $\varphi$, therefore, if all the admissible classifiers compatible with her knowledge verify $\varphi$. In figure 6.2, $\Box$ ranges over input instances $1, 3, 7, \ldots$ and $\blacksquare$ ranges over $f_0, f_1, \ldots$ The agent knows that $\blacksquare(\neg\Box\neg 3 \wedge \Box(3 \to \mathbf{1}))$, namely there is an input 3 and it is classified as 1.[4] It results in a proper extension of the product modal logic $S5 \times S5 = S5^2$ [Gabbay *et al.* 2003] we call PLC (Product modal Logic of binary-input Classifiers).

The chapter is structured as follows. Section 2 introduces the modal language and semantic model of PLC which we name multi-classifier model (MCM) and is visualized as Figure 6.2. Its axiomatics along with the completeness and complexity results for the satisfiability checking problem are given in Section 3. In Section 4, we will exemplify the logic's application by using it to represent the notion of black box and to formalize different notions of classifier explanation. A dynamic extension is given in Section 6 to capture the process of acquisition of new knowledge about the classifier. Some non-routine proofs are given in Appendix D.

## 6.2 Language and Semantics

Let $Atm_0 = \{p, q, \ldots\}$ be a countable set of atomic propositions which intend to denote input variables (features) of a classifier. We introduce a finite set $Val$ to denote the possible output values (classifications, decisions) of the classifier. Elements of $Val$ are noted $c, c', \ldots$ where $c$ for *c*lassification. For any $c \in Val$, we call $\mathtt{t}(c)$ a decision atom, and have $Dec = \{\mathtt{t}(c) : c \in Val\}$.[5] Finally let $Atm = Atm_0 \cup Dec$.

The modal language $\mathcal{L}$ is defined by the following grammar:

[4] In other words, the agent observes that inputting 3 outputs 1.

[5] Notice that $p$ denotes an input *variable*, while $c$ is an output *value* rather than the output *variable*, which makes sense of the symbolic difference between $p$ and $\mathtt{t}(c)$.

$$\varphi \quad ::= \quad p \mid \mathsf{t}(c) \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \Box\varphi \mid \blacksquare\varphi,$$

where $p$ ranges over $Atm_0$ and $c$ ranges over $Val$.

**Definition 6.1.** *A multi-classifier model (MCM) is a pair $\Gamma = (S, F_S)$ where $S \subseteq 2^{Atm_0}$ and $F_S \subseteq Val^S$ the set of functions with domain $S$ and codomain $Val$. A pointed MCM is a triple $(\Gamma, s, f)$ where $\Gamma = (S, F_S)$ is an MCM, $s \in S$ and $f \in F_S$. The class of all multi-classifier models is noted* **MCM**.

Formulas in $\mathcal{L}$ are interpreted relative to a pointed MCM as follows.

**Definition 6.2** (Satisfaction relation)**.** *Let $\Gamma = (S, F_S)$ be an MCM, $s \in S$ and $f \in F_S$. Then,*

$$\begin{aligned}
(\Gamma, s, f) \models p &\iff p \in s,\\
(\Gamma, s, f) \models \mathsf{t}(c) &\iff f(s) = c,\\
(\Gamma, s, f) \models \neg\varphi &\iff (\Gamma, s, f) \not\models \varphi,\\
(\Gamma, s, f) \models \varphi \wedge \psi &\iff (\Gamma, s, f) \models \varphi \text{ and } (\Gamma, s, f) \models \psi,\\
(\Gamma, s, f) \models \Box\varphi &\iff \forall s' \in S : (\Gamma, s', f) \models \varphi,\\
(\Gamma, s, f) \models \blacksquare\varphi &\iff \forall f' \in F_S : (\Gamma, s, f') \models \varphi.
\end{aligned}$$

Both $\Box\varphi$ and $\blacksquare\varphi$ have standard modal reading but range over different sets. $\Box\varphi$ has to be read "$\varphi$ *necessarily* holds for the actual function, regardless of the input instance", while its dual $\Diamond\varphi =_{def} \neg\Box\neg\varphi$ has to be read "$\varphi$ *possibly* holds for the actual function, regardless of the input instance". Similarly, $\blacksquare\varphi$ has to be read "$\varphi$ *necessarily* holds for the actual input instance, regardless of the function" and its dual $\blacklozenge\varphi$ has to be read "$\varphi$ *possibly* holds for the actual input instance, regardless of the function".

Let $X$ be a finite subset of $Atm_0$. An important abbreviation is the following:

$$[X]\varphi =_{def} \bigwedge_{Y \subseteq X} ((\bigwedge_{p \in Y} \wedge \bigwedge_{p \in X \setminus Y} \neg p) \rightarrow \Box((\bigwedge_{p \in Y} \wedge \bigwedge_{p \in X \setminus Y} \neg p) \rightarrow \varphi)).$$

Complex as it seems, $[X]\varphi$ means nothing but "$\varphi$ necessarily holds, regardless of the values of the input variables outside $X$" or "$\varphi$ necessarily holds, if the values of the input variables in $X$ are kept fixed". It can be justified by checking that $(\Gamma, s, f) \models [X]\varphi$, if and only if $\forall s' \in S$, if $s \cap X = s' \cap X$ then $(\Gamma, s', f) \models \varphi$. Its dual $\langle X\rangle\varphi =_{def} \neg[X]\neg\varphi$ has to be read "$\varphi$ *possibly* holds, if the values of the input variables in $X$ are kept fixed"'. These modalities have a *ceteris paribus* reading and were first introduced in [Grossi *et al.* 2015]. Similar modalities are used in existing logics of functional dependence [Yang & Väänänen 2016, Baltag & van Benthem 2021].

A formula $\varphi$ of $\mathcal{L}$ is said to be satisfiable relative to the class **MCM** if there exists a pointed multi-classifier model $(\Gamma, s, f)$ with $\Gamma \in$ **MCM** such that $(\Gamma, s, f) \models \varphi$. We say that that $\varphi$ is valid in the multi-classifier model $\Gamma = (S, F_S)$, noted $\Gamma \models \varphi$, if $(\Gamma, s, f) \models \varphi$ for every $s \in S, f \in F_S$. It is said to be valid relative to **MCM**, noted $\models_{\mathbf{MCM}} \varphi$, if $\neg\varphi$ is not satisfiable relative to **MCM**.

## 6.3 Axiomatics and Complexity

In this section, we are going to present two axiomatics for the language $\mathcal{L}$ by distinguishing the finite-variable from the infinite-variable case. We will moreover give complexity results for satisfiability checking. Before, we are going to introduce an alternative Kripke semantics for the interpretation of the language $\mathcal{L}$. It will allow us to use the standard canonical model technique for proving completeness. Indeed, this technique cannot be directly applied to MCMs in the infinite-variable case.

### 6.3.1 Alternative Kripke Semantics

The crucial concept of the alternative semantics is multi-decision model (MDM).

**Definition 6.3.** *An MDM is a tuple* $M = (W, \sim_{\Box}, \sim_{\blacksquare}, V)$ *where:*

- $W$ *is a set of worlds,*
- $\sim_{\Box}$ *and* $\sim_{\blacksquare}$ *are equivalence relations on* $W$*,*
- $V : W \longrightarrow 2^{Atm}$ *is a valuation function,*

*and which satisfies the following constraints,* $\forall w, v \in W$, $\forall c, c' \in Val$ :

**(C1)** $\sim_{\Box} \circ \sim_{\blacksquare} = \sim_{\blacksquare} \circ \sim_{\Box}$,

**(C2)** *if* $V_{Atm_0}(w) = V_{Atm_0}(v)$ *and* $w \sim_{\Box} v$ *then* $V_{Dec}(w) = V_{Dec}(v)$,

**(C3)** *if* $w \sim_{\blacksquare} v$ *then* $V_{Atm_0}(w) = V_{Atm_0}(v)$,

**(C4)** *if* $\mathtt{t}(c) \in V(w)$ *and* $c \neq c'$ *then* $\mathtt{t}(c') \notin V(w)$,

**(C5)** $\exists c \in Val$ *such that* $\mathtt{t}(c) \in V(w)$,

*with* $V_Y(w) = (V(w) \cap Y)$ *for every* $w \in W$ *and for every* $Y \subseteq Atm$*, and* $\circ$ *the standard composition operator for binary relations.*

The class of multi-decision models is noted **MDM**. An MDM $M = (W, \sim_{\Box}, \sim_{\blacksquare}, V)$ is called finite if $W$ is finite. The class of finite MDMs is noted finite-**MDM**. Interpretation of formulas in $\mathcal{L}$ relative to a pointed MDM goes as follows. (We omit interpretations for $\neg$ and $\wedge$ which are defined as usual.)

**Definition 6.4** (Satisfaction Relation)**.** *Let* $M = (W, \sim_{\Box}, \sim_{\blacksquare}, V)$ *be an MDM and let* $w \in W$*. Then,*

$$\begin{aligned}
(M, w) \models q &\iff q \in V(w) \text{ for } q \in Atm,\\
(M, w) \models \Box\varphi &\iff \forall v \in W, \text{ if } w \sim_{\Box} v \text{ then } v \models \varphi,\\
(M, w) \models \blacksquare\varphi &\iff \forall v \in W, \text{if } w \sim_{\blacksquare} v \text{ then } v \models \varphi.
\end{aligned}$$

Validity and satisfiability of formulas in $\mathcal{L}$ relative to class **MDM** (resp. finite-**MDM**) are defined in the usual way.

The most important result in this subsection is the semantic equivalence between **MCM** and **MDM**, regardless of $Atm_0$ being finite or infinite. Although a pointed MDM $(M, w)$ looks like a pointed MCM $(\Gamma, s, f)$, it only approximates it. Indeed, unlike an MCM, an MDM $M$ may be redundant, that is, (i) a classifier in $M$ (i.e., a $\sim_{\Box}$-equivalence class) may include multiple copies of the same input instance (i.e., of the same valuation for the atoms in $Atm_0$), or (ii) $M$ may contain multiple copies of the same classifier (i.e., two identical $\sim_{\blacksquare}$-equivalence classes). Moreover, an MDM $M$ may be "defective" insofar as (iii) the intersection between a classifier in $M$ (i.e., a $\sim_{\Box}$-equivalence class) and the set of all possible classifications of a given input instance by the classifiers in $M$ (i.e., a $\sim_{\blacksquare}$-equivalence class) is not a singleton. What makes the proof of the following theorem non-trivial is transforming a possibly redundant or defective MDM into a non-redundant and non-defective one by preserving truth of formulas. A non-redundant and non-defective MDM is then isomorphic to an MCM.

**Theorem 6.1.** *Let $\varphi \in \mathcal{L}$. Then, $\varphi$ is satisfiable relative to the class* **MCM** *if and only if it is satisfiable relative to the class* **MDM**.

### 6.3.2 Finite-Variable Case

We first consider the variant of the logic with finitely many propositional atoms in $Atm_0$. For every finite $X, Y \subseteq Atm_0$ we define:

$$\mathsf{cn}_{X,Y} =_{def} \bigwedge_{p \in X} p \wedge \bigwedge_{p \in (Y \setminus X)} \neg p.$$

**Definition 6.5** (Logic PLC)**.** *Let $Atm_0$ be finite. We define* PLC *as the extension of classical propositional logic given by axioms and rules of inference in Table 6.1.*

Axioms $\mathbf{AtLeast}_{\mathsf{t}(c)}$, $\mathbf{AtMost}_{\mathsf{t}(c)}$ and **Funct** guarantee that every input $Y \subseteq Atm_0$, whose syntactic counterpart is $\mathsf{cn}_{Y,Atm_0}$, has only one decision atom as output. Axioms $\mathbf{K}_{\boxplus}$, $\mathbf{T}_{\boxplus}$, $\mathbf{4}_{\boxplus}$ and $\mathbf{5}_{\boxplus}$ together with the rule of inference $\mathbf{Nec}_{\boxplus}$ indicate that both modal operators $\blacksquare$ and $\Box$ satisfy the principles of the modal logic S5. According to Axioms **Comm** and **Conf** (also known as Churcher-Rosser property), they moreover commute and converge. These make the logic meet the requirement of a product of two S5 modal logics, i.e., $S5^2$ [Gabbay *et al.* 2003]. Nevertheless, the existence of the two "independence" Axioms $\mathbf{Indep}_{\blacksquare p}$ and $\mathbf{Indep}_{\blacksquare \neg p}$ indicates that PLC is stronger than $S5^2$ in general.

Soundness of PLC relative to **MCM** is a simple exercise. To prove the completeness result, we first need to show that PLC is complete relative to **MDM**, which is proven by the canonical model construction.

**Theorem 6.2.** *Let $Atm_0$ be finite. Then, the logic* PLC *is sound and complete relative to the class* **MDM**.

| | |
|---|---|
| $\big(\boxplus\varphi \wedge \boxplus(\varphi \to \psi)\big) \to \boxplus\psi$ | $(\mathbf{K}_{\boxplus})$ |
| $\boxplus\varphi \to \varphi$ | $(\mathbf{T}_{\boxplus})$ |
| $\boxplus\varphi \to \boxplus\boxplus\varphi$ | $(\mathbf{4}_{\boxplus})$ |
| $\neg\boxplus\varphi \to \boxplus\neg\boxplus\varphi$ | $(\mathbf{5}_{\boxplus})$ |
| $\blacksquare\Box\varphi \leftrightarrow \Box\blacksquare\varphi$ | (**Comm**) |
| $\neg\Box\neg\blacksquare\varphi \to \blacksquare\neg\Box\neg\varphi$ | (**Conf**) |
| $\bigvee_{c\in Val} \mathsf{t}(c)$ | $(\mathbf{AtLeast}_{\mathsf{t}(c)})$ |
| $\mathsf{t}(c) \to \neg\mathsf{t}(c')$ if $c \neq c'$ | $(\mathbf{AtMost}_{\mathsf{t}(c)})$ |
| $\big(\mathsf{cn}_{X,Atm_0} \wedge \mathsf{t}(c)\big) \to \Box\big(\mathsf{cn}_{X,Atm_0} \to \mathsf{t}(c)\big)$ | (**Funct**) |
| $p \to \blacksquare p$ | $(\mathbf{Indep}_{\blacksquare p})$ |
| $\neg p \to \blacksquare\neg p$ | $(\mathbf{Indep}_{\blacksquare\neg p})$ |
| $\dfrac{\varphi}{\boxplus\varphi}$ | $(\mathbf{Nec}_{\boxplus})$ |

Table 6.1: Axioms and rules of inference, with $\boxplus \in \{\Box, \blacksquare\}$

Our main result of this subsection becomes a corollary of Theorems 6.1 and 6.2.

**Corollary 6.1.** *Let $Atm_0$ be finite. Then, the logic* PLC *is sound and complete relative to the class* **MCM**.

### 6.3.3 Infinite-Variable Case

We now move to the infinite-variable variant of our logic, under the assumption that the set $Atm_0$ is countably infinite. In order to obtain an axiomatics we just need to drop the "functionality" Axiom **Funct** of Table 6.1. Indeed, when $Atm_0$ is infinite, the construction $\mathsf{cn}_{X,Atm_0}$ cannot be expressed in a finitary way.

**Definition 6.6** (Logic WPLC)**.** *We define* WPLC *(Weak* PLC*) to be the extension of classical propositional logic given by Axioms $\boldsymbol{K}_{\boxplus}$, $\boldsymbol{T}_{\boxplus}$, $\boldsymbol{4}_{\boxplus}$, $\boldsymbol{5}_{\boxplus}$, **Comm**, **Conf**, $\boldsymbol{AtLeast}_{\mathsf{t}(c)}$, $\boldsymbol{AtMost}_{\mathsf{t}(c)}$, $\boldsymbol{Indep}_{\blacksquare p}$ and $\boldsymbol{Indep}_{\blacksquare\neg p}$, and the rule of inference $\boldsymbol{Nec}_{\boxplus}$ in Table 6.1.*

Soundness of the logic WPLC is a straightforward exercise. For completeness, we need to distinguish MDMs from quasi-MDMs that are obtained by removing the "functionality" Constraint **C2** from Definition 6.3.

**Definition 6.7** (Quasi-MDM)**.** *A quasi-MDM is a tuple $M = (W, \sim_{\Box}, \sim_{\blacksquare}, V)$ where $W$, $\sim_{\Box}$, $\sim_{\blacksquare}$ and $V$ are defined as in Definition 6.3 and which satisfies all constraints of Definition 6.3 except* **C2**.

The class of quasi-MDMs is noted **QMDM**. A quasi-MDM $M = (W, \sim_{\Box}, \sim_{\blacksquare}, V)$ is said to be finite if $W$ is finite. The class of finite quasi-MDMs is noted finite-**QMDM**. Semantic interpretation of formulas in $\mathcal{L}$ relative to quasi-MDMs is

analogous to semantic interpretation relative to MDMs given in Definition 6.4. Moreover, validity and satisfiability of formulas in $\mathcal{L}$ relative to class **QMDM** (resp. finite-**QMDM**) is again defined in the usual way.

The first crucial result of this subsection is that when $Atm_0$ is infinite the language $\mathcal{L}$ cannot distinguish finite MDMs from finite quasi-MDMs.

**Theorem 6.3.** *Let $\varphi \in \mathcal{L}$ with $Atm_0$ infinite. Then, $\varphi$ is satisfiable relative to the class finite-***MDM** *if and only if it is satisfiable relative to the class finite-***QMDM**.

The second result is that satisfiability for formulas in $\mathcal{L}$ relative to the class **QMDM** is equivalent to satisfiability relative to the class finite-**QMDM**.

**Theorem 6.4.** *Let $\varphi \in \mathcal{L}$. Then, $\varphi$ is satisfiable relative to the class* **QMDM** *if and only if it is satisfiable relative to the class finite-***QMDM**.

The following theorem is provable by standard canonical model argument. Note that like Theorems 6.1 and 6.4, it does not rely on $Atm_0$ being infinite or finite.

**Theorem 6.5.** *The logic* WPLC *is sound and complete relative to the class* **QMDM**.

The fact that the logic WPLC is sound and complete relative to the class **MCM** is a direct corollary of Theorems 6.1, 6.3, 6.4 and 6.5.

**Corollary 6.2.** *Let $Atm_0$ be infinite. Then, the logic* WPLC *is sound and complete relative to the class* **MCM**.

### 6.3.4 Complexity Results

We now move to complexity of satisfiability checking. As for the axiomatics, we distinguish the finite-variable from the infinite-variable case. When $Atm_0$ is finite, the problem of verifying whether a formula is satisfiable is polynomial. The latter problem mirrors the satisfiability checking problem for the finite-variable modal logic S5 which is also known to be polynomial [Halpern 1995].

**Theorem 6.6.** *Let $Atm_0$ be finite. Then, checking satisfiability of $\mathcal{L}$-formulas relative to the class* **MCM** *can be done in polynomial time.*

We know that when moving from the finite-variable to the infinite-variable case complexity of satisfiability checking is in NEXPTIME.

**Theorem 6.7.** *Let $Atm_0$ be infinite. Then, checking satisfiability of $\mathcal{L}$-formulas relative to the class* **MCM** *is in NEXPTIME.*

In [Bezhanishvili & Hodkinson 2004] (see also [Bezhanishvili & Marx 2003]) it is proved that all proper normal extensions of the product modal logic $S5^2$ are in NP. In future work, we plan to verify whether these results are applicable to our setting in order to improve our complexity upper bound. The problem is that Axioms $\mathbf{Indep}_{\blacksquare p}$, $\mathbf{Indep}_{\blacksquare \neg p}$, $\mathbf{AtMost}_{\mathsf{t}(c)}$ and $\mathbf{AtLeast}_{\mathsf{t}(c)}$ are not axiom schemata in the proper sense.

## 6.4 Application

As mentioned, the ■ operator is interpreted as partial knowledge about the classifier properties.[6] In this section, we are going to exemplify how to use it for representing abductive explanations of a black box classifier.

### 6.4.1 An Example of Classification Task

Consider a selection function which specifies whether a paper submitted to a conference is acceptable for presentation (1) or not (0) depending on its feature profile composed of four input features: significance (si), originality (or), clarity of the presentation (cl) and fulfillment of the anonymity requirement (an). For the sake of simplicity, we assume each feature in a paper profile is binary: si means the paper is significant while ¬si means the paper is not significant, or means the paper is original while ¬or means the paper is not original, and so on. We say that a first paper profile dominates a second paper profile, if all conditions satisfied by the second profile are satisfied by the first profile, and there exists a condition satisfied by the first profile which is not satisfied by the second profile. For example if the first profile is si $\wedge$ ¬or $\wedge$ cl $\wedge$ an and the second profile is si $\wedge$ ¬or $\wedge$ ¬cl $\wedge$ an, then the first dominates the second.

The selection function is implemented in a classifier system that has to automatically split papers into two sets, the set of acceptable papers and the set of non-acceptable ones. We assume a certain agent (e.g., the author of a paper submitted to the conference) has only partial knowledge of the classifier system. In particular, she only knows that the classifier complies with the following three constraints: (1) submissions that satisfy the four conditions should be automatically accepted, (2) if a first paper profile dominates a second paper profile and the second paper profile is acceptable, then the first paper profile should also be acceptable, and (3) submissions that violate the anonymity requirement should be automatically rejected. In this case, the classifier is a black box for the agent.

**Example 6.1.** *The multi-classifier model (MCM) representing the previous situation is the tuple* $\Gamma = (S, F_S)$ *such that* $S = 2^{\{\mathrm{si},\mathrm{or},\mathrm{cl},\mathrm{an}\}}$ *and*

$$\begin{aligned} \forall f \in \mathit{Val}^S, f \in F_S \text{ iff } &(i)\ \forall s \in S,\ \text{if } \{\mathrm{si},\mathrm{or},\mathrm{cl},\mathrm{an}\} \subseteq s \text{ then } f(s) = 1, \\ &(ii)\ \forall s, s' \in S,\ \text{if } s \subset s' \text{ and } f(s) = 1 \text{ then } f(s') = 1. \\ &(iii)\ \forall s \in S,\ \text{if } \mathrm{an} \notin s \text{ then } f(s) = 0. \end{aligned}$$

*The agent does not know which function in* $F_S$ *corresponds to the actual classifier, i.e., they are epistemically indistinguishable for her.*

[6] In the real world, partial knowledge may come from the data set as well as from the training process. For example, through learning, we may acquire knowledge that certain input features behave monotonically [You *et al.* 2017], also in [Kearns & Roth 2019] the authors provide several methods to implement some constraints in training the algorithm to meet ethical requirements.

### 6.4.2 Explanations

We exemplify explanations for white and black box classifiers by showing the dichotomy global vs. local explanation and the notion of *abductive explanation* based on *prime implicant*.

Recall that $s$ is called an *instance*, $\lambda$ denotes a set of consistent literals and is called a *term* or *property* (of the instance). The set of terms is noted *Term*. Moreover, let $Atm(\varphi)$ denote the atoms occurring in $\varphi$. Finally, notice that the abbreviations $[X]\varphi$ and $\langle X\rangle\varphi$ introduced in Section 2 will be used.

Recall *prime implicant*, a key concept in the theory of Boolean functions since [Quine 1955]. It can be presented in the language $\mathcal{L}(Atm)$ as follows:

$$\mathsf{PImp}(\lambda, c) =_{def} \Box\Big(\lambda \rightarrow (\mathsf{t}(c) \wedge \bigwedge_{p\in Atm(\lambda)} \langle Atm(\lambda) \setminus \{p\}\rangle \neg \mathsf{t}(c))\Big).$$

The abbreviation $\mathsf{PImp}(\lambda, c)$ has to be read "$\lambda$ is a prime implicant for the classification $c$". Roughly speaking, the latter means that (i) $\lambda$ necessarily leads to the classification $c$ (why $\lambda$ is an *implicant*), and (ii) for any of its proper subsets $\lambda'$, possibly there is a state where $\lambda'$ holds but the classification is different from $c$ (why $\lambda$ is *prime*).

Prime implicant counts as a "global" explanation, in the sense that it is a property of the classifier and holds at *all* its input instances. The localized version of prime implicant is called abductive explanation. An abductive explanation is not only a prime implicant, but also a *property of the actual instance*. It is expressed in $\mathcal{L}$ as follows:

$$\mathsf{AXp}(\lambda, c) =_{def} \lambda \wedge \mathsf{PImp}(\lambda, c).$$

$\mathsf{AXp}(\lambda, c)$ just means that $\lambda$ is an abductive explanation of the actual classification $c$. Let us instantiate the notions of prime implicant and abductive explanation in the paper example we introduced in Section 6.4.1.

**Example 6.2.** *Take the MCM* $\Gamma = (S, F_S)$ *in Example 6.1, and let* $s_1 = \{\mathrm{si}, \mathrm{or}, \mathrm{an}\} \in S$. *Consider the function* $f_1$ *s.t.* $\forall s \in S : f_1(s) = 1$ *iff* $\mathrm{an} \in s$ *and* $\{\mathrm{or}, \mathrm{cl}\} \cap s \neq \emptyset$. *The function* $f_1$ *is syntactically expressed by the formula* $\Box(\mathsf{t}(1) \leftrightarrow ((\mathrm{or} \wedge \mathrm{an}) \vee (\mathrm{cl} \wedge \mathrm{an})))$. *Clearly* $f_1 \in F_S$ *for it satisfies the three constraints. Hence, we have:*

$$(\Gamma, s_1, f_1) \models \mathsf{AXp}(\mathrm{or} \wedge \mathrm{an}, 1) \wedge \mathsf{PImp}(\mathrm{or} \wedge \mathrm{an}, 1) \wedge \mathsf{PImp}(\mathrm{cl} \wedge \mathrm{an}, 1).$$

*Meanwhile* $(\Gamma, s_1, f_1) \not\models \mathsf{AXp}(\mathrm{cl} \wedge \mathrm{an}, 1)$, *because* $(\Gamma, s_1, f_1) \not\models \mathrm{cl} \wedge \mathrm{an}$. *But consider* $s_2 = \{\mathrm{si}, \mathrm{cl}, \mathrm{an}\} \in S$. *We have* $(\Gamma, s_2, f_1) \models \mathsf{AXp}(\mathrm{cl} \wedge \mathrm{an}, 1)$.

Now we investigate what happens when facing a black box model $\Gamma = (S, F_S)$. The agent has uncertainty about the actual classifier's properties. Therefore, it is interesting to draw the distinction between objective and subjective (or epistemic) explanation. Objective explanation coincides with the notion of explanation in the

context of white box classifiers defined above. Subjective explanation refers to the agent's interpretation of the classifier and her explanation of the classifier's decision in the light of her partial knowledge.

We say the term $\lambda$ is a *subjective* prime implicant for $c$, noted $\mathsf{SubPImp}(\lambda, c)$, if the agent knows that $\lambda$ is a prime implicant for $c$, that is:

$$\mathsf{SubPImp}(\lambda, c) =_{def} \blacksquare\mathsf{PImp}(\lambda, c).$$

Similarly, we say $\lambda$ is a *subjective* abductive explanation of the actual classification $c$, noted $\mathsf{SubAXp}(\lambda, c)$, if the agent knows that $\lambda$ is an abductive explanation of the actual classification $c$, that is:

$$\mathsf{SubAXp}(\lambda, c) =_{def} \blacksquare\mathsf{AXp}(\lambda, c).$$

It is worth noting that in the case of a white box classifier, where the set of input instances $S$ is finite, we can always find an abductive explanation of the actual classification. That is, for every $\Gamma = (S, F_S) \in \mathbf{MCM}$, $s \in S$ and $f \in F_S$:

$$\textit{if } S \textit{ is finite then } \exists \lambda \in \textit{Term} \text{ such that } (\Gamma, s, f) \models \mathsf{AXp}(\lambda, f(s)).$$

Nonetheless, this result cannot be generalized to the black box case. Indeed, as the following example shows, there is no guarantee for the existence of a subjective explanation of the actual classification. The problem is that the minimality condition can collapse when moving from objective to subjective explanation, since the agent can have more than one classifier in her epistemic state.

**Example 6.3.** *Let $\Gamma = (S, F_S)$, $f_1$ and $s_1$ be the same as in Example 6.2. There is no $\lambda$ such that $(\Gamma, s_1, f_1) \models \blacksquare\mathsf{AXp}(\lambda, 1)$. To see this, consider $f_2$ s.t. $\forall s \in S$ : $f_2(s) = 1$ iff $\{\mathrm{si}, \mathrm{an}\} \subseteq s$. The function $f_2$ is syntactically expressed by the formula $\Box(\mathsf{t}(1) \leftrightarrow (\mathrm{si} \wedge \mathrm{an}))$. Clearly $f_2 \in F_S$ for it satisfies the three constraints. We have $(\Gamma, s_1, f_2) \models \mathsf{AXp}(\mathrm{si} \wedge \mathrm{an}, 1)$. But there is no term which minimally explains both $f_1(s_1)$ and $f_2(s_1)$. Indeed,* $\mathrm{or} \wedge \mathrm{an}$ *is not enough for explaining $f_2(s_1)$,* $\mathrm{si} \wedge \mathrm{an}$ *is not enough for explaining $f_1(s_1)$, and* $\mathrm{si} \wedge \mathrm{or} \wedge \mathrm{an}$ *fails the minimality condition for both. Therefore, we have*

$$(\Gamma, s_1, f_1) \models \mathsf{AXp}(\mathrm{or} \wedge \mathrm{an}, 1) \wedge \bigwedge_{\lambda \in \mathit{Term}(\{\mathrm{si,or,cl,an}\})} \neg\mathsf{SubAXp}(\lambda, 1).$$

*However, this does not mean that the agent knows nothing about the classifier. For instance, she knows that violating the anonymity requirement is a prime implicant for rejection, that is,* $(\Gamma, s_1, f_1) \models \mathsf{SubAXp}(\neg\mathrm{an}, 0)$.

To sum up, the four notions of explanation we introduced can be organized in Table 6.2 along the two dimensions objective vs subjective and local vs global.

| | Local | Global |
|---|---|---|
| Objective | $\mathsf{AXp}(\lambda, c)$ | $\mathsf{PImp}(\lambda, c)$ |
| Subjective | $\mathsf{SubAXp}(\lambda, c)$ | $\mathsf{SubPImp}(\lambda, c)$ |

Table 6.2: Notions of prime implicant and abductive explanation

## 6.5 Dynamic Extension

Before concluding, we are going to present a simple dynamic extension of the language $\mathcal{L}$ by operators of the form $[\varphi]$. They describe the consequences of removing from the actual model all classifiers that do not *globally* satisfy the constraint $\varphi$. More generally, they allow us to model the process of gaining new knowledge about the classifier's properties. The extended modal language $\mathcal{L}^{dyn}$ is defined by the following grammar:

$$\varphi \quad ::= \quad p \mid \mathsf{t}(c) \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \Box\varphi \mid \blacksquare\varphi \mid [\varphi]\psi,$$

where $p$ ranges over $Atm_0$ and $c$ ranges over $Val$.

The new formula $[\varphi]\psi$ has to be read "$\psi$ holds after having discarded all classifiers that do not globally satisfy the property $\varphi$". Notice the similar but different notations $[X]$ and $[\varphi]$. For example, $[\{p\}], [\{p, q\}]$ are abbreviations with ceteris paribus meaning, while $[p], [p \wedge \neg q]$ are dynamic operators.

The interpretation of the operators $[\varphi]$ relative to a pointed MCM $(\Gamma, s, f)$ with $\Gamma = (S, F_S)$, $s \in S$ and $f \in F_S$ goes as follows:

$$(\Gamma, s, f) \models [\varphi]\psi \quad \Longleftrightarrow \quad \text{if } (\Gamma, s, f) \models \Box\varphi \text{ then } (\Gamma^{\varphi}, s, f) \models \psi,$$

where $\Gamma^{\varphi} = (S^{\varphi}, F_S^{\varphi})$ is the MCM such that:

$$S^{\varphi} = S,$$
$$F_S^{\varphi} = \{f' \in F_S : \forall s' \in S, (\Gamma, s', f') \models \varphi\}.$$

The previous update semantics for the operator $[\varphi]$ is reminiscent of the semantics of public announcement logic (PAL) [Plaza 2007, van Ditmarsch *et al.* 2007]. However, there is an important difference. While PAL has a one-dimensional state elimination semantics, our update semantics operates on a single dimension of the product in an MCM. In particular, it only removes classifiers that do not globally satisfy the constraint $\varphi$, without modifying the set $S$ of input instances.

The logics DPLC and WDPLC (Dynamic PLC and WDPLC) extend the logic PLC and WPLC by the dynamic operators $[\varphi]$. They are defined as follows.

**Definition 6.8** (Logics DPLC and WDPLC)**.** *We define* DPLC *(resp.* WDPLC*) to be the extension of* PLC *(resp.* WPLC*) of Definition 6.5 (resp. Definition 6.6)*

*generated by the following reduction axioms for the dynamic operators* $[\varphi]$*:*

$$\begin{aligned}
[\varphi]p &\leftrightarrow (\Box\varphi \to p)\\
[\varphi]\mathsf{t}(c) &\leftrightarrow (\Box\varphi \to \mathsf{t}(c))\\
[\varphi]\neg\psi &\leftrightarrow (\Box\varphi \to \neg[\varphi]\psi)\\
[\varphi](\psi_1 \wedge \psi_2) &\leftrightarrow ([\varphi]\psi_1 \wedge [\varphi]\psi_2)\\
[\varphi]\Box\psi &\leftrightarrow (\Box\varphi \to \Box[\varphi]\psi)\\
[\varphi]\blacksquare\psi &\leftrightarrow (\Box\varphi \to \blacksquare[\varphi]\psi)
\end{aligned}$$

*and the following rule of inference:*

$$\frac{\varphi_1 \leftrightarrow \varphi_2}{\psi \leftrightarrow \psi[\varphi_1/\varphi_2]} \qquad (\mathbf{RE})$$

It is a routine exercise to verify that the equivalences in Definition 6.8 are valid for the class **MCM** and that the rule of replacement of equivalents (**RE**) preserves validity. We show the validity of the sixth equivalence as an example:

$$\begin{aligned}
(\Gamma, s, f) \models [\varphi]\blacksquare\psi &\iff \text{if } (\Gamma, s, f) \models \Box\varphi \text{ then} (\Gamma^{\varphi}, s, f) \models \blacksquare\psi;\\
&\iff \text{if } (\Gamma, s, f) \models \Box\varphi \text{ then} \forall f' \in F_S^{\varphi}, (\Gamma^{\varphi}, s, f') \models \psi;\\
&\iff \text{if } (\Gamma, s, f) \models \Box\varphi \text{ then } \forall f' \in F_S,\\
&\qquad (\text{if } \forall s' \in S, (\Gamma, s', f') \models \psi \text{ then } (\Gamma^{\varphi}, s, f') \models \psi);\\
&\iff \text{if } (\Gamma, s, f) \models \Box\varphi \text{ then } \forall f' \in F_S,\\
&\qquad (\text{if } (\Gamma, s, f') \models \Box\psi \text{ then } (\Gamma^{\varphi}, s, f') \models \psi);\\
&\iff \text{if } (\Gamma, s, f) \models \Box\varphi \text{ then } \forall f' \in F_S, (\Gamma, s, f') \models [\varphi]\psi;\\
&\iff (\Gamma, s, f) \models \Box\varphi \to \blacksquare[\varphi]\psi.
\end{aligned}$$

The completeness of DPLC and WDPLC for this class of models follows from Theorem 6.2 and Corollary 6.1, in view of the fact that the reduction axioms and the rule of replacement of proved equivalents can be used to find, for any $\mathcal{L}^{dyn}$-formula, a provably equivalent $\mathcal{L}$-formula.

**Theorem 6.8.** *Let* $Atm_0$ *be finite. Then, the logic* DPLC *is sound and complete relative to the class* **MCM***.*

**Theorem 6.9.** *Let* $Atm_0$ *be infinite. Then, the logic* WDPLC *is sound and complete relative to the class* **MCM***.*

The following decidability result is a consequence of Theorem 6.7 and the fact that via the reduction axioms in Definition 6.8 we can find a reduction of satisfiability checking of $\mathcal{L}^{dyn}$-formulas to satisfiability checking of $\mathcal{L}$-formulas.

**Theorem 6.10.** *Checking satisfiability of* $\mathcal{L}^{dyn}$*-formulas relative to* **MCM** *is decidable.*

Let us end up with the paper example to illustrate to expressive power of our dynamic extension.

**Example 6.4.** *Let* $\Gamma = (S, F_S), f_1$ *and* $s_1$ *be the same as in Example 6.2. We have*

$$(\Gamma, s_1, f_1) \models [(\mathrm{or} \wedge \mathrm{an}) \to \mathsf{t}(1)]\blacksquare \bigvee_{\lambda \subseteq (\mathrm{or} \wedge \mathrm{an})} \mathsf{AXp}(\lambda, 1).$$

*This means that after having discarded all classifiers which do not take* $(\mathrm{or} \wedge \mathrm{an})$ *as an implicant for acceptance of a paper, the agent knows that there must be a part of* $\mathrm{or} \wedge \mathrm{an}$ *that abductively explains the acceptance of the paper* $s_1$.

## 6.6 Conclusion

We have presented a product modal logic which supports reasoning about (i) partial knowledge and uncertainty of a classifier's properties and, (ii) objective and subjective explanations of a classifier's decision. Moreover, we have studied a dynamic extension of the logic which allows us to represent the event of gaining new knowledge about the classifier's properties.

Our logic is intrinsically single-agent: it models the uncertainty of one agent about the actual classifier's properties. In future work, we plan to generalize our framework to the multi-agent setting. The extension would result in a multi-relational product semantics in which every agent has her own epistemic indistinguishability relation which commutes with the input instance dimension (the equivalence relation $\sim_\square$ in Definition 6.3 of MDM). We also plan to enrich this semantics with a knowledge update mechanism in the spirit of Section 6.5. This would allow us to represent exchange of information between agents with an explanatory purpose, which is named dialogical explanation by philosophers [Walton 2004] and interactive explanation by researchers in the XAI domain [Amershi *et al.* 2014, Miller 2019].

Chapter 7

# Perspectives

In this chapter we deal with two problems which rose in the previous chapters. Although these are interesting as technical issues in their own right, there are also conceptual reasons that make them worth studying. The first problem concerns the complexity of satisfiability problem in **MCM**. The second one concerns whether we can define and axiomatize model classes that even when the language has infinite atoms, every classifier therein is definite with respect to some finite subset of atoms.

## 7.1 How Hard is Black Box Explanation? A Complexity Study

It is clear that black box classifiers are hard to explain since without opening up them and looking inside the only way is based on perturbation. But how hard is it in general from the computational complexity viewpoint? In Chapter 6 we have shown that the satisfiability problem in **MCM** is in NEXPTIME, since the filtration construction of **MDM** gives an NEXPTIME algorithm for it, and every MDM can be transformed into an MCM in linear time. We will study further on this issue, and give a new lower-bound.

### 7.1.1 Some hints

The complexity of satisfiability problem in **MCM** is sensitive to the cardinality of $Dec$. If $|Dec| = 1$, then whatever we write it, $\mathtt{t}(0), \mathtt{t}(?)$ etc., it trivially equals $\top$. Then the whole logic collapses to mono-dimensional, which makes every MCM just a CM. Therefore, what interesting is when $|Dec| > 1$.

It can be shown that the problem is PSPACE-hard already when $|Dec| = 2$. The reason is that, in product modal logic at latest from Marx [Marx 1999] we know that two S5-operators can mimic a K-operator. We can thus define $[K]\varphi := \blacksquare(\mathtt{t}(0) \to \Box(\mathtt{t}(1) \to \varphi))$. To see that it is indeed a K-operator, one can show that axioms like T, D, B, 4 are not valid.

Different with [Marx 1999] who used arbitrary two variables $p, q$ into the abbreviation, it seems that we have to use decision atoms, otherwise the formula reduces and fails to invalidate those axioms, for the formula will "collapese" because of axioms $p \to \blacksquare p$ and $\neg p \to \blacksquare \neg p$. This is one reason that at least two decision atoms are needed.

On the other hand, the normal proof strategy for NEXPTIME-hardness appears unable to be directly brought into **MCM**. In [Marx 1999], Marx showed the

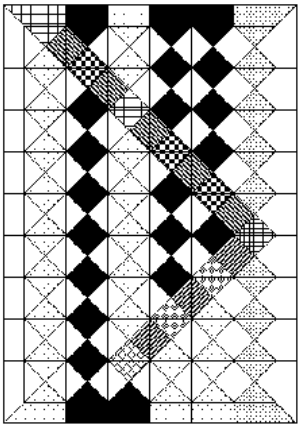


Figure 7.1: An example in [van Emde Boas 1997] of the tiling representing the computation 11+1=12

NEXPTIME-hardness of the satisfiability problem in $S5^2$ by reduction of a tiling problem. (We will introduce tiling problems soon.) The key difference is that, while Marx can use infinite supply of atomic propositions to reduce the increasing $2^n \times 2^n$ tiling, in our case it reaches a limit since $|Dec|$ is a finite number. Hence I will work with a more modest goal, namely to prove an EXPTIME lower-bound.

### 7.1.2 Tiling problems

Tiling problem, or domino problem, is a mathematical problem first proposed by logician and philosopher Wang Hao. A Wang tile, or Wang domino, is a square with four colors on its four sides. Tiling problems take the form as "given a set of tile types and a plane, can they tile it in such a way that, the adjacent sides of any two tiles share the same color?". It has been shown that various Turing machines have a visualized way to reduce to corresponding tiling problems. Therefore tiling problems have different computability and complexity with respect to 1) the size of the plane, e.g. infinite columns and rows, $n \times n$ square, $m \times n$ corridor, etc., 2) the type of the game, i.e. whether it is a single player or two-player game and what the winning condition is.

In virtue of the variety of computational degree of tiling problems, they are used widely in the complexity of modal logics. Van Emde Boas [van Emde Boas 1997] argues that using tiling problem in complexity is a *master reduction*, in the sense that it provides "an intended directly visible correspondence between records of accepting computations for the computational problem and solutions of the combinatorial problem under consideration". The reduction of Turing machine is a prime example of master reduction. Tiling problems, because of their visualization, are also good tools of master reductions. We will reduce the satisfiability problem in **MCM** to the tiling problem below.

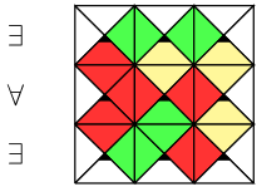



Figure 7.2: The two-player tiling game example in [Schwarzentruber 2019]

**Two-player tiling game** The tiling problem that we aim to reduce is a two-player game. Each *role* is tiled by a player and the two players take turns. We call the first player Eloise and the second Abelard (nicknames for $\exists$ and $\forall$ respectively). The winning condition for Eloise is to tile a $2^n \times n$ corridor, where $n$ is given by the input. Namely, $2^n$ rows and $n$ columns. For technical reason, we suppose there are a 0th row, a 0th column and an $n+1$th column, and they are previously tiled (by a third party and not shown in the plane) by a white domino tile type.

We say that Eloise has a *winning strategy*, if whatever Abelard tiles, Eloise can tile after Abelard's row until finish tiling the corridor. The complexity of *deciding whether Eloise has a winning strategy for this tiling game* is EXPTIME-complete.

Recall that Eloise has a winning strategy, if there is a game tree such that

1. each node denotes a tiling of a row;
2. all nodes in the odd levels (including the root) belong to Eloise, the rest to Abelard;
3. every node encodes a tiling of a row such that if a node is a successor of another one, then these two rows are compatible in all their tiles;
4. at each level of the tree, all possible tiling of Abelard are encoded by a node;
5. the tree has $2^n$ depth.

Every branch from the root to a leaf is called a game. The game tree guarantees that whatever Abelard plays, Eloise can keep tiling to let the game go on until the $2^n - 1$-st round.

### 7.1.3 Tiling for lower bound

**Variables and operator needed** The basic thought is that $S$ consists of two disjoint sets of states, *tiling states* and *counting states*, that tell which tile in a cell of a row and which step the current row is respectively. Each member of $F_S$ represents a row of a tiling game – and hence a node of the intended game tree.

The variables we need are partitioned as $\{eloise, aberlard\} \uplus \{p_\#, p_{til}\} \uplus \{p_i : 1 \leq i \leq n\} \uplus \{pos_i : 0 \leq i \leq n+1\} \uplus \{p_d : d \in T\} \uplus \{col_i(d) : 0 \leq i \leq n+1, d \in T\} \subseteq Atm_0$

and $\{\mathtt{t}(1), \mathtt{t}(0)\} \subseteq Dec$.[1]

Clearly, *eloise* and *aberlard* refer to the two players. The intended meanings of $p_\#$ and $p_{til}$ are to denote whether a state is a counting state or tiling state, i.e. the state conveys the information of counting (of the whole row), or the information of tiling (of some cell). $T$ stands for the set of all *d*omino tile types. We always assume a special domino type $p_{white} \in T$, which has all four sides colored white. This domino type tiles exclusively the cells in the 0th row, the 0th column, and the $n+1$st column, which mark the boundary of the plane. The variable $col_i(d)$ has the reading "the *previous* tiling of position $i$ is domino type $d$". *Dec* has double roles regarding tiling and counting states which we will explicate soon.

**Intuition** Suppose Eloise has a winning stragtegy, viz, there is a game tree for her. We want the reduction formula (of the game tree) defined later satisfied in the desired model defined later $\Gamma = (S, F_S)$, such that each $f \in F_S$ contains the following information

1. WHO tiles the row;
2. HOW (s)he tiles each cell
   - for any position $pos_i$, a domino type is put $p_d$, only if it fits the right-side's color of $col_{i-1}(d')$ (i.e. the $i-1$-st column's tile) and the top-side's color of $col_i(d'')$ (i.e. the $i$-th column's tile *of the previous row* $d''$);
3. WHICH round the game is at.

Therefore there are two types of states in $S$, responsible for tiling and for counting respectively. And/but they work in different ways regarding $f$. For any tiling state $s$, $f(s) = 1$ means that state is "activated" in order to tell us which position to put what domino type; thus $f(s) = 0$ means "inactivated". For counting states the interpretation of classification is different. We will prepare $n$ counting states each representing some natural number $1 \leq i \leq n$,[2] i.e. a counting state is like $\{p_\#, p_i\}$. Then $f(\{p_\#, p_i\}) = 1$ means the $i$-th digit has number one and $f(\{p_\#, p_i\}) = 0$ means it has number zero in the row $f$.

It would be a shame if I said tiling problems are easily visualized without providing an illustration here. Figure 7.3 exemplifies the intuition. The left part is just the example in [Schwarzentruber 2019] of the tiling game we use. We take a dynamic perspective to construct a tiling game row by row, cell by cell. In the right part we concentrate on the central cell, namely the second row of the second column. Suppose that all the cells before it are already tiled and now is its turn. Since it is the second row, it is Abelard's round. Three possibilities of tiling will be represented as the following states in the desired model

[1] The difference of $i$ ranging from 1 through $n$ and from 0 through $n+1$ is only for the technical trick of letting the 0th row, the 0th and $n+1$st columns be tiled by $p_{white}$.

[2] Do not confuse this with the $i$ of $pos_i$.

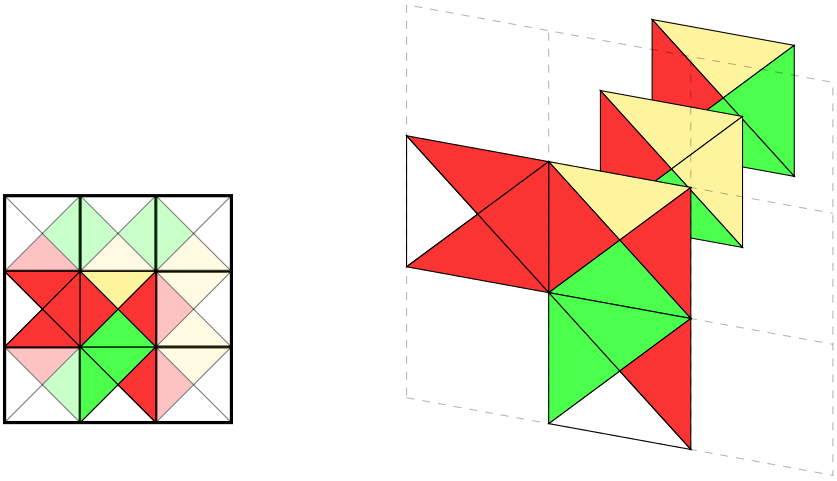


Figure 7.3: Tiling the center cell

$$s = \{p_{til}, abelard, pos_2, col_1(p_{rrrw}), col_2(p_{grwg}), p_{yrgr}\}$$

$$s' = \{p_{til}, abelard, pos_2, col_1(p_{rrrw}), col_2(p_{grwg}), p_{yygr}\}$$

$$s'' = \{p_{til}, abelard, pos_2, col_1(p_{rrrw}), col_2(p_{grwg}), p_{yggr}\}$$

where domino types here are named after their colors clockwise as up-right-down-left. For instance, $p_{yrgr}$ means the up-color is yellow, right-color red, down-color green and left-color red. All three states are legal, since the colors match. However, in every classifier, which intends to encode a row, at most one of them can be classified as 1. In our case it is $f(s) = 1$ while $f(s') = f(s'') = 0$.

Also $f$ must encode the information of which round it is at. Hence we shall have two counting states $\{p_{\#}, p_1\}, \{p_{\#}, p_2\}$ s.t. $f(\{p_{\#}, p_1\}) = 1$ and $f(\{p_{\#}, p_2\}) = 0$ encoding binary 01. It is the decimal 1 which represents round 2.

Thus we finish the construction of $f$. The desired MCM is a model of many such $f$s. It satisfies the game tree for Eloise's winning strategy. A game is therefore a branch from the root to a leaf with depth $2^n$, which is in turn a series of classifiers $f_0, f_1, \dots f_{2^n}$ s.t. $f_0$ denotes root (i.e. the initial row that Eloise tiles), and for each $i > 0$, $f_i$ encodes the tiling information of itself and $f_{i-1}$ (because of variables $col_i(d'')$). Since Eloise has a winning strategy, whatever Abelard tiles, a game will be done.

**Reduction** So the formula consists of the following parts, recall that we define $[K]\varphi := \blacksquare(\mathsf{t}(0) \to \square(\mathsf{t}(1) \to \varphi))$ and use it to denote "in all the possible tilings for the next round, $\varphi$ holds".

0. Each state represents either a number of row or a tiling of position

$$\blacksquare\square((p_{\#} \vee p_{til}) \wedge \neg(p_{\#} \wedge p_{til})) \tag{7.1}$$

1. Tiling

   (a) each cell position $i$ encodes exactly one tile, one previous tile for $i$ and one previous tile for $i-1$ [3]

$$\blacksquare\Box(p_{til}\wedge \mathtt{t}(1)) \leftrightarrow \bigvee_{i,d_1,d_2,d_3} \Big( (pos_i \wedge p_{d_3} \wedge col_{i-1}(d_2) \wedge col_i(d_1)) \wedge \bigwedge_{(i',d'_1,d'_2,d'_3)\neq(i,d_1,d_2,d_3)} \neg(pos_{i'} \wedge p_{d'_3} \wedge col_{i-1}(d'_2) \wedge col_i(d'_1)) \Big) \tag{7.2}$$

   (b) start configuration: all previous tiles are $p_{white}$ and the number of row is 0

$$\bigwedge_i \Diamond(pos_i \wedge p_{til} \wedge col_i(p_{white}) \wedge col_{i-1}(p_{white}) \wedge \mathtt{t}(1) \wedge eloise) \wedge \bigwedge_i \Diamond(p_i \wedge p_\# \wedge \mathtt{t}(0)) \tag{7.3}$$

   (c) color matches both horizontally and vertically, where $right(d')$ denotes the set of dominos whose left-color is the right-color of $d'$, and similar to $top(d'')$

$$\blacksquare\Box \bigwedge_{i,d',d''} ((p_{til} \wedge pos_i \wedge \mathtt{t}(1) \wedge col_i(d'') \wedge col_{i-1}(d')) \rightarrow \bigvee_{d\in right(d')\cap top(d'')} p_d) \tag{7.4}$$

   (d) The *current* tiling of the current position in the current row is the *previous* tiling of the current position in whatever *next* row

$$\blacksquare\Box \bigwedge_{i,d} ((p_{til} \wedge pos_i \wedge \mathtt{t}(1) \wedge p_d) \rightarrow [K]\Box((p_{til} \wedge pos_i \wedge \mathtt{t}(1)) \rightarrow col_i(d))) \tag{7.5}$$

2. Counting: we use $p_{i,\#,c}$ to denote $(p_i \wedge p_\# \wedge \mathtt{t}(c))$ with $1 \leq i \leq n, c \in \{1,0\}$

   (a) every digit of every row has at least a value

$$\blacksquare \bigwedge_{1\leq i\leq n} \Diamond(p_{i,\#,1} \vee p_{i,\#,0}) \tag{7.6}$$

[3] To save space we write $\bigvee_i$ instead of $\bigvee_{1\leq 0\leq n+1}$, $\bigvee_d$ instead of $\bigvee_{d\in T}$ for all the formulas below, and similar for the conjunction cases. We will write explicitly $\bigwedge_{1\leq i\leq n}$ when $i$ ranges from 1 through $n$ instead of from 0 through $n+1$.

(b) start counting with 0

$$\Box \bigwedge_{1\leq i\leq n} ((p_i \wedge p_\#) \to \mathsf{t}(0)) \tag{7.7}$$

(c) in an incremental way

$$inc := \blacksquare \bigwedge_{1\leq k\leq n} \Big(\Diamond p_{k,\#,0} \wedge \bigwedge_{1\leq i<k} \Diamond p_{k,\#,1}) \to \Big([K]\Box((p_k \wedge p_\#) \to \mathsf{t}(1)) \wedge \bigwedge_{1\leq i<k} \Box((p_k \wedge p_\#) \to \mathsf{t}(0)) \wedge \bigwedge_{n\geq i>k} store(p_i)\Big)\Big) \tag{7.8}$$

where

$$store(p_i) := \bigwedge_{c\in\{0,1\}} (\Diamond p_{i,\#,c} \to [K]\Box((p_k \wedge p_\#) \to \mathsf{t}(c)))$$

(d) not terminates before $2^n - 1$

$$\blacksquare([K]\bot \to \bigwedge_{1\leq i\leq n} \Diamond p_{i,\#,1}) \tag{7.9}$$

3. Players

(a) Each whole row belongs to a player

$$\blacksquare(\Diamond(eloise \wedge \mathsf{t}(1)) \to \Box(abelard \to \mathsf{t}(0))) \tag{7.10}$$

(b) players alternate

$$\begin{aligned}\blacksquare(&\Diamond(eloise \wedge \mathsf{t}(1)) \to [K]\Box(abelard \to \mathsf{t}(1)) \wedge \\ &\Diamond(abelard \wedge \mathsf{t}(1)) \to [K]\Box(eloise \to \mathsf{t}(1)))\end{aligned} \tag{7.11}$$

(c) winning strategy: Abelard can play whatever possible until the $2^n$-th round

$$\begin{aligned}\Big(\blacksquare\Box \bigwedge_{i,d',d''} &(eloise \wedge pos_i \wedge \mathsf{t}(1) \wedge col_i(d'') \wedge col_i(d')) \\ &\to \bigwedge_{d\in right(d')\cap top(d'')} \langle K\rangle\Diamond(pos_i \wedge \mathsf{t}(1) \wedge p_d))\Big) \vee \bigwedge_{1\leq i\leq n} \Diamond p_{i,\#,1}\end{aligned} \tag{7.12}$$

**The desired model (for a game tree)** If Eloise has a winning strategy in the two-player corridor tiling game we present above, then there exists a game tree, so that Eloise tiles according to its instruction.

Given that game tree, the desired model $\Gamma = (S, F_S)$ is defined as follows

- $S = S_{til} \cup S_{\#}$ is partitioned as sets of tiling and counting states respectively, where
  - $S_{til} = \{\{p_{til}, \pi, pos_i, p_d, col_{i-1}(d'), col_i(d'')\} : \pi \in \{eloise, abelard\}, 0 \leq i \leq n+1, d, d', d'' \in T\}$
  - $S_{\#} = \{\{p_{\#}, p_i\} : 1 \leq i \leq n\}$
- $f \in F_S$, if and only if there is a node in the game tree, which encodes a row, such that
  - it is played by player $\pi$ iff $(\exists s \in S, \pi \in s, f(s) = 1)$ implies that $(\forall s \in S,$ if $\pi' \in s$ then $f(s) = 0)$, for $\pi, \pi' \in \{eloise, abelard\}$ and $\pi \neq \pi'$.
  - for every row which is tiled by some $\pi$, every $i$-th column of the row, it is tiled by $d$, the $i-1$-st of the row tiled by $d'$, the $i$-th column of its previous row tiled by $d''$, iff $f(\{p_{til}, \pi, pos_i, p_d, col_{i-1}(d'), col_i(d'')\}) = 1$
  - it is at the $k$-th level of the tree, iff the sequence $f(p_n)f(p_{n-1})\dots f(p_1)$ binary encodes $k-1$.[4]

How many states we have? Exactly $2n^3|T|^3 + n$ many, where they refer to the number of tiling and counting states respectively. Hence how many possible functions in $F_S$ at most? We have $2^n \times 2 \times |pos_1| \times \dots |pos_n|$, where $2^n$ is the possible numbers $n$ in binary encodes; 2 the number of players; $|pos_i| = \{s : pos_i \in S\}$ the tiling states encoding the tiling of position $i$. For each $i$, $|pos_i| \leq |T| \times |T| \times |T|$, namely the choices of $d, d', d''$ of $p_d, col_i(d''), col_{i-1}(d')$. How many pointed model, i.e. the pair $(s, f)$ we have? It is $(n + 2n^2(n-1)|T|^3) \times 2^{n+1}|T|^{3n}$, exponential in $n$.

It is not hard to see that the model indeed satisfies the game tree. Let us take a look anyways by checking conditions one by one. The root of the tree is a row, which is encoded by a classifier $f$, that belongs to Eloise. And $\forall s \in S_{\#}, f(s) = 0$, therefore it represents $000\dots0$ ($n$ many), which encodes 0; $\forall pos_i, \exists! s \in S, s = \{eloise, pos_i, col_{i-1}(white), col_i(white)\}$ and $f(s) = 1$. It indicates that all the previous tiling is the white one, which fits our supposition of the additional white tiling. Moreover, for that $s$, $p_d \in s$ if and only if Eloise tiles domino type $d$ on cell $i$ at the root of the game tree. This is the first round of a tiling game.

In round two, Abelard has many possible returns. Each of them will be encoded by a classifier $f' \in F_S$, such that the number $f'$ encodes is 1, namely $00\dots01$. Now, every legal tile of Abelard should be compatible with the tiling in $f$. That is to say, for any $pos_i$, if in $f$ we have $f(s) = 1$ for some $s$ s.t. $eloise, pos_i, p_d \in s$ for some $i, d$, then we shall have $f'(s') = 1$ for some $s'$ s.t. $abelard, pos_i, col_i(d) \in s'$. And moreover, if $col_{i-1}(d'), p_{d''} \in s'$, then it must be $d \in top(d'') \cap right(d')$.

For all the rest of the rounds it is similarly checked one by one. Finally, the tree ends up with nodes which have depth $2^n$, while the model has classifiers which encode the number $2^n - 1$, and no classifier shall encode the number larger than it.

[4]Notice again we start counting game rounds from 1 while the binary encoding starts from 0.

**Lemma 7.1.** *Eloise has a winning strategy in the two-player $2^n \times n$ corridor tiling game if and only if the reduction formula is satisfiable in* **MCM**.

*Proof.* Suppose Eloise has a winning strategy. Then there shall be a game tree for Eloise, which is satisfied by the desired model for it. We need show that the model also satisfies the reduction formula $\varphi$. Obviously, the tiling and counting conjuncts are satisfied, otherwise either some row is not legally tiled or some player cannot tile after some $k$-round for $1 \leq k < 2^n$. In both cases the game tree would not exist. For the player conjunct, that each player holds a whole row and players alternate are clearly satisfied. To see the winning strategy formula is satisfied, notice that if it is not satisfied in the desired model, then either Abelard has some move in some $k$-th round ($1 \leq k < 2^n$) that is "unexpected" by the game tree, or the game does not terminate after $2^n$ rounds. In both case Eloise would have no winning strategy.

Let the formula be satisfiable. We first reduce any model satisfying the formula to a tree model in terms of the $K$-accessibility relation. Then Eloise starts the game from the pointed model that satisfies the formula. Whatever Abelard plays in his round, say $k$ with $1 \leq k < 2^n$, what Eloise need do is to find a classifier $f'$ in the model whose number is $k-1$ and whose tiling is exactly what Abelard tiles. Then, she chooses any of its $K$-successor $f''$, and tiles as $f''$'s instruction. The formula guarantees that whatever Abelard tiles, Eloise does not stop tiling before the $2^n$-th round. □

With the lemma above and the knowledge that the two player tiling game is EXPTIME-complete, we have the main result of this section.

**Theorem 7.1.** *Let $Atm_0$ be infinite. Then deciding satisfiability of $\mathcal{L}$-formulas relative to the class* **MCM** *is EXPTIME-hard.*

## 7.2 Finitely Definite Classifier Models

### 7.2.1 Motivation

By finite definiteness we mean a property of classifier being $X$-definite for some finite $X \subseteq Atm_0$ (recall Definition 3.4). In this section we deal with the class of finitely definite CMs and the classs of finitely definite MCMs respectively, where finite definiteness will be defined as $X$-definite for some finite $X$. The main technical challenge is to find proper axiomatics. It will turn out that all we need is not an axiom but a rule of inference informally saying that

*if the state is not $X$-definite for all finite $X$, then falsum.*

To prove the completeness results we, as usual, will make use of decision models (DMs) and multi-decision models (MDMs). Though the main part of this section will be proofs, the interests are both technical and conceptual.

**Bridge the axiomatic gap** In the past chapters we have provided several modal logics for classifier systems including (single) classifiers and multi-classifiers. In both axiomatics a salient inconvenience is that the axiomatics are always presented as two instead of a unified one, namely cases of finite variables and of countably infinite variables. As we mentioned, the focus is always on the axiom **Funct**, which informally says that

$$\textit{there is an instance } s \textit{ with classification } c \to \Box(\textit{inputting } s \textit{ outputs } c)$$

In the finite-variable case this axiom guarantees the property of functionality, i.e. every CM is indeed a function. Nonetheless, in the infinite-variable case such a proposition would be illegal (not a well-formed formula) because an instance $s$ would be represented as a conjunction of infinite literals. In other words, there the axiomatics cannot capture functionality from a syntactic viewpoint.

By contrast, we will show that in light of the rule of inference informally given above and formalized later, the axiomatics for finitely definite (single or multi-) classifier models is unified regardless of the cardinality of $Atm_0$. This is because **Funct** will be derived from the rule when $Atm_0$ is finite.

**Possible syntactic expression** We have briefly mentioned that the inner mechanism of a binary-input classifier can be recognized as its syntactic expression, i.e. some formula that expresses it. In our language, when $Atm_0$ is finite we can express a classifier $f$ by the following formula

$$\bigwedge_{s\in dom(f)} \Diamond \mathsf{cn}_{s,Atm_0} \wedge \bigwedge_{s'\notin dom(f)} \Box\neg \mathsf{cn}_{s',Atm_0} \wedge \bigwedge_{(s,c)\in f} \Box(\mathsf{cn}_{s,Atm_0} \to \mathsf{t}(c)) \tag{7.13}$$

which one may call a modal normal form of $f$. The first two conjuncts encode the domain of $f$, while the third one encodes the mapping of $f$.

If the classifier is $X$-definite, even if in the infinite-variable case, we have the expression for $f$

$$\bigwedge_{Y\subseteq X, c\in Val, \exists (s,c)\in dom(f), s\cap X=Y} \Big(\Diamond \mathsf{cn}_{Y,X} \wedge \Box(\mathsf{cn}_{Y,X} \to \mathsf{t}(c))\Big) \wedge \bigwedge_{Z\subseteq X, \forall s\in dom(f), s\cap X\neq Z} \Box\neg \mathsf{cn}_{Z,X}. \tag{7.14}$$

which can be seen as a relativized version of the formula above.

The problem is when $f$ is not finitely definite, as we pointed out in Fact 3.3 by giving an easy diagonal proof. Those classifiers cannot be expressed by any formula, since again the illegal infinite conjunction would be yielded.[5] Therefore, by restricting to finitely definite classifiers we can ensure that for any such classifier

[5] One can even argue that those classifiers have in principle unkownable inner mechanisms and are unexplainable. We will leave this as a further topic.

there always exists some formula to express it, which is what happens in real life.[6]

**Strong completeness** Last, the technical benefit is that we can expect a strong completeness result, instead of the previous weak one for the target aximatics. The reason is that, take the one-dimensional logic for example, we used QDM (quasi-decision model) instead of decision model directly to do the canonical model argument for the completeness proof. Then, we showed that for any $\varphi$, if it is satisfied in some QDM, it is satisfied in some DM. Since the transformation from QDM to DM depends on the fixed $\varphi$, eventually we only reached $\models_{\mathbf{DM}} \varphi \Longrightarrow \vdash_{\mathsf{BCL}} \varphi$ but not $\Phi \models_{\mathbf{DM}} \varphi \Longrightarrow \Phi \vdash_{\mathsf{BCL}} \varphi$, i.e. weak rather than strong completeness for **DM** (recall Definition 2.29), and consequently for **CM**.

The difficulty is that when $Atm_0$ is infinite, we cannot guarantee the "functionality" from the proof theory. The *quasi*-DM is exactly inferior in this property and thus easier to handle. Nevertheless, with the inference rule guaranteeing the finite definiteness, we no more need QDMs but can directly work with DMs. For this reason we can enhance the completeness result from weak to strong. The resulting logic has therefore compactness. The case of multi-classifiers is similar.

### 7.2.2 Semantics

In Chapter 3 and Chapter 6, we dealt with different languages. Here for the notional convenience, we use $\mathcal{L}$ to denote the language of $\mathsf{PLC}$, and $\mathcal{L}^{\setminus\blacksquare}$ to denote the language of $\mathsf{BCL}$.

**Definition 7.1** (Finitely definite classifier)**.** *For any classifier $f : S \longrightarrow Val$, we say $f$ is* finitely definite*, if $\exists X \subseteq^{\text{fin}} Atm_0$, s.t. $\forall s, s' \in S$, if $s \cap X = s' \cap X$ then $f(s) = f(s')$.*

**Definition 7.2** (Finitely definite CM)**.** *A finitely definite CM is a model $C = (S, f)$ where $f$ is finitely definite. The class of finitely definite MCMs is noted as $\mathbf{CM}_{fd}$.*

**Definition 7.3** (Finitely definite DM)**.** *A finitely definite DM $M = (W, \sim_{\square}, V)$ is a DM, if it satisfies the following constraint*

**C$_{\mathbf{d}}$** $\forall w \in W, \exists X \subseteq^{\text{fin}} Atm_0$ *s.t.* $\forall v \in W$, *if* $V_X(w) = V_X(v)$ *then* $V_{Dec}(w) = V_{Dec}(V)$.

*The class of finitely definite MCMs is noted as $\mathbf{DM}_{fd}$.*

**Definition 7.4** (Finitely definite MCM)**.** *Let $\Gamma = (S, F_S)$ be an MCM. We say that $\Gamma$ is finitely definite, iff $\exists X \subseteq^{\text{fin}} Atm_0$ s.t. $\forall f \in F_S, f$ is $X$-definite. The class of finitely definite MCMs is noted as $\mathbf{MCM}_{fd}$.*

**Definition 7.5** (Finitely definite MDM)**.** *Let $M = (W, \sim_{\square}, \sim_{\blacksquare}, V)$ be an MDM. We say that $M$ is is finitely definite, if it satisfies the following constraint*

[6] To be more precise, real life is that we have $Atm_0$ neither fixed nor infinite, but unbounded and finite.

**C$_{\mathbf{md}}$** $\exists X \subseteq^{\text{fin}} Atm_0$ *s.t.* $\forall w, v \in W$, *if* $w \sim_{\Box} v$ *and* $V_X(w) = V_X(v)$, *then* $V_{Dec}(w) = V_{Dec}(v)$.

*The class of finitely definite MDMs is noted as* $\mathbf{MDM}_{fd}$.

The satisfaction relation and validity for each modal classes here are defined in exactly the same way as previous, and hence omitted. The theorem below is proven in the same way as we did previously, and hence omitted.

**Theorem 7.2.** *We have the following equivalence results:*

- *let* $\varphi \in \mathcal{L}^{\setminus\blacksquare}$, *then* $\varphi$ *is satisfiable in* $\mathbf{CM}_{fd}$ *if and only if* $\varphi$ *is satisfiable in* $\mathbf{DM}_{fd}$;
- *let* $\varphi \in \mathcal{L}$, *then* $\varphi$ *is satisfiable in* $\mathbf{MCM}_{fd}$ *if and only if* $\varphi$ *is satisfiable in* $\mathbf{MDM}_{fd}$.

### 7.2.3 Axiomatics and strong completeness for single classifiers

In this subsection our language is restricted to the language of BCL.

Axiomatically the finite definiteness is characterized by not an axiom, but a rule of inference.

$$\frac{\varphi \to \neg \mathsf{Defin}(X) \text{ for all } X \subseteq^{\text{fin}} Atm_0}{\Box\neg\varphi} \qquad (\textbf{Findef})$$

Recall that in Definition 3.2, we have

$$\mathsf{Defin}(X) =_{def} \bigwedge_{c \in Val} \Box\Big((\mathsf{t}(c) \to [X]\mathsf{t}(c)) \wedge (\neg\mathsf{t}(c) \to [X]\neg\mathsf{t}(c))\Big)$$

which can be simplified as

$$\bigwedge_{c \in Val} \Box\Big(\langle X\rangle\mathsf{t}(c) \to [X]\mathsf{t}(c)\Big).$$

The rule we propose is a kind of *infinitary inference rule* which has been originally studied in logics with names in e.g. [Gargov *et al.* 1987], [Gargov & Goranko 1993] and [Lorini 2019].

**Definition 7.6.** *We write* BCL + **Findef** *for the axiomatics resulting from adding* **Findef** *into* BCL.

The definitions below and following in this subsection are all defined relative to BCL + **Findef**, and we omit all the subscripts BCL + **Findef** since the context is clear.

**Definition 7.7** ($\vdash$)**.** *Let* $\Delta$ *be a set of formulas and* $\varphi$ *a formula. We use* $\Delta \vdash \varphi$ *to denote that* $\varphi$ *is derivable from* $\Delta$ *with the axioms and inference rules in* BCL + **Findef**; $\Delta \nvdash \varphi$ *to denote that it is not the case* $\Delta \vdash \varphi$; $\vdash \varphi$ *means that* $\varphi$ *is a theorem of* BCL + **Findef**.

#### 7.2.3.1 Axiom Funct is derivable

As we mentioned, an advantage of this framework is unifying two former axiomatics w.r.t. different cardinality of $Atm_0$. This can be established by the fact that the only axiom differing the two, namely **Funct**, turns to be derivable here.

**Lemma 7.2.** $\vdash \mathsf{Defin}(X) \to \mathsf{Defin}(X \cup Y)$

*Proof.* It is easy to prove that $\langle X \cup Y\rangle\varphi \to \langle X\rangle\varphi$ and $[X]\varphi \to [X \cup Y]\varphi$ for any $X, Y \subseteq^{\text{fin}} Atm_0$ and formula $\varphi$. Thus together with $\langle X\rangle\mathsf{t}(c) \to [X]\mathsf{t}(c)$ for any $c \in Val$ which is derived from $\mathsf{Defin}(X)$ and modus ponens, we have $\bigwedge_{c\in Val} \Box(\langle X \cup Y\rangle\mathsf{t}(c) \wedge \mathsf{Defin}(X)) \to [X \cup Y]\mathsf{t}(c)$, which is what we want. □

**Proposition 7.1.** $\vdash$ **Funct**, *when the language has* $|Atm_0|$ *finite.*

*Proof.* Followed from the lemma above it is easy to prove $\vdash \neg$**Funct** $\to \neg\mathsf{Defin}(X)$ for all $X \subseteq Atm_0$, therefore $\vdash \Box\neg\neg$**Funct**, which by $\mathbf{T}_\Box$ follows $\vdash$ **Funct**. □

#### 7.2.3.2 Proof of strong completeness

We give the canonical model proof for the completeness of $\mathsf{BCL}$ + **Findef** relative to $\mathbf{DM}_{fd}$ in detail. The completeness relative to $\mathbf{CM}_{fd}$ follows as a corollary in virtue of Theorem 7.2. As usual it starts with the definition of maximal consistent set (relative to the current logic, i.e. $\mathsf{BCL}$ + **Findef**).

**Definition 7.8** (MCT)**.** *A theory is a set of formulas containing all axioms above and closed under* **MP**, **Nec**, **Findef**. *A consistent theory is a set of formulas does not contain* $\bot$. *A maximal consistent theory (MCT)* $\Delta$ *is a set of formulas s.t. for any consistent theory* $\Delta'$, *we have* $\Delta \not\subset \Delta'$.

The main difference between the standard proof and ours here is the Lindenbaum-type lemma. We have to extend the consistent set of formulas more carefully to ensure that at least one $\mathsf{Def}(X)$ is in the eventual MCT, so that **Findef** will not be "triggered" to cause inconsistency.

**Lemma 7.3** (Lindenbaum-type)**.** *Let* $\Delta_0$ *be a consistent theory and* $\psi \notin \Delta_0$. *Then, it can extend to an MCT* $\Delta$, *s.t.* $\psi \notin \Delta$.

*Proof.* The proof is in the same spirit of the works of Gargov, Passy and Tinchev, but easier. The proof structure is constructing a chain of extensions from $\Delta_0$. The main concern will be to avoid that every $\Delta_i$ is consistent, while $\bigcup_{i\in\mathbb{N}} \Delta_i$ is inconsistent using **Findef**. We can avoid it by adding some $\mathsf{Defin}(X)$ which is consistent with $\Delta_0$.

Claim: for all theory $\Delta'$, $\exists X \subseteq^{\text{fin}} Atm_0$ s.t. $\neg\mathsf{Defin}(X) \notin \Delta'$.

Suppose not towards a contradiction. Then we have $\neg\mathsf{Defin}(X) \in \Delta'$ for all $X \subseteq^{\text{fin}} Atm_0$. Since $(\top \to \neg\mathsf{Defin}(X)) \leftrightarrow \neg\mathsf{Defin}(X)$ is a theorem, by **MP** we have $\top \to \neg\mathsf{Defin}(X) \in \Delta'$ for all $X \subseteq^{\text{fin}} Atm_0$. Hence by **Findef** and **T** we have $\neg\top \in \Delta'$, i.e. $\bot \in \Delta'$, which contradicts the assumption that $\Delta'$ is consistent.

So we extend $\Delta_0$ step by step as follows. Let $Th(\Delta_i \cup \{\varphi_i\}) =_{\mathsf{Defin}} \{\chi : \Delta_i \cup \{\varphi_i\} \vdash \chi\}$.

- $\Delta_1 = Th(\Delta_0 \cup \{\neg\psi\})$;
- $\Delta_2 = Th(\Delta_1 \cup \{\mathsf{Defin}(X')\})$ for some $X' \subseteq^{\text{fin}} Atm_0$ s.t. $\neg\mathsf{Defin}(X') \notin \Delta_1$, which necessarily exists as we showed;
- Enumerate all formulas as $\varphi_3, \ldots, \varphi_n, \ldots$, and $\forall\Delta_{i+1}$ for $i > 2$,
  - $\Delta_{i+1} = Th(\Delta_i \cup \{\varphi_i\})$, if the latter is consistent;
  - $\Delta_{i+1} = \Delta_i$, otherwise;
- $\Delta = \bigcup_{i\in\mathbb{N}} \Delta_i$.

We claim that $\Delta$ is an MCT. All others are the same as standard Lindenbaum lemma proof, the only interesting potential danger is that $\Delta$ contains $\bot$ by using **Findef**. That is to say, there exists some $\varphi$ s.t. $\varphi \to \neg\mathsf{Defin}(X) \in \Delta$ for all $X \subseteq^{\text{fin}} Atm_0$, and $\varphi \in \Delta$.

We show this is impossible by contradiction. Suppose there were such a $\varphi$. First, we know by the claim before and the construction, $\Delta_2$ is consistent. Then, we would have $\varphi_m := \varphi \to \neg\mathsf{Defin}(X'), \varphi_n := \varphi$ with some $m, n > 1$. By the construction, $\varphi_m \in \Delta_m, \varphi_n \in \Delta_n$. Then, we would derive a contradiction by **MP** in $\Delta_{max\{m,n\}}$, since it contains both $\mathsf{Defin}(X')$ and $\neg\mathsf{Defin}(X')$. But it contradicts the construction of $\Delta_i$. □

**Definition 7.9** (Canonical Model)**.** *We define the canonical finitely definite DM* $M^c = (W^c, \sim^c_\Box, V^c)$ *as follows:*

- $W^c = \{w : w \text{ is an MCT}\}$
- $\forall w, v, w \sim^c_\Box v \iff \{\Box\varphi : \Box\varphi \in w\} = \{\Box\varphi : \Box\varphi \in v\}$
- $V^c(w) = w \cap Atm_0$.

**Lemma 7.4** (Indeedness)**.** *$M^c$ is indeed a finitely definite DM.*

*Proof.* Comparing with the previous lemma for QDM in Chapter 3, we have two more properties to prove. Let us begin by dealing with $\mathbf{C_d}$. we need show that it is captured by **Findef**. Suppose the latter is not satisfied, then there is an MCT $w$, s.t. $\forall X \subseteq^{\text{fin}} Atm_0$, there is another MCT $v$ s.t. $w \sim_\Box v, V_X(w) = V_X(v)$ but $V_{Dec}(w) \neq V_{Dec}(v)$. Let $V_{Dec}(w) = \{\mathsf{t}(c)\}$. We have that $\neg\mathsf{Defin}(X) \in w$ for all $X \subseteq^{\text{fin}} Atm_0$. Since $\neg\mathsf{Defin}(X) \leftrightarrow (\top \to \neg\mathsf{Defin}(X))$ is a theorem, we have $\top \to \neg\mathsf{Defin}(X) \in w$ for all $X \subseteq Atm_0$. Since $w$ is closed under **Findef**, we have $\Box\neg\top \in w$, which derives $\bot \in w$, contradicting that $w$ is consistent. Then we need show it satisfies the constraint **C4** of DM, recall that requires if $w \cap Atm_0 = v \cap Atm_0$ then $w \cap Dec = v \cap Dec$. This is however obvious now in light of **Findef**. □

The proof of the existence lemma below is a bit different from the normal one. Since we have **Findef**, there are two possible ways to derive the inconsistency instead of one. And we need prove both are actually impossible.

**Lemma 7.5** (Existence)**.** *Let $M^c$ be the canonical DM. Then $\forall w \in W, \Diamond\varphi \in w$, $\exists v \in W$ s.t. $w \sim_{\Box} v$ and $\varphi \in v$.*

*Proof.* We claim the set $w^{\Box} := \{\Box\psi : \Box\psi \in w\}$ is consistent with $\{\varphi\}$. Suppose not towards a contradiction, two possibilities. First, there are some $\vdash (\Box\psi_1 \wedge \cdots \wedge \Box\psi_n) \to \neg\varphi$ without using **Findef**, where $\forall i \in \{1, \ldots n\}, \Box\varphi_i \in w^{\Box}$. Then using **Nec**, **K**, **T** and **MP** it is not hard to derive $\Box\neg\varphi$ from $w^{\Box}$, therefore $\Box\neg\varphi \in w$, contradicting $\Diamond\varphi \in w$. Another possibility is that $\forall X \subseteq^{\text{fin}} Atm_0, \Box(\varphi \to \neg\mathsf{Defin}(X)) \in w^{\Box}$, hence by **Findef** we have $\Box\neg\varphi \in w$, hence by **T**, $w^{\Box}$ is inconsistent with $\varphi$. However, again $\Box\neg\varphi \in w$ contradicts $\Diamond\varphi \in w$. Use Lemma 7.3 to obtain $v$ from $w^{\Box} \cup \{\varphi\}$. □

With help of the existence lemma, the truth lemma is straightforwardly provable.

**Lemma 7.6** (Truth)**.** *Let $M^c$ be the canonical DM and $w \in W$. Then for any $\varphi$, $\varphi \in w \iff (M^c, w) \models \varphi$.*

**Theorem 7.3.** *The logic* $\mathsf{BCL} + \mathbf{Findef}$ *is sound and strongly complete relative to the modal class* $\mathbf{DM}_{fd}$.

*Proof.* Soundness is a straightforward test. To show strong completeness, suppose $\Phi$ be a set of consistent formulas and $\varphi$ a formula, and $\Phi \not\models \varphi$. By Lemma 7.3 we have an MCT $\varphi \notin w \supseteq \Phi$. By Lemma 7.6 we have $\forall\psi \in \Phi$, $(M^c, w) \models \psi$, while $(M^c, w) \not\models \varphi$. □

**Corollary 7.1.** *The logic* $\mathsf{BCL} + \mathbf{Findef}$ *is sound and strongly complete relative to the modal class* $\mathbf{CM}_{fd}$.

### 7.2.4 Axiomatics and strong completeness for multi-classifiers

Now we turn to the two-dimensional case. The motivation and proof strategy are the same as the one-dimensional case. However, we need to enhance the inference rule accordingly as the following.

$$\frac{\varphi \to \neg\mathsf{Defin}(X) \text{ for all } X \subseteq^{\text{fin}} Atm_0}{\blacksquare\Box\neg\varphi} \qquad (\mathbf{FinDef})$$

The definitions of MCT, canonical model are defined in the similar way as the case of single classifiers. Also the Lindenbaum-type lemma is proven in a similar way. The only difference is that in Step two we add some $\blacksquare\mathsf{Defin}(X')$ into the set.

**Lemma 7.7** (Indeedness)**.** *$M^c$ is indeed a finitely definite MDM.*

*Proof.* It is easy to see the correspondence between axioms and constraints: **Comm** and $\mathbf{C_1}$, $\mathbf{Indep}_{\blacksquare p}$, $\mathbf{Indep}_{\blacksquare\neg p}$ and $\mathbf{C_3}$, $\mathbf{AtMost}_{\mathsf{t}(c)}$ and $\mathbf{C_4}$, $\mathbf{AtLeast}_{\mathsf{t}(c)}$ and $\mathbf{C_5}$, **FinDef** and $\mathbf{C_{md}}$. Finally, $\mathbf{C_2}$ automatically holds because of $\mathbf{C_{md}}$. □

**Lemma 7.8** (Existence)**.** *Let $M^c$ be the canonical finitely definite MDM. Then $\forall w \in W, \neg \boxplus \neg\varphi \in w, \exists v \in W$, s.t. $w \sim_{\boxplus} v$ and $\varphi \in v$.*

*Proof.* Let $\boxplus$ be $\blacksquare$. We claim the set $w^{\blacksquare} := \{\blacksquare\psi : \blacksquare\psi \in w\}$ is consistent with $\varphi$. Suppose not towards a contradiction, two possibilities. First, there are some $\vdash (\blacksquare\psi_1 \wedge \cdots \wedge \blacksquare\psi_n) \to \neg\varphi$ without using **FinDef**, where $\forall i \in \{1, \ldots, n\}, \blacksquare\varphi_i \in w^{\blacksquare}$. Then it is not hard to derive $\blacksquare\neg\varphi$ from $w^{\blacksquare}$, contradicting $\blacklozenge\varphi \in w$. Another possibility is that $\{\blacksquare(\varphi \to \neg\mathsf{Defin}(X)) : X \subseteq^{\text{fin}} Atm_0\} \subseteq w^{\blacksquare}$, thus by $\mathbf{T}_{\boxplus}$ and **FinDef**, $\blacksquare\Box\neg\varphi \in w$. Then by $\mathbf{T}_{\boxplus}$, $\neg\varphi \in w$ which lets $w^{\blacksquare}$ inconsistent with $\varphi$. However, by $\mathbf{T}_{\boxplus}$ and **Comm** we have $\blacksquare\varphi \in w$, which already contradicts $\blacklozenge\varphi \in w$. Now since the claim holds, the Lindenbaum-type lemma permits us to obtain a $v$ from $w^{\blacksquare} \cup \{\varphi\}$. The case of $\boxplus$ being $\Box$ is proven in the same way. $\Box$

The following theorem directly follows from the lemmas above.

**Theorem 7.4.** *The logic* $\mathsf{PLC} + \mathbf{FinDef}$ *is sound and strongly complete relative to the modal class* $\mathbf{MDM}_{fd}$.

The main result now turns to be a corollary.

**Corollary 7.2.** *The logic* $\mathsf{PLC} + \mathbf{FinDef}$ *is sound and strongly complete relative to the modal class* $\mathbf{MCM}_{fd}$.

# Conclusions and Future Work

In the thesis I have investigated a variety of logics with the aim of advancing explainable AI. The investigation began with a simple yet powerful idea that a Boolean function can be represented, instead of a propositional formula in the traditional way, an S5 Kripke model. Generalizing this idea from finite-airty to infinite-arity, from binary output to finitary output, we came up with the binary-input classifier logic BCL, a logic for (binary-input) classifier models in Chapter 3. With this logic we have been able to express the most important notions of classifier explanation in symbolic XAI, namely abductive explanation, contrastive explanation, counterfactual explanation, and concepts of bias and fairness.

As an application, we used BCL to legal case-based reasoning (CBR). The guiding principle was that a case base can be viewed as a classifier. Specifically, we represented Horty's factor-based models for CBR as classifier models of BCL. By this step allowed us to apply many notions from Boolean functions and XAI to CBR. These constituted of Chapter 4.

In developing BCL we introduced a counterfactual conditional operator using Hamming distance as distance between worlds. In the finite-variable case, the operator was found to be definable by the universal modality $\Box$. A natural question rose: does this still hold, when the variables are infinite? And if not, what else axiom would we need? Our study in Chapter 5 demonstrated that answers to both questions are negative. Among other findings, we showed that the class of VC models using Hamming distance satisfies exactly the same set of formulas as the class of all VC models of Lewis, and the same for VCU. This suggests that, in XAI we can safely use Hamming distance as the "canonical" measure of distance without loss of generality. The reason is that, any other measure can be reinterpreted as Hamming distance by adding "hidden variables".

The primary targets of XAI are black box classifiers, which we had not addressed yet. In Chapter 6, we took a closer look at the black box metaphor by highlighting that what black is not the learning algorithm, but the learned model. The term black box is an epistemic notion to denote the agent's uncertainty of what is the "real" classifier due to the lack of complete knowledge. In light of that, we represented a black box classifier as a set of classifiers compatible with the agent's partial knowledge. To achieve this we needed a two-dimensional product modal logic. Thus besides $\Box$ ranging over all possible instances, we introduced $\blacksquare$ to range over all possible classifiers. The resulting class of models is called multi-classifier models **MCM** and the logic is called product modal logic for classifiers PLC. The notions of explanation expressed in Chapter 3 thus found their correspondences in the black box setting, and the latter were, without surprise, harder than the former. For example, in the black box case we may not know the abductive explanation, even though it exists.

Explanations for black box classifiers are hard. But how hard? In the first half

of Chapter 7, we studied the complexity of deciding satisfiability problem in **MCM** and obtained a new lower-bound as EXPTIME. The second half of Chapter 7 was dedicated to bridging an axiomatic gap between logics with different cardinality of atomic propositions. This allowed us to study a particular subclass of **CM** (and **MCM**, respectively) such that even their number of arguments is infinite, they are always finitely-definite. This guarantees that all such classifier models (multi-classifier models, respectively) can be expressed by some formula.

As for future work, there are several interesting open questions have arisen in the previous study. For instance, what is the complexity of model checking problem in Hammingized models? Is the the lower bound result of satisfiability problem in MCMs optimal, i.e. is there an EXPTIME algorithm to decide the satisfiability, or there is an NEXPTIME lower bound for it? Beyond classifiers, which are the main objects of model-agnostic XAI, there are also topics in model-specific XAI. Popular systems include Bayesian network and causal graphs, which can be seen as compositions of classifiers. It would be interesting to investigate whether the current logical frameworks can extend to them.

# Appendices

Appendix A

# Proofs for Chapter 3

### Proof of Proposition 3.2

*Proof.* Suppose $C$ is $X$-definite but $(C,s) \models \neg\mathsf{Defin}(X)$, which means that $\exists c \in Val$ s.t. $(C,s) \models \neg = (X, \mathsf{t}(c))$. W.l.o.g., we assume that $(C,s) \models \neg\Box(\neg\mathsf{t}(c) \to [X]\neg\mathsf{t}(c))$. That is to say, $\exists s' \in S$, s.t. $f(s') \neq c$ but $(C,s') \models \langle X\rangle\mathsf{t}(c)$. The latter indicates that $\exists s'' \in S$, s.t. $s'' \cap X = s' \cap X$ but $f(s'') = c$, which violates $X$-definiteness.

Let $(C,s) \models \mathsf{Defin}(X)$, and assume $f(s) = c$ Then since $(C,s) \models \Box(\mathsf{t}(c) \to [X]\mathsf{t}(c))$, we have $\forall s' \in S$ if $s' \cap X = s \cap X$ then $f(s') = c = f(s)$, which is what $X$-definiteness says. □

### Proof of Theorem 3.1

*Proof.* For the left to right direction, given a CM $C = (S,f)$ and $s_0 \in S$ s.t. $(C,s_0) \models \varphi$, we construct a DM $M^\flat = (W^\flat, (\equiv^\flat_X)_{X \subseteq^{\text{fin}} Atm_0}, V^\flat)$ as follows

- $W^\flat = S$
- $s \equiv^\flat_X s'$ if $s \cap X = s' \cap X$
- $V^\flat(s) = s \cup \{\mathsf{t}(f(s))\}$.

It is easy to check that $M^\flat$ is indeed a DM and $(M^\flat, s_0) \models \varphi$.

For the other direction, given a DM $(W, (\equiv_X)_{X \subseteq^{\text{fin}} Atm_0}, V)$ and $w_0 \in W$ s.t. $(M, w_0) \models \varphi$, we construct a CM $C^\sharp = (S^\sharp, f^\sharp)$ as follows

- $S^\sharp = \{V_{Atm_0}(w) : w \in W\}$
- $\forall V_{Atm_0}(w) \in S^\sharp, f^\sharp(V_{Atm_0}(w)) = c$, if $V_{Dec}(w) = \{\mathsf{t}(c)\}$.

It is routine to check that $C^\sharp$ is a CM, and $(C^\sharp, V_{Atm_0}(w_0)) \models \varphi$.

□

### Proof of Theorem 3.2

*Proof.* The proof is conducted by constructing the canonical model.

**Definition A.1** (Theory)**.** *A set of formulas $\Delta$ is said to be a* BCL-theory *if it contains all theorems of* BCL *and is closed under Modus Ponens and* ***Nec***$_\Box$*. It is said to be a* consistent BCL*-theory if it is a theory and $\bot \notin \Delta$. It is said to be a* maximal consistent BCL*-theory (MCT for short), if it is a consistent theory and for all consistent theory $\Delta'$, if $\Delta \subseteq \Delta'$ then $\Delta = \Delta'$.*

**Lemma A.1** (Lindenbaum-type)**.** *Let $\Delta$ be a consistent* BCL*-theory and $\varphi \notin \Delta$ Then, there is a maximal consistent* BCL*-theory $\Delta'$ s.t. $\Delta' \subseteq \Delta$ and $\varphi \notin \Delta$.*

The proof is standard and omitted (see, e.g. [Blackburn *et al.* 2001, p. 197] ).

**Definition A.2** (Canonical model)**.** *The canonical decision model $\mathfrak{M} = (W^c, (\equiv^c_X)_{X \subseteq^{\text{fin}} Atm_0}, V^c)$ is defined as follows*

- $W^c = \{w : w \text{ is a maximal consistent } \mathsf{BCL} \text{ theory.}\}$
- $w \equiv^c_X v \iff \{[X]\varphi : [X]\varphi \in w\} = \{[X]\varphi : [X]\varphi \in v\}$
- $V^c(w) = \{p : p \in w\}$

**Lemma A.2.** *Let $w$ be an MCT. Then $[X]\varphi \to \varphi \in w$.*

*Proof.* Suppose $[X]\varphi \to \varphi \notin w$, then by the maximality of $w$ and $\mathbf{Red}_{[\emptyset]}$, we have $\bigwedge_{Y \subseteq X} (\mathsf{cn}_{Y,X} \to [\emptyset](\mathsf{cn}_{Y,X} \to \varphi)) \wedge \neg\varphi \in w$. Since $w$ is maximally consistent, there is exactly one $Z \subseteq X$ s.t. $\mathsf{cn}_{Z,X} \in w$. By Modus Ponens we have $\Box(\mathsf{cn}_{Z,X} \to \varphi) \in w$, and by $\mathbf{K}_{[\emptyset]}$ and Modus Ponens we have $\varphi \in w$. But than $w$ is inconsistent, since $\varphi \wedge \neg\varphi \in w$. Hence the supposition fails, which means $[X]\varphi \to \varphi \in w$. □

**Lemma A.3.** *The canonical model $\mathfrak{M}$ is indeed a decision model.*

*Proof.* Check the conditions one by one. For **C1**, we need show $w \equiv^c_X v$, if $\forall p, p \in w \cap X$ implies $p \in v$. Suppose not, then w.l.o.g. we have some $q \in w \cap X, q \notin v$, by maximality of $v$ namely $\neg q \in v$. However, we have $[q]q \in w$, for $q \to [q]q$ is a theorem, and by definition of $\equiv^c_X$, $[q]q \in v$, hence $q \in v$, since $[q]q \to q \in v$. But now we have a contradiction. **C2-4** hold obviously due to axioms **AtLeast**, **AtMost**, **Def** and **Funct** respectively. □

**Lemma A.4** (Existence)**.** *Let $\mathfrak{M} = (W^c, (\equiv^c_X)_{X \subseteq^{\text{fin}} Atm_0}, V^c)$ be the canonical model, $w$ be an MCT, $X \subseteq^{\text{fin}} Atm_0$. Then, if $\langle X\rangle\varphi \in w$ then $\exists v \in W^c$ s.t. $w \equiv^c_X v$ and $\varphi \in v$.*

*Proof.* We first claim that $\{\psi : [X]\psi \in w\} \cup \{\varphi\}$ is consistent. The proof is following the same line in e.g. [Blackburn *et al.* 2001, p. 198-199] and omitted. We claim then that $\{[X]\psi : [X]\psi \in w\} \cup \{\varphi\}$ is consistent. Suppose not towards a contradiction. It must be the case that $[X]\psi_1 \wedge \cdots \wedge [X]\psi_m \vdash \neg\varphi$. It is not hard to prove that $\Box\psi \to [X]\psi$ is a theorem. Hence we have $\Box\xi_i \in w$ for all $i \in \{1, \ldots m\}$. Moreover, $\vdash \Box\psi_1 \wedge \Box\psi_m \to \neg\varphi$. Then use $\mathbf{Nec}_{\Box}$ $\mathbf{T}_{\Box}$ and Modus Ponens we have $\Box\neg\varphi \in w$. However, because $\vdash \Box\neg\varphi \to [X]\neg\varphi$ we have $[X]\neg\varphi \in w$, contradicting $\langle X\rangle\varphi \in w$. □

**Lemma A.5** (Truth)**.** *Let $\mathfrak{M}$ be the canonical model, $w$ be an MCT, $\varphi \in \mathcal{L}(Atm_0)$. Then $(\mathfrak{M}, w) \models \varphi \iff \varphi \in w$.*

*Proof.* By induction on $\varphi$. We only show the interesting case when $\varphi$ takes the form $[X]\psi$.

For $\Longleftarrow$ direction, if $[X]\psi \in w$, since for any $v \equiv_X w$, $[X]\psi \in v$, then thanks to $[X]\psi \to \psi \in v$ we have $\psi \in v$. By induction hypothesis this means $(\mathfrak{M}, v) \models \psi$, therefore $(\mathfrak{M}, w) \models [X]\psi$.

For $\Longrightarrow$ direction, suppose not, namely $[X]\psi \notin w$. Then by the exitence lemma $\exists v \equiv_X^c w, \neg\psi \in v$. Hence $(\mathfrak{M}, v) \not\models \psi$ by induction hypothesis. However, this contradicts $(\mathfrak{M}, w) \models [X]\psi$. □

Now the completeness of **DM** w.r.t. BCL is a corollary of Lemma A.3 and A.5. □

**Alternative proof of Corollary 3.1** We slightly modify the canonical model technique to prove the completeness result in the finite-variable case directly via CMs.

The definition of MCTs and Lindenbaum-type lemma are the same and omitted.

**Definition A.3** (Canonical CM for $w$)**.** *Let $w$ be an MCT. We define its corresponding canonical CM $C^w = (S^w, f^w)$ s.t.*

- $S^w = \{w' \cap Atm_0 : \{\Box\varphi : \Box\varphi \in w'\} = \{\Box\varphi : \Box\varphi \in w\}\}$
- $f^w = \{(w' \cap Atm_0, x) : w' \cap Atm_0 \in S^w \text{ and } \mathsf{t}(x) \in w\}$.

The satisfaction relation is defined as the one of CM and omitted.

**Lemma A.6.** *Let $w$ be an MCT, then $C^w$ is indeed a CM.*

*Proof sketch.* We only need show the functionality of $f^w$, which is guaranteed by the axiom **Funct**. □

**Lemma A.7** (Existence)**.** *Let $w$ be an MCT and $C^w = (S^w, f^w)$ be its corresponding CM. Then $\forall\Diamond\varphi \in \mathcal{L}(Atm)$, if $\Diamond\varphi \in w$, then there exists a $w' \cap Atm_0 \in S^w$ s.t. $(w' \cap Atm_0, x) \in f^w$ and $\varphi \in w'$.*

*Proof.* Let $\Delta = \{\varphi\} \cup \{\psi : \Box\psi \in w\}$. We claim that $\Delta$ is consistent. For if no, then there are $\psi_1 \dots \psi_m \in \{\psi : \Box\psi \in w\}$, s.t. $(\psi_1 \wedge \dots \wedge \psi_m) \to \neg\varphi$ is a BCL theorem. It is derivable that $\Box(\psi_1 \wedge \dots \wedge \psi_m) \to \Box\neg\varphi$ is a BCL theorem. Hence we have $\Box\neg\varphi \in w$, which contradicts that $w$ is consistent, since $\Diamond\varphi \in w$. Now, by Lindenbaum-type lemma we can extend $\Delta$ to an MCT $w'$. By definition of $\Delta$ and $C^w$, obviously $w' \cap Atm_0 \in S^w$. □

**Lemma A.8** (Truth)**.** *Let $w$ be an MCT, then $\forall\varphi \in \mathcal{L}(Atm_0)$, $\varphi \in w \iff (C^w, w \cap Atm_0) \models \varphi$.*

*Proof.* By induction on $\varphi$. We prove the case when $\varphi$ takes the form $\Box\psi$. Suppose $\Box\psi \in w$, we need show that $\forall w' \cap Atm_0 \in S^w, (C^w, w' \cap Atm_0) \models \psi$, which is shown by induction hypothesis and the definition of $S^w$.

For the other direction we show its countraposition. Suppose $(C^w, w \cap Atm_0) \nvDash \Box\psi$, by definition $\exists w' \cap Atm_0 \in S^w$, s.t. $(C^w, w' \cap Atm_0) \models \neg\psi$. By induction hypothesis we have $\neg\psi \in w'$. Then by the existence lemma we have $\Diamond\neg\psi \in w'$, which by the maximality of $w$ means $\Box\psi \notin w$. □

The theorem of strong completeness turns out to be a corollary of the lemmas above.

**Proof of Theorem 3.3**

*Proof.* Let $(W, (\equiv_X)_{X \subseteq^{\text{fin}} Atm_0}, V)$ be a QDM and $w_0 \in W$ s.t. $(M, w_0) \models \varphi$. Let $sf(\varphi)$ be the set of all subformulas of $\varphi$ and let $sf^+(\varphi) = sf(\varphi) \cup Dec$. Moreover, $\forall v, u \in W$, we define $v \simeq u \iff \forall \psi \in sf^+(\varphi)$, $(M, v) \models \psi$ iff $(M, u) \models \psi$. Finally, we define $[v] = \{u \in W : v \simeq u\}$.

Now we construct a filtration through $sf^+(\varphi)$, $M' = (W', (\equiv'_X)_{X \subseteq^{\text{fin}} Atm_0}, V')$ as follows

- $W' = \{[v] : v \in W\}$
- $\forall X \subseteq^{\text{fin}} Atm_0$, $[v] \equiv'_X [u]$, iff $V'_X([v]) = V'_X([u])$
- $V'([v]) = V_{sf^+(\varphi) \cap Atm_0}(v)$

$M'$ is indeed a filtration. We need show that it satisfies the two conditions.

1) $v \equiv_X u \iff V_X(v) = V_X(u) \implies V'_X([v]) = V'_X([u]) \iff [v] \equiv'_X [u]$. Suppose $v \equiv_X u$. By construction of $V'$, $\forall p \in X \cap sf^+(\varphi), p \in V'_X([v]) p \in V(v) \iff p \in V(u) \iff p \in V'_X([u])$, and $\forall p \in X \setminus sf^+(\varphi), p \notin V'_X([v])$ and $p \notin V'_X([u])$. As a result, $V'_X([v]) = V'_X([u])$ which means $[v] \equiv'_X [u]$.

2) If $[v] \equiv'_X [u]$, then $\forall [X]\psi \in sf^+(\varphi)$: if $(M, v) \models [X]\psi$ then $(M, u) \models \psi$. The crucial point is that $\forall v, v' \in [v], \forall u, u' \in [u]$, $\forall [X]\psi \in sf^+(\varphi)$, if $[v] \equiv'_X [u]$, then $v \equiv_X v' \equiv_X u \equiv_X u'$ by the definitions of $V'$ and $\simeq$. Hence by satisfaction relation of $M$ we have if $(M, v) \models [X]\psi$ then $(M, u) \models \psi$.

Moreover, $M'$ is a finite-QDM. For **C1** it is given as the definition of $V'$. **C2** and **C3** hold because of $sf^+(\varphi) = sf(\varphi) \cup Dec$.

Finally, we need prove $(M, w_0) \models \varphi$ iff $(M', [w_0]) \models \varphi$. We only show when $\varphi$ takes the form $[X]\psi$. Given $(M, w_0) \models [X]\psi$, i.e. $\forall v \in W$, if $w_0 \equiv_X v$ then $(M, v) \models \psi$. By definitions of $\equiv'_X$ and **C1** we have $V'_X([w_0]) = V'_X([v])$, by induction hypothesis $(M', [v]) \models \psi$, which means $(M', [w_0]) \models [X]\psi$. If $(M', [w_0]) \models [X]\psi$, i.e. $\forall [v] \in W'$, if $[v] \equiv'_X [w_0]$ then $(M', [v]) \models \psi$. Then by definitions of $V'$ and $\simeq$ we have $w_0 \equiv_X v$, by induction hypothesis $(M, v) \models \psi$. □

**Proof of Theorem 3.4**

*Proof.* The right to left direction is obvious since any finite-DM is a finite-QDM. For the other direction, suppose there is a finite-QDM $(W, (\equiv_X)_{X \subseteq^{\text{fin}} Atm_0}, V)$ and $w \in W$ s.t. $(M, w) \models \varphi$. Since $Atm_0$ is infinite, we can construct an injection $\iota : W \longrightarrow$

$Atm_0 \setminus Atm(\varphi)$. Then, we construct a finite-DM $M' = (W', (\equiv'_X)_{X \subseteq^{\text{fin}} Atm_0}, V')$ as follows

- $W' = W$
- $w \equiv'_X v$ iff $V'_X(w) = V'_X(v)$
- $V'(w) = (V(w) \cup \{\iota(w)\}) \setminus \{p : \exists v \in W, v \neq w \ \& \ \iota(v) = p\}$.

It is easy to check that $M'$ is indeed a finite-DM. By induction we show that $(M', w) \models \varphi$. When $\varphi$ is some $p$, we have $V(w) = V'(w)$ since the injection $\iota$ has nothing to do with $\varphi$. The case of $\mathsf{t}(c)$ is the same. The Boolean cases are straightforward. Finally when $\varphi$ takes form $[X]\psi$. Again since $\iota$ does not change valuation in $\varphi$, we have $\forall v \in W, V_X(v) = V'_X(v)$. Hence we have $(M, w) \models [X]\psi \iff \forall v \in W$, if $V_X(w) = V_X(v)$ then $(M, v) \models \psi \iff \forall v \in W$, if $V'_X(w) = V'_X(v)$ then $(M', v) \models \psi \iff (M', w) \models [X]\psi$. □

**Proof of Theorem 3.7**

*Proof.* Suppose $Atm_0$ is finite and fixed. In order to determine whether a formula $\varphi$ is satisfiable for the class **CM**, we are going to verify whether $\varphi$ is satisfied in each CM, by doing this sequentially one CM after the other. The corresponding algorithm runs in polynomial time in the size of $\varphi$ since: (i) there is a finite, constant number of CMs and (ii) model checking for the language $\mathcal{L}(Atm)$ relative to a pointed CM is polynomial. This means that, when $Atm_0$ is finite and fixed, satisfiability checking has the same complexity as model checking. Regarding (i), the finite, constant number of CMs in the class **CM** is $\sum_{S \subseteq 2^{Atm_0}} |Val|^{|S|}$. Indeed, for every $S \subseteq 2^{Atm_0}$, we consider the number of functions from $S$ to $Val$. Regarding (ii), it is easy to build a model checking algorithm running in polynomial time. It is sufficient to adapt the well-known "labelling" model checking algorithm for the basic multimodal logics and CTL [Clarke & Schlingloff 2001]. The general idea of the algorithm is to label the states of a finite model step-by-step with sub-formulas of the formula $\varphi$ to be checked, starting from the smallest ones, the atomic propositions appearing in $\varphi$. At each step, a formula should be added as a label to just those states of the model at which it is true.

□

**Proof of Theorem 3.8**

*Proof.* As for NEXPTIME-hardness, in [Grossi *et al.* 2015] the following *ceteris paribus* modal language, noted $\mathcal{L}_{\mathsf{CP}}(Prop)$, is considered with $Prop$ a countable set of atomic propositions:

$$\varphi \quad ::= \quad p \mid \neg\varphi \mid \varphi \wedge \varphi \mid [X]\varphi,$$

where $p$ ranges over $Prop$ and $X$ is a finite set of atomic propositions from $Prop$. Formulas for this language are interpreted relative to a *simple model* $S \subseteq 2^{Atm_0}$

and a state $s \in S$ in the expected way as follows (we omit boolean cases since they are interpreted in the usual way): $(S, s) \models p$ iff $p \in s$; $(S, s) \models [X]\varphi$ iff $\forall s' \in S$ : if $s \cap X = s' \cap X$ then $(S, s') \models \varphi$. It is proved that, when *Prop* is countably infinite, satisfiability checking for formulas in $\mathcal{L}_{\mathsf{CP}}(Prop)$ relative to the class **SM** of simple models is NEXPTIME-hard [Grossi *et al.* 2015, Lemma 2 and Corollary 2]. It follows that satisfiability checking for formulas in our language $\mathcal{L}(Atm)$ with $Atm_0$ countably infinite is NEXPTIME-hard too.

As for membership, let $tr$ be the following translation from the language $\mathcal{L}(Atm)$ to the language $\mathcal{L}_{\mathsf{CP}}(Atm_0 \cup \{p_{\mathsf{t}(c)} : c \in Val\})$:

$$
\begin{aligned}
&tr(p) = p,\\
&tr(\mathsf{t}(c)) = p_{\mathsf{t}(c)},\\
&tr(\neg\varphi) = \neg tr(\varphi),\\
&tr(\varphi \wedge \psi) = tr(\varphi) \wedge tr(\psi),\\
&tr([X]\varphi) = [X]tr(\varphi).
\end{aligned}
$$

By induction on the structure of $\varphi$, it is routine to verify that $\varphi \in \mathcal{L}(Atm)$ is satisfiable for the class **QDM** of Definition 3.9 if and only if $[\emptyset](\varphi_1 \wedge \varphi_2) \wedge tr(\varphi)$ is satisfiable for the class **SM** of simple models, with

$$
\begin{aligned}
\varphi_1 &=_{def} \bigvee_{c \in Val} p_{\mathsf{t}(c)},\\
\varphi_2 &=_{def} \bigwedge_{c,c' \in Val : c \neq c'} (p_{\mathsf{t}(c)} \rightarrow \neg p_{\mathsf{t}(c')}).
\end{aligned}
$$

Hence, by Theorem 3.5 we have that, when $Atm_0$ is countably infinite, $\varphi \in \mathcal{L}(Atm)$ is satisfiable for the class **CM** of classifier models if and only if $[\emptyset](\varphi_1 \wedge \varphi_2) \wedge tr(\varphi)$ is satisfiable for the class **SM** of simple models. Since the translation $tr$ is linear and satisfiability checking for formulas in $\mathcal{L}_{\mathsf{CP}}(Atm_0 \cup \{p_{\mathsf{t}(c)} : c \in Val\})$ relative to the class **SM** is in NEXPTIME in the infinite-variable case [Grossi *et al.* 2015, Lemma 2 and Corollary 1], checking satisfiability of formulas in $\mathcal{L}(Atm)$ relative to the class **CM** is in NEXPTIME too, with $Atm_0$ countably infinite.

□

### Proof of Theorem 3.9

*Proof.* NP-hardness follows from the NP-harndess of propositional logic.

In order to prove NP-membership, we can use the translation given in the proof of Theorem 3.8 to give a polynomial reduction of satisfiability checking of formulas in $\mathcal{L}^{\{[\emptyset]\}}(Atm)$ relative to **CM** to satisfiability checking in the modal logic S5. The latter problem is known to be in NP in the infinite-variable case [Ladner 1977]. □

### Proof of Proposition 3.3

*Proof.* For the right direction, we have $closest_C(s,\varphi,X) \subseteq ||\psi||_C$ from the antecedent. Suppose towards a contradiction that the consequent does not hold. Then, $\exists k \in \{0,\ldots,|X|\}, Y_1, Y_2 \subseteq X$ with $|Y_1| = |Y_2| = k$, s.t. $(C,s) \models \langle Y_1\rangle\varphi \wedge \bigwedge_{Y\subseteq X: k<|Y|}[Y]\neg\varphi \wedge \langle Y_2\rangle(\varphi \wedge \neg\psi)$. The last conjunct means that $\exists s' \in S, s' \cap X = s \cap X = Y_2$ and $(C,s') \models \varphi \wedge \neg\psi$. But the conjuncts together guarantee that $s' \in closest_C(s,\varphi,X)$, because $sim_C(s,s',X) = k$, and it is an argmax by definition of $closest_C(s,\varphi,X)$. It is the desired contradiction, since $s' \notin ||\psi||_C$.

For the other direction, we need show $closest_C(s,\varphi,X) \subseteq ||\psi||_C$, given the antecedent. Suppose the opposite towards a contradiction. Then by definition, $\exists s^* \in closest_C(s,\varphi,X), s' \notin ||\psi||_C$. Let $s\cap X = s^*\cap X = Y^*$, and $sim_C(s,s',X) = k^*$. Then we have $(C,s) \models \mathsf{maxSim}(\varphi,X,k^*) \wedge \langle Y^*\rangle(\varphi \wedge \neg\psi)$, which contradicts the antecedent. To see that, notice the second conjunct is because of $(C,s^*) \models \varphi \wedge \neg\psi$, and the first conjunct because of $sim_C(s,s^*,X) = k^*$ and $s^* \in closest_C(s,\varphi,X)$. □

**Proof of Proposition 3.4**

*Proof.* The first validity is obvious, since if $closest_C(s,\varphi,X) \subseteq ||\mathsf{t}(c)||_C$ then $closest_C(s,\varphi,X) \nsubseteq ||\mathsf{t}(c')||_C$ given $c' \neq c$. For the second validity, notice that $\{s\} = closest_C(s,\varphi,Atm_0)$, if $(C,s) \models \varphi$. Hence if $(C,s) \models \mathsf{t}(c)$, then we have $closest_C(s, \bigvee_{c'\in Val: c'\neq ?}, Atm_0) = \{s\} \subseteq ||\mathsf{t}(c)||_C$. □

**Proof of Proposition 3.5**

*Proof.* Let $(C,s)$ be a pointed CM and $(C,s) \models \mathsf{AXp}(\lambda,c)$, which directly gives us $(C,s) \models \lambda$. Now since $\lambda$ is an implicant of $c$, $(C,s) \models [Atm(\lambda)]\mathsf{t}(c)$, for otherwise $\exists s'$, s.t. $(C,s') \models \lambda \wedge \neg\mathsf{t}(c)$; and since $\lambda$ is prime, we have $(C,s) \models \bigwedge_{p\in Atm(\lambda)}\langle Atm(\lambda) \setminus \{p\}\rangle\neg\mathsf{t}(c))$, otherwise $\exists\lambda'$, s.t. $\lambda' \subset \lambda$ and $\lambda'$ is also an implicant of $c$. The other direction is proven in the same way and omitted. □

**Proof of Proposition 3.6**

*Proof.* Suppose towards a contradiction that $C$ is finitely-definite, but $\exists c \in Val$, s.t. $\forall\lambda \in Term$, if $(C,s) \models \lambda$ then $(C,s) \models \neg\mathsf{PImp}(\lambda,c)$. That is to say, $\exists s_1 \in S$ s.t. $(M,s_1) \models \lambda$ but either $f(s_1) \neq c$ or $\exists s_2 \in S$ s.t. $\exists p \in Atm(\lambda)$, $s_1 \cap (Atm(\lambda \setminus \{p\}) = s_2 \cap (Atm(\lambda \setminus \{p\})$ but $f(s_2) \neq c$. Hence $C$ is neither $Atm(\lambda)$-definite nor $(Atm(\lambda \setminus \{p\})$-definite. Either case $C$ is not finitely-definite, since $\lambda$ is arbitrarily selected from *Term*. □

**Proof of Proposition 3.7**

*Proof.* For the first validity, let $C = (S,f) \in \mathbf{CM}$ and $s \in S$ and suppose $(C,s) \models \mathsf{CXp}(\lambda,c)$. By definition of $\mathsf{CXp}(\lambda,c)$ we have $(C,s) \models \mathsf{t}(c)$. We need to show $(C,s) \models \overline{\lambda} \Rightarrow \neg\mathsf{t}(c)$. By the antecedent, $\exists s' \in S$, s.t. $s\triangle s' = Atm(\lambda)$ and $f(s') \neq c$. It is not hard to show that $closest_C(s,\overline{\lambda},Atm) = \{s'\}$. Therefore $(C,s) \models \overline{\lambda} \Rightarrow \neg\mathsf{t}(c)$, since $closest_C(s,\overline{\lambda},Atm_0) \subseteq ||\neg\mathsf{t}(c)||_C$. For the second validity, the right direction of

the iff is a special case of the first validity. To show the left direction, from $Atm_0$-completeness and the counterfactual conditional we have $\exists s' \in S$, s.t. $s' \triangle s = Atm(l)$ and $\{s'\} = closest_C(s,l,Atm_0)$. Hence $(C,s) \models l \wedge \langle Atm_0 \setminus Atm(l)\rangle \neg \mathsf{t}(c) \wedge [Atm_0]\mathsf{t}(c)$, which is by definition $(C,s) \models \mathsf{CXp}(l,c)$. □

**Proof of Proposition 3.8**

*Proof.* We show that for any $C = (S,f) \in \mathbf{CM}$, both directions are satisfied in $(C,s)$ for some $s \in S$. The right to left direction is obvious, since from the antecedent we know there is a property $\lambda'$ s.t. $\exists s' \in S, s \triangle s' = Atm(\lambda') \subseteq \mathsf{PF}$ and $(C,s') \models \neg \mathsf{t}(c)$, which means $(C,s) \models \mathsf{Bias}(c)$. The other direction is proven by contraposition. Suppose for any $\lambda$ s.t. $Atm(\lambda) \subseteq \mathsf{PF}$, $(C,s) \models \neg \mathsf{CXp}(\lambda, c)$, then it means $\forall s' \in S$, if $s \triangle s' = Atm(\lambda)$, then $f(s') = c$, which means $(C,s) \models \neg \mathsf{Bias}(c)$. □

**Proof of Theorem 3.14**

*Proof.* Suppose $|Atm_0|$ is finite. As in the proof of Theorem 3.7, we can show that the size of the model class **ECM** is bounded by some fixed integer. Thus, in order to determine whether a formula $\varphi$ $\mathcal{L}^{epi}(Atm)$ is satisfiable for this class, it is sufficient to repeat model checking a number of times which is bounded by some integer. Model checking for the language $\mathcal{L}^{epi}(Atm)$ with respect to a pointed ECM is polynomial. □

Appendix B

# Proofs for Chapter 4

### Proof of Theorem 4.1

*Proof.* Suppose $CB$ is consistent. We construct a classifier model $C = (S, f)$ s.t. $S = 2^{Atm_0}$, and $\forall s \in S$, we have

$$f(s) = \begin{cases} c \text{ for } c \in \{0,1\}, & \text{if } \exists (s', X, c) \in CB \text{ s.t. } s \cap Atm_0^c \supseteq X \text{ and } s \cap Atm_0^{\overline{c}} \subseteq s' \cap Atm_0^{\overline{c}}; \\ ? & \text{otherwise.} \end{cases}$$

Obviously $(C, s) \models \texttt{Compl}$ since $S = 2^{Atm_0}$. We need show that $(C, s) \models \texttt{2Mon}$. Suppose the opposite towards a contradiction. W.l.o.g., suppose $\exists s = X \cup Y \in S, f(s) = 0$, where $X \subseteq Dfd, Y \subseteq Plt$ and $\exists s' = X' \cup Y'$ s.t. $X' \supseteq X, Y' \subseteq Y$ but $f(s') \neq 1$. According to the construction of $f$, since $f(s) = 0, \exists c_0 = (s_0, X_0, 0) \in CB$ s.t. $X \supseteq X_0$ and $Y \subseteq s_0 \cap Plt$. By transitivity of $\subseteq$, we have $X' \supseteq X_0$ and $Y' \subseteq s_0 \cap Plt$. According to the construction of $f$ again it has to be $f(s') = c$, a contradiction.

For the other direction, suppose $CB$ is inconsistent, we show that $tr_2(CB) \wedge \texttt{Compl} \wedge \texttt{2Mon}$ is unsatisfiable. Since $CB$ is inconsistent, by definition we shall have $Y_0 <_{CB} Y_1$ and $Y_1 <_{CB} Y_0$. W.l.o.g., assume in $CB$ there are two precedents $c_0 = (s_0, X_0, 0), \mathfrak{c}_1 = (s_1, X_1, 1)$ s.t. $Y_0 <_{c_1} Y_1, Y_1 <_{c_0} Y_0$. Unravel the definition we have $Y_0 \subseteq s_1 \cap Dfd$ and $X_1 \subseteq Y_1$; $Y_1 \subseteq s_0 \cap Plt$ and $X_0 \subseteq Y_0$.

Now towards a contradiction suppose $(C, s)$ be a pointed CM s.t. $(C, s) \models tr_2(CB) \wedge \texttt{Compl} \wedge \texttt{2Mon}$. Consider the state $s_2 = Y_0 \cup Y_1$. Since $(C, s) \models \texttt{Compl}$ we always have $s_2 \in S$. Then by $\texttt{2Mon}$, we have $f(s_2) = 0$ with respect to $s_0$, since $s_2 \cap Dfd = Y_0 \supseteq X_0 \supseteq s_0 \cap Dfd$ and $s_2 \cap Plt = Y_1 \subseteq s_0 \cap Plt$. But also by $\texttt{2Mon}$ we have $f(s_2) = 1$ with respect to $s_1$. Hence $f$ fails to be functional, a contradiction that we want. □

### Proof of Proposition 4.3

*Proof.* Suppose towards a contradiction that $\exists \lambda, (C, s') \models \mathsf{PImp}(\lambda, \overline{c}), X \cap Atm(Lit^-(\lambda)) = \emptyset$ and $s \cap Atm_0^{\overline{c}} \supseteq Atm(Lit^+(\lambda))$. Then $\lambda \wedge \mathsf{cn}_{X,X}$ is consistent. By $\texttt{Compl}$ we have some $s^\dagger \in S, (C, s^\dagger) \models \lambda \wedge \mathsf{cn}_{X,X}$ and $f(s^\dagger) = \overline{c}$ since $\lambda$ is a PImp. However, by virtue of $\texttt{2Mon}$ according to $s$ we shall have $f(s^\dagger) = c$, a contradiction that we want. □

### Proof of Proposition 4.4

*Proof.* If it were no such $\lambda$, then we would have some $s' \in S$ s.t. $X \subseteq s'$, $s' \cap Atm_0^{\bar{c}} \subseteq s \cap Atm_0^{\bar{c}}$ and $f(s') \neq c$. However, this contradicts `2Mon`. Notice that if the classifier is trivial, i.e. $\forall s' \in S, f(s') = c$, then we have $(C, s) \models \mathsf{AXp}(\top, c)$ and by definition of term, $X \supseteq Atm(Lit(\top)) = \emptyset$. □

**Proof of Proposition 4.5**

*Proof.* Let $\varsigma$ abbreviate $\bigwedge_{p \in X} p \wedge \bigwedge_{q \in Atm_0^{\bar{c}} \setminus s} \neg q$. W.l.o.g., we consider the case when $c = 1$. Let $\mathfrak{c}$ be a landmark and suppose towards a contradiction that $(C, s) \models \neg\mathsf{PImp}(\varsigma, 1)$. There are three possibilities.

1) $\varsigma$ is not even an implicant, i.e. $(C, s) \models \neg\mathsf{Imp}(\varsigma, 1)$. Hence $\exists s'$ s.t. $s' \cap Plt \supseteq X$, and $s' \cap Dfd \subseteq s \cap Dfd$, while $f(s') = 0$. But according to the definition of genuine classifier, there must $\exists \mathfrak{c}_0 = (s_0, Y, 0) \in CB$ s.t. $s_0 \cap Y \subseteq s' \cap Dfd$ and $s_0 \cap Plt \supseteq s' \cap Plt$. By transitivity of $\subseteq$ it is easy to see that $CB$ is inconsistent, contradicting the assumption.

Then it is still possible that $\varsigma$ is an implicant but not prime. That is to say, "drawing" a literal from the conjunction $\varsigma$, the resulting conjunction is still an implicant. Two possibilities regarding whether a positive or negative literal is drawn.

2) $\varsigma - p'$ is still an implicant, where $\varsigma - p'$ denotes the resulting conjunction of drawing some $p'$ from $\varsigma$.[1] Then, from $(C, s) \models \mathsf{Imp}(\varsigma - p', 1)$ we know that for the state $s' = (X \setminus \{p'\}) \cup (s \cap Dfd)$, $f(s') = 1$. By the definition of genuine classifier, the reason of $f(s') = 1$ is that there must $\exists \mathfrak{c}_1 = (s_1, X_1, 1) \in CB$ s.t. $X_1 \subseteq X \setminus \{p'\}, s_1 \cap Dfd \supseteq s \cap Dfd$. However, by transitivity we have $X_1 \subseteq X$ as well, which indicates that $s$ is forced to be 1 by $\mathfrak{c}_1$, contradicting the supposition.

3) This time we draw a negative literal and claim the resulting $\varsigma - \neg q'$ is still an implicant. The proof is similar and omitted.

For the other direction, let $(C, s) \models \mathsf{PImp}(\varsigma, 1)$, we need show $\mathfrak{c}$ is a landmark. Suppose towards a contradiction that $\exists \mathfrak{c}_1 = (s_1, X_1, 1) \in CB$ a landmark and forces $s$. By definition, $X_1 \subseteq s \cap Plt$ and $s_1 \cap Dfd \supseteq s \cap Dfd$. Furthermore, we claim $(C, s) \models \mathsf{Imp}(\varsigma - \bigwedge_{p \in (s \cap Plt) \setminus X_1} \wedge \bigwedge_{q \in (Dfd \setminus s) \cap s_1} \neg q, 1)$. Namely we draw from $\varsigma$ the plaintiff-factors present in $s$ but absent from $X_1$, and the defendant-factors absent from $s$ but present in $s_1$. They are all literals that seem "redundant" in the eye of $s_1$. This contradicts the assumption that $\varsigma$ is a prime implicant. □

**Proof of Proposition 4.6**

*Proof.* We use $s - \lambda$ to denote $(s \setminus Lit^+(\lambda)) \cup Lit^-(\lambda)$, i.e. the state resulting from flipping the literals in $\lambda$ of $s$. W.l.o.g., assume $c = 1$.

We first show that $Lit^+(\lambda) \subseteq Plt, Lit^-(\lambda) \subseteq Dfd$. If not, suppose $\exists p \in Lit^+(\lambda)$ s.t. $p \in Dfd$. Then we claim that the $\lambda' \subset \lambda$, which results from retracting $p$ from $\lambda$, is still a WCXp for 1, contradicting the minimality condition of $\lambda$ being a CXp.

Since $\lambda$ is a CXp, we have $f(s - \lambda) = 0$. By the definition of genuine classifier, there must exists a landmark $\mathfrak{c}_0 = (s_0, X, 0) \in CB$ s.t. $X \subseteq (s - \lambda) \cap Dfd, s_0 \cap Plt \supseteq$

[1] Formally, $\varsigma - p' := \bigwedge_{p \in X \setminus \{p'\}} p \wedge \bigwedge_{q \in Dfd \setminus s} \neg q$.

$(s - \lambda) \cap \mathit{Plt}$. Now suppose a contradiction that $\mathit{Lit}^-(\lambda) \not\subseteq X$, i.e. $\exists p \in \mathit{Dfd} \cap \mathit{Atm}(\lambda), p \notin X$. But then $p$ is "unnecessary" since it is not a part of the reason of $\mathfrak{c}$ which forces $s$. That means the $\lambda' \subset \lambda$ resulting from retracting $\neg p$ from $\lambda$, is also a WCXp for 1. Hence it contradicts the minimality condition of $\lambda$ being a CXp.

Suppose towards a contradiction $\mathit{Lit}^+(\lambda) \not\subseteq \mathit{Plt} \backslash s_0$, which means $p \in (s \cap s_0) \cap \mathit{Plt}$. The rest of the proof is similar, namely $p$ is "unnecessary" which makes $\lambda$ a WCXp but not a CXp. □

**Proof of Proposition 4.7**

*Proof.* Suppose towards a contradiction that $p \in \mathit{Plt}$ but $p$ is non-positive. That is, $\exists s \in 2^{\mathit{Atm}_0}, f(s) < f(s \setminus \{p\})$. There are two possibilities: $f(s) = 0 \neq f(s \setminus \{p\})$, and $f(s \setminus \{p\}) = 1 \neq f(s)$. For the first, since $f(s) = 0$, by definition of genuine classifier we know a precedent $(s', Y, 0)$ s.t. $Y \subseteq s \cap \mathit{Dfd}, s' \cap \mathit{Plt} \supseteq s \cap \mathit{Plt}$ that forces $f(s) = 0$. But since $p \in \mathit{Plt}$, we have $Y \subseteq s \cap \mathit{Dfd} = (s \setminus \{p\}) \cap \mathit{Dfd}$. On the other hand, $s' \cap \mathit{Plt} \supseteq s \cap \mathit{Plt}$, a fortiori $s' \cap \mathit{Plt} \supseteq (s \setminus \{p\}) \cap \mathit{Plt}$. Hence by a fortiori reasoning $f(s \setminus \{p\}) = 0$, a contradiction. The second possibility is proven similarly.

Let $p$ be non-negative, we show $p \in \mathit{Plt}$. Then $\exists s \in \mathit{dom}(f), f(s) > f(s \setminus \{p\})$. There are two possibilities: $f(s) = 1 \neq f(s \setminus \{p\})$, and $f(s) \neq 0 = f(s \setminus \{p\})$. For the first, by definition of genuine classifier we know a landmark precedent $(s', X, 1)$ s.t. $X \subseteq s \cap \mathit{Plt}$ and $s \cap \mathit{Dfd} \supseteq s' \cap \mathit{Dfd}$, which forces $f(s) = 1$. The fact that the landmark cannot force $f(s \setminus \{p\})$ indicates that either $X \not\subseteq (s \setminus \{p\}) \cap \mathit{Plt}$ or $(s \setminus \{p\}) \cap \mathit{Dfd} \not\supseteq s' \cap \mathit{Dfd}$. The latter is impossible due to the transitivity of $\supseteq$. Hence it has to be $X \not\subseteq (s \setminus \{p\}) \cap \mathit{Plt}$. We know therefore $p \in X \subseteq \mathit{Plt}$. For the second, from $f(s \setminus \{p\})$ we know a landmark precedent $(s'', Y, 0)$ s.t. $Y \subseteq (s \setminus \{p\}) \cap \mathit{Dfd}, s'' \cap \mathit{Plt} \supseteq (s \setminus \{p\}) \cap \mathit{Plt}$ forcing $f(s \setminus \{p\}) = 0$. The fact that it does not force $f(s) = 0$ indicates that either $Y \not\subseteq s \cap \mathit{Dfd}$ or $s'' \cap \mathit{Plt} \not\supseteq s \cap \mathit{Plt}$. The former is impossible by the transitivity of $\subseteq$. The latter means that $p \notin s'' \cap \mathit{Plt}$. As a result, we have not only shown $p \in \mathit{Plt}$, but also $p$ must be present as a part of the reason $X$ in the landmark $(s', X, 1)$, or absent from the con-factors of the landmark $(s'', Y, 0)$.

The other half of the proposition is proven similarly. □

**A syntactic proof of Proposition 4.4** The proof exercises the axiomatics BCL in Definition 3.7. Recall Proposition 4.4 as below.

**Proposition B.1.** *We have the following validity*

$$\models_{\mathbf{CM}} tr_2(s, X, c) \to \bigvee_{\lambda \in \mathit{Term}} (\mathsf{Imp}(\lambda, c) \wedge (\lambda \to \mathsf{cn}_{X,X})).$$

*Proof.* We prove by deriving $\vdash_{\mathsf{BCL}} tr_2(s, X, c) \to \bigvee_{\lambda \in \mathit{Term}} (\mathsf{Imp}(\lambda, c) \wedge (\lambda \to \mathsf{cn}_{X,X})$. For readability we write $s^x_X$ for $X \cup (s \cap \mathit{Atm}_0^{\bar{c}})$.

1. $\vdash_{\mathsf{BCL}} \bigwedge_{\lambda \in Term} \neg(\mathsf{Imp}(\lambda, c) \rightarrow \mathsf{cn}_{X,X}) \rightarrow \neg(\mathsf{Imp}(\mathsf{cn}_{s^x_X, Atm_0}, c) \wedge (\mathsf{cn}_{s^x_X, Atm_0} \rightarrow \mathsf{cn}_{X,X})$
   by the fact that $\mathsf{cn}_{s^x_X, Atm_0}$ is a term

2. $\vdash_{\mathsf{BCL}} \neg(\mathsf{Imp}(\mathsf{cn}_{s^x_X, Atm_0}, c) \wedge (\mathsf{cn}_{s^x_X, Atm_0} \rightarrow \mathsf{cn}_{X,X}) \rightarrow \neg\mathsf{Imp}(\mathsf{cn}_{s^x_X, Atm_0}, c)$
   by the fact that $\mathsf{cn}_{s^x_X, Atm_0} \rightarrow \mathsf{cn}_{X,X}$

3. $\vdash_{\mathsf{BCL}} \neg\mathsf{Imp}(\mathsf{cn}_{s^x_X, Atm_0}, c) \rightarrow \neg\Box(\mathsf{cn}_{s^x_X, Atm_0} \rightarrow \mathsf{t}(c))$
   by definition of Imp

4. $\vdash_{\mathsf{BCL}} tr_2(s, X, c) \rightarrow \Box(\mathsf{cn}_{s^x_X, Atm_0} \rightarrow \mathsf{t}(c))$
   by definition of $tr_2(s, X, c)$, **Funct**, Modus Ponens and a theorem proven below

5. $\vdash_{\mathsf{BCL}} (tr_2(s, X, c) \wedge \neg\mathsf{Imp}(\mathsf{cn}_{s^x_X, Atm_0}, c)) \rightarrow \bot$
   from 3, 4 by propositional logic

6. $\vdash_{\mathsf{BCL}} tr_2(s, X, c) \rightarrow \bigvee_{\lambda \in Term} (\mathsf{Imp}(\lambda, c) \wedge (\lambda \rightarrow \mathsf{cn}_{X,X})$
   from 2, 5 by propositional logic

The theorem used in 4 is $\vdash_{\mathsf{BCL}} \Diamond(\mathsf{cn}_{Y, Atm_0} \wedge \mathsf{t}(c)) \rightarrow \Box(\mathsf{cn}_{Y, Atm_0} \rightarrow \mathsf{t}(c))$ and is derived as follows.

1. $\vdash_{\mathsf{BCL}} \mathsf{t}(y) \rightarrow \neg\mathsf{t}(c)$ for $c \neq y$

2. $\vdash_{\mathsf{BCL}} (\neg\mathsf{cn}_{Y, Atm_0} \vee \mathsf{t}(y)) \rightarrow (\neg\mathsf{cn}_{Y, Atm_0} \vee \neg\mathsf{t}(c))$
   by propositional logic

3. $\vdash_{\mathsf{BCL}} \Box(\neg\mathsf{cn}_{Y, Atm_0} \vee \mathsf{t}(y)) \rightarrow \Box(\neg\mathsf{cn}_{Y, Atm_0} \vee \neg\mathsf{t}(c))$
   from 1, 2 by **Nec**, **K** and Modus Ponens

4. $\vdash_{\mathsf{BCL}} \Box(\mathsf{cn}_{Y, Atm_0} \rightarrow \mathsf{t}(y)) \rightarrow \Box(\mathsf{cn}_{Y, Atm_0} \rightarrow \neg\mathsf{t}(c))$

5. $\vdash_{\mathsf{BCL}} \bigvee_{c' \in Val \setminus \{c\}} \Box(\mathsf{cn}_{Y, Atm_0} \rightarrow \mathsf{t}(c')) \rightarrow \Box(\mathsf{cn}_{Y, Atm_0} \rightarrow \neg\mathsf{t}(c))$

6. $\vdash_{\mathsf{BCL}} \Diamond(\mathsf{cn}_{Y, Atm_0} \wedge \mathsf{t}(c)) \rightarrow \bigwedge_{c' \in Val \setminus \{c\}} \Diamond(\mathsf{cn}_{Y, Atm_0} \wedge \neg\mathsf{t}(c'))$
   by countraposition of 5

7. $\vdash_{\mathsf{BCL}} \bigwedge_{c' \in Val \setminus \{c\}} (\Diamond(\mathsf{cn}_{Y, Atm_0} \wedge \neg\mathsf{t}(c)) \rightarrow \Box(\mathsf{cn}_{Y, Atm_0} \rightarrow \neg\mathsf{t}(c'))$
   by propositional logic and countraposition of **Funct**

8. $\vdash_{\mathsf{BCL}} \Diamond(\mathsf{cn}_{Y, Atm_0} \wedge \mathsf{t}(c)) \rightarrow \bigwedge_{c' \in Val \setminus \{c\}} \Box(\mathsf{cn}_{Y, Atm_0} \rightarrow \neg\mathsf{t}(c'))$
   from 6, 7 by Modus Ponens

9. $\vdash_{\mathsf{BCL}} \bigwedge_{c' \in Val \setminus \{c\}} \Box(\mathsf{cn}_{Y, Atm_0} \rightarrow \neg\mathsf{t}(c')) \leftrightarrow \Box(\mathsf{cn}_{Y, Atm_0} \rightarrow \mathsf{t}(c))$
   by **AtLeast**, **AtMost**, **Nec**, **K** and propositional logic

10. $\vdash_{\mathsf{BCL}} \Diamond(\mathsf{cn}_{Y, Atm_0} \wedge \mathsf{t}(c)) \rightarrow \Box(\mathsf{cn}_{Y, Atm_0} \rightarrow \mathsf{t}(c))$
    from 8, 9 by Modus Ponens

□

Appendix C

# Three ways of defining monotone variables in pBFs

The partial domain introduces a salient difference in the definition of monotone variables, a distinction that is hidden in the total domain. As we will see, in this context there are not one, but three possible ways to define monotone variables and functions, which "collapse" into a single one in (total) Boolean functions. To the best of my knowledge, they are not yet studied in literature.

**Definition C.1** (Three definitions of monotone variable in pBF)**.** *Let* $f : 2^{Atm_0} \longrightarrow \{0, 1, 0.5\}$ *and* $dod(f)$ *(domain of definition) denotes* $f^{-1}(1) \cup f^{-1}(0)$*. We say that* $f$ *is* positive, weak positive*, and* even weaker positive *in* $p$*, if the followings hold respectively*

$$\left.\begin{array}{r} \forall s \in dom(f), \\ \forall s \in dod(f), \\ \forall s, s \setminus \{p\} \in dod(f), \end{array}\right\} f(s) \geq f(s \setminus \{p\}).$$

*We say that* $f$ *is* negative, weak negative *and* even weaker negative *in* $p$*, if the following hold respectively*

$$\left.\begin{array}{r} \forall s \in dom(f), \\ \forall s \in dod(f), \\ \forall s, s \setminus \{p\} \in dod(f), \end{array}\right\} f(s) \leq f(s \setminus \{p\})$$

*We say that* $f$ *is* (weak / even weaker) monotone *in* $p$*, if* $f$ *is either (weak / even weaker) positive or (weak / even weaker) negative in* $p$*.*

**Definition C.2** (Three definitions of monotone pBF)**.** *Let* $f : 2^{Atm_0} \longrightarrow \{1, 0, ?\}$*. We say that* $f$ *is monotone*$^{\cdot}$*, if it is monotone*$^{\cdot}$ *in all its variables, where* $\cdot$ *is either blank,* $w$ *or* $ew$*.*

Clearly, a Boolean function $f$ belongs to a special sort of partial Boolean function with $dod(f) = dom(f)$. We say sometimes total Boolean function to denote the Boolean function in usual sense; and non-trivial partial Boolean function to denote the pBF whose domain of definition is a proper subset of its domain.

**Fact C.1.** *1. In total Boolean functions,*

(a) *A variable is positive (negative, monotone) iff it is positive$^{\cdot}$, for* $\cdot \in \{w, ew\}$

(b) *a variable can be both positive and negative, iff it is inessential;*[1]

(c) *the function is trivial, namely* $\forall s, s' \in dom(f), f(s) = f(s')$*, iff every variable is both positive and negative;*

(d) *a variable can be both non-negative,and non-positive.*

2. *In partial Boolean functions,*

(a) *a variable is positive implies that it is positive$^{w}$, which implies that it is positive$^{ew}$, and same to negative;*

(b) *a variable can be both positive$^{\cdot}$ and negative$^{\cdot}$, iff it is inessential, where* $\cdot$ *is blank or* $w$*;*

(c) *possibly all variables are both positive$^{ew}$ and negative$^{ew}$, while the function is non-trivial*

- *Example:* $dod(f) = \{11, 00\}$ *s.t.* $f(11) = 1, f(00) = 0$*;*

(d) *a variable can be both non-negative$^{\cdot}$,and non-positive$^{\cdot}$, where* $\cdot$ *is blank,* $w$ *or* $ew$*.*

The following example demonstrates that how $s \in dom(f)$ but $s \notin dod(f)$ can affect at which level of the "hierarchy of monotonicity" a variable locates.

**Example C.1.** *Let* $f : 2^{\{p_1,p_2,p_3,q,r\}} \longrightarrow \{0, 1, ?\}$ *s.t.* $f(\emptyset) = f(\{r\}) = f(\{p_2, q\}) =?$ *and* $f(\{p_3, r\}) = 0$*, and* $f(s) = 1$ *for all other states. Then,* $f$ *is positive in* $p_1$*; positive$^{w}$ but non-positive in* $p_2$ *because of* $\{p_2, q\}$ *and* $\{q\}$*; positive$^{ew}$ but non-positive$^{w}$ in* $p_3$ *because of* $\{p_3, r\}$ *and* $\{r\}$*.*

A natural question is, can we have "sharper" results using the other definitions of monotone variable in analyzing Horty case base? For the "if $p \in Plt$" part we have achieved the best, since that $p$ is positive implies that $p$ is positive$^{w}$, which further implies that $p$ is positive$^{ew}$, as Fact C.1 says. And same for $p \in Dfd$. Also, it is not surprising that, e.g., $p$ is non-negative$^{w}$ implies $p \in Plt$, since by Fact C.1 being non-negative$^{w}$ implies being non-negative. It is rather the correspondences between a variable being non-weakly-monotone and its role playing in the case base that can be more informative.

**Proposition C.1.** *Fix a language with* $Atm_0 = Plt \cup Dfd$*. Let* $f$ *be the genuine classifier of some consistent Horty case base. Then* $\forall p \in Atm_0$*,*

- $p$ *is non-negative$^{w}$, if and only if,* $p$ *is present in the reason of some landmark precedent for* 1*;*
- $p$ *is non-positive$^{w}$, if and only if,* $p$ *is present in the reason of some landmark precedent for* 0*.*

[1] Recall the definition of essential variable Definition 2.16. Plainly speaking, it means $f(s) = f(s \setminus p)$ for all $s$. The only adjustment we need to make here is that $f$ does not need to be total.

*Proof.* We prove the second half, and the first half is proven in the same way. Let $p$ by non-positive$^w$. By definition it promises there $\exists s \in dod(f), f(s) < f(s \setminus \{p\})$. We thus know it has to be $f(s) = 0 \neq f(s \setminus \{p\})$. By the definition of genuine classifier, from $f(s) = 0$ we know there is a precedent $(s', Y, 0)$ s.t. $Y \subseteq s$ and $s' \cap Plt \supseteq s \cap Plt$. But $f(s \setminus \{p\} \neq 0$, meaning $Y \nsubseteq s \setminus \{p\}$ or $s' \cap Plt \not\supseteq (s \setminus \{p\}) \cap Plt$. The latter is impossible in light of the monotonicity of $\supseteq$. We therefore have shown $p \in Y \subseteq Dfd$, otherwise it is impossible to have $Y \nsubseteq s \setminus \{p\}$. □

At this point comes a subtlety. The proposition holds, because in the definition of, e.g. weak negative variable we use the expression like $\forall s \in dod(f), f(s) \leq f(s \setminus \{p\})$. If we redefine it as comparing $f(s)$ with $f(s \cup \{p\})$, the proposition above will be revised as $p$ is absent from some landmark precedent for 0. This observation implies that the notion of weak monotone variable should better be divided as two definitions. However, it is too pedantic to go through that regarding our purpose here. So we treat it as transitional and move on to the last definition.

**Proposition C.2.** *Fix a language with $Atm_0 = Plt \cup Dfd$. Let $f$ be the genuine classifier of some consistent Horty case base. Then $\forall p \in Atm_0$,*

- *$p$ is non-negative$^{ew}$, if and only if*
  - *$p$ is present in the reason of some landmark precedent for* 1 and *$p$ is absent from some landmark precedent for* 0*;*
- *$p$ is non-negative$^{ew}$, if and only if*
  - *$p$ is present in the reason of some landmark precedent for* 0 and *$p$ is absent from some landmark precedent for* 1.

*Proof.* Again we prove the first half part only. By definitions of positive$^{ew}$ variable and genuine classifier, we have some $s, s \setminus \{p\}$ s.t. $f(s) = 0$ and $f(s \setminus \{p\}) = 1$. There thus must exist two landmarks $(s_0, Y, 0), (s_1, X, 1)$ s.t.

- $s \setminus \{p\} \not\supseteq Y \subseteq s$, though $s_0 \cap Plt \supseteq s \cap Plt \supseteq (s \setminus \{p\}) \cap Plt$;
- $s \cap Dfd \not\supseteq s_1 \cap Dfd \supseteq (s \setminus \{p\}) \cap Dfd$, though $X \subseteq (s \setminus \{p\}) \cap Plt \subseteq s \cap Plt$.

Hence we have $p \in Y \subseteq Dfd$ and $p \notin s_1 \cap Dfd$, which are literally what the proposition says. □

It becomes not a surprise that by strengthening $p$,[2] e.g. from non-positive to non-positive$^{ew}$, we simply replace the "or" in Proposition 4.7 with an "and". In those (pairs of) cases, $p$ plays the pivotal role not once but twice, which indicates these cases, jointly, are even more informative than the usual landmarks.

In a nutshell, we show the natural and nice correspondences between the three definitions of monotone variables on one hand, and whether $p$ is partitioned in *Plt* or *Dfd*, and plays a pivotal role in some landmark(s) of the case base (by its presence and/or absence) on the other hand.

[2]Since positive$^{ew}$ness is weaker than positiveness, certainly the negation of the former is stronger than the negation of the latter.

Appendix D

# Proofs for Chapter 6

**Proof of Theorem 6.1**

*Proof.* We start with the left-to-right direction of the proof. Let $(\Gamma, s_0, f_0)$ be a pointed MCM with $\Gamma = (S, F_S)$, $S \subseteq 2^{Atm_0}$ and $F_S \subseteq Val^S$ such that $(\Gamma, s_0, f_0) \models \varphi$. We define the tuple $M = (W, \sim_\square, \sim_\blacksquare, V)$ as follows:

- $W = \{(s, f) : s \in S \text{ and } f \in F_S\}$,
- $\forall (s, f), (s', f') \in W$, $(s, f) \sim_\square (s', f')$ iff $f = f'$
- $\forall (s, f), (s', f') \in W$, $(s, f) \sim_\blacksquare (s', f')$ iff $s = s'$,
- $\forall (s, f) \in W$, $V(s, f) = s \cup \{\mathtt{t}(f(s))\}$.

It is routine exercise to verify that $M$ so defined is an MDM. Moreover, by induction on the structure of $\varphi$, it is easy to prove that "$(\Gamma, s, f) \models \varphi$ iff $(M, (s, f)) \models \varphi$" for every $s \in S$ and $f \in F_S$. Thus, $(M, (s_0, f_0)) \models \varphi$ since $(\Gamma, s_0, f_0) \models \varphi$.

Let us now prove the right-to-left direction. Let $M = (W, \sim_\square, \sim_\blacksquare, V)$ be an MDM and $w_0 \in W$ such that $(M, w_0) \models \varphi$. Given $v \in W$, let $|v| = \{u \in W : v \sim_\square u \text{ and } V(v) = V(u)\}$. We transform the MDM $M$ into a tuple $M' = (W', \sim'_\square, \sim'_\blacksquare, V')$ such that:

- $W' = \{|v| : v \in W\}$,
- $\forall |v|, |u| \in W'$, $|v| \sim_\square |u|$ iff $\exists v' \in |v|, u' \in |u|$ such that $v' \sim_\square u'$,
- $\forall |v|, |u| \in W'$, $|v| \sim'_\blacksquare |u|$ iff $\exists v' \in |v|, u' \in |u|$ such that $v' \sim_\blacksquare u'$,
- $\forall |v| \in W'$, $V'(|v|) = V(v)$.

Like what we did for $V$, let $V'_Y(|v|) = V'(|v|) \cap Y$ for all $Y \subseteq Atm$.

It is a routine exercise to verify that $M'$ is an MDM and, by induction on the structure of $\varphi$, that "$(M, v) \models \varphi$ iff $(M', |v|) \models \varphi$" for every $v \in W$. Thus, $(M, |w_0|) \models \varphi$ since $(M, w_0) \models \varphi$. Finally, because of Constraints **C2** and **C3** in Definition 6.3, the following property holds:

$$(\mathbf{C6}) \quad (\sim'_\square \cap \sim'_\blacksquare) = id_{W'},$$

where $id_{W'}$ is the identity relation on $W'$.

Let $W'/\sim'_\square$ be the quotient set of $W'$ by the equivalence relation $\sim'_\square$. We note $\tau, \tau', \ldots$ its elements. Given $\tau, \tau' \in W'/\sim'_\square$, we write $\tau \approx_F \tau'$ if and only

if, $\forall|v| \in \tau, \forall|u| \in \tau'$, if $V'_{Atm_0}(|v|) = V'_{Atm_0}(|u|)$ then $V'_{Dec}(|v|) = V'_{Dec}(|u|)$. Given $|v|, |u| \in W'$, we write $|v| \simeq |u|$ if and only if $\exists \tau, \tau' \in W'/\sim'_{\Box}$ such that $|v| \in \tau, |u| \in \tau'$, $\tau \approx_F \tau'$ and $V'_{Atm_0}(|v|) = V'_{Atm_0}(|u|)$. Clearly, $\approx_F$ and $\simeq$ are equivalence relations.

We are going to transform the MDM $M'$ into an MDM which does not contain multiple copies of the same function and which satisfies the same formulas as $M'$. We define it to be a tuple $M'' = (W'', \sim''_{\Box}, \sim''_{\blacksquare}, V'')$ such that:

- $W'' = \{\simeq(|v|) : |v| \in W'\}$,
- $\forall \simeq(|v|), \simeq(|u|) \in W''$, $\simeq(|v|) \sim''_{\Box} \simeq(|u|)$ iff $\exists |v'| \in \simeq(|v|), |u'| \in \simeq(|u|)$ such that $|v'| \sim'_{\Box} |u'|$,
- $\forall \simeq(|v|), \simeq(|u|) \in W''$, $\simeq(|v|) \sim''_{\blacksquare} \simeq(|u|)$ iff $\exists |v'| \in \simeq(|v|), |u'| \in \simeq(|u|)$ such that $|v'| \sim'_{\blacksquare} |u'|$,
- $\forall \simeq(|v|) \in W''$, $V''(\simeq(|v|)) = V'(|v|)$.

Again, it is routine to verify that $M''$ is an MDM which satisfies the previous Constraint **C6**. Moreover, by induction on the structure of $\varphi$, it is easy to prove that "$(M', |v|) \models \varphi$ iff $(M'', \simeq(|v|)) \models \varphi$" for every $|v| \in W'$. Thus, $(M'', \simeq(|w_0|)) \models \varphi$ since $(M', |w_0|) \models \varphi$.

We can easily build an MCM isomorphic to $M''$.

□

**Proof of Theorem 6.3**

*Proof.* The left-to-right direction is trivial. We prove the right-to-left direction. Suppose $Atm_0$ is infinite. Moreover, let $M = (W, \sim_{\Box}, \sim_{\blacksquare}, V)$ be a finite quasi-MDM and $w_0 \in W$ such that $(M, w_0) \models \varphi$. Since $Atm_0$ is infinite and $W$ is finite, we can define an injection $g : W \longrightarrow Atm_0 \setminus Atm(\varphi)$. We define the tuple $M' = (W', \sim'_{\Box}, \sim'_{\blacksquare}, V')$ as follows:

- $W' = W$;
- $\sim'_{\Box} = \sim_{\Box}$
- $\sim'_{\blacksquare} = \sim_{\blacksquare}$;
- for every $w \in W'$,

$$V'(w) = (V(w) \setminus \{g(v) : v \in W \text{ and } w \neq v\}) \cup \{g(w)\}.$$

It is routine to verify that $M'$ is a finite MDM. Indeed, $V'_{Atm_0}(w) \neq V'_{Atm_0}(v)$ for all $w, v \in W'$ such that $w \neq v$. This guarantees that $M'$ satisfies the "functionality" constraint **C2**. Moreover, by induction on the structure of $\varphi$, it is straightforward to prove that "$(M, v) \models \varphi$ iff $(M', v) \models \varphi$" for every $v \in W$. The crucial point of the proof is that $\sim'_{\Box} = \sim_{\Box}$ and $\sim'_{\blacksquare} = \sim_{\blacksquare}$. Thus, $(M', w_0) \models \varphi$ since $(M, w_0) \models \varphi$. □

**Proof of Theorem 6.4**

*Proof.* The right-to-left direction is clear. We are going to prove the left-to-right direction by a filtration argument.

Let $M = (W, \sim_{\Box}, \sim_{\blacksquare}, V)$ be a quasi-MDM and $w_0 \in W$ such that $(M, w_0) \models \varphi$. It is routine to verify that $(\sim_{\Box} \cup \sim_{\blacksquare})^* = \sim_{\Box} \circ \sim_{\blacksquare} = \sim_{\blacksquare} \circ \sim_{\Box}$. Thus, we can define $M' = (W', \sim'_{\Box}, \sim'_{\blacksquare}, V')$ to be the submodel of $M$ generated from $w_0$ through the relation $\sim_{\Box} \circ \sim_{\blacksquare}$. $M'$ is a quasi-MDM and $(M', w_0) \models \varphi$.

Let $sf(\varphi)$ be the set of all subformulas of $\varphi$ and let $sf^+(\varphi) = sf(\varphi) \cup Dec$. Moreover, for every $v \in W'$, let $\Theta(v) = \{\psi \in sf^+(\varphi) : (M', v) \models \psi\}$. For every $v, u \in W'$, we define

$$v \simeq u \text{ iff } \Theta(v) = \Theta(u).$$

Moreover, we define $[v] = \{u \in W' : v \simeq u\}$.

We construct a new model $M'' = (W'', \sim''_{\Box}, \sim''_{\blacksquare}, V'')$ where:

- $W'' = \{[v] : v \in W'\}$;
- $[v] \sim''_{\Box} [u]$ iff

$$\forall \Box\psi \in sf(\varphi), ((M', v) \models \Box\psi \text{ iff } (M', u) \models \Box\psi);$$

- $[v] \sim''_{\blacksquare} [u]$ iff

$$\forall \blacksquare\psi \in sf(\varphi), ((M', v) \models \blacksquare\psi \text{ iff } (M', u) \models \blacksquare\psi) \text{ and}$$
$$\forall p \in sf(\varphi) \cap Atm_0, ((M', v) \models p \text{ iff } (M', u) \models p);$$

- $V''([v]) = V'_{sf(\varphi) \cap Atm_0}(v) \cup V'_{Dec}(v)$.

$M''$ is indeed a filtration, for it satisfies that if $v \sim_{\boxplus} u$, then $[v] \sim_{\boxplus} [u]$; and if $\boxplus\psi \in sf(\varphi)$ and $(M', v) \models \boxplus\psi$, then $(M', u) \models \psi$, for $\boxplus \in \{\Box, \blacksquare\}$. Additionally, the valuation function is defined in the standard way.

To check that $M''$ is a finite quasi-MDM, we go through all constraints. For **C1** a crucial fact is that $M'$ generated from $w_0$ through $\sim_{\Box} \circ \sim_{\blacksquare}$, viz. $\forall v, u \in W', v \sim_{\Box} \circ \sim_{\blacksquare} u$ and $v \sim_{\blacksquare} \circ \sim_{\Box} u$. **C3** holds because of the definition of $\sim''_{\blacksquare}$. **C4**, **C5** hold, since $V''$ not only considers $sf(\varphi) \cap Atm_0$ but also $Dec$.

It is routine to verify that $M'' = (W'', \sim''_{\Box}, \sim''_{\blacksquare}, V'')$ is a filtration of $M'$ and is a finite quasi-MDM. Therefore, $(M'', [w_0]) \models \varphi$. □

**Proof of Theorem 6.6**

*Proof.* Suppose $|Atm_0|$ is finite. Then, the class **MCM** is bounded by some integer $k$. So, in order to determine whether a formula $\varphi$ is satisfiable for the class **MCM**, it is sufficient to verify whether $\varphi$ is satisfied in one of these MCMs. This verification takes a polynomial time in the size of $\varphi$ since it is a repeated model checking and model checking in the product modal logic $S5^2$ is polynomial. □

**Proof of Theorem 6.7**

*Proof.* We know that satisfiability checking for the product modal logic $S5^2$ with two S5 modalities $\Box_1$ and $\Box_2$ is NEXPTIME-complete [Gabbay *et al.* 2003]. We have a polynomial reduction of satisfiability checking for $\mathcal{L}$-formulas relative to the class **MCM** to the latter problem. In particular, given a formula $\varphi \in \mathcal{L}$, we can translate it into a formula $tr(\varphi)$ of $S5^2$ where the translation $tr$ is defined as follows: (i) $tr(q) = q$ for $q \in Atm$, (ii) $tr(\neg\varphi) = \neg tr(\varphi)$, (iii) $tr(\varphi_1 \wedge \varphi_2) = tr(\varphi_1) \wedge tr(\varphi_2)$, (iv) $tr(\Box\varphi) = \Box_1 tr(\varphi)$, (v) $tr(\blacksquare\varphi) = \Box_2 tr(\varphi)$. We have that $\varphi$ is satisfiable for the class **MCM** if and only $\bigwedge_{\chi\in\Delta} \Box_1\Box_2\chi \wedge tr(\varphi)$ is a satisfiable formula of the product modal logic $S5^2$, where $\Delta$ is the following finite theory corresponding to the Axioms $\mathbf{Indep}_{\blacksquare p}$, $\mathbf{Indep}_{\blacksquare\neg p}$, $\mathbf{AtMost}_{\mathsf{t}(c)}$ and $\mathbf{AtLeast}_{\mathsf{t}(c)}$ of the logic WPLC:

$$\Delta = \{ \bigvee_{c\in Val} \mathsf{t}(c)\} \cup \{\mathsf{t}(c) \rightarrow \neg\mathsf{t}(c') : c \neq c'\} \cup \{p \rightarrow \Box_2 p : p \in Atm_0(\varphi)\} \cup$$
$$\{\neg p \rightarrow \Box_2\neg p : p \in Atm_0(\varphi)\},$$

and $Atm_0(\varphi)$ is the set of atoms in $Atm_0$ which occur in $\varphi$. □

# Bibliography


[Aiguier *et al.* 2018] Marc Aiguier, Jamal Atif, Isabelle Bloch and Céline Hudelot. *Belief revision, minimal change and relaxation: A general framework based on satisfaction systems, and applications to description logics.* Artificial Intelligence, vol. 256, pages 160–180, 2018. (Cited on page 68.)

[Aleven 2003] Vincent Aleven. *Using background knowledge in case-based legal reasoning: a computational model and an intelligent learning environment.* Artificial Intelligence, vol. 150, no. 1-2, pages 183–237, 2003. (Cited on pages 52 and 65.)

[Amershi *et al.* 2014] Saleema Amershi, Maya Cakmak, William.B. Knox and Todd Kulesza. *Power to the people: The role of humans in interactive machine learning.* AI Magazine, vol. 35, no. 4, pages 105–120, 2014. (Cited on page 102.)

[Amgoud & Ben-Naim 2022] Leila Amgoud and Jonathan Ben-Naim. *Axiomatic Foundations of Explainability.* In 31st International Joint Conference on Artificial Intelligence (IJCAI 2022), 2022. (Cited on page 24.)

[Ashley 1990] Kevin D. Ashley. Modeling legal argument: Reasoning with cases and hypotheticals. MIT, 1990. (Cited on pages 52 and 65.)

[Atkinson *et al.* 2020] Katie Atkinson, Trevor Bench-Capon and Danushka Bollegala. *Explanation in AI and law: Past, present and future.* Artificial Intelligence, vol. 289, page 103387, 2020. (Cited on pages 52 and 66.)

[Audemard *et al.* 2021] Gilles Audemard, Steve Bellart, Louenas Bounia, Frédéric Koriche, Jean-Marie Lagniez and Pierre Marquis. *On the Computational Intelligibility of Boolean Classifiers.* In Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning, volume 18, pages 74–86, 2021. (Cited on page 24.)

[Baltag & van Benthem 2021] Alexandru Baltag and Johan van Benthem. *A simple logic of functional dependence.* Journal of Philosophical Logic, vol. 50, no. 5, pages 939–1005, 2021. (Cited on pages 27 and 92.)

[Bezhanishvili & Hodkinson 2004] Nick. Bezhanishvili and Ian. M. Hodkinson. *All Normal Extensions of S5-squared Are Finitely Axiomatizable.* Studia Logica, vol. 78, no. 3, pages 443–457, 2004. (Cited on page 96.)

[Bezhanishvili & Marx 2003] Nich. Bezhanishvili and Maarten. Marx. *All Proper Normal Extensions of S5-square have the Polynomial Size Model Property.* Studia Logica, vol. 73, no. 3, pages 367–382, 2003. (Cited on page 96.)

[Biran & Cotton 2017] Or Biran and Courtenay Cotton. *Explanation and justification in machine learning: A survey.* In IJCAI-17 workshop on explainable AI (XAI), volume 8(1), pages 8–13, 2017. (Cited on page 24.)

[Blackburn *et al.* 2001] Patrick Blackburn, Maarten de Rijke and Yde Venema. Modal logic. Cambridge University Press, Cambridge, Massachusetts, 2001. (Cited on pages 19, 21 and 124.)

[Borgida 1985] Alexander Borgida. *Language features for flexible handling of exceptions in information systems.* ACM Transactions on Database Systems (TODS), vol. 10, no. 4, pages 565–603, 1985. (Cited on page 38.)

[Burgess 1981] John P Burgess. *Quick completeness proofs for some logics of conditionals.* Notre Dame Journal of Formal Logic, vol. 22, no. 1, pages 76–84, 1981. (Cited on page 69.)

[Card 2017] Dallas Card. *The black box metaphor in machine learning.* https://dallascard.medium.com/the-black-box-metaphor-in-machine-learning-4e57a3a1d2b0, 2017. Accessed: 2023-07-05. (Cited on pages 4 and 5.)

[Caridroit *et al.* 2017] Thomas Caridroit, Jean-Marie Lagniez, Daniel Le Berre, Tiago de Lima and Valentin Montmirail. *A SAT-Based Approach for Solving the Modal Logic S5-Satisfiability Problem.* In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17), pages 3864–3870. AAAI Press, 2017. (Cited on page 48.)

[Carnap 1967] Rudolf Carnap. The logical structure ofthe world. Berkeley-Los Angeles, Univ, 1967. (Cited on page 21.)

[Charrier *et al.* 2016] Tristan Charrier, Andreas Herzig, Emiliano Lorini, Faustine Maffre and François Schwarzentruber. *Building Epistemic Logic from Observations and Public Announcements.* In Proceedings of the Fifteenth International Conference on Principles of Knowledge Representation and Reasoning (KR 2016), pages 268–277. AAAI Press, 2016. (Cited on page 46.)

[Clarke & Schlingloff 2001] Edmund M. Clarke and Bernd-Holger Schlingloff. *Model checking.* In Alan J. A. Robinson and Andrei Voronkov, editors, Handbook of automated reasoning, pages 1635–1790. Elsevier, 2001. (Cited on page 127.)

[Crama & Hammer 2011] Yves Crama and Peter L Hammer. Boolean functions: Theory, algorithms, and applications. Cambridge University Press, 2011. (Cited on pages 11, 12, 13, 14 and 63.)

[Cunningham & Delany 2022] Padraig Cunningham and Sarah Jane Delany. *K-Nearest Neighbour Classifiers - A Tutorial.* ACM Computing Surveys, vol. 54, no. 6, pages 1–25, 2022. (Cited on page 45.)

[Dalal 1988] Mukesh Dalal. *Investigations into a theory of knowledge base revision: preliminary report.* In Proceedings of the Seventh National Conference on Artificial Intelligence, volume 2, pages 475–479. Citeseer, 1988. (Cited on pages 38, 68 and 73.)

[Darwiche & Hirth 2020] Adnan Darwiche and Auguste Hirth. *On the Reasons Behind Decisions.* In 24th European Conference on Artificial Intelligence (ECAI 2020), volume 325 of *Frontiers in Artificial Intelligence and Applications*, pages 712–720. IOS Press, 2020. (Cited on pages 2, 3, 18, 19, 24, 25, 41 and 43.)

[Darwiche 2020] Adnan Darwiche. *Three modern roles for logic in AI.* In Proceedings of the 39th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, pages 229–243, 2020. (Cited on page 2.)

[Delgrande & Peppas 2015] James P Delgrande and Pavlos Peppas. *Belief revision in Horn theories.* Artificial Intelligence, vol. 218, pages 1–22, 2015. (Cited on page 68.)

[Dhurandhar *et al.* 2018] Amit Dhurandhar, Pin-Yu Chen, Ronny Luss, Chun-Chen Tu, Paishun Ting, Karthikeyan Shanmugam and Payel Das. *Explanations based on the missing: Towards contrastive explanations with pertinent negatives.* In Advances in Neural Information Processing Systems, pages 592–603, 2018. (Cited on page 24.)

[Dizadji-Bahmani & Bradley 2014] Foad Dizadji-Bahmani and Seamus Bradley. Lewis' account of counterfactuals is incongruent with Lewis' account of laws of nature. available in `http://philsci-archive.pitt.edu/10875/`, 2014. (Cited on page 68.)

[Eiter & Gottlob 1992] Thomas Eiter and Georg Gottlob. *On the Complexity of Propositional Knowledge Base Revision, Updates, and Counterfactuals.* Artif. Intell., vol. 57, no. 2-3, pages 227–270, 1992. (Cited on page 73.)

[Fagin *et al.* 1995] Ronald Fagin, Yoram Moses, Joseph Y Halpern and Moshe Y Vardi. Reasoning about knowledge. MIT Press, 1995. (Cited on page 27.)

[Floridi 2010] Luciano Floridi. *Information, possible worlds and the cooptation of scepticism.* Synthese, vol. 175, no. Suppl 1, pages 63–88, 2010. (Cited on pages 68 and 74.)

[Friedman & Halpern 1994] Nir Friedman and Joseph Y Halpern. *On the complexity of conditional logics.* In Principles of Knowledge Representation and Reasoning, pages 202–213. Morgan Kaufmann, 1994. (Cited on pages 71, 75, 79 and 81.)

[Gabbay *et al.* 2003] D. M. Gabbay, A. Kurucz, F. Wolter and M. Zakharyaschev. Many-dimensional modal logics: theory and applications. Elsevier, 2003. (Cited on pages 91, 94 and 142.)

[Gärdenfors 1984] Peter Gärdenfors. *Epistemic importance and minimal changes of belief.* Australasian Journal of Philosophy, vol. 62, no. 2, pages 136–157, 1984. (Cited on page 68.)

[Gargov & Goranko 1993] G. Gargov and V. Goranko. *Modal logic with names.* Journal of Philosophical Logic, vol. 22, pages 607–636, 1993. (Cited on page 114.)

[Gargov *et al.* 1987] George Gargov, Solomon Passy and Tinko Tinchev. *Modal Environment for Boolean Speculations: preliminary report.* Mathematical logic and its applications, pages 253–263, 1987. (Cited on page 114.)

[Girard & Triplett 2016] Patrick Girard and Marcus Anthony Triplett. *Ceteris paribus logic in counterfactual reasoning.* In Proceedings of the Fifteenth Conference on Theoretical Aspects of Rationality and Knowledge (TARK 2015), pages 176–193, 2016. (Cited on page 38.)

[Goldszmidt & Pearl 1992] Moisés Goldszmidt and Judea Pearl. *Rank-based systems: A simple approach to belief revision, belief update, and reasoning about evidence and actions.* KR, vol. 92, pages 661–672, 1992. (Cited on page 71.)

[Goodman 1955] Nelson Goodman. Fact, fiction, and forecast. Harvard University Press, 1955. (Cited on page 25.)

[Grahne 1998] Gösta Grahne. *Updates and counterfactuals.* Journal of Logic and Computation, vol. 8, no. 1, pages 87–117, 1998. (Cited on page 68.)

[Grossi *et al.* 2015] Davide Grossi, Emiliano Lorini and François Schwarzentruber. *The Ceteris Paribus Structure of Logics of Game Forms.* Journal of Artificial Intelligence Research, vol. 53, pages 91–126, 2015. (Cited on pages 26, 27, 92, 127 and 128.)

[Grove 1988] A. Grove. *Two modellings for theory change.* J. of Philosophical Logic, vol. 17, pages 157–170, 1988. (Cited on page 68.)

[Halpern 1995] Joseph Y. Halpern. *The Effect of Bounding the Number of Primitive Propositions and the Depth of Nesting on the Complexity of Modal Logic.* Artificial Intelligence, vol. 75, no. 2, pages 361–372, 1995. (Cited on pages 36 and 96.)

[Halpern 2016] Joseph Y Halpern. Actual causality. MiT Press, 2016. (Cited on page 25.)

[Hempel & Oppenheim 1948] Carl G. Hempel and Paul Oppenheim. *Studies in the Logic of Explanation.* Philosophy of science, vol. 15, no. 2, pages 135–175, 1948. (Cited on pages 14, 15 and 24.)

[Herzig *et al.* 2015] Andreas Herzig, Emiliano Lorini and Faustine Maffre. *A Poor Man's Epistemic Logic Based on Propositional Assignment and Higher-Order Observation.* In Proceedings of the 5th International Workshop on Logic, Rationality, and Interaction, Lecture Notes in Computer Science, pages 156–168. Springer, 2015. (Cited on page 46.)

[Herzig 1998] Andreas Herzig. *Logics for belief base updating.* In Didier Dubois, Dov Gabbay, Henri Prade and Philippe Smets, editors, Handbook of defeasible reasoning and uncertainty management, volume 3 - Belief Change, pages 189–231. Kluwer, 1998. (Cited on page 73.)

[Horty & Bench-Capon 2012] John F. Horty and Trevor J. M. Bench-Capon. *A factor-based definition of precedential constraint.* Artificial intelligence and Law, vol. 20, pages 181–214, 2012. (Cited on pages 52 and 66.)

[Horty 2004] John F. Horty. *The Result Model of Precedent.* Legal Theory, vol. 10, pages 19–31, 2004. (Cited on page 52.)

[Horty 2011] John F. Horty. *Rules and reasons in the theory of precedent.* Legal theory, vol. 17, pages 1–33, 2011. (Cited on pages 52, 53, 54, 55 and 66.)

[Horty 2017] John Horty. *Reasoning with Dimensions and Magnitudes.* In International Conference on Artificial Intelligence and Law, ICAIL2017. ACM, 2017. (Cited on page 52.)

[Huang *et al.* 2022] Xuanxiang Huang, Yacine Izza, Alexey Ignatiev, Martin Cooper, Nicholas Asher and Joao Marques-Silva. *Tractable explanations for d-DNNF classifiers.* In Proceedings of the AAAI Conference on Artificial Intelligence, volume 36, pages 5719–5728, 2022. (Cited on pages 40 and 62.)

[Ignatiev *et al.* 2019] Alexey Ignatiev, Nina Narodytska and Joao Marques-Silva. *Abduction-based explanations for machine learning models.* In Proceedings of the Thirty-third AAAI Conference on Artificial Intelligence (AAAI-19), volume 33, pages 1511–1519, 2019. (Cited on pages 3, 18, 24, 25, 26 and 40.)

[Ignatiev *et al.* 2020a] Alexey Ignatiev, Martin C. Cooper, Mohamed Siala, Emmanuel Hebrard and Joao Marques-Silva. *Towards Formal Fairness in Machine Learning.* In International Conference on Principles and Practice of Constraint Programming, pages 846–867. Springer, 2020. (Cited on page 43.)

[Ignatiev *et al.* 2020b] Alexey Ignatiev, Nina Narodytska, Nicholas Asher and Joao Marques-Silva. *From contrastive to abductive explanations and back again.* In

International Conference of the Italian Association for Artificial Intelligence, pages 335–355. Springer, 2020. (Cited on pages 2, 3, 17, 18, 24, 26 and 41.)

[Katsuno & Mendelzon 1991] Hirofumi Katsuno and Alberto O Mendelzon. *Propositional Knowledge Base Revision and Minimal Change*. Artificial Intelligence, vol. 52, 1991. (Cited on page 68.)

[Kearns & Roth 2019] Michael Kearns and Aaron Roth. The ethical algorithm: The science of socially aware algorithm design. Oxford University Press, 2019. (Cited on pages 5, 6, 7, 89 and 97.)

[Kment 2006] Boris Kment. *Counterfactuals and explanation*. Mind, vol. 115, no. 458, pages 261–310, 2006. (Cited on page 24.)

[Kraus *et al.* 1990] Sarit Kraus, Daniel Lehmann and Menachem Magidor. *Nonmonotonic reasoning, preferential models and acumulative logics*. Artificial Intelligence, vol. 44, no. 1-2, pages 167–207, 1990. (Cited on pages 68 and 71.)

[Ladner 1977] Richard E. Ladner. *The computational complexity of provability in systems of modal propositional logic*. SIAM journal on computing, vol. 6, no. 3, pages 467–480, 1977. (Cited on page 128.)

[Lewis 1973] David K. Lewis. Counterfactuals. Harvard University Press, 1973. (Cited on pages 37, 71 and 73.)

[Lewis 1979] David K. Lewis. *Counterfactual dependence and time's arrow*. Noûs, pages 455–476, 1979. (Cited on page 25.)

[Lewis 1986] David K. Lewis. *Causal Explanation*. In Philosophical Papers, volume 2, pages 214–240. Oxford University Press, 1986. (Cited on page 27.)

[Lewis 1995] David K. Lewis. *Causation*. Journal of Philosophy, vol. 70, no. 17, pages 556–567, 1995. (Cited on pages 25 and 71.)

[Liu & Lorini 2021] Xinghan Liu and Emiliano Lorini. *A Logic for Binary Classifiers and Their Explanation*. In P. Baroni, C. Benzmüller and Y. N. Wáng, editors, Logic and Argumentation - 4th International Conference, CLAR 2021, Hangzhou, China, 2021, Proceedings, Lecture Notes in Computer Science, pages 302–321. Springer, 2021. (Cited on page 7.)

[Liu & Lorini 2022] Xinghan Liu and Emiliano Lorini. *A logic of "black box" classifier systems*. In Logic, Language, Information, and Computation: 28th International Workshop, WoLLIC 2022, Iași, Romania, 2022, Proceedings, pages 158–174. Springer, 2022. (Cited on page 8.)

[Liu & Lorini 2023] Xinghan Liu and Emiliano Lorini. *A unified logical framework for explanations in classifier systems*. Journal of Logic and Computation, vol. 33, no. 2, pages 485–515, 2023. (Cited on page 7.)

[Liu *et al.* 2022] Xinghan Liu, Emiliano Lorini, Antonino Rotolo and Giovanni Sartor. *Modelling and Explaining Legal Case-Based Reasoners Through Classifiers.* In Legal Knowledge and Information Systems, pages 83–92. IOS Press, 2022. (Cited on page 8.)

[Lorini 2019] Emiliano Lorini. *Reasoning about cognitive attitudes in a qualitative setting.* In Logics in Artificial Intelligence: 16th European Conference, JELIA 2019, Rende, Italy, May 7–11, 2019, Proceedings 16, pages 726–743. Springer, 2019. (Cited on page 114.)

[Lundberg & Lee 2017] Scott Lundberg and Su-In Lee. *A unified approach to interpreting model predictions.* arXiv preprint arXiv:1705.07874, 2017. (Cited on pages 1, 16 and 17.)

[Marques-Silva & Ignatiev 2023] Joao Marques-Silva and Alexey Ignatiev. *No silver bullet: interpretable ML models must be explained.* Frontiers in Artificial Intelligence, vol. 6, page 1128212, 2023. (Cited on page 91.)

[Marques-Silva 2023] Joao Marques-Silva. *Logic-based explainability in machine learning.* In Reasoning Web. Causality, Explanations and Declarative Knowledge: 18th International Summer School 2022, Berlin, Germany, September 27–30, 2022, Tutorial Lectures, pages 24–104. Springer, 2023. (Cited on page 2.)

[Marx 1999] Maarten Marx. *Complexity of products of modal logics.* Journal of Logic and Computation, vol. 9, no. 2, pages 197–214, 1999. (Cited on page 103.)

[Mertes *et al.* 2022] Silvan Mertes, Christina Karle, Tobias Huber, Katharina Weitz, Ruben Schlagowski and Elisabeth André. *Alterfactual Explanations–The Relevance of Irrelevance for Explaining AI Systems.* arXiv preprint arXiv:2207.09374, 2022. (Cited on page 24.)

[Miller *et al.* 2022] Tim Miller, Robert Hoffman, Ofra Amir and Andreas Holzinger. *Special issue on Explainable Artificial Intelligence (XAI).* Artificial Intelligence, vol. 307, 2022. (Cited on page 52.)

[Miller 2019] Tim Miller. *Explanation in artificial intelligence: Insights from the social sciences.* Artificial intelligence, vol. 267, pages 1–38, 2019. (Cited on pages 24 and 102.)

[Miller 2021] Tim Miller. *Contrastive explanation: A structural-model approach.* The Knowledge Engineering Review, vol. 36, 2021. (Cited on page 24.)

[Mittelstadt *et al.* 2019] Brent Mittelstadt, Chris Russell and Sandra Wachter. *Explaining explanations in AI.* In Proceedings of the 2019 conference on Fairness, Accountability, and Transparency, pages 279–288, 2019. (Cited on page 24.)

[Molnar 2023] Christoph Molnar. Interpretable machine learning. Lulu. com, 2023. Available in https://christophm.github.io/interpretable-ml-book/index.html. (Cited on page 1.)

[Mothilal *et al.* 2020] Ramaravind K. Mothilal, Amit Sharma and Chenhao Tan. *Explaining machine learning classifiers through diverse counterfactual explanations*. In Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, pages 607–617, 2020. (Cited on page 24.)

[Pearl 2009] Judea Pearl. Causality. Cambridge university press, 2009. (Cited on page 27.)

[Plaza 2007] J. Plaza. *Logics of public communications*. Synthese, vol. 158, no. 2, pages 165–179, 2007. (Cited on page 100.)

[Pozos-Parra *et al.* 2013] Pilar Pozos-Parra, Weiru Liu and Laurent Perrussel. *Dalal's revision without Hamming distance*. In Advances in Artificial Intelligence and Its Applications: 12th Mexican International Conference on Artificial Intelligence, MICAI 2013, Mexico City, Mexico, November 24-30, 2013, Proceedings, Part I 12, pages 41–53. Springer, 2013. (Cited on page 68.)

[Prakken & Sartor 1998] Henry Prakken and Giovanni Sartor. *Modelling Reasoning with Precedents in a Formal Dialogue Game*. Artificial Intelligence and Law, vol. 6, pages 231–87, 1998. (Cited on page 66.)

[Prakken 2021] Henry Prakken. *A formal analysis of some factor- and precedent-based accounts of precedential constraint*. Artificial Intelligence and Law, 2021. (Cited on pages 53 and 54.)

[Quine 1950] Willard Van Orman Quine. Methods of logic. Harvard University Press, 1950. (Cited on page 71.)

[Quine 1955] Willard V. Quine. *A way to simplify truth functions*. The American mathematical monthly, vol. 62, no. 9, pages 627–631, 1955. (Cited on pages 40 and 98.)

[Ribeiro *et al.* 2016] Marco Tulio Ribeiro, Sameer Singh and Carlos Guestrin. *"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, pages 1135–1144, 2016. (Cited on pages 1, 15 and 16.)

[Rudin 2019] Cynthia Rudin. *Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead*. Nature machine intelligence, vol. 1, no. 5, pages 206–215, 2019. (Cited on page 90.)

[Schwarzentruber 2019] François Schwarzentruber. *The Complexity of Tiling Problems*. arXiv preprint arXiv:1907.00102, 2019. (Cited on pages 105 and 106.)

[Segerberg 1989] Krister Segerberg. *Notes on conditional logic.* Studia Logica, pages 157–168, 1989. (Cited on page 81.)

[Shi *et al.* 2020] Weijia Shi, Andy Shih, Adnan Darwiche and Arthur Choi. *On tractable representations of binary neural networks.* arXiv preprint arXiv:2004.02082, 2020. (Cited on page 24.)

[Shih *et al.* 2018] Andy Shih, Arthur Choi and Adnan Darwiche. *Formal verification of Bayesian network classifiers.* In International Conference on Probabilistic Graphical Models, pages 427–438. PMLR, 2018. (Cited on pages 24 and 41.)

[Sokol & Flach 2019] Kacper Sokol and Peter A. Flach. *Counterfactual explanations of machine learning predictions: opportunities and challenges for AI safety.* In SafeAI@ AAAI, 2019. (Cited on pages 42 and 43.)

[Stalnaker 1968] Robert C Stalnaker. *A theory of conditionals.* In Ifs, pages 41–55. Springer, 1968. (Cited on page 71.)

[Strumbelj & Kononenko 2010] Erik Strumbelj and Igor Kononenko. *An efficient explanation of individual classifications using game theory.* The Journal of Machine Learning Research, vol. 11, pages 1–18, 2010. (Cited on page 16.)

[Van Benthem *et al.* 2006] Johan Van Benthem, Jan Van Eijck and Barteld Kooi. *Logics of communication and change.* Information and Computation, vol. 204, no. 11, pages 1620–1662, 2006. (Cited on pages 27 and 44.)

[Van Der Hoek *et al.* 2011] Wiebe Van Der Hoek, Nicolas Troquard and Michael J Wooldridge. *Knowledge and control.* In Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), pages 719–726. IFAAMAS, 2011. (Cited on page 46.)

[van der Hoek *et al.* 2012] Wiebe van der Hoek, Petar Iliev and Michael J Wooldridge. *A logic of revelation and concealment.* In Proceedings of the International Conference on Autonomous Agents and Multiagent Systems, (AAMAS 2012), pages 1115–1122. IFAAMAS, 2012. (Cited on page 46.)

[van Ditmarsch *et al.* 2005] Hans P van Ditmarsch, Wiebe van der Hoek and Barteld P Kooi. *Dynamic epistemic logic with assignment.* In Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2005), pages 141–148. ACM, 2005. (Cited on page 44.)

[van Ditmarsch *et al.* 2007] Hans van Ditmarsch, Wiebe van Der Hoek and Barteld Kooi. Dynamic epistemic logic, volume 337 of *Synthese Library.* Springer, 2007. (Cited on pages 27 and 100.)

[van Emde Boas 1997] Peter van Emde Boas. *The Convenience of Tilings.* In Complexity, Logic, and Recursion Theory, pages 331–363. CRC Press, 1997. (Cited on page 104.)

[Van Fraassen 1980] Bas C Van Fraassen. The scientific image. Oxford University Press, 1980. (Cited on page 14.)

[Van Woerkom *et al.* 2022] Wijnand Van Woerkom, Davide Grossi, Henry Prakken and Bart Verheij. *Landmarks in case-based reasoning: From theory to data.* In HHAI2022: Augmenting Human Intellect, pages 212–224. IOS Press, 2022. (Cited on page 61.)

[Verma *et al.* 2020] Sahil Verma, John Dickerson and Keegan Hines. *Counterfactual Explanations for Machine Learning: A Review.* arXiv preprint arXiv:2010.10596, 2020. (Cited on pages 24 and 42.)

[Wachter *et al.* 2017] Sandra Wachter, Brent Mittelstadt and Chris Russell. *Counterfactual explanations without opening the black box: Automated decisions and the GDPR.* Harv. JL & Tech., vol. 31, page 841, 2017. (Cited on page 24.)

[Walton 2004] D. Walton. *A new dialectical theory of explanation.* Philosophical Explorations, vol. 7, no. 1, pages 71–89, 2004. (Cited on page 102.)

[Williamson 1988] Timothy Williamson. *First-order logics for comparative similarity.* Notre Dame Journal of Formal Logic, vol. 29, no. 4, 1988. (Cited on pages 68 and 77.)

[Woodward & Hitchcock 2003] James Woodward and Christopher Hitchcock. *Explanatory generalizations, part I: A counterfactual account.* Noûs, vol. 37, no. 1, pages 1–24, 2003. (Cited on page 25.)

[Woodward 2000] James Woodward. *Explanation and invariance in the special sciences.* The British journal for the philosophy of science, vol. 51, no. 2, pages 197–254, 2000. (Cited on pages 24, 25, 26 and 27.)

[Woodward 2003] James Woodward. Making things happen: a theory of causal explanation. Oxford University Press, 2003. (Cited on page 25.)

[Yang & Väänänen 2016] Fan Yang and Jouko Väänänen. *Propositional logics of dependence.* Annals of Pure and Applied Logic, vol. 167, no. 7, pages 557–589, 2016. (Cited on pages 31 and 92.)

[You *et al.* 2017] Seungil You, David Ding, Kevin Canini, Jan Pfeifer and Maya Gupta. *Deep lattice networks and partial monotonic functions.* Advances in neural information processing systems, vol. 30, 2017. (Cited on page 97.)

[Zednik 2021] Carlos Zednik. *Solving the black box problem: A normative framework for explainable artificial intelligence*. Philosophy & technology, vol. 34, no. 2, pages 265–288, 2021. (Cited on page 88.)

[Zheng *et al.* 2020a] Heng Zheng, Davide Grossi and Bart Verheij. *Case-based reasoning with precedent models: Preliminary report*. In Computational Models of Argument, pages 443–450. IOS Press, 2020. (Cited on page 66.)

[Zheng *et al.* 2020b] Heng Zheng, Davide Grossi and Bart Verheij. *Precedent comparison in the precedent model formalism: theory and application to legal cases*. In Proceedings of the EXplainable and Responsible AI in Law (XAILA) Workshop at JURIX, 2020. (Cited on page 66.)