# Improving Formal Explanations in AI 

## Niku Gorji



## THE UNIVERSITY OF SYDNEY

Supervisor: Sasha Rubin

A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy

School of Computer Science
Faculty of Engineering
The University of Sydney
Australia

11 June 2023

## Statement of Originality

This is to certify that the content of this thesis is my own work. This thesis has not been submitted for any degree or other purposes.

I certify that the intellectual content of this thesis is the product of my own work and that all the assistance received in preparing this thesis and sources have been acknowledged.

Name:
Signature:

## Authorship Attribution Statement

This thesis contains published materials:
(1) A major part of Chapter 3 was published as "Sufficient Reasons for Classifier Decisions in the Presence of Domain Constraints" (Gorji and Rubin 2022), and presented at the Association for the Advancement of Artificial Intelligence Conference 2022 (AAAI'22). I co-designed the study with the co-author, analysed the data and wrote the drafts of the manuscript.
(2) A major part of Chapter 6 was published as "A Configurational Analysis of Risk Patterns for Predicting the Outcome After Traumatic Brain Injury" (Gorji, Zador and Poon 2017), and presented at the American Medical Informatics Association Conference 2017 (AMIA'17). It was nominated for the "Distinguished Paper Award" at the conference. I co-designed the study with the co-authors, analysed the data and wrote the drafts of the manuscript.

I certify that the aforementioned authorship attribution statements are correct and I have received permissions from the other authors to include the published materials.
Name:
Signature:
Date:

As supervisor for the candidature upon which this thesis is based, I can confirm that the authorship attribution statements above are correct.

Name: Signature: Date:

## Acknowledgements

This thesis would not exist without the guidance and support of many people: Sasha Rubin, Simon Poon, Joachim Gudmundsson, Gary Weng, Baptiste Lepers, Irena Koprinska, Judy Kay, Joseph Davis, Alrik Theim, Lusine Mkrtchyan. Thank you!

It would also not exist without the encouragement and the spirit of perseverance instilled in me by my parents. Thank you!

I would also like to thank the reviewers for their time and thoughtful comments, and Carol Ong for proofreading this manuscript. Thank you!


#### Abstract

Within the field of explainable AI, a considerable drawback of the current explanation methods is that they do not take background knowledge into account to improve the quality of explanations. We study this problem and present a mechanism to include arbitrary background knowledge on the input domain as constraints into the reasoning process. We show, theoretically and empirically, that the quality of explanations can be enhanced by 1) using domain constraints to improve the parsimony of explanations, and 2) producing more focused explanations by specifying a "context" for an explanation (i.e. a cover and a partial world). Further, we investigate the close connection between explanations and causality by formalising a few relevant concepts and notions from the social science literature. We illustrate the usefulness of these formalised notions for making causal arguments over some canonical examples from the causality literature. Finally, we provide the details of a quantitative approach to improving explanation quality by using a real-life example from medical domains.


## Contents

Statement of Originality ..... ii
Authorship Attribution Statement ..... iii
Acknowledgements ..... iv
Abstract ..... v
Contents ..... vi
List of Figures ..... ix
Chapter 1 Introduction ..... 1
1.1 Contributions ..... 6
Chapter 2 Related Work ..... 10
Chapter 3 Sufficient Reasons in the Presence of Domain Constraints ..... 17
3.1 Preliminaries ..... 18
3.2 Problem Setting ..... 20
3.3 Sufficient Reasons in the Presence of Constraints ..... 21
3.3.1 Variations and Parsimony of Reasons ..... 24
3.4 Computing Sufficient Reasons ..... 25
3.4.1 Illustration ..... 27
3.5 Case Studies and Validation ..... 28
3.6 Discussion ..... 33
Chapter 4 Sufficient Reasons in the Multi-value Setting ..... 34
4.1 Preliminaries ..... 35
4.2 Extending to the Multi-Value Setting ..... 38
4.2.1 Decision Function of a Multi-Value Classifier ..... 39
4.3 Sufficient Reasons in the Multi-Value Setting ..... 41
4.4 Computing Sufficient Reasons in the Multi-value Setting ..... 43
4.4.1 One-hot Encoding ..... 44
4.4.2 Restricting Prime Implicants of $\Delta^{H}$ to be Negative Boolean Terms ..... 45
4.4.3 Size of a reason ..... 48
4.5 Constraint Subsumption in General ..... 50
4.6 Discussion ..... 51
Chapter 5 Causal Arguments and Explanations Within a Context ..... 53
5.1 Preliminaries ..... 54
5.2 Irredundant Prime-Covers as Causal Models ..... 56
5.3 Using Irredundant Prime-Covers to Infer Actual Causation and Causal Explanations ..... 62
5.3.1 Example 1: Loader - General illustration ..... 63
5.3.2 Example 2: Window - Overdetermination ..... 64
5.3.3 Example 3: Backup - Early Preemption ..... 65
5.3.4 Example 4: Inevitable Shock - Switch ..... 66
5.3.5 Example 5: Purple Flame - (In)transitivity of Causation ..... 66
5.3.6 Example 6: Classifier Function ..... 67
5.4 Discussion ..... 68
Chapter 6 Application of Covers and Explanations Within a Cover ..... 70
6.1 Introduction ..... 71
6.2 Prognostic Models and Predictor Variables in TBI. ..... 73
6.3 Qualitative Comparative Analysis ..... 76
6.4 Variable Selection and Dimensionality Reduction ..... 77
6.5 Results ..... 77
6.6 Comparative Analysis Using QCA ..... 79
6.7 Analysis of the 9-Var Model ..... 79
6.8 Analysis of the 7-Var Model ..... 80
6.9 Predicting Outcome with QCA ..... 81
6.10 Discussion ..... 84
Chapter 7 Conclusion ..... 86
Reference for Important Notations and Definitions ..... 89
Bibliography ..... 95

## List of Figures

3.1 Workflow - finding sufficient reasons in the presence of constraints. The focus ofthis chapter is on the right of the dashed line.173.2 RPART decision tree for Case Study 1. ..... 29
3.3 Distribution of shortest reasons, restricted to instances without length $\leqslant 2$ reasons (i.e., 5440 of 109120 instances). Percentages are rounded to the nearest decimal. ..... 30
3.4 RPART decision tree for case study 2 . We drop $V$ and write, e.g., $4 o$ instead of $V_{4, O}$ for readability. ..... 31
4.1 MV Workflow ..... 34
4.2 Decision tree for IRIS ..... 40
6.1 The 6 steps in our analysis ..... 77

## Chapter 1

## Introduction

With increasing deployment of AI systems in safety-critical and sensitive domains, there is a surge of interest in techniques that can better explain the internal decision-making rationale and outputs of these systems to their end users. Explanations were always needed by the developers and the deployers of machine learning (ML) models to examine the internal validity, fairness and correctness of the outputs of the models before deploying them into real-life settings. In the past couple of years, however, the need for explanations has intensified by the need to provide the end users of these systems with the "right to be informed" (Wachter, Mittelstadt and Floridi 2017) attested by the 2016 regulations passed in the EU (GDPR, European Commission 2016), which made explanations a requirement for systems with algorithmic decision-making.

Therefore, for real-world applications of ML models, there is a legal obligation and a social need to ensure that the results, decisions or outputs of these systems can be explained to their end users.

Some ML models are arguably inherently interpretable. Often called intrinsic, transparent, or glass-box models, these include the linear models, decision-tree models, and rule-based models (Linardatos, Papastefanopoulos and Kotsiantis 2020).

Inherently interpretable models can be contrasted with black-box models that are not transparent, nor easily interpretable. Black-box models include deep neural networks (Adadi and Berrada 2018) and the ensemble methods such as bagging, stacking, boosting and random forests (Polikar 2012). Black-box models usually require explicit explanations.

Explicit explanations that accompany an ML model's output after it is generated are often called post-hoc explanations.

Post-hoc explanation techniques can be classified based on the type of problem they apply to (i.e., classification, regression, etc.), whether their scope is global or local, and their algorithmic properties such as search technique, level of approximation (e.g. heuristic vs. exact) and the final outputs (e.g. feature importance vs. rules, and type of rules) (Sokol and Flach 2020; Agarwal et al. 2021).

Black-box models are the most obvious candidates among ML models that need post-hoc explanations; however, even models that are known to be inherently interpretable may become too complex to be easily interpreted by their end users. Studies show that even simple decision trees may have many unnecessary complexities in the rules, and that typically, the rules of a decision tree could be simplified further through logic-based explanation (Izza, Ignatiev and Marques-Silva 2020).

Unnecessary complexities in explanations can hinder their usefulness, so for practical applicability of explanations, it is crucial to ensure that Occam's razor is applied in principle to the explanations before they are returned to a user. Occam's razor dictates that explanations should be devoid of any unnecessary complexities, striving for simplicity and straightforwardness. But simplicity is not the only factor in providing good explanations. It is also important to ensure that explanations align with the knowledge, understanding and expectations of the end users about the problem domain and hence are deemed relevant by them (Miller 2019). A notable drawback of many of the prior explanation methods (Ribeiro, Singh and Guestrin 2018; Ribeiro, Singh and Guestrin 2016; Lundberg and Lee 2017; Shih, Choi and Darwiche 2018; Ignatiev, Narodytska and Marques-Silva 2019a; Ignatiev, Pereira et al. 2018; Huang et al. 2021) is that they do not provide an in-depth evaluation of the role of background knowledge on providing explanations, nor provide methods to systematically incorporate arbitrary domain constraints into the process of finding explanations. This has warranted further research into addressing this problem in the past couple of years (Deutch and Frost 2018; Shrotri et al. 2022) and is the focus of our work in this thesis.

Incorporating background knowledge as domain constraints. Domain constraints are ubiquitous in ML and arise from the structure and inter-dependencies between features present in data (Darwiche 2020). As a simple example, consider a medical setting in which some combinations of drugs are never prescribed together and thus will not appear in any dataset: if we know that "drug A and drug B are never prescribed together" (i.e., the constraint), then an explanation of the form "drug A was prescribed and drug B was not prescribed" is overly redundant; in this situation, it is more parsimonious to supply "drug A was prescribed" instead.

Although it might be obvious how to process explanations to take simple constraints into account, it is by no means obvious how to handle constraints represented by arbitrarily complex formulas. For example, given some well-known but complex rules of a structured game (such as chess, checkers or even the seemingly simple rules of a tic-tac-toe (TTT) game), it is not clear how the background knowledge about the rules of these games can be used to produce better explanations. For instance, how would one incorporate "players take turns" and "X goes first" into the reasoning process? And what changes to explanations can one expect as the result of incorporating this background knowledge?

We can expect background knowledge to help to enrich our explanations and enable us to reveal insights that would not be immediately available from the model itself. For instance, in a TTT game, there may be several reasons for why a player wins the game, but not all explanations are well-known. The most common explanation is the three-in-a-row explanation - that one of the players managed to put three marks in a horizontal, vertical, or diagonal row. But in some situations, "isolated marks" can also be valid explanations - although not commonly known.

Consider a finished TTT game in which player X has won in 5 moves (depicted in Table 1.1.a). The fact that player O has placed "an isolated" mark on the board (Table 1.1.c Reason 2 ) is also an explanation for $X$ winning the game; i.e. a player can never be the winner in a game in which at most 6 marks are placed on the board and one of his marks is "isolated" (surrounded by empty cells); there is no way that the player can have three-in-a-row in this situation.

1 InTRODUCTION

| a. TTT Game |  |  |  |  |  |  | b. Reason 1 |  |  |  | c. Reason 2 |  |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :--- | :--- | :--- | :--- | :---: | :---: | :---: | :---: | :---: |
| $\mathbf{X}$ | $\mathbf{X}$ | $\mathbf{X}$ | $\mathbf{X}$ | $\mathbf{X}$ | $\mathbf{X}$ |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  | $\varnothing$ | $\varnothing$ |  |  |  |  |  |
| O |  | O |  |  |  |  |  | $\varnothing$ | $\mathbf{O}$ |  |  |  |  |  |

Table 1.1. a: A TTT board in which player X has won. b : The common three-in-a-row explanation for why X has won. c: Another possible explanation that does not mention any position of X. Marks in the highlighted cells (in grey) form an explanation: $\varnothing$ depicts an empty cell, while $\mathbf{X}$ and $\mathbf{O}$ show the position of the marks of players X and O that are part of the explanation.

To show that Reason 2 actually explains why X won, notice that the game could not have been a draw (because there are still empty cells on the board), so if the game has ended validly (which is our assumption), then someone has won the game. With 5 empty cells (unmarked) on the board of Reason 2, we know that player O can have either two or three moves on the board (since in a valid game, players take turns). No one wins with two moves, so we consider the case that O has put three marks on the board. From Reason 2, we know the position of one of O's moves, and we see that it is an isolated move. If one move out of three was isolated, then it is not possible to make a 3-in-a-row, therefore O could not have won. Since someone has won this game, then X must be the winner.

Providing more relevant explanations. Different stakeholders may have different perspectives and requirements when it comes to explaining ML decisions. For instance, if one is interested in player styles and strategy, then examining different types of reasons may yield interesting insights. Sometimes it is necessary to examine all valid reasons, for instance to assess the fairness and biases of an ML model (Darwiche and Hirth 2020). So the ability to generate all of the valid explanations for a given context is an important task.

Too many explanations however, pose another significant challenge for explainable AI; the existence of multiple competing explanations for a given ML decision or prediction is problematic both in terms of computing explanations and for returning them to the end users (Lakkaraju et al. 2019).

Studies show that (following the required verifications of a model as mentioned earlier) not all explanations may be needed by end users, as people's cognitive biases often lead them to
prefer certain types of explanations over others (Miller 2019). It is important to consider the relevance of an explanation to the end user and their existing knowledge about the problem domain (Miller 2019).

In line with considering background knowledge, another important aspect is to focus explanations on a "context" that is meaningful for the end user. This context can be seen as an acceptable explanation framework from which all explanations are derived. This approach aligns with social science research, where explanations are often generated based on a pre-selected "causal model" or "explanatory model" that provides a relevant context for understanding phenomena (Baumgartner and Falk 2018; Ragin 2014a; Duşa and Thiem 2015).

For illustration, consider the example of a professor who is shortlisting student candidates for a research project. She considers students who have taken at least one and at most two out of the following three courses: Logic (L), AI (A) and Statistics (S). The professor has an artificial agent assistant that sends her qualified applications, each accompanied with an explanation. As required by the professor, the only selection criterion used by the AI assistant is that candidates have taken at least one and no more than two courses.

There are a total of 6 possible explanations: $a .(\neg L \wedge S), b .(L \wedge \neg A), c .(A \wedge \neg S), d .(\neg L \wedge$ $A), e .(\neg A \wedge S), f .(L \wedge \neg S)$.

Table 1.2 shows a truth table for the combinations of courses ( $\mathrm{L}, \mathrm{A}, \mathrm{S}$ ) that a student could have taken, a column Y indicating whether a candidate matches the selection criteria, followed by the possible explanations for that row. On the right hand side, we see 5 possible explanatory models, each can explain all of rows of the truth table. Setting aside the details of how these models are derived, notice that the models differ in the combination of variables that are explicitly mentioned by them. For example, in explanatory model 1, variable L is not explicitly mentioned in all of the explanations. But model 4 mentions L in every explanation.

If the professor is happy with only one explanation per candidate, and requires that all of the explanations to explicitly mention whether the candidate took logic or not, then the AI assistant can return explanations from explanatory model 4.


Table 1.2. The truth table on the left shows rows with combinations of courses that candidates could have taken and the possible explanations for the row (explanations: a: $(\neg L \wedge S)$, b: $(L \wedge \neg A)$, c: $(A \wedge \neg S), \mathrm{d}:(\neg L \wedge A)$, $\mathrm{e}:(\neg A \wedge S), \mathrm{f}:(L \wedge \neg S))$. The list on the right hand side shows 5 possible explanatory models.

For instance, explanations for row $(0,1,0)$ are $(\neg L \wedge A)$ and $(A \wedge \neg S)$. Given explanatory model 1 , the explanation is $(A \wedge \neg S)$, and given explanatory model 4 , the explanation is $(\neg L \wedge A)$.

Choosing an explanatory model from which explanations are generated, can help to improve explanations that are returned to an end user by focusing them to a context that the end user finds acceptable and relevant.

### 1.1 Contributions

Against this backdrop, we now provide a summary of our contributions in this thesis in more technical terms.

First, we study the problem of incorporating background knowledge as domain constraints into the process of finding explanations. We propose a straightforward approach to effectively consider and incorporate arbitrary domain constraints.

First contribution: In summary the contributions of the first part of this thesis are:
(1) Focusing on the sufficient reasons notion of explanation (Darwiche and Hirth 2020), we provide a crisp formalisation of the notion of sufficient reasons for classifier decisions that take domain constraints into account.

We show that this results in sufficient reasons that are at least as (and sometimes more) parsimonious, i.e., more general and more succinct, than not taking constraints into account. The central insight, both simple and powerful, is to treat a classifier as a partial function by making it undefined on input instances that do not satisfy the constraint, and then to use the classic definition of prime implicant on partial functions (Coudert 1994) as the instantiation of "sufficient reason".

This immediately and naturally generalises the state-of-the-art from the unconstrained setting to the constrained setting.

We do this for Boolean classifiers in Chapter 3 and the more general multi-value classifiers in Chapter 4.
(2) We provide a simple reduction of the computational problem of finding all sufficient reasons of a classifier's decision for a given instance in the presence of constraints to the same problem in the unconstrained setting. This allows one to reuse existing algorithms and tools from the unconstrained setting.

The idea is that if the constraint is given by the formula $\kappa$ (which is a formula that constrains the instances we ask decisions about), and the decision-function is given by the formula $\varphi$, then sufficient reasons of decisions (made by decisionfunction represented by $\varphi$ ) that take $\kappa$ into account are exactly the sufficient reasons of the decision function represented by the formula $(\kappa \rightarrow \varphi)$. We prove that all other variations, including the natural variation $(\kappa \wedge \varphi)$ which expresses exactly the instances of interest (i.e., positive instances that satisfy the constraints), provide no more, and sometimes less, parsimonious sufficient reasons.
(3) We show, both theoretically and empirically on synthetic classifiers and classifiers learnt from data, that approaches that ignore constraints may supply sufficient reasons that are unnecessarily long since they redundantly encode knowledge already described in the constraints.

These contributions are based on our publication (Gorji and Rubin 2022).

Second contribution: We then address the problem of explanation relevancy and having too many competing explanations.

We narrow the possible explanations in two dimensions. First, we take inspiration from a formal computational model of causal reasoning that has been applied in diverse fields (such as medical diagnosis, social sciences, natural language processing and intention inference in robotics (Peng and Reggia 1990; Katz et al. 2016)) and use covers of the decision functions of classifiers as causal models with reference to which explanations are produced. Second, at the time of reasoning, we further narrow down explanations to those that satisfy a given partial world (which is a generalisation of an instance to a partial instance), and highlight the difference makers within the given partial world.

We formalise the definitions of actual causal inference within this framework and prove that causes are minimal difference makers of their effects. We then show how our definitions deal with some of the subtleties associated with causal inference.

Third contribution: Finally, we show that these theoretical works can be applied in practice by analysing the dataset of Corticosteroid Randomization after Significant Head Injury (CRASH) (Collaborators et al. 2008a) trials in Chapter 6. We show that the simplified examples presented in the previous paragraphs actually have real-life counterparts. For instance, focusing on a subset of the inputs allowed specifying the combinations of admission parameters that are likely to result in a given clinical outcome for different subsets of patients. This is appealing from a clinician's perspective and helps to design simplified assessment protocols in small medical centres without specialist capacity. Therefore, our results have translational value. Further, in Chapter 6, we also demonstrated how explanations can be accompanied by quantitative support metrics.

In summary, the thesis is organized as follows:

- Chapter 2 presents the related work.
- Chapters 3 and 4 present the first contribution of the thesis.
- In Chapter 3 we define the notion of sufficient reasons for classifier decisions in the presence of domain constraints.
- In Chapter 4 we extend our notions to finding sufficient reasons for multi-value classifiers.
- Chapter 5 focuses on providing explanations within a specific context.
- Chapter 6 reports the results of a real-life application of explanations within a context.
- Finally, we offer some closing remarks in Chapter 7.


## Chapter 2

## Related Work

In this chapter we review the closely related work on formal and heuristic approaches to providing explanations that are prevalent in $\mathrm{AI} / \mathrm{ML}$ and causality analysis domains. Before reviewing the related work, we will clarify a few competing notions of explanation that are used in the literature.

Explanations. The literature offers many definitions for an "explanation". Informally, an explanation is a reason for a belief or an action (Miller 2018) and has roots in causality. Halpern and Pearl (Joseph Y. Halpern and Pearl 2005b) call it a "fact that is not known for certain but, if found to be true, would constitute an actual cause" of something that needs to be explained. Operationally, it has been described as the "result of abductive inference conducted to identify the causes for a certain event" (Miller 2018; Joseph Y. Halpern and Pearl 2005b).

The literature also characteristically distinguishes explanations from various other (related) concepts such as inference, justification, scrutability and interpretability/explainability (Kay 2000; Kay and Kummerfeld 2006; Miller 2019). Inference is concerned with determining the truth of an explanation, or finding a suitable explanation among competing explanations (Harman 1968). Justification explains why a decision is good, but does not necessarily aim to give an explanation of the actual decision making process (Miller 2019). Scrutability can be described as the degree to which a system can be examined to understand what the system believes about the user, and how the system arrives at its beliefs about other (different) users. It is also used to describe the degree to which a system can be modified by the user (Kay and Kummerfeld 2006). Interpretability/explainability is the degree to which an observer can
understand the cause of a decision (Miller 2019; Linardatos, Papastefanopoulos and Kotsiantis 2020).

There are three main types of explanations or reasons for outputs of ML models. Namely:
(1) Sufficient reasons which are also known as prime implicant (PI)-explanations (Shih, Choi and Darwiche 2018) and abductive explanations (Ignatiev, Narodytska and Marques-Silva 2019a),
(2) Necessary reasons (Darwiche and Hirth 2022) which are also known as contrastive explanations (Ignatiev, Narodytska, Asher et al. 2020), and
(3) Counterfactual explanations (Mothilal, Sharma and Tan 2020).

Sufficient reasons provide answers to "plain-fact" questions which ask questions of the form: "Why does object $A$ have property $P$ ?" (Van Bouwel and Weber 2002). For explaining the decisions of classifiers, a sufficient reason is a minimal set of features and their values that is sufficient for the classifier to return a specific decision. This is the notion of explanation that we use in this thesis.

Contrastive explanations provide an answer to contrastive questions (Van Bouwel and Weber 2002). A contrastive question can be "alternative" or "congruent" or "time-contrastive" (Miller 2018). Paraphrasing Miller 2018: an alternative contrastive question asks: Why does object $a$ have property $P$, rather than property $Q$ ? A congruent contrastive question asks: Why does object $a$ have property $P$, while object $b$ has property $Q$ ? A time-contrastive question asks : Why does object $a$ have property $P$ at time $t$, but $Q$ at time $t^{\prime}$ ? For explanations of the individual decisions of a classifier, contrastive explanations (Marques-Silva and Ignatiev 2022) provide the minimal set of features that are needed to change (take different values from their domain) in order to change the computed decision (Marques-Silva and Ignatiev 2022).

Counterfactual explanations provide answers to counterfactual questions, which are defined based on the definition of counterfactual statements (Pearl 2009). Counterfactual questions ask: Would object $a$ having property $P$, have property $Q$ under condition $X$ ? For explanations
of the individual decisions of a classifier, a counterfactual explanation usually has one of the following forms:
(1) "The decision on <the given instance $>$ is $<$ decision $>$. Had a small subset of features been different $<$ foil $>$, the decision would have been $<$ different decision> instead" (Sokol and Flach 2019).
(2) "The decision on <the given instance> will remain the same, even if a small subset of features had been different $<$ foil $>$, because of $<$ necessary set of features>" (Darwiche and Hirth 2022).

Both of these counterfactual forms highlight what could have happened had the input to a given model been altered in a particular way.

To compute explanations for individual decisions of a classifier, various formal and heuristic methods of explanation are available.

Formal methods of explanation. Formal explanation methods have their origins in the logic synthesis methods of 1950s (Quine 1952a; Nelson 1955; McCluskey 1956a) that were used for circuit optimisation in digital systems (Brayton, Hachtel et al. 1984; Brayton and Somenzi 1989; Bryant 1992; Brayton, McGeer et al. 1993; McGeer et al. 1993), and were later adapted and leveraged in expert system diagnosis (Reiter 1987; De Kleer, Mackworth and Reiter 1992; Peng and Reggia 1990) and for analysing the causes of failures of large electrical systems (Coudert and Madre 1993; Coudert and Madre 1992; Coudert 1995).

The most notable formal explanation tools in AI are STEP (Shih, Choi and Darwiche 2018) which uses Knowledge Compilation (Darwiche and Marquis 2002; Darwiche 2011), and eXplainer (Ignatiev, Narodytska and Marques-Silva 2019a) and its family of tools which use SAT (Ignatiev, Pereira et al. 2018), SMT (Ignatiev, Narodytska and Marques-Silva 2019a) and ILP (Ignatiev, Narodytska and Marques-Silva 2019b) solvers to find explanations for the outputs of a classifier on individual input instances.

All of these methods are focused on providing (sufficient or necessary) reasons behind an ML model's output (prediction/decision) on individual input instances, but they can also be used to form counterfactual explanations (Darwiche and Hirth 2022).

Formal explanation methods compute provably minimal explanations, but also have the ability to be used in model enumeration and to answer other queries such as identifying irrelevant, forbidden or mandatory features for a given class (Audemard, Koriche and Marquis 2020). The main limitation of formal explanation methods is their scalability. Indeed, if one is interested in producing all possible explanations, these methods can hardly scale beyond a dozen variables for complex problems (Shih, Choi and Darwiche 2018; Ignatiev, Narodytska and Marques-Silva 2019a; Marques-Silva and Ignatiev 2022). However, in the last couple of years, numerous theoretical and practical improvements for efficient computation of explanations have been made, by focusing on specific classifiers, instead of remaining modelagnostic. For instance, (Huang et al. 2021) introduced "explanation graphs" which allows for finding explanations for a range of classifiers, including decision trees, graphs and diagrams in polynomial time. (Marques-Silva and Ignatiev 2022) provide a thorough summary of some other recent developments in formal explanations in AI.

Our first contribution (Chapters 3 and 4) is based on the formal explanation methods for ML models. In particular, we focus on improving sufficient reasons by including domain constraints into the reasoning process.

Apart from finding explanations for individual instances, formal explanation methods can be used to summarise the input-output behaviour of an ML model into a single (simplified) explanatory model, by using various covers. A cover of a function is a (minimal) set of terms that can collectively explain the function as a whole (Karp 1972). Covers are useful when one wants to reduce the number of possible explanations by focusing explanations on a subset of terms (i.e. those that appear in a cover). Covers are sometimes used as explanatory causal models in social science research (Baumgartner 2013; Baumgartner and Falk 2018; Graßhoff and May 2001).

The use of covers as causal models was popularised through the works of (Ragin 2014a; Baumgartner 2009; Duşa and Thiem 2015) on Configurational Comparative Analysis Methods (CCMs), the most well-known of which is called Qualitative Comparative Analysis (QCA). QCA uses covers as causal models for a set of observations that are presented by a Boolean function (Ragin 2014a; Ragin 2009a; Rihoux and Ragin 2008). Following the logical minimisation procedure of the input Boolean function, QCA measures each term in a cover with various statistical measures (Ragin 2009a). These measures are used by analysts to evaluate the importance of contributory factors to causing a phenomenon of interest.

Generating various covers from prime implicants (be they irredundant or of minimum cardinality) is a foundational research topic in computer science and engineering. The first application of covers was for obtaining smaller digital circuits while optimising for cost and improving the computing speed. Since then, exact and approximate covering methods have found several other application domains such as fault tree analysis (Coudert and Madre 1993), bioinformatics (Acuña et al. 2012), model-based diagnosis (De Kleer, Mackworth and Reiter 1990; Peng and Reggia 1990), test generation (Ghosh, Devadas and Newton 1991), data compression (Amarú et al. 2014) and automated and non-monotonic reasoning (Brewka, Niemelä and Truszczyński 2008).

In Chapter 5 and 6 we focus on finding explanations from within a particular cover. We also move beyond explanations for individual input instances. By generalising the notion of instance we allow for finding explanations for partial instances which was not (so far) discussed in the related literature (i.e. (Shih, Choi and Darwiche 2018; Ignatiev, Narodytska and Marques-Silva 2019a)).

Formal explanation methods can be contrasted with heuristic approaches that are widely used to explain larger ML models. We now review some of the most well-known heuristic approaches.

Heuristic methods of explanation. The ML literature provides many techniques for producing post-hoc explanations for the outputs of complex models. For example, Saliency Map (Simonyan, Vedaldi and Zisserman 2013) is one of the most widely used methods to visualise
feature importance and contribution to the output of a deep neural network. Saliency Maps are not explanations per se, but they may help construct explanations by highlighting the "important" parts of an image. To explain neural networks, backpropagation-based methods are commonly used to find important features as explanations of an instance. These methods include gradients-based methods (Simonyan, Vedaldi and Zisserman 2013; Sundararajan, Taly and Yan 2017), DeepLIFT (Shrikumar, Greenside and Kundaje 2017), and influence functions (Koh and Liang 2017). All of these methods are heavily susceptible to adversarial attacks; (Ghorbani, Abid and Zou 2019) highlight issues with explanation robustness and sensitivity of these methods to small changes to input data. They show how explanations with these methods can be manipulated (specially for the cases where the importance of small subsets of features are being evaluated).

Anchors (Ribeiro, Singh and Guestrin 2018), LIME (Ribeiro, Singh and Guestrin 2016) and SHAP (Lundberg and Lee 2017) are the other most widely used post-hoc explanation methods. LIME and SHAP are local approximation (perturbation)-based approaches (Agarwal et al. 2021). Using these tools, the outputs of an arbitrary ML model for each individual instance (as well as the model as a whole) can be explained by "perturbing the inputs in the neighbourhood of a given instance to observe effects of perturbations on the model's output" (Agarwal et al. 2021). Several challenges can be associated with perturbation-based methods. (Slack et al. 2020) report how LIME and SHAP can be "fooled" to not flag sensitives features (such as gender or race) as important features of classifiers that heavily depend on those features to make decisions.

The main reason behind the shortcomings of these heuristic methods is that these methods do not take the whole feature space, or the complete decision function of a black-box into account for producing explanations, and therefore heuristic methods may produce differing results depending on what subset of feature space was included as input to their explanation algorithms.

Another drawback is that they do not take background knowledge into account. For instance, (Izza and Marques-Silva 2022) show that the correctness of Anchors, SHAP and LIME may improve significantly when background knowledge is available. They showed that the
correctness of these methods (when compared with the results of formal explanation methods) is higher when domain constraints are applied. However, (Izza and Marques-Silva 2022) also report that these heuristic approaches were not able to achieve $100 \%$ correctness for the majority of the datasets that they have examined, even when domain constraints were used.

Explanation relevancy and the possibility of having too many competing explanations is another problem (Lakkaraju et al. 2019). The work of (Deutch and Frost 2018) and (Shrotri et al. 2022) show that end users can make explanations more relevant by repeatedly focusing on their desired area of the input space.

Despite these challenges, heuristic methods are widely used because they are more scalable than their rival formal-explanation methods; thus, the current trade-off is for guarantees on the quality and validity of produced explanations in exchange for tractability.

## ChAPTER 3

## Sufficient Reasons in the Presence of Domain Constraints

Remark: A major part of this chapter was published as "Sufficient Reasons for Classifier Decisions in the Presence of Domain Constraints" (Gorji and Rubin 2022) and presented at the AAAI'22 conference (Gorji and Rubin 2022) and is an almost verbatim copy of the published work. Some sections and examples have been edited to fit the flow and style of the thesis.

In this chapter we study the problem of providing explanations for classifier decisions in the presence of domain constraints. We provide a principled answer, as well as theoretical and empirical evidence suggesting that ignoring constraints can result in explanations that are unnecessarily long. In particular, ignoring constraints may result in sufficient reasons that redundantly encode the background knowledge that is already described in the constraints.

The workflow is illustrated in Figure 3.1. A classifier, typically learnt from data, is transformed into a decision function $F$. Domain constraints $C$ are taken into account to produce a partial


Figure 3.1. Workflow - finding sufficient reasons in the presence of constraints. The focus of this chapter is on the right of the dashed line.

Boolean function $F_{C}$, which is used to compute sufficient reasons for the classifier's decisions on a particular instance $\mathbf{x}$.

### 3.1 Preliminaries

We begin by recalling the logical background needed to explain our theory in this chapter.

Boolean logic. Let $\mathbf{X}=\left\{X_{1}, X_{2}, \cdots, X_{n}\right\}$ be a set of $n$ Boolean variables (aka features). The set of Boolean formulas is generated from $\mathbf{X}$, the constants $\top$ (true) and $\perp$ (false), and the logical operations $\wedge$ (conjunction), $\vee$ (disjunction),$\neg$ (negation), $\rightarrow$ (conditional) and $\leftrightarrow$ (bi-conditional). Variables $X$ and their negations $\neg X$ are called literals. A term $t$ is a conjunction of literals; the empty-conjunction is also denoted T. The size of a term $t$ is the number of literals that occur in it. An instance (over $\boldsymbol{X}$ ) is an element of $\{0,1\}^{n}$, and is denoted $\mathbf{x}$ (intuitively, it is an instantiation of the variables $\mathbf{X}$ ). An instance $\mathbf{x}$ satisfies a formula $\varphi$ if $\varphi$ evaluates to true when the variables in $\varphi$ are assigned truth-values according to $\mathbf{x}$. The set of instances that satisfy the formula $\varphi$ is denoted $[\varphi]$, and is called the set represented by $\varphi$, i.e., a set $C$ of instances is represented by $\varphi$ if $C=[\varphi]$. If $[\varphi]=[\psi]$ then we say that $\varphi, \psi$ are logically equivalent, i.e., they mean the same thing. For terms $s, t$, we say that $s$ subsumes $t$ if $[t] \subseteq[s]$, i.e., if every instance that satisfies $t$ also satisfies $s$. If $[t] \subset[s]$ then we say that $s$ properly subsumes $t$; depending on the context, we also describe this by saying that $s$ is more general or more parsimonious than $t$, or $s$ is more succinct than $t$ (note that $s$ is smaller than $t$ ).

Partial Boolean functions, and prime implicants. A partial Boolean function F (over X) is a function $\{0,1\}^{n} \rightarrow\{0,1, *\}$. For $i \in\{0,1, *\}$ define $F^{i}$ to be the set $F^{-1}(i)$. The instances in $F^{1}, F^{0}, F^{*}$ are called, respectively, the positive, negative, undefined instances of $F$. If the set $F^{*}$ is empty, then $F$ is a total Boolean function. If $[\varphi]=F^{1}$ we say that the formula $\varphi$ represents the total Boolean function $F$. A term $t$ is an implicant of $F$ if $[t] \subseteq F^{1} \cup F^{*}$; it is prime if no other implicant of $F$ subsumes $t$. Intuitively, $t$ is prime if removing any literal from $t$ results in a term that is no longer an implicant. This generalises the notion of
implicant and prime implicant from total Boolean functions, cf. (Quine 1952a; Shih, Choi and Darwiche 2018; Darwiche and Hirth 2020), to partial Boolean functions, cf. (McCluskey 1956a; Coudert 1994).

We state a simple but useful lemma:

Lemma 1. If $F, G$ are partial functions over $\boldsymbol{X}$ such that $F^{1} \cup F^{*} \subseteq G^{1} \cup G^{*}$, then every prime implicant of $F$ is subsumed by some prime implicant of $G$.

Proof. Clearly, every implicant of $F$ is an implicant of $G$. Now, apply the fact that every implicant of a function is subsumed by some prime implicant of that function.

Decision-functions. Total Boolean functions naturally arise as the decision-functions of threshold-based binary classifiers (Choi, Shi et al. 2019; Shih, Choi and Darwiche 2018): the decision-function $F$ of a threshold-based classifier is the function that maps an instance $\mathbf{x}$ to 1 if $\operatorname{Pr}(d=1 \mid \mathbf{x}) \geqslant T$, and to 0 otherwise; here $d$ is a binary class variable, and $\operatorname{Pr}$ is the distribution specified by the classifier, and $T$ is a user-defined classification threshold.

Table 3.1. demonstrates the truth table representing a hypothetical classifier's decision function where $T=0.5$

| $X_{1}$ | $X_{2}$ | $X_{3}$ | $\operatorname{Pr}(d=1 \mid \mathbf{x})$ | $\operatorname{Pr}(d=1 \mid \mathbf{x}) \geqslant 0.5$ | d |
| :---: | :---: | :---: | :---: | :---: | :---: |
| 0 | 0 | 0 | 0.1 | 0 | 0 |
| 0 | 0 | 1 | 0.3 | 0 | 0 |
| 0 | 1 | 0 | 0.3 | 0 | 0 |
| 0 | 1 | 1 | 0.6 | 1 | 1 |
| 1 | 0 | 0 | 0.3 | 0 | 0 |
| 1 | 0 | 1 | 0.6 | 1 | 1 |
| 1 | 1 | 0 | 0.6 | 1 | 1 |
| 1 | 1 | 1 | 0.9 | 1 | 1 |

Table 3.1. Truth table representing a classifier's decision function

Definition 1 (Sufficient reasons for total functions). (Darwiche and Hirth 2020) Let F be a total Boolean function and let $\boldsymbol{x}$ be a positive instance of $F$. A term $t$ is a sufficient reason of the decision $F(\mathbf{x})=1$ if $(i) t$ is a prime implicant of $F$, and (ii) $t$ is satisfied by $\boldsymbol{x}$.

Sufficient reasons are called PI-explanation in (Shih, Choi and Darwiche 2018), and abductive explanations in (Ignatiev, Narodytska and Marques-Silva 2019a).

Standard convention. We freely interchange between total Boolean functions and the formulas that represent them. In particular, if $\varphi$ represents the total Boolean function $F$, we may refer to implicants, prime implicants, and sufficient reasons of $\varphi$ (instead of $F$ ).

### 3.2 Problem Setting

The problem we address is how to define reasons behind the decisions of a classifier in the presence of domain constraints. As we will show below, by domain constraints we mean the relationships that we assume holds between variables (features) present in data.

Definition 2. A constraint is a set $C$ of instances over $\boldsymbol{X}$.

We typically represent constraints by Boolean formulas. Here are just a few examples that show that constraints are ubiquitous. In a medical setting, constraints of the form $\left(X_{1} \rightarrow X_{2}\right)$ may reflect that people with condition $X_{1}$ also have condition $X_{2}$, e.g., $X_{1}$ may mean "is pregnant" and $X_{2}$ may mean "is a woman". In a university degree structure: constraints of the form $X_{1} \rightarrow\left(X_{2} \wedge X_{3}\right)$ may reflect that $X_{2}$ and $X_{3}$ are prerequisites to $X_{1}$; constraints of the form $X_{1} \rightarrow \neg\left(X_{2} \vee X_{3}\right)$ may reflect prohibitions; and constraints of the form $X_{1} \wedge X_{2}$ may reflect compulsory courses. In configuration problems, e.g., that arise when users purchase products, the user may be required to configure their product subject to certain constraints, and constraints of the form $\left(X_{1} \vee X_{2}\right) \wedge \neg\left(X_{1} \wedge X_{2}\right)$ may reflect that the user needs to choose between two basic models. These constraints also result from one-hot encodings of categorical variables, e.g., if $M$ is a 12 -valued variable representing months, and $X_{i}$ for $i=1, \cdots, 12$ is Boolean variable, then the induced constraint is $\left(\bigvee_{i} X_{i}\right) \wedge\left(\bigwedge_{i \neq j} \neg\left(X_{i} \wedge X_{j}\right)\right)$. Combinatorial objects have natural constraints, e.g., rankings are ordered sets, trees are acyclic graphs, and games have rules, see the Case Studies and Validation section. Finally, the assumption in this chapter is that constraints are hard, i.e.,
instances that are not in $C$ will not appear in any data and can be ignored (e.g., they will not appear in training or testing data).

Recall that we denote decision functions over X by $F$. Given $F$ and a constraint $C$, we ask:

How should one define reasons for decision-functions in the presence of constraints?

We posit that a suitable notion of "reason" that takes constraints into account:

D1. does not depend on the representations of $F$ or $C$, i.e., it is a semantic notion;
D2. does not depend on the values $F(\mathbf{x})$ for $\mathbf{x} \notin C$, i.e., if $F, G$ agree on $C$ (and perhaps disagree on the complement of $C$ ), then reasons for $F$ given constraint $C$ should be the same as reasons for $G$ given constraint $C$;

D3. in case there are no constraints, i.e., $C=\{0,1\}^{n}$, recovers the notion of sufficient reasons from Definition 1;

D4. eliminates redundancies that are captured by the constraints.

We offer a formalisation that satisfies these desiderata.

### 3.3 Sufficient Reasons in the Presence of Constraints

In this section we provide the main definition of reasons in the presence of constraints (Definition 4) and show that it satisfies all of the desired properties D1-D4 listed in the Problem Setting section.

D1 and D2 motivate the insight that decision-functions in the presence of constraints should be treated as partial Boolean functions:

Definition 3. For a decision-function $F$ and a constraint $C$, let $F_{C}$ be the partial Boolean function that maps $\boldsymbol{x}$ to $F(\boldsymbol{x})$ if $\boldsymbol{x} \in C$, and to $*$ otherwise.

Technically, * means undefined. We sometimes call $F_{C}$ a constrained decision function.

We now define sufficient reasons that take constraints into account by considering the partial function $F_{C}$.

DEFINITION 4 (Sufficient reasons that take constraints into account). Let $F$ be a decision function, and $C$ a constraint. Let $\boldsymbol{x}$ be a positive instance of $F$ such that $\boldsymbol{x} \in C$. A term $t$ is a sufficient reason of the decision $F(\mathbf{x})=1$ that takes the constraint $C$ into account if (i) $t$ is $a$ prime implicant of the partial Boolean function $F_{C}$, and (ii) $t$ is satisfied by $\boldsymbol{x}$.

In this case, we will also say that $t$ is a sufficient reason of the decision $F_{C}(\mathbf{x})=1$. We will also call prime implicants of $F_{C}$, sufficient reasons using $F_{C}$.

Notice that it does not matter what $F$ is for $F$ not in $C$.

To see that D3 holds, simply note that if $C=\{0,1\}^{n}$ then $F_{C}=F$ is a total function. Thus, a term $t$ is a sufficient reason of the decision $F(\mathbf{x})=1$ that takes $C$ into account iff it is a sufficient reason of the decision $F(\mathbf{x})=1$ according to Definition 1 .

Remark 1. Sufficient reasons of negative instances $\boldsymbol{x}$ can be defined and handled dually: a term $t$ is a sufficient reason of the decision $F_{C}(\mathbf{x})=0$ if it is a sufficient reason of the decision $\bar{F}_{C}(\boldsymbol{x})=1$ where $\bar{F}$ is the "negation of $F$ ", i.e., $\bar{F}(\boldsymbol{x}):=0$ if $F(\boldsymbol{x})=1$, and $\bar{F}(\boldsymbol{x}):=1$ if $F(\boldsymbol{x})=0$. Thus, we can reduce reasoning about negative instances of $F_{C}$ to reasoning about positive instances of $\bar{F}_{C}$.

Example 1. Consider the total Boolean function Fover $\boldsymbol{X}=\left\{X_{1}, X_{2}\right\}$ represented by the formula $\left(X_{1} \leftrightarrow X_{2}\right)$. Suppose a constraint $C$ is represented by the formula $\left(X_{1} \rightarrow X_{2}\right)$, thus $C=\{(0,0),(0,1),(1,1)\}$. Table 3.2 provides both $F$ and the partial Boolean function $F_{C}$. The prime implicants of $F$ are $\left(X_{1} \wedge X_{2}\right)$ and $\left(\neg X_{1} \wedge \neg X_{2}\right)$. The only sufficient reason of the decision $F(0,0)=1$ is the term $\left(\neg X_{1} \wedge \neg X_{2}\right)$, and the only sufficient reason of the decision $F(1,1)=1$ is the term $\left(X_{1} \wedge X_{2}\right)$. The prime implicants of $F_{C}$ are $\neg X_{2}$ and $X_{1}$. The only sufficient reason of the decision $F_{C}(0,0)=1$ is $\neg X_{2}$, and the only sufficient reason of the decision $F_{C}(1,1)=1$ is $X_{1}$.

| $X_{1}$ | $X_{2}$ | $F$ | $F_{C}$ |
| :---: | :---: | :---: | :---: |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | $*$ |
| 1 | 1 | 1 | 1 |

Table 3.2. The row corresponding to the instance not in the constraint is greyed out.

Finally, we provide a simple theorem that formalises D4. We prove that every sufficient reason that does not take constraints into account is subsumed by some sufficient reason that does.

TheOrem 1. Suppose $\boldsymbol{x}$ is a positive instance of $F_{C}$. Then every sufficient reason of the decision $F(\boldsymbol{x})=1$ is subsumed by some sufficient reason of the decision $F_{C}(\boldsymbol{x})=1$.

Proof. Let $\mathbf{x}$ be a positive instance of $F_{C}$. In particular, it is a positive instance of $F$. Let $t$ be a prime implicant of $F$ that is satisfied by $\mathbf{x}$. We show that there is some prime implicant $t^{\prime}$ of $F_{C}$ that subsumes $t$ (and thus is satisfied by $\mathbf{x}$ ). To see this, apply Lemma 1 taking $G=F_{C}$. The hypothesis of the Lemma holds (i.e., that $F^{1} \cup F^{*} \subseteq G^{1} \cup G^{*}$ ) since $F^{1} \cup F^{*}=F^{1}$ (since $F$ is total) and $\left(F_{C}\right)^{1} \cup\left(F_{C}\right)^{*}=\left(F^{1} \cap C\right) \cup \bar{C}$ (by definition of $F_{C}$ ).

To complement this theorem, we show that simply considering reasons of the total Boolean function $F$ (and ignoring the constraint $C$ ), may actually supply strictly less succinct reasons.

Example 2. Continuing Example 1, note that the only sufficient reason for $F(0,0)=1$ is subsumed by a sufficient reason of $F_{C}(0,0)=1$, i.e., $\left(\neg X_{1} \wedge \neg X_{2}\right)$ is subsumed by $\neg X_{2}$. Similarly, the only sufficient reason for $F(1,1)=1$ is subsumed by a sufficient reason of $F_{C}(1,1)=1$, i.e., $\left(X_{1} \wedge X_{2}\right)$ is subsumed by $X_{1}$. This accords with the intuition that, in light of the constraint $\left(X_{1} \rightarrow X_{2}\right)$, reason $X_{1}$ is preferred to reason $\left(X_{1} \wedge X_{2}\right)$.

Note: Example 1 is reused and made more intuitive in Example 10 of Chapter 5. The reader can refer to that example for more details.

It is not hard to find examples where every sufficient reason of $F(\mathbf{x})=1$ is much larger than every sufficient reason of $F_{C}(\mathbf{x})=1$. E.g., let $F$ be the function $X_{1} \wedge X_{2} \wedge \cdots \wedge X_{n}$, and $C$ be the constraint $X_{1} \rightarrow\left(X_{2} \wedge X_{3} \wedge \cdots \wedge X_{n}\right)$; then the only reason of the decision $F(1,1, \cdots, 1)=1$ is $X_{1} \wedge X_{2} \wedge \cdots \wedge X_{n}$, which is subsumed by the reason $X_{1}$ of the decision $F_{C}(1,1, \cdots, 1)=1$.

Constraint-equivalent reasons. If one is interested in the meaning of a reason, and not its syntactic structure, then one should consider sufficient reasons up to logical-equivalence modulo the constraints. That is, terms $t, s$ are $C$-equivalent (or simply, constraint-equivalent when the constraint is understood), if $C \cap[s]=C \cap[t]$. For instance, if $C$ is represented by $\left(X_{1} \vee X_{2}\right) \wedge \neg\left(X_{1} \wedge X_{2}\right)$ then $t=\neg X_{1}$ is $C$-equivalent to $s=X_{2}$, and thus $s$ and $t$ may be identified as the same reason in the presence of $C$.

### 3.3.1 Variations and Parsimony of Reasons

Subtle changes in the definition of sufficient reasons result in radically different types of reasons. First, we have seen in the Examples that ignoring the constraints does not provide the most parsimonious reasons. Second, consider the variation in which, instead of using reasons of the partial function $F_{C}$, one uses reasons of the total function that agrees with $F$ on $C$ and assigns 0 to instances not in $C$. Although seemingly natural, it is not hard to see using Lemma 1, that this results in less parsimonious reasons. Moreover, if $F, C$ are represented by the Boolean formulas $\varphi$ and $\kappa$ respectively, then this total function is represented by the formula $\kappa \wedge \varphi$. In the next section we will see that sufficient reasons using $F_{C}$ are the same as using the total function corresponding to the formula $\kappa \rightarrow \varphi$. We find it striking that this change of perspective drastically changes the parsimony of the produced reasons; we provide an example of this difference in the discussion of Case Study 1.

### 3.4 Computing Sufficient Reasons

In this section we discuss how to computationally find sufficient reasons in the presence of constraints. In particular, we show how to reduce this to the unconstrained case.

Definition 5 (Computational problems). Given a decision-function F, a constraint $C$ and a positive instance $\boldsymbol{x}$ of $F_{C}$, find all (resp. one) sufficient reasons for the decision $F_{C}(\boldsymbol{x})=1$.

As usual (see the Preliminaries, Section 3.1), we can think of the total function $F$ and the set of instances $C$ as Boolean formulas, say $F^{1}=[\varphi]$ and $C=[\kappa]$ (we are agnostic about exactly how to represent these formulas until we discuss complexity and the experiments). The following proposition says that we can reduce the computational problem of the constrained case to the unconstrained case using the formula $(\kappa \rightarrow \varphi)$.

Proposition 1. Suppose $\varphi$ represents $F$ and $\kappa$ represents $C$. For a positive instance $\boldsymbol{x}$ of $F_{C}$, the sufficient reasons of the decision $F_{C}(\boldsymbol{x})=1$ are exactly the sufficient reasons of the decision $G(\boldsymbol{x})=1$ where $G$ is the total function represented by the Boolean formula $(\kappa \rightarrow \varphi)$.

Proof. First, note that $\mathbf{x}$ is a positive instance of $G$. Indeed, since $F_{C}(\mathbf{x})=1$ we know that $\mathbf{x} \in C \cap F^{1}$, i.e., $\mathbf{x} \models \kappa \wedge \varphi$, and thus also $\mathbf{x} \models \kappa \rightarrow \varphi$. Thus, it is sufficient to show that a term $t$ is an implicant of $F_{C}$ iff it is an implicant of $G$. By definition, $t$ is an implicant of $F_{C}$ iff $[t] \subseteq\left(F_{C}\right)^{1} \cup\left(F_{C}\right)^{*}$. But $\left(F_{C}\right)^{1}=F^{1} \cap C$ and $\left(F_{C}\right)^{*}=\bar{C}$ (Definition 3). On the other hand, $t$ is an implicant of the total function $G$ iff $[t] \subseteq G^{1}$. But $G^{1}=\bar{C} \cup F^{1}$. Thus $G^{1}=\left(F_{C}\right)^{*} \cup\left(F_{C}\right)^{1}$.

The significance of Proposition 1 is that it shows how to reuse algorithms and tools that are already developed for reasoning about total Boolean functions. Indeed, as long as the formulas $\kappa, \varphi$ are represented in a language that allows one to form the conditional $\kappa \rightarrow \varphi$ formula in polynomial time in the sizes of $\kappa, \varphi$, we have a polynomial time reduction of the problem of finding reasons with constraints to those without. On the other hand, reasoning without constraints is a special case of reasoning with constraints, i.e., there is a trivial reduction in
the other direction too, simply take $\kappa=$ true. We summarise this important computational fact as follows:

THEOREM 2. Assume that formulas are represented in a formalism that allows one to form the conditional of two formulas in polynomial time. Then, the problem of finding all (resp. one) sufficient reasons for a decision that takes constraints into account is polynomial time interreducible with the problem of finding all (resp. one) sufficient reasons for a decision (without constraints).

Thus, if one uses representations that also allow one to compute sufficient reasons of total Boolean functions in polynomial time, then, by first applying the reduction in Theorem 2 one can find sufficient reasons for constrained decision-functions in polynomial time too.

We mention the two main approaches comprising the state of the art for computing sufficient reasons for total Boolean functions. First, (Shih, Choi and Darwiche 2018) represent formulas using OBDDs, which support polynomial negation and conjunction (and thus implication). Their approach provides a polynomial time procedure for finding all sufficient reasons, using the fact that OBDDs support polynomial-time validity and entailment checking. To reuse their algorithm in our setting, simply run it on the OBDD representation of the formula $(\kappa \rightarrow \varphi)$. Second, (Ignatiev, Narodytska and Marques-Silva 2019a) take an agnostic view on the representation of formulas, and only require that the chosen representation allows polynomial time entailment checking. To reuse their approach in the presence of constraints, one may use it on formulas of the form $(\kappa \rightarrow \varphi)$ instead of $\varphi$.

Note that if a representation also allows (a) polynomial time validity checking, and (b) forming the conjunction of a term and formula in polynomial time, then one can decide if two terms are constraint-equivalent in polynomial time. Thus, if one is interested in computing reasons up to constraint-equivalence one can compute a set of representatives by, for instance, checking each pair of reasons for constraint-equivalence.

### 3.4.1 Illustration

To clarify the introduced concepts, we illustrate sufficient reasons on a complete synthetic example of a learnt classifier, inspired by an example in (Kisa et al. 2014).

Consider a tech-company that is shortlisting recent CS graduates for a job interview. The company considers candidates who took courses on Probability (P), Logic (L), Artificial Intelligence (A) or Knowledge Representation (K) during their studies. Suppose that the company uses data on candidates who were hired in the past to learn a threshold-based classifier, and let $F$ be the associated total decision-function over $\mathbf{X}=\{L, K, P, A\}$ with $F^{1}=\{(0011),(0110),(0111),(1100),(1101),(1110),(1111)\}$.

Consider an instance $\mathbf{x}=(0011)$ corresponding to candidates that did not take L or K , but did take P and A . Note that $F(\mathbf{x})=1$, i.e., the classifier decides to grant such candidates interviews. What is the reason behind this decision? Table 3.3 gives the reasons which were computed using (Shih, Choi and Darwiche 2018). We see that the only reason behind the decision of $F$ for $\mathbf{x}=(0011)$ is $(\neg L \wedge P \wedge A)$, i.e., that the candidate did not take L , but did take P and A .

| L | K P | A | Reasons using $F$ | using $F_{C}$ |  |
| :--- | :--- | :--- | :--- | :--- | :--- |
| 0 | 0 | 1 | 1 | $(\neg L \wedge P \wedge A)$ | $(\neg L \wedge A)$ |
| 0 | 1 | 1 | 1 | $(\neg L \wedge P \wedge A),(K \wedge P)$ | $(\neg L \wedge A), K$ |
| 1 | 1 | 0 | 0 | $(L \wedge K)$ | $K$ |
| 1 | 1 | 1 | 0 | $(L \wedge K),(K \wedge P)$ | $K$ |
| 1 | 1 | 1 | 1 | $(L \wedge K),(K \wedge P)$ | $K$ |

Table 3.3. Rows list the positive instances that satisfy the constraints, along with their reasons using $F$ and using $F_{C}$.

Suppose, that a student's enrolments must satisfy the following constraints $C$ : a student must take P or $\mathrm{L},(P \vee L)$; the prerequisite for A is $\mathrm{P},(A \rightarrow P)$; the prerequisite for K is A or L , $(K \rightarrow(A \vee L))$. Reasons of the constrained decision-function $F_{C}$ are given in Table 3.3. Note $(\neg L \wedge A)$ and $K$ are not constraint-equivalent.

Consider the reason behind the decision $F_{C}(\mathbf{x})=1$ where $\mathbf{x}=(0011)$, i.e., $\neg L \wedge A$. This reason strictly subsumes the reason $\neg L \wedge P \wedge A$ used by the original (unconstrained)
classifier $F$. This phenomenon, that for every positive instance $\mathbf{x}$ in $C$, every sufficient reason of $F(\mathbf{x})=1$ is subsumed by some sufficient reason of $F_{C}(\mathbf{x})=1$, can be seen in all other rows of Table 3.3. This illustrates that our notion of sufficient reason (Definition 4) eliminates such redundancies, a fact we formalised in Theorem 1.

### 3.5 Case Studies and Validation

In this section we validate our theory on classifiers learnt from binary data given constraints $C .{ }^{1}$ We provide a prototype using a type of classifier that is often considered interpretable, i.e., decision trees. The purpose of the prototype is to provide a proof of concept that shows that by using constrained decision-functions $F_{C}$ : (1) we get no less succinct, and sometimes more succinct, reasons compared with the unconstrained setting; (2) we can seamlessly integrate two major types of constraints that can arise in AI, namely constraints due to pre-processing of data (e.g. one-hot, or other categorical, encodings), and semantic constraints that are inherent to the input domain.

Representation. As discussed earlier, we can compute reasons by reducing to the unconstrained case. We reuse the algorithms in (Shih, Choi and Darwiche 2018) by simply building an OBDD representing $\kappa \rightarrow \varphi$ (using the OBDD operations for complementation and disjunction), and pass this OBDD as input to their tool that computes sufficient reasons for a given instance.

Case Study 1. We used the dataset of Corticosteroid Randomization after Significant Head Injury (CRASH) trial (Collaborators et al. 2008b) to predict the condition of a patient after a traumatic head injury. There are eleven clinically relevant input variables, including demographics, injury characteristics and image findings, see (Zador, Sperrin and King 2016a) for a detailed description of the dataset. Six variables are categorical, and the rest are Boolean. ${ }^{2}$

[^0]The outcome variable indicates moderate or full recovery at 6 months versus death or severe disability.

Categorical variables are encoded using a one-hot encoding, which induces the constraint $C$ as follows. For a categorical variable $X$, let $D$ denote a set of Boolean variables corresponding to the set of categories of $X$. The corresponding constraint says that exactly one of the variables in $D$ must be true. For example, variable Eye (shortened to $E$ ) has 4 categories, which we encode by the Boolean variables in $D_{E}=\left\{E_{1}, E_{2}, E_{3}, E_{4}\right\}$. The corresponding constraint is $\bigvee_{i} E_{i} \wedge \bigwedge_{i \neq j} \neg\left(E_{i} \wedge E_{j}\right)$, where $i, j$ vary over $\{1,2,3,4\}$. The constraint $C$ is the conjunction of all such constraints, one for each categorical variable.

Following (Steyerberg et al. 2008a) we base our example on 6945 cases with no missing values. RPART (seed: 25 , train: 0.75 , cp: 0.005 ) correctly classifies $75.69 \%$ of instances in the test set (ROC 0.77). Figure 3.2 shows the model.


Figure 3.2. RPART decision tree for Case Study 1.

Consider the instance $\mathbf{x}$ that maps $A_{1}, E_{1}, M_{5}, V_{2}, P_{1}, O B, M D$ to 1 , and the remaining four variables to 0 . The decision-rule in the decision tree that explains why $\mathbf{x}$ is positive is $E_{1} \wedge P_{1} \wedge \neg A_{7} \wedge M_{5} \wedge \neg A_{6}$ (size: 5). There is one sufficient reason using $F: \neg A_{6} \wedge \neg A_{7} \wedge$ $M_{5} \wedge P_{1}$ (size: 4). Up to constraint-equivalence there are two sufficient reasons using $F_{C}$ : (i) $A_{1} \wedge M_{5} \wedge P_{1}$ (size: 3), (ii) $\neg A_{6} \wedge \neg A_{7} \wedge M_{5} \wedge P_{1}$ (size: 4).

Discussion of Case-Study 1. The explanation using the decision tree is strictly subsumed by the sufficient reason using $F$. This shows that decision-rules may not be the most succinct reasons. Further, incorporating constraints resulted in having a smaller reason which would be missed if one just used $F$. The reason using $F$ is subsumed by some reason using $F_{C}$, in fact it appears as reason (ii); cf. Theorem 1.

Note that reasons (i) and (ii) are not constraint-equivalent (and thus should be considered different reasons). Which reason should one prefer? On the one hand, (i) is more succinct, but on the other hand (ii) strictly constraint-subsumes (i), i.e., it applies to more instances. Without additional preferences there is no basis to prefer one over the other, and thus we report both of them. ${ }^{3}$

If one incorporated constraints by instead using the function represented by the formula $(\kappa \wedge \varphi)$ one would get one sufficient reason for this decision that is highly redundant in light of the constraint (as discussed in the Variations section), i.e., $\left(A_{1} \wedge E_{1} \wedge M_{5} \wedge V_{2} \wedge P_{1} \wedge \wedge_{X} \neg X\right)$ where the conjunction is over all the remaining variables $A_{2}, A_{3}, \cdots, E_{2}, E_{3}, \cdots$.

Finally, the histogram in Figure 3.3 compares the sizes of shortest reasons using $F$ and $F_{C}$ (omitting size 2 reasons which would dominate the graph). Note that the percentage of reasons using $F$ increases with size, while those using $F_{C}$ decreases with size.


Figure 3.3. Distribution of shortest reasons, restricted to instances without length $\leqslant 2$ reasons (i.e., 5440 of 109120 instances). Percentages are rounded to the nearest decimal.

In summary, this case study empirically validates that reasons that take constraints into account may be more succinct.

Case Study 2. To study semantic constraints, we used the Tic-Tac-Toe (TTT) Endgame dataset from the UCI machine learning repository (Dua and Graff 2017). This dataset contains the complete set of board configurations that result from X going first, until the game ends. The target concept is "player X has three-in-a-row".

[^1]i. | 0 | 1 | 2 |
| :---: | :---: | :---: |
| 3 | 4 | 5 |
| 6 | 7 | 8 |

ii. | X | X | X |
| :---: | :---: | :---: |
| O |  | 0 |

iii. | 01 | 01 | 01 |
| :--- | :--- | :--- |
| 00 | 00 | 00 |
| 10 | 00 | 10 |

Table 3.4. i. TTT board; ii. Positive instance; iii. Encoded instance (cell $i$ is labelled by the values of $\left.V_{i, O} V_{i, X}\right)$.

We binarize the dataset as in (Verwer and Zhang 2019). For each of the 9 board positions (labelled as in Table 3.4i.) introduce variables $V_{i, O}$ (resp. $V_{i, X}$ ) capturing whether or not O (resp. X) was placed in position $i$. We trained a classifier on this dataset using RPART (seed 1, train: 0.7 , cp 0.01); with $93 \%$ accuracy for the test set (ROC 0.97), see Figure 3.4.


Figure 3.4. RPART decision tree for case study 2. We drop $V$ and write, e.g., $4 o$ instead of $V_{4, O}$ for readability.

Let $F$ be the corresponding decision-function. In what follows we focus on sufficient reasons for the instance in Table 3.4iii. The sufficient reasons using $F$ are given in Table 3.5.

| 1. | $\begin{aligned} & 0---0- \\ & --0--- \\ & --0--- \end{aligned}$ | 2. | $\begin{aligned} & 0----- \\ & --0-0- \\ & --0--- \end{aligned}$ | 3. | $\begin{aligned} & ----0- \\ & 0-0--- \\ & --0--- \end{aligned}$ | 4. | $\begin{gathered} --0--- \\ 0-0-0- \\ --0--- \end{gathered}$ |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 5. | $\begin{gathered} 01-101 \\ ------ \\ --0--- \end{gathered}$ | 6. | $\begin{aligned} & 01-1-1 \\ & ----0- \\ & -0-0-- \end{aligned}$ | 7. | $\begin{aligned} & -1-101 \\ & 0----- \\ & --0--- \end{aligned}$ | 8. | $\begin{aligned} & -101-1 \\ & 0---0- \\ & --0--- \end{aligned}$ |

Table 3.5. Reasons using $F$

Simple constraints for TTT. The encoding induces a constraint $C$ that expresses that no position contains both an O and an X , although, unlike the one-hot-constraints (as in Case

Study 1), it may have neither, i.e., $C$ is given by $\bigwedge_{0 \leqslant i \leqslant 8} \neg\left(V_{i, O} \wedge V_{i, X}\right)$. Again, consider the positive instance in Table 3.4iii. The reasons for the decision using $F_{C}$ include Reasons 1-4 in Table 3.5, as well as Reason A from Table 3.6 which strictly subsumes Reasons 5-8 in Table 3.5.

$$
\begin{array}{ll} 
& -1-1-1 \\
----- \\
--0---
\end{array} \quad \text { B. } \quad \begin{aligned}
& ------0000 \\
& --001-
\end{aligned}
$$

Table 3.6. (A) a reason using $F_{C}$, (B) a reason using $F_{C^{\prime}}$

This shows that some reasons of $F$ are redundant in light of the constraint $C$, e.g., as witnessed by the inclusion of the literals $\neg V_{0, O}$ and $V_{0, X}$ in reason 5 .

More complex constraints: adding game rules. Define the constraint $C^{\prime}$ to include $C$ as well as saying that the board is the result of valid play, i.e., that X moves first and players alternate moves. The additional constraint is $\bigvee_{S, T}\left(\psi_{S} \wedge \varphi_{T}\right)$ where $S, T$ vary over all subsets of $U=\{0,1,2, \cdots, 8\}$ such that $S \cap T=\varnothing$, and $0 \leqslant|S|-|T| \leqslant 1$, and $\psi_{S}$ is $\left(\bigwedge_{i \in S} V_{i, X}\right) \wedge\left(\bigwedge_{i \in U \backslash S} \neg V_{i, X}\right)$ and $\varphi_{T}$ is $\left(\bigwedge_{i \in T} V_{i, O}\right) \wedge\left(\bigwedge_{i \in U \backslash T} \neg V_{i, O}\right)$. The formula expresses that the set $S$ of positions where X has played is disjoint from the set $T$ where O has played, and that either there are the same number of moves, or X has played one more. Using $F_{C^{\prime}}$, the sufficient reasons for the instance above include Reason B in Table 3.6. This reason can be interpreted as follows: in light of the constraint $C^{\prime}$, which says that the board is the result of a valid play, if positions 4,5,7 are blank and position 8 has an O , then player X must have won. This is indeed correct: player O could not have won since with 5 moves in the game player O can only move twice, and there could not be a draw because not all positions were filled yet.

Discussion of Case-Study 2. This case study illustrates how our framework seamlessly takes complex semantic constraints, such as combinatorial constraints, into account when producing reasons. This should be contrasted with potential ad-hoc algorithms for incorporating any fixed constraint.

### 3.6 Discussion

The crux of this chapter shows how to handle constraints in a principled manner and establishes that ignoring constraints could result in unnecessarily long/complex sufficient reasons, as well as missing some reasons altogether. For computing sufficient reasons, our approach reduces the constrained case to the unconstrained case. Thus, any advance in the efficiency of tools for solving the latter will yield benefits for the former.

A general critique of the prime implicant based approach is that reasons may become too large to comprehend when the number of variables is large. Notice that our method is a step towards improving this problem in the presence of constraints. If the shortest sufficient reason in the presence of constraints is still too large to comprehend, not taking constraints into account may result in reasons that are even larger and even harder to comprehend. Observe, from the case studies, that while adding constraints may decrease or increase the number of sufficient reasons, it never increases the size of the shortest sufficient reasons (a fact that is guaranteed by Theorem 1). In cases of multiple (constraint-inequivalent) reasons for a decision (even amongst the shortest ones), we do not supply a way to pick one reason over another, a challenging problem (Lakkaraju et al. 2019). Indeed, preferring one reason over another would require additional assumptions about preferred reasons, e.g., favouring shorter reasons (Miller 2019).

Our framework for handling constraints is model-agnostic, i.e., it supplies the underlying principle for handling domain constraints for decision-functions that correspond to binary classifiers, no matter how the classifiers were learnt or modelled. As a proof-of-concept, we illustrated this by compiling decision trees into OBDDs. The general problem of compiling ML models into compact circuits is being actively researched, e.g., (Shih, Choi and Darwiche 2018) for Bayesian networks, (Choi, Shi et al. 2019) for Neural Networks, and (Audemard, Koriche and Marquis 2020) for Random Forests.

Finally, our work opens up applications that are currently only available in the unconstrained setting, including the study of classifier bias and counterfactual decisions (Darwiche and Hirth 2020).

## ChAPTER 4

## Sufficient Reasons in the Multi-value Setting

In this chapter, we formally extend the required notions for sufficient reasons to the multi-value setting. We also explore the reasons for constraint-subsumption that arise when constraints are due to one-hot encoding of variables, as we have seen within the descriptions of the Case Study 1 (the CRASH example) in Chapter 3.

Constraint subsumption means that in light of the constraints, a reason may be less general than it could be, so this form of subsumption needs to be considered when producing sufficient reasons.

Figure 4.1 depicts the full workflow. The focus of this chapter is on everything that follows obtaining the classification formula $\Delta$ (clarified by the dashed line), namely on encoding an instance and the classification formula, and then including encoding constraints and domain constraints to obtain sufficient reasons for the decisions of a multi-value classifier.


Figure 4.1. MV Workflow

### 4.1 Preliminaries

Here we recap the logical background that are needed to explain our theory in this chapter.

Variables, values and domains. Capital letters denote multi-value variables. We may use subscripts, e.g. $X_{i}$ denotes the $i^{\text {th }}$ variable. Bold capital letters denote a finite set of multivalue variables $\left\{X_{1}, X_{2}, \cdots, X_{n}\right\}$ or $\{X, Y, \ldots, Z\}$. Lower case letters denote (nominal) variable values. We may also use numbers for variable values. We use $D_{X}$ to denote the domain of variable $X$ that contains the set of values that the variable can take. In the running examples, we use two multi-value variables $X$ and $Y$ where $D_{X}=\left\{x_{1}, x_{2}, x_{3}\right\}$ and $D_{Y}=\left\{y_{1}, y_{2}, y_{3}\right\}$. The feature space $\mathcal{U}$ is the product of the domains of the variables, i.e., $\mathcal{U}=\prod_{X \in \mathbf{X}} D_{X}$. Elements of $\mathcal{U}$ are called assignments, typically denoted $\alpha$. We may also use functional notation, i.e., $\alpha\left(X_{i}\right) \in D_{X_{i}}$.

MV-formulas. An MV-formula $\Delta$ is a formula generated from atomic predicates of the form $X=v$ where $X \in \mathbf{X}$ and $v \in D_{X}$, the constants $\top$ (true) and $\perp$ (false), and the logical operations $\wedge$ (conjunction), $\vee($ disjunction $), \neg$ (negation), $\rightarrow$ (conditional) and $\leftrightarrow$ (bi-conditional).

For an MV-formula $\Delta$ and an assignment $\alpha$, define $\alpha \models \Delta$ (read $\alpha$ satisfies $\Delta$ or $\alpha$ is a model of $\Delta$ ), inductively as usual:

- $\alpha \models \mathrm{T}$,
- $\alpha \models(X=x)$ if $\alpha(X)=x$.
- $\alpha \models\left(\Delta_{1} \wedge \Delta_{2}\right)$ if $\alpha \models \Delta_{i}$ for all $i=1,2$,
- $\alpha \models \neg \Delta_{1}$ if it is not the case that $\alpha \models \Delta_{1}$.

A formula $\Delta$ is consistent if there is some assignment that satisfies $\Delta$, otherwise it is inconsistent. A set $\Phi$ of formulas is consistent if there is some assignment $\alpha$ that satisfies every formula in $\Phi$.

A formula $\Delta_{i}$ logically implies another formula $\Delta_{j}$, written $\Delta_{i} \models \Delta_{j}$, iff every assignment satisfying $\Delta_{i}$ satisfies $\Delta_{j}$. If we also have $\Delta_{j} \models \Delta_{i}$ then the two formulas are logically equivalent, written $\Delta_{i} \equiv \Delta_{j}$.

We may write $[\Delta]$ for the set of assignments that satisfy $\Delta$.

A formula $\Delta$ is trivial if neither $\Delta$ nor $\neg \Delta$ is consistent.

Literals and terms and relations between terms. Following (Choi, Shih et al. 2020), we define an MV-literal to be a non-trivial MV-formula that mentions a single variable.

Let $D_{X}=\left\{x_{1}, x_{2}, x_{3}\right\}$. The following are literals:

- $X=x_{1}$,
- $X \neq x_{1}$,
- $X=x_{1} \vee X=x_{3}$,
- $X \neq x_{1} \wedge X \neq x_{2}$.

The following are not literals, as they are trivial:

- $X=x_{1} \vee X=x_{2} \vee X=x_{3}$,
- $X=x_{1} \wedge X \neq x_{1}$.

Intuitively, even-though a literal is an arbitrary formula, it just determines a strict subset of $D_{X}$. For instance if $D_{X}=\{0,1\}$ semantically there are 2 literals (which can be written in many different ways). E.g. $(X=1),(X \neq 0)$ and $(X=1 \wedge X=1)$ are semantically the same.

An $M V$-Term $\tau$ is a conjunction of MV-literals over distinct variables. Let $D_{X}=\left\{x_{1}, x_{2}, x_{3}\right\}$ and $D_{Y}=\left\{y_{1}, y_{2}, y_{3}\right\}$. The following is an MV-term: $\left(X=x_{1} \vee X=x_{2}\right) \wedge\left(Y \neq y_{1}\right)$. We emphasis that although having disjunctions in literals may seem unnatural, semantically, terms can be expressed as conjunctions of negations of atoms.

An MV-literal that specifies a single value, i.e., the cardinality of set $S$ is 1 , is called simple. An MV-term is simple if all of its MV-literals are simple. The following MV-terms are simple:

- $X=x_{1} \wedge Y=y_{1}$,
- $X=x_{2} \wedge Y=y_{1}$.

The following MV-terms are not simple:

- $X \neq x_{1} \wedge Y=y_{1}$,
- $X=x_{2} \wedge\left(Y=y_{1} \vee Y=y_{3}\right)$.

Logical entailment applied to MV-terms is called subsumption. That is, an MV-term $\tau_{i}$ subsumes MV-term $\tau_{j}$ if $\tau_{j} \models \tau_{i}$. If we also have $\tau_{i} \not \equiv \tau_{j}$, then $\tau_{i}$ strictly subsumes $\tau_{j}$. For example, the MV-term $X=x_{1} \wedge\left(Y=y_{1} \vee Y=y_{3}\right)$ is strictly subsumed by the MV-terms $X \neq x_{3} \wedge\left(Y=y_{1} \vee Y=y_{3}\right)$ and $X=x_{1}$.

Moreover, when logical entailment is applied to literals we use general instead of "subsumes", i.e., if $l_{j} \models l_{i}$ we say that $l_{i}$ is more general than $l_{j}$.

To understand subsumption better, consider the following example. Let $X$ denote the day of the week, $D_{X}=\{$ Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday \}. Let $Y$ denote month of the year $D_{Y}=\{1,2,3,4,5,6,7,8,9,10,11,12\}$. The MVterm $\tau_{1}:(X=$ Monday $\wedge(Y=1 \vee Y=2))$ is subsumed by another term $\tau_{2}:(X \neq$ Saturday $\wedge(Y=1 \vee Y=2 \vee Y=3))$ since $\tau_{2}$ is more general and contains $\tau_{1}$.

An MV-term $\tau$ is an implicant of MV-formula $\Delta$ if $\tau \models \Delta$.

An MV-term $\tau$ is a prime implicant of $\Delta$ if $\tau$ is an implicant of $\Delta$ that is not strictly subsumed by another implicant of $\Delta$.

## Constraints.

DEFInition 6. An MV-constraint $\kappa$ is a set of MV-instances that can be represented by a formula.

Specifically, $\kappa$ captures background knowledge.

Let $\kappa$ be an MV-constraint. For two terms $\tau_{i}, \tau_{j}$ we say that $\tau_{j}$ is $\kappa$-subsumed by $\tau_{i}$ if $\tau_{j} \wedge \kappa \models \tau_{i} \wedge \kappa$. If also have that $\tau_{i} \wedge \kappa \not \models \tau_{j} \wedge \kappa$ we say that $\tau_{j}$ is strictly $\kappa$-subsumed by $\tau_{i}$. We can think of this as inducing an order on terms: $\tau_{j} \leqslant \tau_{i}$ if $\tau_{j}$ is $\kappa$-subsumed by $\tau_{i}$. Two terms $\tau_{i}, \tau_{j}$ that are not logically equivalent may still be logically equivalent modulo $\kappa$, i.e., each $\kappa$-subsumes the other. Note that in this case $[\kappa] \cap\left[\tau_{i}\right]=[\kappa] \cap\left[\tau_{j}\right]$.

### 4.2 Extending to the Multi-Value Setting

Classifiers that accept numeric or categorical variables as input are called multi-value-input classifiers. Multi-value-input classifiers can be divided into separate groups based on the type of their outputs. Those with a single Boolean output variable are called Boolean-output. Those that have a single multi-value output are multi-value-output. And those that have multiple outputs are called multi-output classifiers.

Example 3 (Demonstrative Example ${ }^{1}$ ). Consider the study of the expression level of five genes from a sample of 500 hypothetical individuals, 200 of whom are diagnosed with cancer and the rest are controls. The columns in Table 4.1. show the expression levels of genes ( $g_{1}$ to $g_{5}$ ), patient's sex and age, and two outcome variables (C1-stage) and (C2-stage) for the stage of two different types of comorbidity for each hypothetical case.

A classifier that is trained on the above dataset and predicts C1-stage and C2-stage for a new patient based on the values of 7 inputs \{g1-g5, sex, age\} is an example of a multi-value-input, multi-output classifier. If we dropped C2-stage, and trained a classifier that only predicts C1-stage, then we have a multi-value-input, multi-value-output classifier. If we dropped both C1-stage and C2-stage, and created a new dichotomised outcome variable indicating the presence of C1 (yes,no), regardless of its stage, then the classifier is an example of a multi-value-input, Boolean-output classifier.

[^2]| Case | $\mathbf{g 1}$ | $\mathbf{g 2}$ | $\mathbf{g 3}$ | $\mathbf{g 4}$ | $\mathbf{g 5}$ | sex | age | C1-stage | C2-stage |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| $\mathbf{1}$ | 0.17 | 0.51 | 0.83 | 0.51 | 0.51 | $F$ | 34 | 0 | 0 |
| $\mathbf{2}$ | 0.17 | 0.51 | 0.83 | 0.51 | 0.51 | $F$ | 26 | 0 | 0 |
| $\mathbf{3}$ | 0.17 | 0.51 | 0.83 | 0.51 | 0.17 | $M$ | 18 | 0 | 0 |
| $\mathbf{4}$ | 0.51 | 0.51 | 0.17 | 0.51 | 0.51 | $F$ | 46 | 0 | 0 |
| $\mathbf{5}$ | 0.51 | 0.51 | 0.17 | 0.51 | 0.51 | $M$ | 20 | 0 | 0 |
| $\mathbf{6}$ | 0.17 | 0.83 | 0.51 | 0.17 | 0.83 | $M$ | 19 | 0 | 0 |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ |
| $\mathbf{4 9 5}$ | 0.51 | 0.17 | 0.83 | 0.51 | 0.83 | $F$ | 50 | 1 | 2 |
| $\mathbf{4 9 6}$ | 0.51 | 0.17 | 0.83 | 0.51 | 0.83 | $M$ | 49 | 0 | 1 |
| $\mathbf{4 9 7}$ | 0.83 | 0.17 | 0.51 | 0.83 | 0.51 | $M$ | 34 | 0 | 3 |
| $\mathbf{4 9 8}$ | 0.83 | 0.51 | 0.51 | 0.17 | 0.83 | $M$ | 67 | 2 | 0 |
| $\mathbf{4 9 9}$ | 0.83 | 0.51 | 0.51 | 0.17 | 0.83 | $F$ | 34 | 1 | 3 |
| $\mathbf{5 0 0}$ | 0.17 | 0.83 | 0.51 | 0.17 | 0.83 | $F$ | 23 | 3 | 0 |

Table 4.1. Demonstrative Dataset.

The case of multi-value-output and multi-output classifiers is briefly discussed in Section 4.6 (Discussions). In the remaining sections, our focus is on multi-value-input Boolean-output classifiers, and we refer to these as multi-value classifiers or MV-classifiers for short.

### 4.2.1 Decision Function of a Multi-Value Classifier

The input-output behaviour of multi-value classifiers can be captured by total functions.
Definition 7. Given a set $\boldsymbol{X}$ of variables $X$ and their corresponding domains $D_{X}$, $a$ multivalue decision function over $\boldsymbol{X}$ is a function $F$ that maps instances over $\boldsymbol{X}$ to classes $\{0,1\}$, i.e., a total function of the form $F: \mathcal{U} \rightarrow\{0,1\}$. If $F(\alpha)=1$ (resp. $=0$ ) we say that the classifier's decision is positive (resp. negative) on that instance.

We represent an MV-classifier by a single formula $\Delta$, where the models of $\Delta$ capture the instances in class 1 , and the models of formula $\neg \Delta$ capture instances in class 0 . That is, $\alpha$ satisfies $\Delta$ if and only if $F(\alpha)=1$; and thus, $\alpha$ satisfies $\neg \Delta$ iff $F(\alpha)=0$.

Example 4 (Illustrating the decision function of multi-value classifiers). Consider the following example with the Iris dataset (Dua and Graff 2017) The Iris dataset contains measurements for features distinguishing three classes of Iris flowers, with 50 instances for


Figure 4.2. Decision tree for IRIS
each class. We considered a dichotomised class of 'Versicolor vs. Not-versicolor', and trained the decision tree shown in Figure 4.2.

From the Figure 4.2 decision tree, we see that the classifier uses only two variables, Petal Width $(W)$ and Petal Length $(L)$, and discretised the variables as follows:

- Petal Width into 5 intervals $(-\infty, 0.08],(0.08,1.55],(1.55,1.65],(1.65,1.75]$, $(1.75,+\infty)$, and
- Petal Length to 3 intervals $(-\infty, 4.95]$, $(4.95,5.45],(5.45,+\infty]$.

Let $\boldsymbol{X}=\{W, L\}$ and let their values represent these intervals, i.e., $D_{W}=\left\{x_{1}, \ldots, x_{5}\right\}$ and $D_{L}=\left\{y_{1}, \ldots, y_{3}\right\}$. The decision function $F$ of the MV-classifier is then over the 15 $\left(\left|D_{W}\right| \times\left|D_{L}\right|\right)$ possible combinations of the values of variables $W$ and $L$ :

| $W$ | $x_{1}$ | $x_{2}$ | $x_{3}$ | $x_{4}$ | $x_{5}$ | $x_{1}$ | $x_{2}$ | $x_{3}$ | $x_{4}$ | $x_{5}$ | $x_{1}$ | $x_{2}$ | $x_{3}$ | $x_{4}$ | $x_{5}$ |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| $L$ | $y_{1}$ | $y_{1}$ | $y_{1}$ | $y_{1}$ | $y_{1}$ | $y_{2}$ | $y_{2}$ | $y_{2}$ | $y_{2}$ | $y_{2}$ | $y_{3}$ | $y_{3}$ | $y_{3}$ | $y_{3}$ | $y_{3}$ |
| $F(W, L)$ | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |

Table 4.2. Truth table representing the classifier's decision function

Therefore, while the inputs to the classifier in Figure 4.2 are continuous variables, the input-output behaviour of the classifier can be captured by the MV-formula:
$\Delta=\left(W=x_{2} \wedge L=y_{1}\right) \vee\left(W=x_{3} \wedge L=y_{1}\right) \vee\left(W=x_{3} \wedge L=y_{2}\right) \vee\left(W=x_{4} \wedge L=y_{2}\right)$

### 4.3 Sufficient Reasons in the Multi-Value Setting

Given a multi-value decision function $\Delta$ and an instance $\alpha$ where $\Delta(\alpha)=1$ we are interested in finding the reasons behind the classifier's positive decision on this instance.

The notion of prime implicant as sufficient reasons is straightforwardly applicable in the multi-value setting. We recall the extension of the notion of prime implicants from Boolean to multi-value from (Marquis 1991; Ignatiev, Narodytska and Marques-Silva 2019a).

In the Boolean setting, given a decision function $F$, and an instance $\mathbf{x}$, a sufficient reason for decision $F(\mathbf{x})=1$ is a prime implicant of $F$ that is satisfied by instance $\mathbf{x}$ (Definition 4, Chapter 3).

The same principles can be applied in the multi-value setting. Let us recall the definition of a multi-value prime implicant:

Definition 8. [CF. (Marquis 1991; Ignatiev, Narodytska and Marques-Silva 2019a)]. An $M V$-term $\tau$ is a prime implicant of $F$ if $\tau$ is an implicant of $F$ that is not strictly subsumed by another implicant of $F$.

A sufficient reason for a positive decision on $\alpha$ can be defined to be a prime implicant of $F$ that is satisfied by instance $\alpha$. More formally:

Definition 9. [CF. (Choi, Shih et al. 2020)]. Let F be a total multi-value function and $\alpha$ be an instance decided positively by F. An MV-term $\tau$ is a sufficient reason of the decision $F(\alpha)=1$ if $(i) \tau$ is a prime implicant of $F$ and (ii) $\tau$ is satisfied by $\alpha$.

Note that this definition does not talk about domain constraints.

Multi-value sufficient reasons in the presence of domain constraints. Similar to the Boolean case, the process of finding sufficient reasons in the multi-value setting should ideally be conducted while including any background knowledge that may be available on the problem domain into the reasoning process. Including the domain constraints into the
reasoning process ensures that sufficient reasons are devoid of any unnecessary complexities that would be simplified by the background knowledge as shown in the previous chapter.

In the presence of domain constraints, the notion of sufficient reasons for a constrained decision function can be straightforwardly extended to the multi-value setting. We do so now.

Let $P I(\Delta)$ denote the set of all prime implicants of $\Delta$. Let $\kappa$ be an MV-constraint (Definition 6).

DEFINITION 10. An MV-term term $\tau$ is a multi-value sufficient reason, or simply sufficient reason, for instance $\alpha$ if $\alpha \models \tau$ and $\tau \in P I(\kappa \rightarrow \Delta)$.

Example 5 (Illustrating the definition of multi-value sufficient reason (with terms with complex literals)). Consider two variables $X$ and $Y$ with $D_{X}=\left\{x_{1}, x_{2}, x_{3}\right\}$ and $D_{Y}=$ $\left\{y_{1}, y_{2}, y_{3}\right\}$ and the formula $\Delta:\left(X=x_{1} \vee X=x_{2}\right) \wedge\left(Y=y_{2} \vee Y=y_{3}\right)$ representing the decision function of a classifier over these two variables.

Given an instance $\alpha: X=x_{1} \wedge Y=y_{2}$, the only prime implicant (and multi-value sufficient reason) for the decision $\Delta(\alpha)=1$ is $X \neq x_{3} \wedge Y \neq y_{1}$. Now consider a constraint $\kappa: X \neq x_{3} \rightarrow Y \neq y_{1}$, then the sufficient reason for $\alpha$ is $X \neq x_{3}$.

Multi-value sufficient-reasons, when applied to Boolean formulas, recover the notion of sufficient reasons from the previous chapter (i.e. Definition 8, Chapter 3).

Suppose in the MV setting that all of the variables are Boolean (i.e., $D_{X}=\{0,1\}$ for every variable $X$ ). Then the satisfaction relation $\models$, and thus also subsumption and sufficient reasons, coincide in the two settings. To see this, note that $\alpha \models X=1$ in the MV setting iff $\alpha \models X$ in the Boolean setting, and $\alpha \models X=0$ in the MV setting iff $\alpha \models \neg X$ in the Boolean setting. Thus, $\alpha \models \varphi$ in the MV setting iff $\alpha \models \varphi$ in the Boolean setting. Thus, by treating the MV-literal $X=1$ as the Boolean literal $X$, and treating the MV-literal $X=0$ as the Boolean literal $\neg X$, we can treat MV-formulas as Boolean formulas. Then, $\varphi \models \varphi^{\prime}$ in the MV setting iff $\varphi \models \varphi^{\prime}$ in the Boolean setting, i.e., subsumption coincides in the two settings. Also, if $\tau$ is an MV-term, then each literal in $\tau$ is of the form $X=0, X=1$, or $(X=0) \vee(X=1)$. However, the latter is logically equivalent to $\top$, and thus can be dropped
from the term. Thus, an MV-term $\tau$ can be considered a Boolean term. Thus, $\tau$ is a (prime) implicant of $\varphi$ in the MV setting iff $\tau$ is a (prime) implicant of $\varphi$ in the Boolean-setting, and so sufficient reasons in the two setting coincide.

To summarise this section, the notions of prime implicants as sufficient reasons can be straightforwardly extended from the Boolean to multi-value setting: (1) if variables in the multi-value case happen to be Boolean, we get the same notion as in the Boolean setting; (2) prime implicants are computed by maximal generalisation of terms that satisfy an instance (while ensuring that they remain an implicant).

In the next section, we discuss the practical aspects and considerations for computing multivalue sufficient reasons.

### 4.4 Computing Sufficient Reasons in the Multi-value Setting

There are some important issues that arise when computing sufficient reasons in the multivalued case as compared with the Boolean case:
(1) Terms (and therefore prime implicants) in the multi-value setting can contain "complex" literals, i.e. in the multi-value setting literals can be arbitrary formulas over a single variable (containing disjunctions, conjunctions, equalities or inequalities). This is a notable difference with the Boolean case where there are only two non-trivial literals on a variable $X$ (i.e. $X$ and $\neg X$ ).
(2) An MV-term $\tau_{1}$ subsumes another term $\tau_{2}$ if every literal in $\tau_{1}$ subsumes some literal in $\tau_{2}$. In comparison, in the Boolean case, this condition is equivalent to the condition that every literal in $\tau_{2}$ appears in $\tau_{1}$.

As we are interested in using the apparatus that is available in a Boolean setting (such as Knowledge Compilation or SAT solvers), in the remaining part of this chapter, we focus on reducing the problem of producing sufficient reasons for a multi-value classifier to the problem of producing sufficient reasons for a Boolean classifier. This is operationalised through encoding the required multi-value expressions in Boolean logic.

### 4.4.1 One-hot Encoding

One-hot encoding is one of the most common pre-processing techniques for representing multi-value variables with Boolean variables.

Consider a multi-value variable $X$ with $m$ values in its domain $D_{X}:\left\{x_{1}, \ldots, x_{m}\right\}$.

Encoding MV-variables. To represent an MV-variable $X$ in the Boolean setting, we introduce $m$ Boolean variables $X_{x_{1}}, \ldots, X_{x_{m}}$. We encode an MV-atom $X=x_{i}$ by the Boolean term that is a conjunction of $X_{x_{i}}$ and $\neg X_{x_{j}}$ for all $j \neq i$. For example, if $D_{X}=\left\{x_{1}, x_{2}, x_{3}\right\}$, the three values of $X$ are encoded as $\left(X_{x_{1}} \wedge \neg X_{x_{2}} \wedge \neg X_{x_{3}}\right),\left(\neg X_{x_{1}} \wedge X_{x_{2}} \wedge \neg X_{x_{3}}\right)$ and $\left(\neg X_{x_{1}} \wedge \neg X_{x_{2}} \wedge X_{x_{3}}\right)$.

Encoding MV-formulas. We extend this encoding to MV-formulas $\Delta$ as follows: define $\Delta^{b}$ to be the BV-formula formed from $\Delta$ by replacing every occurrence of an MV-atom by its encoding term. For example, if the domain of $X$ is $\left\{x_{1}, x_{2}, x_{3}\right\}$ and the domain of $Y$ is $\left\{y_{1}, y_{2}, y_{3}\right\}$, then the encoding of $X=x_{1} \vee Y=y_{2}$ is

$$
\left(X_{x_{1}} \wedge \neg X_{x_{2}} \wedge \neg X_{x_{3}}\right) \vee\left(\neg Y_{y_{1}} \wedge Y_{y_{2}} \wedge \neg Y_{y_{3}}\right)
$$

and the encoding of $\left(X=x_{1} \vee X=x_{2}\right) \wedge Y=y_{2}$ is

$$
\left(\left(X_{x_{1}} \wedge \neg X_{x_{2}} \wedge \neg X_{x_{3}}\right) \vee\left(\neg X_{x_{1}} \wedge X_{x_{2}} \wedge \neg X_{x_{3}}\right)\right) \wedge\left(\neg Y_{y_{1}} \wedge Y_{y_{2}} \wedge \neg Y_{y_{3}}\right)
$$

Once $\Delta^{b}$ is obtained, one may mistakenly assume that it is all one needs to compute sufficient reasons. One-hot encoding is known for producing "illegal states" (Golson 1993), i.e. combinations of Boolean variables that are not representative of any combination of MV-variables. There is an important constraint on the Boolean feature space that needs to be considered.

Adding the necessary encoding constraint. We need to ensure that $\Delta^{b}$ rules out the useless combinations of Boolean variables that are not mapped to any MV-term properly. For example, ( $X_{x_{1}} \wedge X_{x_{2}} \wedge \neg X_{x_{3}}$ ) would not represent any term of $\Delta$ because it says that $X$ takes the value
$x_{1}$ and the value $x_{2}$, but each MV-variable can take just one value) and needs to be ruled out. Similarly ( $\neg X_{x_{1}} \wedge \neg X_{x_{2}} \wedge \neg X_{x_{3}}$ ) needs to be ruled since it cannot represent any MV-literal.

This is done by employing a one-hot constraint $\kappa_{X}^{b}$ that ensures $\Delta^{b}$ represents only the terms of $\Delta$ and nothing more, i.e.,

$$
\begin{equation*}
\kappa_{X}^{b}=\bigvee_{i} X_{x_{i}} \wedge \bigwedge_{i \neq j} \neg\left(X_{x_{i}} \wedge X_{x_{j}}\right) \tag{4.1}
\end{equation*}
$$

where $i, j$ vary over $m$.
Let $\kappa_{\mathbf{X}}^{b}$ be the conjunction of one-hot constraints for all variables, i.e.,

$$
\begin{equation*}
\kappa_{\mathbf{X}}^{b}=\bigwedge_{X \in \mathbf{X}} \kappa_{X}^{b} \tag{4.2}
\end{equation*}
$$

Then, $\Delta$ in Boolean form is represented by

$$
\begin{equation*}
\Delta^{H}=\kappa_{\mathbf{X}}^{b} \rightarrow \Delta^{b} \tag{4.3}
\end{equation*}
$$

We use $H$ to denote the embedded one-hot constraint.

### 4.4.2 Restricting Prime Implicants of $\Delta^{H}$ to be Negative Boolean Terms

(Choi, Shih et al. 2020) prove that for one-hot encoded multi-value variables, prime implicants should be limited to terms that only contain negative Boolean literals.

In the next section, we will illustrate the need for such an approach with an example of an MV-formula and its Boolean encoding that has prime implicants with positive literals whose corresponding MV-implicants are not prime. That is, Boolean prime implicants with positive literals need not be prime in the MV setting.

A negative Boolean term $\tau^{n e g}$ is formed from an MV-term $\tau$ by replacing every MV-literal in $\tau$ with a negative Boolean term. An MV-literal specifies a subset $S_{\tau}$ of values in $D_{X}$. It is encoded using the negative Boolean term $\bigwedge_{x_{i} \notin S_{\tau}} \neg X_{x_{i}}$. For example, if $D_{X}=\left\{x_{1}, x_{2}, x_{3}\right\}$,
then literal $X=x_{2}$ is encoded using the negative Boolean term $\neg X_{x_{1}} \wedge \neg X_{x_{3}}$ and literal $X=x_{1} \vee X=x_{2}$ is encoded using the negative Boolean term $\neg X_{x_{3}}$.

Further, (Choi, Shih et al. 2020) observe that negative Boolean terms have two useful properties:
(1) A single negative Boolean term can represent an MV-term with complex literals (whereas in normal one-hot encoding we may need multiple Boolean terms to represent the arbitrary formula of an MV-term).
(2) If the negative Boolean representation of an MV-term $\tau_{1}$ subsumes the negative Boolean representation of another MV-term $\tau_{2}$ (i.e. if $\bigwedge_{x_{i} \notin S_{\tau_{2}}} \neg X_{x_{i}} \models \bigwedge_{x_{i} \notin S_{\tau_{1}}} \neg X_{x_{i}}$ ), then we have that $\tau_{2}$ is subsumed by $\tau_{1}$.

By restricting PIs to be negative Boolean terms, given a term $\tau$, (Choi, Shih et al. 2020) show that:

$$
\begin{equation*}
\tau \in P I(\Delta) \Leftrightarrow \tau^{n e g} \in P I\left(\Delta^{H}\right) \tag{4.4}
\end{equation*}
$$

We now show how to take domain constraints into account.

Recall that $\kappa_{\mathbf{X}}^{b}$ denotes the one-hot encoding constraints for all variables (Equation 4.2), and let $\Psi^{b}$ be the Boolean encoded background knowledge capturing the relationships between variables. A negative Boolean term $\tau^{\text {neg }}$ is a sufficient reason (Definition 9) of $\Delta^{H}$ for instance $\alpha^{b}$ in presence of background knowledge $\Psi^{b}$ if :

$$
\begin{gather*}
\alpha^{b} \models \tau^{n e g} \text { and }  \tag{4.5}\\
\tau^{n e g} \in P I\left(\Psi^{b} \rightarrow \Delta^{H}\right) \tag{4.6}
\end{gather*}
$$

We remark that the formula $\Psi^{b} \rightarrow \Delta^{H}$ is $\Psi^{b} \rightarrow\left(\kappa_{\mathbf{X}}^{b} \rightarrow \Delta^{b}\right)$, which is equivalent to ( $\kappa_{\mathbf{X}}^{b} \wedge$ $\left.\Psi^{b}\right) \rightarrow \Delta^{b}$, i.e., constraints that capture background knowledge can be conjoined with the encoding constraints to compute sufficient reasons.

EXAMPLE 6 (Demonstrating constraint-subsumption with $\Delta^{H}$ and showing that negative terms are not constraint-subsumed). Consider two $M V$-variables $X$ and $Y$ with $D_{X}=$
$\left\{x_{1}, x_{2}, x_{3}\right\}$ and $D_{Y}=\left\{y_{1}, y_{2}, y_{3}\right\}$. Let the following truth table (Table 4.3.) represent the decision function of a classifier over these two variables.

| $X$ | $x_{1}$ | $x_{2}$ | $x_{3}$ | $x_{1}$ | $x_{2}$ | $x_{3}$ | $x_{1}$ | $x_{2}$ | $x_{3}$ |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| $Y$ | $y_{1}$ | $y_{1}$ | $y_{1}$ | $y_{2}$ | $y_{2}$ | $y_{2}$ | $y_{3}$ | $y_{3}$ | $y_{3}$ |
| $F(X, Y)$ | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |

TABLE 4.3. decision function

The formula $\Delta:\left(X=x_{1} \wedge Y=y_{2}\right) \vee\left(X=x_{2} \wedge Y=y_{2}\right) \vee\left(X=x_{1} \wedge Y=y_{3}\right) \vee(X=$ $\left.x_{2} \wedge Y=y_{3}\right)$ can represent this classifier.

Given an instance $\alpha: X=x_{1} \wedge Y=y_{2}$, the four $M V$-terms that are listed in the first column of Table 4.4 are all implicants of $\Delta$, and are consistent with $\alpha$. However, notice that between these MV-terms, the one at row I2 subsumes all other terms as its literals are the weakest (or most abstract, or most general) (i.e., the values of $X$ and $Y$ in I2 contain all the values of $X$ and $Y$ in all other $M V$-terms).

|  | A. MV-terms | B. Boolean terms |
| :---: | :---: | :---: |
| I1. | $X \neq x_{3} \wedge Y=y_{2}$ | $\neg X_{x_{3}} \wedge Y_{y_{2}}$ |
| I2. | $X \neq x_{3} \wedge Y \neq y_{1}$ | $\neg X_{x_{3}} \wedge \neg Y_{y_{1}}$ |
| I3. | $X=x_{1} \wedge Y=y_{2}$ | $X_{x_{1}} \wedge Y_{y_{2}}$ |
| I4. | $X=x_{1} \wedge Y \neq y_{1}$ | $X_{x_{1}} \wedge \neg Y_{y_{1}}$ |

Table 4.4. Multi-value terms in Column A. and their Boolean encoding listed in column B.

In other words, $\left(X=x_{1}\right) \models X \neq x_{3}$ and $\left(X=x_{2}\right) \models X \neq x_{3}$ and similarly, $\left(X=x_{1}\right) \models$ $X \neq x_{2}$ and $\left(X=x_{3}\right) \models X \neq x_{2}$.

The MV-term $X \neq x_{3} \wedge Y \neq y_{1}$ is the only prime implicant of $\Delta$. But all four terms in the second column of Table 4.4 are prime implicants of the Boolean encoding of $\Delta$ (i.e., $\Delta^{H}$ ).

Some of the Boolean prime implicants in the second column of Table 4.4 correspond to MV-terms that are subsumed by some other MV-terms (and therefore are not prime) in the multi-value setting.

For example, the MV-term $X=x_{1} \wedge Y=y_{2}$ is subsumed by the MV-term $X \neq x_{3} \wedge Y \neq y_{1}$, however its Boolean encoding, $X_{x_{1}} \wedge Y_{y_{2}}$ is not subsumed by $\neg X_{x_{3}} \wedge \neg Y_{y_{1}}$.

By restricting sufficient reasons to be negative Boolean terms, the only sufficient reason for $\Delta^{H}$ in Example 6 is $\neg X_{x_{3}} \wedge \neg Y_{y_{1}}$, which corresponds to the only prime implicant of $\Delta$, i.e., $X \neq x_{3} \wedge Y \neq y_{1}$.

Example 7 (Adding to the Case Study 1 of Chapter 3, where up to constraint-equivalence there were two sufficient reasons (i) $A_{1} \wedge M_{5} \wedge P_{1}$, (ii) $\neg A_{6} \wedge \neg A_{7} \wedge M_{5} \wedge P_{1}$ ). By using the $\Delta^{H}$ and subsequently computing (negative) prime implicants we obtain only one sufficient reason:

$$
\tau: \neg A_{6} \wedge \neg A_{7} \wedge \neg M_{1}, \neg M_{2}, \neg M_{3}, \neg M_{4} \wedge \neg M_{6} \wedge \neg P_{2} \wedge \neg P_{3} .
$$

Therefore restricting prime implicants to the negative Boolean terms, provides a reason that is not constraint-subsumed.

### 4.4.3 Size of a reason

We now clarify the notion of the size of a sufficient reason in the multi-value setting. The size of a reason (term) in the Boolean setting is the number of literals that occur in it. Similarly in the multiple-value setting, the size of a reason is the number of MV-literals that appear in it. If Boolean encoding is used to compute sufficient reasons for multi-value classifiers, it is important to note that, the size of a reason in the multi-value setting is not the number of encoding literals in the Boolean setting, rather it is the number of literals of the MV-term. In other words, reasons need to be transformed back into their multi-value form to compute the size of a reason.

Therefore the size of the reason $\tau$ given above is 3 as the MV-term that that is encoded by this PI has three MV-literals. For example:

$$
\begin{aligned}
& (A \neq 6 \wedge A \neq 7) \wedge M=5 \wedge P=1, \\
& (A=1 \vee A=2 \vee A=3 \vee A=4 \vee A=5) \wedge M=5 \wedge P=1
\end{aligned}
$$

both have three literals. All other MV-terms that $\tau$ can represent have 3 MV-literals and have a size=3.

Example 8 (A complete example of computing sufficient reasons in the presence of background knowledge in the multi-value setting through Boolean encoding). Consider the illustrative example in section 3.4.1. This time, let all variables be multi-value variables, each with 2 values in their domain $D_{X_{i}}=\{0,1\}$. let the associated classification function over $\boldsymbol{X}=\{L, K, P, A\}$ be represented by $\Delta:$

$$
\begin{aligned}
& (L=0 \wedge K=0 \wedge P=1 \wedge A=1) \vee(L=0 \wedge K=0 \wedge P=1 \wedge A=1) \vee \\
& (L=1 \wedge K=1 \wedge P=0 \wedge A=0) \vee(L=1 \wedge K=1 \wedge P=1 \wedge A=0) \vee \\
& (L=1 \wedge K=1 \wedge P=1 \wedge A=1)
\end{aligned}
$$

Following the original example, a student's enrolments must satisfy the following constraints $\Psi$ : they must take $P$ or $L,(P=1 \vee L=1)$; the prerequisite for taking $A$ is $P,(A=1 \rightarrow$ $P=1)$; the prerequisite for taking $K$ is $A$ or $L,(K=1 \rightarrow(A=1 \vee L=1))$.

We follow the example in section 3.4.1 and compare the sufficient reasons that were obtained in the Boolean setting with those that are obtained from the MV-encoding for the instance $\alpha:(L=0 \wedge K=0 \wedge P=1 \wedge A=1)$ that is decided positively. This instance corresponds to a candidate that did not take L or K, but did take P and A. Results in Table 4.5 were computed by reusing the tool proposed in (Shih, Choi and Darwiche 2018).

| $L$ | $K$ | $P$ | $A$ | Reasons | Negative PIs of $\Delta^{H}$ consistent with $\left(\alpha^{b}\right)$ |
| :--- | :--- | :--- | :--- | :--- | :--- |
| 0 | 0 | 1 | 1 | $(\neg L \wedge A)$ | $\left(L_{1}=0 \wedge A_{0}=0\right)$ |
| 0 | 1 | 1 | $l$ | $(\neg L \wedge A), K$ | $\left(L_{1}=0 \wedge A_{0}=0\right),\left(K_{0}=0\right)$ |
| 1 | 1 | 0 | 0 | $K$ | $\left(K_{0}=0\right)$ |
| 1 | 1 | 1 | 0 | $K$ | $\left(K_{0}=0\right)$ |
| 1 | 1 | $l$ | 1 | $K$ | $\left(K_{0}=0\right)$ |

TAble 4.5. Rows list the instances for which the classifier made a positive decision, and that satisfy the constraints. The other two columns show reasons that were obtained in the Boolean and encoded multi-value setting respectively.

The results confirm that we can reuse the apparatus from the Boolean setting in the MV setting.

### 4.5 Constraint Subsumption in General

So far, we have demonstrated that in some cases (specifically for when constraints are due to the Boolean encoding of MV-variables) it is required that the sufficient reasons supplied are not constraint-subsumed. Indeed, in Example 6 we have seen that constraint-subsumed reasons when constraints are due to the Boolean encoding of MV-variables, would correspond to reasons that are not prime in the MV setting.

However, so far, it is not clear whether we should allow constraint-subsumed prime implicants as sufficient reasons in general. For instance, in the following example, it is not immediately clear whether sufficient reasons should be limited to those that are not constraint-subsumed.

Example 9. Consider three $M V$-variables $X$ and $Y$ and $Z$ with $D_{X}=\left\{x_{1}, x_{2}, x_{3}\right\}$ and $D_{Y}=\left\{y_{1}, y_{2}, y_{3}, y_{4}\right\}$ and $D_{Z}=\left\{z_{1}, z_{2}, z_{3}\right\}$.

Let $\Delta:\left(X=x_{1} \vee\left(Y \neq y_{3} \wedge Y \neq y_{4}\right)\right) \wedge Z=z_{3}$ represent the decision function of a classifier over these three variables.

Given a constraint $\kappa: X=x_{1} \rightarrow Y=y_{1}$, there are two prime implicants of $\kappa \rightarrow \Delta$ : $X=x_{1} \wedge Z=z_{3}$ and $\left(Y \neq y_{3} \wedge Y \neq y_{4}\right) \wedge Z=z_{3}$.

The first prime implicant is $\kappa$-subsumed by the second one.

This example is an illustration of the fact that with respect to the part of the feature space that we "care about" (i.e., those that satisfy $\kappa$ ), constraint-subsumed prime implicants are less general than those prime implicants that are not constraint-subsumed. Indeed, they cover less instances.

While it could be argued that without having further background knowledge (e.g., user preference), we cannot choose one reason over the other, from our perspective, taking constraints into account is an indication of user preference. It shows that the user is interested in those explanations that are most parsimonious (and general) with respect to the constraints. Therefore, unless there is a strong reason not to do so, we propose not to return constraintsubsumed prime implicants as sufficient reasons to the user. Hence, taking constraints into
account means sufficient reasons are both not subsumed, and not constraint-subsumed by any other potential reason.

Therefore, a sufficient reason for a positive decision on $\alpha$ can be defined to be a prime implicant of $F$ that is satisfied by instance $\alpha$, and is not constraint-subsumed by any other prime implicant of $F$. More formally:

Definition 11. (Refining Definition 9) An MV-term term $\tau$ is a multi-value sufficient reason, or simply sufficient reason, for instance $\alpha$ if all of the following conditions are satisfied:
(1) $\alpha \models \tau$
(2) $\tau \in P I(\kappa \rightarrow \Delta)$
(3) there is no other prime implicant $\tau^{\prime}$ in $P I(\kappa \rightarrow \Delta)$ such that $\tau \wedge \kappa \models \tau^{\prime} \wedge \kappa$

In Example 9, the only sufficient reason is the prime implicants that is not constraint-subsumed, i.e. $\left(Y \neq y_{3} \wedge Y \neq y_{4}\right) \wedge Z=z_{3}$.

### 4.6 Discussion

In this chapter, we have extended our notion of sufficient reasons in the presence of constraints to the multi-value setting. We then focused on addressing the constraint subsumption issues that were due to the encoding of multi-value variables into Boolean variables, and took inspiration from the negative encoding of (Choi, Shih et al. 2020) to limit sufficient reasons to negative prime implicants, and thus resolve constraint subsumptions that were due to the one-hot encoding of multi-value literals.

It is important to note that without constraints, by definition, sufficient reasons are not subsumed, but in the presence of constraints, sufficient reasons may be constraint-subsumed.

In the general case, we believe that unless there is a good reason to allow constraint-subsumed reasons, by default, taking constraints into account means that reasons should not be constraintsubsumed.

In this chapter we did not investigate providing sufficient reasons for multi-value-output and multi-output classifiers, however, we showed that one can represent a multi-value classifier with a Boolean classifier. On the other hand (Sasao 1978; Coudert 1994) showed that one can reduce the minimization of both, multi-value-output and multi-output Boolean functions to the minimisation of a single-output Boolean function with added variables. This allows reusing the same apparatus from the Boolean setting in the most general case of multi-value multi-output classifiers. However, based on our preliminary analysis, some post-processing (or extra constraints that capture the nuances of the Boolean encoding) are needed in this reduction. For instance, to clean up explanations by removing the added dummy variables that are needed for the reduction, or prevent returning the dummy variables in the explanations. We will leave the investigation of multi-value multi-output classifiers to future work.

## Chapter 5

## Causal Arguments and Explanations Within a Context

There is a close connection between explanations and causality, as (Miller 2019) puts it, "good" explanations should address the causes of events.

In this chapter, we investigate the connection between causes and prime implicants, and motivate the definition of irredundant prime-covers of the decision functions of a classifier as a causal model. Selecting a causal model prior to the reasoning process allows one to reduce the number of possible explanations. This is useful in situations where there are too many competing explanations, and not all explanations are needed, or when there is a preference for providing explanations from within a certain (predefined) subset of explanations.

We formalise some concepts and notions from the causality literature, in particular, we formalise the notion of cause, actual cause and causal difference makers, and prove that difference makers track the outcome. We then illustrate the usefulness of finding difference makers with some examples and show how our definitions deal with some of the subtleties associated with causal inference. Analysing such subtleties has great value for application in sensitive domains, such as health care or legal policy, where the cause-and-effect relations between variables need to be investigated in depth, prior to making conclusions.

The results of our investigation demonstrate that our method performs as well as (Bochman 2018) compared with Structural Causal Models (Joseph Y Halpern and Pearl 2005a) when causes for one effect at a time are being modelled and analysed. A limitation of our work is that we do not model a sequence of cause-and-effect relations, and therefore cannot compare the performance of our method with some other criteria and subtleties that are studied in the literature.

### 5.1 Preliminaries

The preliminaries listed in this section are the same as the preliminaries in Chapter 3. The reader can skip to the new notion introduced in Definition 12.

We will represent the decision function of a Boolean classifier by partial Boolean functions, and causal models by certain disjunctive normal-forms of Boolean formulas that cover these functions.

Notation. We use the set-theoretic notation: subset $\subseteq$, strict-subset $\subset$, complement $\bar{X}$.

Boolean Formulas. Let $\mathbf{X}$ be a finite set of Boolean variables, say $\left\{X_{1}, X_{2}, \cdots, X_{n}\right\}$. The set of Boolean formulas is generated from $\mathbf{X}$ and the constants true and false, by the operations $\neg$ (logical negation), $\vee$ (logical disjunction), and $\wedge$ (logical conjunction). An assignment of the variables is a function $\mathbf{x}: \mathbf{X} \rightarrow\{0,1\}$. For a Boolean formula $\varphi$ and an assignment $\mathbf{x}$, define $\mathbf{x} \models \varphi$, read $\boldsymbol{x}$ satisfies $\varphi$, inductively as usual:

- $\mathbf{x} \models X_{i}$ if $\mathbf{x}_{i}=1$
- $\mathbf{x} \models \neg \varphi$ if $\mathbf{x} \not \vDash \varphi$,
- $\mathbf{x} \models \varphi_{1} \wedge \varphi_{2}$ if $\mathbf{x} \models \varphi_{i}$ for all $i=1,2$,
- $\mathbf{x} \models \varphi_{1} \vee \varphi_{2}$ if $\mathbf{x} \models \varphi_{i}$ for some $i=1,2$.

If $\Phi$ is a set of Boolean formulas, write $\Phi \models \phi$ if every assignment satisfying all the formulas in $\Phi$ satisfies $\phi$. In this case we say that $\Phi$ logically implies $\phi$, and that $\phi$ logically follows from $\Phi$. In case $\Phi$ is a singleton, we may write $\phi$ instead of the more precise $\{\phi\}$. A set $\Phi$ of formulas is consistent if $\Phi \not \vDash$ false, i.e., if there is some assignment $v$ that satisfies every formula in $\Phi$.

A literal is a variable $X_{i}$ or the negation of a variable $\neg X_{i}$. A term is a set of literals in which each variable occurs at most once (either as $X_{i}$ or $\neg X_{i}$ ). By convention, the empty term is the constant true. Implicitly, a term $t$ is the conjunction $\bigwedge_{X \in t} X$, and so $\neg t$ means $\bigvee_{X \in t} \neg X$. A formula $\varphi$ that is a disjunction of conjunctions of literals is said to be in disjunctive-normal form ( $D N F$ ). We represent DNFs as sets $\Sigma$ of terms. Implicitly, a set of terms is a disjunction
$\bigvee_{t \in \Sigma} t$ (and so $\neg \Sigma$ means $\bigwedge_{t \in \Sigma} \neg t$ ), and a set $\Sigma$ of terms represents the Boolean formula $\bigvee_{t \in \Sigma} \bigwedge_{l \in t} l$.

Partial Boolean Functions and Covers. A partial Boolean function $F$ (over $\boldsymbol{X}$ ) is a function $\{0,1\}^{n} \rightarrow\{0,1, *\}$. For $i \in\{0,1, *\}$ define $F^{i}$ to be the set $F^{-1}(i)$. The instances in $F^{1}, F^{0}$, $F^{*}$ are called, respectively, the positive, negative, undefined instances of $F$. We name these sets as follows: $F^{1}$ is the function's onset, $F^{0}$ the offset, and $F^{*}$ the don't-care set. If the set $F^{*}$ is empty, then $F$ is a total Boolean function or simply, a Boolean function. Note that in this case, the function $F$ is determined by its onset.

DEFInition 12 (Cover). (Coudert and Sasao 2002) A Boolean function $G$ covers a partial Boolean function $F$ if $F^{1} \subseteq G^{1}$ and $F^{0} \subseteq G^{0}$.

An equivalent formulation is that $F^{1} \subseteq G^{1} \subseteq F^{1} \cup F^{*}=\overline{F^{0}}$. In other words, $G^{1}$ separates $F^{1}$ from $F^{0}$.

Example 10. Consider a voting system where three people represented by $\boldsymbol{X}=\left\{X_{1}, X_{2}, X_{3}\right\}$ vote on a particular proposal. A proposal is accepted or rejected based on the majority vote. A Boolean function $G$ determines whether the proposal is accepted or rejected. Now suppose voter $X_{2}$ copies the vote of $X_{1}$, captured by the formula $\left(X_{1} \leftrightarrow X_{2}\right)$. A partial Boolean function $F$ determines the fate of the proposal in this situation. Table 5.1 provides both $G$ and the partial Boolean function $F$.

| $X_{1}$ | $X_{2}$ | $X_{3}$ | $G$ | $F$ |
| :---: | :---: | :---: | :---: | :---: |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | $*$ |
| 0 | 1 | 1 | 1 | $*$ |
| 1 | 0 | 0 | 0 | $*$ |
| 1 | 0 | 1 | 1 | $*$ |
| 1 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 |

Table 5.1. Grey rows highlight the difference between functions $G$ and $F$.

Boolean function $G$ covers partial function $F$, notice that $G^{1} \subseteq F^{1} \cup F^{*}$ and $G^{0} \subseteq F^{0} \cup F^{*}$.

Important Notation. For clarifying some arguments, we may freely swap between formula notation, set notation, and function notation. In particular, a Boolean formula $\varphi$ determines the total Boolean function $F$ such that $F(\mathbf{x})=1$ iff $\mathbf{x} \models \varphi$; every total Boolean function can be expressed by a Boolean formula, say in DNF; a Boolean formula $\varphi$ covers the partial Boolean function $F$ if $F^{1} \models \varphi$ and $F^{0} \models \neg \varphi$; a (partial or total) assignment $\mathbf{x}$ is identified with the set of literals it makes true, i.e., $\{X: \mathbf{x}(X)=1\} \cup\{\neg X: \mathbf{x}(X)=0\}$; for a term $t$ and an assignment $\mathbf{x}$ we write $\mathbf{x} \models t$ and $t(\mathbf{x})=1$ interchangeably. Finally, in the examples we will use capital letters $A, B, C$ for variables, and instead of writing $A=1$ (resp. $A=0$ ) we may write $A$ (resp. $\neg A$ ).

Irredundant Prime-Covers. The problem of finding "simple" sets $\Sigma$ of terms that cover a given partial Boolean function $F$ is sometimes called two-level logic minimisation (Coudert and Sasao 2002). Here "simple" can be instantiated in a number of ways. In this work it means that $\Sigma$ is prime and irredundant, which we now define.

An implicant of $F$ is a term $t$ such that $t^{1} \subseteq F^{1} \cup F^{*}$, i.e., $t(\mathbf{x})=1$ implies $F(\mathbf{x}) \in\{1, *\}$. An implicant of $F$ is prime if there is no proper subset of $t$ is an implicant of $F$. Write $\operatorname{PI}(F)$ for the set of all prime implicants of $F$.

A cover $\Sigma$ of $F$ is prime if $\Sigma \subseteq \operatorname{PI}(F)$. A cover $\Sigma$ of $F$ is irredundant if no strict subset $\Sigma^{\prime} \subset \Sigma$ covers $F$. A cover $\Sigma$ is irredundant prime if it is both irredundant and prime.

Note that the only irredundant prime-cover of the Boolean formula true is the cover consisting of a single term, i.e., \{true\}. In particular, the cover does not mention any literals.

In the following section we import from social science literature, the idea that irredundant prime-covers are causal models of the underlying data.

### 5.2 Irredundant Prime-Covers as Causal Models

In this section we show how to describe a dataset with a partial Boolean function and illustrate that irredundant prime-covers of such functions serve as causal models.

Consider a dataset containing all possible observations for a set of variables $\left\{X_{1}, X_{2}, \cdots, X_{n}\right\}$ that result in a specific phenomenon/outcome. Suppose the data is given as two disjoint sets $P, N \subseteq\{0,1\}^{n}$. This induces the partial Boolean function $F$ that maps $\mathbf{x} \in\{0,1\}^{n}$ as follows: $\mathbf{x} \mapsto 1$ if $\mathbf{x} \in P ; v \mapsto 0$ if $\mathbf{x} \in N$; and $\mathbf{x} \mapsto *$ otherwise. Irredundant prime-covers of such partial Boolean functions serve as causal models.

We illustrate with the following running example, taken from (Joseph Y Halpern and Pearl 2005a).

EXAMPLE 11 (Arsonists). Two arsonists drop lit matches $L_{1}, L_{2}($ so $n=2)$ in two different parts of the forest. The issue is whether the whole forest burns down.

In the first setting, called "disjunctive", it is enough that one of the lit matches is dropped to burn down the whole forest. The data is $P=\{(1,0),(0,1),(1,1)\}$ and $N=\{(0,0)\}$. The corresponding (total) Boolean function $F_{\vee}$ maps assignments in $P$ to 1 and assignments in $N$ to 0. The only irredundant prime-cover of $F_{\vee}$ is $\Sigma_{\vee}:=L_{1} \vee L_{2}$.

In the second setting, called "conjunctive", both lit matches need to be dropped to burn down the whole forest. The corresponding Boolean function is denoted $F_{\wedge}$. The only irredundant prime-cover of $F_{\wedge}$ is $\Sigma_{\wedge}:=L_{1} \wedge L_{2}$.

Note that it is sometimes convenient to denote the value of a given Boolean function by a variable. In this example, we will use the variable $B$ (for either setting).

We now discuss how one may justify that irredundant prime-covers can be considered causal models.

Covers (i.e., formulas in DNF) can be seen as (two-level) circuits. According to (Pearl 2002), "Circuits qualify as causal models because a circuit can contain the information to confirm or refute all action, counterfactual and explanatory sentences concerned with the operation of the circuit", and "logical functions (Boolean input-output relation) is insufficient for answering such queries".

By the above reasoning, any cover can be taken as a causal model, but not according to (Graßhoff and May 2001) who limit the definition to specific types of covers, i.e. those covers that are irredundant and prime. Indeed, (Graßhoff and May 2001) argue that irredundant prime-covers satisfy four basic intuitions of causation, i.e.,
(1) the principle of causal determinism which says that the same cause is always accompanied by the same effect,
(2) the principle of causality which says that if no cause is present no effect is caused,
(3) the principle of causal relevance which says that every type of cause is indispensable in at least one situation, and
(4) the principle of persistent relevance which says that a causal factor maintains its causal relevance when additional factors are taken into account.

We will not say more about such justifications and for the rest of this chapter we simply take irredundant prime-covers to be causal models. Instead, we focus on two intuitions from this literature that try and identify properties of causes.
(Baumgartner 2015) stresses that "causes are difference makers of outcomes". More formally, after defining the notions of sufficiency and necessity ( $A$ is sufficient for $E$ if and only if $A \rightarrow E$; and, $A$ is necessary for $E$ if and only if $E \rightarrow A$.) (Baumgartner 2015) defines the notion of a Boolean difference maker as follows: "A factor $A$ is a Boolean difference-maker of an outcome $E$ if, and only if, $A$ is contained in a minimally sufficient condition $A X$ of $E$ such that $A X$, in turn, is contained in a minimally necessary condition of $E$." We formalise these statements in the following definition (Definition 13 ); we call it "cause" and reserve the phrase "difference maker" for a property that we define in (Definition 14) based on the intuitions discussed in (Baumgartner 2022).

DEFINITION 13 (Cause). Let $\Sigma$ be an irredundant prime-cover of partial Boolean function $F$. A literal $l$ is a cause of $F$ wrt model $\Sigma$ if there is a term $t \in \Sigma$ containing $l$. We say $l$ is a cause of $F$ if it is a cause of $F$ wrt some irredundant prime-cover $\Sigma$ of $F$.

Example 12 (Arsonists Continued). In the disjunctive case, $L_{1}$ is a cause of $B$ (by symmetry, also $L_{2}$ is also a cause of $B$ ). Also in the conjunctive case $L_{1}$ is a cause of $B$ (by symmetry $L_{2}$ is a cause of $\left.B\right)$.

Baumgartner has another intuition: "Causes explain outcomes", and as such, a change in a cause $X$ "is associated with a change in the outcome when everything else stays the same" (Baumgartner 2022). We formalise this intuition in the following definition which we call "difference maker" (note that (Baumgartner 2015) does not formalise this and uses "Boolean difference maker" to refer to what we call "cause" in Definition 13).

Definition 14 (Difference Maker). Let $F$ be a partial Boolean function. A literal $l$ is a Difference Maker for $F$ if there is a Boolean formula $T$, called a context, such that
(1) $T \models F^{1} \rightarrow l$,
(2) $T \models F^{0} \rightarrow \neg l$,
(3) $T, F^{1}$ is consistent,
(4) $T, F^{0}$ is consistent,

The first two items express that in context $T$, the value of the literal $l$ "tracks" the outcome of $F$, and we call these the tracking conditions. The second two items express that the context is not trivial, and we call these the consistency conditions.

Note that in the special case that $F$ is a total Boolean function, then we get that $T \models F^{1} \leftrightarrow l$. The next theorem connects the two notions:

Theorem 3 (Cause implies Difference Maker). Let F be a partial Boolean function such that $F^{1}$ is consistent and $F^{0}$ is consistent. If a literal $l$ is a cause for $F$ then $l$ is a difference maker for $F$.

Proof. Assume that the literal $l$ is a cause for $F$, i.e., $l$ in a term $t$ of an irredundant prime-cover $\Sigma$ of $F$.

Recall that $\Sigma$ covers $F$ means:

$$
\begin{align*}
& F^{1} \models \bigvee_{c \in \Sigma} c  \tag{5.1}\\
& F^{0} \models \bigwedge_{c \in \Sigma} \neg c \tag{5.2}
\end{align*}
$$

and that we view terms both as sets of literals and conjunctions of literals. In particular, $t \backslash\{l\}$ is shorthand for $\bigwedge_{l^{\prime} \in t, l^{\prime} \neq l} l$.

We will show that $l$ is a difference maker for $F$. Define $T$ to be the following formula:

$$
\begin{equation*}
\left(\bigwedge_{c \in \Sigma, c \neq t} \neg c\right) \wedge(t \backslash\{l\}) . \tag{5.3}
\end{equation*}
$$

Informally, $T$ says that none of the $c \neq t$ are true and all of the terms other than $l$ in $t$ are true. Note that if $t=l$ then $T$ is simply

$$
\begin{equation*}
\left(\bigwedge_{c \in \Sigma, c \neq t} \neg c\right) . \tag{5.4}
\end{equation*}
$$

We begin by showing that the tracking conditions hold.
Combining (5.1) and (5.3) we see that $T, F^{1} \models t$ and so also $T, F^{1} \models l$ since $l$ is a conjunct in $t$. Thus, $T \models F^{1} \rightarrow l$. Combining (5.2) and (5.3), we see that $T, F^{0} \models \neg t \wedge(t \backslash\{l\})$, and so also $T, F^{0} \models \neg l$. Thus $T \models F^{0} \rightarrow \neg l$.

We now show that the consistency conditions hold.
Suppose, for a contradiction, that $F^{1}, T$ is not consistent. Thus, $F^{1} \models \neg T$. That is, every assignment that satisfies $c$ for some $c \in \Sigma$ also either (a) satisfies $c^{\prime}$ for some $c^{\prime} \in \Sigma \backslash\{t\}$ or (b) satisfies $\neg l^{\prime}$ for some $l^{\prime} \neq l$ a literal in $t$. We will show that $\Sigma \backslash\{t\}$ is a cover of $F$, thus contradicting that $\Sigma$ is irredundant. First, we show that $F^{1} \models \bigvee_{c^{\prime} \in \Sigma \backslash\{t\}} c^{\prime}$. Note that if $t=l$ then (b) disappears and (a) states exactly what we want. On the other hand, if $t \backslash\{l\}$ is non-empty, then if an assignment makes $F^{1}$ true, but does not make any $c^{\prime} \in \Sigma \backslash\{t\}$
true, then it must make $t$ true by (5.1), which is impossible by (b). Second, we show that $F^{0} \models \bigwedge_{c \in \Sigma \backslash\{t\}} \neg c$. But this is immediate since we even have that $F^{0} \models \neg c$ for every $c \in \Sigma$ by (5.2).

Suppose, for a contradiction, that $F^{0}, T$ is not consistent. Thus, $F^{0} \models \neg T$. Note that if $t=l$ then $F^{0} \models \neg T$ is impossible by (5.4). So we may assume that $t \backslash\{l\}$ is not empty. So, $F^{0} \models \bigvee_{c \in \Sigma, c \neq t} c \vee \neg(t \backslash\{l\})$. By (5.2), conclude that $F^{0} \models \neg(t \backslash\{l\})$. Thus $t \backslash\{l\} \models \neg F^{0}$, i.e., every assignment that makes $t \backslash\{l\}$ true is in $F^{1} \cup F^{*}$, i.e., $t \backslash\{l\}$ is an implicant of $F$. But this contradicts that $t$ is prime.

REMARK 2. We remark that slight variations of this definition do not work. For instance, we cannot let $T=t \backslash\{l\}$. Also, we cannot use the conditions $T \models l \models F^{1}$ and $T \models \neg l \models F^{0}$.

Example 13 (Arsonists continued). In the conjunctive setting, the literal $L_{1}$ is a differencemaker. Indeed, let $T:=L_{2}$. Then
(1) in every situation in which the second lit match is dropped and there is a fire, we must have that the first lit match is dropped;
(2) in every situation in which the second lit match is dropped and there is no fire, we must have that the first lit match is not dropped;
(3) there are situations in which the second lit match is dropped and there is a fire;
(4) there are situations in which the second lit match is dropped and there is no fire.

In the disjunctive setting, the literal $L_{1}$ is a difference-maker. Indeed, it is not hard to see that one can take $T:=\neg L_{2}$. Then
(1) in every situation in which the second lit match is not dropped and there is a fire, we must have that the first lit match is dropped;
(2) in every situation in which the second lit match is not dropped and there is no fire, we must have that the first lit match is not dropped;
(3) there are situations in which the second lit match is not dropped and there is a fire;
(4) there are situations in which the second lit match is not dropped and there is no fire.

Finally, observe that a function may have more than one irredundant prime-cover. This means that, given data, the user might have to choose one cover (perhaps using additional background knowledge), or might have to consider all covers (Spirtes 2010). We will see an instance of this in Example 6 (5.3.6).

### 5.3 Using Irredundant Prime-Covers to Infer Actual Causation and Causal Explanations

In this section we show how to do causal inference using irredundant prime-covers as causal models. To do so, we refine the definition of cause (Definition 13) to worlds.

A partial world $w$ is a consistent set of literals. In functional notation, this corresponds to the set of assignments that make $w$ true. A world is a partial world with exactly one assignment making it true, i.e., a maximally consistent set of literals (which can be viewed, equivalently, as an assignment).

In the next definition, we use the following notation. If $\Sigma$ is a cover and $w$ a partial world, write $\Sigma, w$ for the set of formulas $\left\{\bigvee_{t \in \Sigma} \bigwedge_{l \in t} l\right\} \cup\{l: l \in w\}$, i.e., all the literals in $w$ as well as the cover $\Sigma$ in DNF.

Definition 15 (Actual Causes). Let $\Sigma$ be an irredundant prime-cover of partial Boolean function $F$, and let w be a partial world such that $\Sigma, w$ is consistent. A literal $l$ is an actual cause of $F$ in $w$ wrt model $\Sigma$ if there is a term $t \in \Sigma$ such that a) $l \in t$, and b) $\Sigma, w \models t$.

Intuitively, condition a) says that $l$ is a cause of $F$ if the circumstances $t$ hold, while condition b) says that the circumstances $t$ hold in all worlds that are considered possible.

REMARK 3. A weaker definition would replace condition b) by condition b'): $w, t$ is consistent. This says that the circumstances $t$ hold in some possible world. This would allow one to distinguish "necessary actual causes" and "possible actual causes". We will not pursue this distinction further, but merely point out that if l does not satisfy our definition of "actual cause", although we will say that l is not an actual cause, under a finer definition it might still be a possible actual cause.

Example 14 (Arsonists continued). Suppose the forest burned down, and we want to understand the cause. Consider the situation that the first arsonist did not light a match, i.e., $w:=\left\{\neg L_{1}\right\}$. This rules out the conjunctive case (since $\left\{L_{1} \wedge L_{2}, \neg L_{1}\right\}$ is inconsistent). In the disjunctive case, $\left\{L_{1} \vee L_{2}, \neg L_{1}\right\}$ is consistent, its only model is $\left\{\neg L_{1}, L_{2}\right\}$, and a) $L_{2} \in \Sigma_{\vee}$, and b) $L_{2}$ is a logical consequence of $\left\{L_{1} \vee L_{2}, \neg L_{1}\right\}$. So $L_{2}$ is an actual cause, i.e., the second arsonist caused the fire.

In the rest of this section we evaluate our definition of actual cause on a number of subtle examples from the literature, taken from (Bochman 2018).

Our aim is to investigate the kind of causal claims one can make with irredundant prime-covers in some complex causal scenarios that have been discussed in the causality literature. We also provide an example of using our definition for explaining the decision function of a classifier.

### 5.3.1 Example 1: Loader - General illustration

(Hopkins and Pearl 2003) A firing squad consists of shooters $B$ and $C$. It is $A$ 's job to load $B$ 's gun, $C$ loads and fires his own gun. If shot by either $B$ or $C$, the prisoner dies $D$.

A Boolean function $F$ represents all possible observations is in Table 5.2.

| A | B | C | D |  |
| :---: | :---: | :---: | :---: | :---: |
| 0 | 0 | 0 | 0 |  |
| 0 | 0 | 1 | 1 |  |
| 0 | 1 | 0 | 0 | Irredundant prime-cover: |
| 0 | 1 | 1 | 1 | $\Sigma_{D}:=(A \wedge B) \vee C$ |
| 1 | 0 | 0 | 0 |  |
| 1 | 0 | 1 | 1 |  |
| 1 | 1 | 0 | 1 |  |
| 1 | 1 | 1 | 1 |  |

TABLE 5.2. Observations and causal models for Loader

The only irredundant prime-cover of $F$ is $\Sigma:=\{\{A, B\},\{C\}\}$. In DNF this is $(A \wedge B) \vee C$. Thus, for instance, each of $A, B$ and $C$ are causes of $F$.

Suppose the prisoner dies, and we want to understand the cause of her death. We instantiate Definition 15 in a number of partial worlds:

- Suppose $w:=\{A, \neg B, C\}$, i.e. we know that $A$ loaded the gun but only $C$ shot. Then $C$ is the only actual cause.
- Suppose $w:=\{A, C\}$, i.e., we know that $A$ loaded the gun and $C$ shot (but do not know if $B$ shot or not). Then only $C$ is an actual cause.
- Suppose $w:=\{A, B, C\}$, i.e., we know that $A$ loaded $B$ 's gun, and both $B$ and $C$ shot. Then each of $A, B$, and $C$ are actual causes.
- Suppose $w:=\{A, \neg C\}$, i.e., we know that $A$ loaded $B$ 's gun, that $C$ did not shoot, but do not observe whether $B$ shot or not. Then each of $A$ and $B$ are actual causes.
- Suppose $w:=\{A\}$, i.e., we know that $A$ loaded $B$ 's gun. Then none of $A, B$, or $C$ are actual causes.

The cause for the first partial world is discussed in (Bochman 2018) and agrees with our finding here. (Bochman 2018) reports that a Structural Equation Model yielded an incorrect cause ( $A$ ) for this example.

### 5.3.2 Example 2: Window - Overdetermination

Billy $(B)$ and Suzy $(S)$ both throw rocks at a window. The rocks strike the window at exactly the same time. The window breaks $(W)$.

Possible observations and the causal model is in Table 5.3.

We explore the scenario commonly discussed in the literature. Consider the window is broken, we have $w:=\{S, B\}$. Then each of $S$ and $B$ is an actual cause of $W$. In this example, the rocks hit the window at the same time, and the conclusion of the model meets common sense. The finding of (Bochman 2018) for this example is also that both $S$ and $B$ are the actual cause.

| B | S | W |  |
| :---: | :---: | :---: | :--- |
| 0 | 0 | 0 | Irredundant prime-cover: |
| 0 | 1 | 1 | $\Sigma_{W}:=B \vee S$ |
| 1 | 0 | 1 |  |
| 1 | 1 | 1 |  |

TAble 5.3. Observations and causal models for Window

The Structural Equation Models (Joseph Y Halpern and Pearl 2005a) usually need additional variables for dealing with examples of overdetermination.

### 5.3.3 Example 3: Backup - Early Preemption

(Hitchcock 2007) Assassin poisons Victim's coffee ( $A$ ). Victim drinks it and dies ( $D$ ). If Assassin had not poisoned the coffee, Backup would have ( $B$ ), and Victim would have died anyway.

The death of the victim is invariable in this situation. The possible observations are given in Table 5.4.

| A | B | D | Irredundant prime-cover: |  |
| :---: | :---: | :---: | :--- | :---: |
| 1 | 0 | 1 | $\Sigma_{D}:=$ true |  |
| 0 | 1 | 1 |  |  |

Table 5.4. Observations and causal models for Backup

Since there is just one irredundant prime-cover, and it has no literals, we cannot deduce any causes, actual causes, or difference makers.

The Structural Equation Model (Joseph Y Halpern and Pearl 2005a) for this example is represented using two different equations $B=\neg A$ and $D=A \vee B$. Indeed, our method does not deal with analysing actual causes for multiple events at the same time (or a sequence of events). In (Bochman 2018) both $\neg A$ and $B$ are identified as actual cause of the victim's death.

### 5.3.4 Example 4: Inevitable Shock - Switch

(McDermott 1995; Weslake 2015) Two switches are wired to an electrode. The switches are controlled by A and B respectively, and the electrode is attached to $C$. A flips her switch ( $A$ ), which forces $B$ to flip her switch $(B)$ ( $B$ has no other option). The electrode is activated and shocks $C(C)$ iff both switches are in the same position.

The truth table and the causal model is given in Table 5.5.

| A | B | C |  |
| :---: | :---: | :---: | :--- |
| 0 | 0 | 1 | Irredundant prime-cover: |
| 0 | 1 | 0 | $\Sigma_{C}:=A \vee \neg B$ |
| 1 | 1 | 1 |  |

TABLE 5.5. Observations and causal models for Inevitable Shock

Given the partial world $w:=\{A, B\}, A$ is the only actual cause of $C$. Similar conclusion is reached in (Bochman 2018).

### 5.3.5 Example 5: Purple Flame - (In)transitivity of Causation

(Menzies and Beebee 2001) Jones puts potassium salts $(P)$ into a hot fire $(F)$. Because potassium compounds produce a purple flame when heated, the flame changes to a purple colour ( $P F$ ), though everything else remains the same. Both flames ignite some flammable material ( $I$ ).

The truth table and causal model is provided in Table 5.6.

| P | F | PF | I |  |
| :---: | :---: | :---: | :---: | :---: |
| 0 | 0 | 0 | 0 | Irredundant prime-cover: |
| 0 | 1 | 0 | 1 | $\Sigma_{I}:=F$, |
| 1 | 0 | 0 | 0 | $\Sigma_{P F}:=F \wedge P$ |
| 1 | 1 | 1 | 1 |  |

Table 5.6. Observations and causal models for Purple Flame

Considering $w:=\{P, F, I\}$, the actual causes of $P F$ would be $F$ and $P$. But $P$ is not an actual cause of $I$. It should be noted that causation is transitive in regularity theories in general
(Paul 2004), however it is not transitive in the irredundant prime-cover model presented here. Indeed as we have mentioned earlier (in Example 5.3.3), irredundant prime-cover models do not deal with analysing actual causes for multiple events (i.e., we do not analyse causes of a sequence of events).

### 5.3.6 Example 6: Classifier Function

The example in Table 5.7 is taken from (Miller 2018). We use this example to demonstrate how causal difference makers can be used for explaining classifiers in AI.

| Legs | Sting | Eyes | Compound. eyes | Wings | Type |
| :---: | :---: | :---: | :---: | :---: | :---: |
| 8 | 0 | 8 | 0 | 0 | Spider |
| 6 | 0 | 2 | 1 | 2 | Beetle |
| 6 | 1 | 5 | 1 | 4 | Bee |
| 6 | 0 | 5 | 1 | 2 | Fly |

Table 5.7. Arthropods classifier

There are multiple irredundant prime-covers for each Type as displayed under columns in Table 5.8:

| $\Sigma_{\text {Spider }}$ | $\Sigma_{\text {Beetle }}$ | $\Sigma_{\text {Bee }}$ | $\Sigma_{\text {Fly }}$ |
| :--- | :--- | :--- | :--- |
| $8 L$ | $6 L \wedge 2 E$ | $S$ | $5 E \wedge 2 W$ |
| $8 E$ | $2 E \wedge 2 W$ | $4 W$ | $5 E \wedge \neg S$ |
| $\neg C$ | $2 E \wedge C$ |  |  |
| $\neg W$ | $2 E \wedge \neg S$ |  |  |

TABLE 5.8. Arthropods type irredundant prime-covers

Consider the partial world $w:=\{6 L, 5 E, 4 W\}$ and suppose we want to understand the cause of the classifier returning Bee. Then, $4 W$ is the only actual cause, and we should use the cover $4 W$ to see this.

We make a few important observations:
(1) For producing explanations, all of the irredundant prime-covers of a class might need to be evaluated.
(2) Even in the absence of a complete set of observations (i.e., given a partial world) actual causes can be identified and can be used for generating explanations for classifier decisions using a causal model of the classifier.
(3) If explanations address causal difference makers, they are generated with respect to the whole causal model of the classifier, therefore we speculate that they may help produce more robust explanations.

### 5.4 Discussion

In this chapter we motivated the definition of a cover of the decision function of a classifier as a causal model, and then formalised the intuitive notion of a difference maker condition from (Baumgartner 2015) showing that being a cause implies being a difference maker. We did not establish the converse (i.e., whether meeting the difference maker condition implies being a cause). For making actual causation claims, we generalised the notion of an instance to a partial instance in the definition of a partial world, therefore our work in this chapter enables finding explanations for partial observations (which was not discussed in the related literature (Choi, Shih et al. 2020; Ignatiev, Narodytska and Marques-Silva 2019a)).

Many of the subtleties that are studied in the causality literature are related to the issues regarding sequential causation claims. We looked at two such issues in Examples 5.3.3 and 5.3.5, but left the extension and refinements needed for modeling a sequence of cause-effect relations to future work.

The work of (Baumgartner 2015) is fundamentally based on the regularity theoretic account of causality, which is one of the several different accounts of causation (Beebee, Hitchcock and Menzies 2009). In regularity theories, causation requires a "causes before effects" relation (among some other conditions that must be satisfied), i.e., that event $B$ causally depends on event $A$ necessitates that $A$ occurred before $B$. These include Mackie's INUSconditions (Mackie 1965), and Wright's NESS theory (Wright 1985) ${ }^{1}$. Counterfactual

[^3]approaches on the other hand, take that event $B$ causally depends on event $A$ to mean that if $A$ had not occurred then $B$ would not have occurred (Lewis 1974).

Refinements to Mackie's original method, in particular, the work of (Graßhoff and May 2001) reformed the INUS approach with two insights: 1) whether a causal condition is true cannot be judged in isolation but depends on a system of causal conditions; 2 ) while the naive INUS approach understood that causal conditions should be minimal (i.e., implicants should be prime), the refined account also understood that the system of conditions itself should also be minimal (i.e., covers should be irredundant). This model was shown to perform at least on par with the counterfactual accounts (Baumgartner 2013) and was further developed in (Baumgartner and Falk 2018). It is for these reasons that we focused on irredundant prime-covers as causal models in this chapter.

Our work in this chapter leaves us with an open question: how important is the irredundancy condition in being a difference maker? More formally, is it the case that a literal $l$ is a cause (Definition 13) if it appears in some prime implicant $t$ of $F$, even if $t$ does not appear in any irredundant prime-cover of $F$ ?

A prominent account in AI of causal models, actual causation, and explanations, are Structural Equations (Joseph Y Halpern and Pearl 2005a; Joseph Y. Halpern and Pearl 2005b; Miller 2019; Miller 2018). Besides some obvious differences between this setting and ours (e.g., we currently only allow literals to be causes, rather than formulas), there are a number of important questions aimed at comparing the two settings: Do irredundant prime-covers correspond to Structural Equations in a precise sense? Are irredundant prime-covers adequate as causal theories of action and change? (McCain, Turner et al. 1997)

## ChAPTER 6

## Application of Covers and Explanations Within a Cover

Remark: A major part of this chapter was published as "A Configurational Analysis of Risk Patterns for Predicting the Outcome After Traumatic Brain Injury"(Gorji, Zador and Poon 2017) and presented at American Medical Informatics Association 2017 (AMIA’17). The paper was modified to fit the structure and style of the thesis and minor improvements were made.

Note on some special terminology used in this section: set-theoretic logical analysis method refers to the problem of finding an irredundant prime-cover of a function. A configuration is the set of literals in a prime implicant. We follow the methodology of Qualitative Comparative Analysis (QCA)(Ragin 2014b) which is a method of causality analysis that uses an irredundant prime-cover as a causal model.

QCA yields a cover whose prime implicants are weighted empirically with three different measures; consistency, raw coverage and unique coverage. We briefly explain these terms.

Paraphrasing (Ragin 2009a), consistency indicates the "degree to which instances of the outcome agree in displaying each prime implicant", and is defined as the ratio between the number of instances in the original dataset that are satisfied by the prime implicant and belong to the decided outcome class, versus the number of instances in the original dataset that are satisfied by the prime implicant.

Raw coverage "assesses the degree to which instances of the condition are paired with instances of the outcome" (Ragin 2009a) and is calculated by computing the number of instances in the original dataset that can be explained by the given prime implicant versus the number of instances in the original dataset (with the same outcome as the outcome of the
prime implicant). Unique coverage provides further information about a prime implicant's relative empirical importance (Ragin 2009a). It is calculated by subtracting from the raw coverage of a prime implicant, the proportion that is explained by any other prime implicant.

Finally, GCS stands for Glasgow Coma Scale (Marmarou et al. 2007) which is a rating for the state of a patient with a traumatic brain injury and is based on components/features such as "motor response", "pupillary reactivity" and "verbal response".

The rest of the terms are described within the introduction and body of the included work.

### 6.1 Introduction

Traumatic Brain Injury is a significant source of morbidity and mortality. TBI-related disability is quoted to be 5.3 million in the United States (Langlois and Sattin 2005) and 7.7 million in the European Union old member state (Tagliaferri et al. 2006). Furthermore, TBI affects younger population ( $<45$ years), which contributes to the devastating impact on society. Prognostic models have given increasing insight into predictor importance highlighting patient age, motor response and imaging findings as the most influential predictors of outcome3. These findings helped tailor our assessment protocols and pointed out the variables that should be gathered for clinical trials (Murray et al. 2007). In the past, TBI studies have investigated both multi-variable and single-variable models to assess the prognostic strength of variables on TBI outcome (Majdan et al. 2017; Hawley et al. 2017). Multi-variable models (Murray et al. 2007; Steyerberg et al. 2008b; Zador, Sperrin and King 2016b) focus on development and assessment of the combined effect of multiple variables on the outcome, while singlevariable approaches focus on assessing the prognostic strength of one particular variable (Marmarou et al. 2007). More recently machine-learning methods such as Bayesian Networks have been applied to TBI databases, which proved to be an appealing way to formalise intuitive as well as unexpected associations between variables (Zador, Sperrin and King 2016b). Model specification for multi-variate approach using regression, often rely on the inherent assumption that each variable has an independent effect on the outcome (Lingsma et al. 2010). It is known that these techniques assess the net effect (Ragin 2000) of a set of
variables on an outcome and are not generally concerned with configurations and interaction of variables. For understanding complex biological conditions, the interactions between variables needs to be studied in a multi-dimensional manner. In reviewing the methods that assess the effect of multiple variables on TBI outcome, we note that the mainstream techniques are marked by limitations in expressing the interactions between variables, and the role of these interactions in predicting the outcome. This means that in these techniques, the data is assumed to have just one ready answer for the magnitude of a variable's effect on the outcome. When interaction terms are not modelled adequately, the accuracy of estimates in regression approaches can be affected by model misspecification (Gordon 1968). If interaction terms are not modelled, the effect of individual independent variables are likely to be over-estimated. Modeling interactions in a multi-variate analysis is not a straightforward task. Starting from single variables, all possible combinations of variables need to be investigated. Depending on the number of variables, multiple models can be generated, and the validation of these models is non-trivial. Conventional statistical methods cannot account for situations in which only specific combinations of variables reveal their impact on the outcome (conjunctural causation) or all paths that lead to an outcome need to be simultaneously uncovered (equifinality). These methods also fall short in explaining situations in which a given combination of variables contributes to the presence of an outcome but at the same time is irrelevant for the absence of that outcome (causal asymmetry) (Ragin 2014b).

Despite the depth and breadth of recent investigations, there is limited generalised knowledge to model the complex interaction of variables and the prognostic value of these interactions in TBI. In this study our goal is to systematically investigate these interactions. While considering that the predictors of favourable outcome in TBI are not necessarily the negation or reversal of predictors of unfavourable outcome, we study the interaction of variables causative to this asymmetry, in a multi-dimensional, multi-variate manner.

Set-theoretic logical analysis methods can detect recurring causal patterns (Mahoney, Goertz and Ragin 2013), and are well suited to help us explore a configurational model of TBI outcome. For this, we apply the method of Qualitative Comparative Analysis (Ragin 2014b; Ragin 2000; Ragin 2009b) (QCA) which unlike statistical approaches, can address the three
important phenomenon of conjunctural causation, equifinality and causal asymmetry inherent in modelling the concept of configurations (Ragin 2009b). The general assumption behind the configurational approach applied here is that the interaction or combinations of different predictor variables can explain the difference in outcome classes. Hence, in comparison to statistical approach like regression that provides an estimate of impacts of the study variable on outcome in a specified model, rather QCA allows a study factor to participate in difference configurations affecting the outcome.

The chapter proceeds as follows: First we briefly cover the background on TBI and the current state of research in this area and will introduce the explanatory variables included in our study. Next, we explain the analytical framework behind our study, followed by the research design. We then present the QCA results and offer a more substantive interpretation of risk patterns before concluding the chapter with an assessment of the predictive power of the model compared to that of a simple logistic regression model followed by a discussion.

### 6.2 Prognostic Models and Predictor Variables in TBI

The International Mission for Prognosis and Analysis of Clinical Trials in TBI (IMPACT) (Hukkelhoven et al. 2005) set forth three prognostic models with different levels of complexity, using well-known predictors (age, Glasgow Coma Motor Score, and pupillary reactivity), computed tomographic characteristics (CT classification and traumatic subarachnoid hemorrhage), secondary insults (hypoxia or hypotension) and laboratory values on admission ( Hb and glucose) (Murray et al. 2007; Maas et al. 2010). These models can predict 6-month outcome in patients with severe or moderate TBI with good discriminative ability based on the Area Under Curve (AUC) (Hukkelhoven et al. 2005). Assessment and validation of these widely accepted prediction models on different cohorts has been the focus of many investigations. Externally, the IMPACT models were validated against the Corticosteroid Randomization after Significant Head Injury (CRASH) (Collaborators et al. 2008a) trial findings. The CRASH trial included 10008 cases of patients with traumatic head injury within 8 hours of clinical assessment from 239 hospitals in 29 countries.

We have based our current study on the clinically relevant variables from previous studies by IMPACT and CRASH researchers who have identified age, motor score and imaging abnormalities as important predictors of clinical outcome in $\operatorname{TBI}$ (Roozenbeek, Maas and Menon 2013; Murray et al. 2007; Zador, Sperrin and King 2016b). Study variables include demographics, injury characteristics, computed tomography (CT) findings and Glasgow Outcome Scale (GCS, motor, verbal response and eye opening). Outcome measure was dichotomised as death or severe disability at 6 months.

From the 10008 cases in the CRASH dataset, about a third had one or more missing values and were omitted from our analysis. Our analysis is therefore based on the 6945 cases that had no missing values. Table 1 describes the characteristics of patient data in the CRASH dataset. The missing CT findings were responsible majority of the excluded values in the study (2191 of the 10008 patients, $21.9 \%$ ). For the majority of these patients (2063) a CT brain was not performed at all whereas only 128 had one or more imaging findings not recorded in the dataset. We considered multiple imputations of missing data, which would technically be difficult to interface with the subsequent analysis. Furthermore, previous studies with the CRASH trial dataset found no difference between imputed and complete datasets (Steyerberg et al. 2008b). We therefore choose to undertake a complete data analysis rather than imputing missing values. Another consideration regarding the dataset was the better early outcomes (14 days) for high-income countries, compared low-middle income regions. The 6-month outcomes (used in our study) were however similar between income regions.

| Variable category | Variable (abbr.) | Category | Total |
| :---: | :---: | :---: | :---: |
| Epidemiology | Sex (sex) | Male | 5706 |
|  |  | Female | 1239 |
|  | Age (age) | $<20$ | 892 |
|  |  | 20-24 | 1191 |
|  |  | 25-29 | 860 |
|  |  | 30-34 | 754 |
|  |  | 35-44 | 1199 |
|  |  | 45-54 | 899 |
|  |  | $\geqslant 55$ | 1150 |
|  | Injury Cause (cause) | Road traffic accident | 4780 |
|  |  | Fall>2 meters | 920 |
|  |  | Other | 1245 |
|  | Major extracranial injury (ec) | Yes | 1638 |
|  |  | No | 5307 |
| Assessment | Eye opening (eye) | No response | 2680 |
|  |  | To pain | 1261 |
|  |  | To verbal stimulus | 1764 |
|  |  | Spontaneous | 1240 |
|  | Motor response (motor) | No response | 601 |
|  |  | Extension | 407 |
|  |  | Abnormal flexion | 515 |
|  |  | Withdrawal | 933 |
|  |  | Localises | 2723 |
|  |  | Follows commands | 1766 |
|  | Verbal response (verbal) | No response | 2640 |
|  |  | Incomprehensible sounds | 1124 |
|  |  | Single words | 821 |
|  |  | Confused | 2006 |
|  |  | Oriented | 354 |
|  | Pupillary response (pupils) | Both reactive | 5791 |
|  |  | No response unilateral | 496 |
|  |  | No response | 658 |
| Image findings | Petechial haemorrhage (phm) | Yes | 1974 |
|  |  | No | 4971 |
|  | Subarachnoid bleed (sah) | Yes | 2206 |
|  |  | No | 4739 |
|  | Obliterated 3rd ventricle or basal cisterns (oblt) | Yes | 1663 |
|  |  | No | 5282 |
|  | Midline shift (mdls) | Yes | 1021 |
|  |  | No | 5924 |
|  | Hematoma (hmt) | Yes | 2718 |
|  |  | No | 4227 |
| Outcome | Outcome at 6 months | Death or severe disability | 2763 |
|  |  | Moderate disability or good recovery | 4182 |

TABLE 6.1. Characteristics of patient data in the CRASH dataset

### 6.3 Qualitative Comparative Analysis

The method of Qualitative Comparative Analysis (QCA) is used for analysis of complex dependencies in configurational data (Ragin 2014b). Ragin describes QCA as "an analytic technique designed specifically for the study of cases as configurations of aspects, conceived as combinations of set memberships" (Ragin 2014b). A configuration is a combination of variables that consistently produce (i.e. are sufficient for) the outcome (Ragin 2000).

At its core, QCA is based on ideas from the field of logic synthesis (Shannon 1949) to obtain the minimal Boolean sum-of-products (SOP) formulas (in other words DNF formulas) that fully represents a given truth table of variables. The truth table lists all logically possible combinations of the variables based on the dataset included in the study. The core algorithm in QCA, the Quine-McCuskey (Quine 1952b; McCluskey 1956b) algorithm, was established in 1950s and is used for minimisation of Boolean logic formulas to find the smallest, logically valid combination of variables that have the largest coverage over the all cases under investigation.

The Quine-McCluskey algorithm like any other logical analysis method is not concerned with the empirical validity of the formulas that are being discovered. It is the role of the analyst to design a valid foundation for analysis and then to assess the empirical validity of the findings. After listing all variables in a truth table, the analyst needs to select the threshold at which sufficient evidence for the outcome is defined. For example, if the analyst wants to uncover all combinations of variables that lead to a certain outcome $85 \%$ of the time, the sufficiency score needs to be set to $85 \%$. All combinations of conditions that meet this threshold are then included in further analysis. The analyst can also define the minimum number of occurrences of a certain combination for it to be included in the study. This gives the analyst the choice to include for example all combinations that appeared at least two times for favourable outcome.

The parameters of fit in QCA are consistency and coverage (Ragin 2014b). These parameters assess how consistently a combination of conditions appears in the data and the degree to which the findings cover or explain the dataset.

Steps of Analysis. For analysis, we used fsQCA (Ragin, Drass and Davey 2006), a software developed by Ragin (Ragin 2014b) for configurational analysis. An implementation of QCA in $R$ (Thiem and Dusa 2013) was also used for replication and comparison. The steps are schematically shown below (Figure 6.1).


Figure 6.1. The 6 steps in our analysis

### 6.4 Variable Selection and Dimensionality Reduction

Since the computational cost of an exact multi-value logical analysis increases according to the number of variables included in the study, the algorithms used with these methods cannot process a large number of variables. The predictor variables in TBI dataset are nominal and multi valued. When flattened, the total number of variables in the truth table sums up to 36 (including the outcome variable,). An exact analysis of 35 variables and one outcome requires 1.8 Petabytes of memory and could not be analysed on conventional lab computers at the university. (6th Generation Intel® Core ${ }^{\mathrm{TM}} \mathrm{i} 7-6700 \mathrm{~T}$ Processor ( 8 M Cache, up to 3.60 GHz ), 12GB Memory, 1TB hard drive). For this reason, we need to select the most informative variables and consider increasing the granularity of multi-value variables by merging multiple sub-categories.

### 6.5 Results

We employed the binary decision tree algorithm RPART (Therneau, Atkinson, Ripley et al. 2010), which is an implementation of Classification and Regression Trees (Breiman, J. H. Friedman et al. 2017) (CART) in R, to identify the most informative variables and the cut off point for each multi-level variables. We pruned the resulting decision tree using two different complexity parameters ( 0.001 and 0.01 ) and evaluated the predictive power of the resulting
models based on the Area Under Curve (AUC). Table 6.2 compares the AUC of the two models with that of the original CRASH dataset. DeLong's test was used to formally compare the ROC curves for the different models.

| Model | AUC | 95\%CI (DeLong) | DeLong $\mathrm{p}^{1}$ |
| :---: | :---: | :---: | :---: |
| Original CRASH | 0.8348 | $0.8252-0.8444$ | - |
| 11-var Binarized | 0.8235 | $0.8136-0.8334$ | 0.1091 |
| 9-var Binarized | 0.8175 | $0.8073-0.8276$ | 0.01504 |

${ }^{1}$ Compared with Original Crash
TABLE 6.2. Model Evaluation

The 11-var model showed no significant different (Delong $p$ values $>0.05$ ) compared with the original model (non-binarized dataset). The alternative hypothesis was that the true difference in AUC is not equal to 0 . Even though the dataset represents the same population, the paired ROC test is not applicable for comparing the ROC curves of the binarized models with that of the original model since the models are very different and are deemed to be unpaired by the built-in glm (general linear model) algorithm in R. At AUC 0.8175 , the 9 -variable model has a higher AUC than sensitivity based (AUC 0.8149) and specificity based (AUC 0.8132) models reported in earlier studies7. The Delong p-value for the 9 -var model is less than 0.05 showing a more significant difference to the AUC of the original model compared to the 11-var model.

We test two models. The first model includes the 9 most informative variables based on the application of RPART, and the second model includes only 7 variables. Variable importance ranking for the two models is given in Table 3.

| Variables | motor | verbal | eye | pupils | age | mdls | oblt | hmt | sah | ec | phm | sex | cause |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 9-var Model | 30 | 18 | 15 | 13 | 11 | 5 | 4 | 2 | 2 | 1 | 1 | - | - |
| 7-var model | 35 | 15 | 14 | 16 | 11 | 4 | 3 | 1 | 1 | - | - | - | - |

Table 6.3. Variable importance based on RPART for two models

### 6.6 Comparative Analysis Using QCA

The first step in an exact analysis of a dataset using the configurational approach is to construct a truth table of variables. Each case in the CRASH dataset will correspond to a row in truth table. A truth table represents a binary tree in which every input variable takes either a zero or one for value. The truth table for our dataset is constructed by replacing for each label $i$ in variable $X$ the $i$ th label of $X$ with a new variable $X_{i}$. This means that multi-level variables are flattened into binary variables by expanding column wise. Given that our dataset includes 6945 cases, and in QCA terms this represents a Large-N analysis, it is unlikely that we can find perfectly sufficient causal combinations. We tested multiple levels and decided to set the sufficiency threshold (Ragin 2009b) to $70 \%$. The inclusion cut off point is kept at 1 , meaning a single occurrence of a combination is enough to include it for further analysis.

Raw coverage (RC), unique coverage (UC) and consistency (CONS) are the parameters of fit and assess how consistently a combination of conditions appears in the data and the degree to which the findings cover or explain the dataset (Ragin 2009b). The dashes (-) in the result tables mean that presence or absence of the variable does not matter for the outcome of that configuration.

### 6.7 Analysis of the 9-Var Model

As shown in Table 6.4, $67.8 \%$ of the Configurations for favourable outcome with a consistency of $84.9 \%$ could be explained by 40 combinations. The top 6 configurations for favourable outcome based on this model are reported. The first four conditions cover more cases in the dataset based on their RC and UC. Table 6.5 shows top 6 configurations for unfavourable outcome.

As shown in Table 6.5, $42.9 \%$ of the Configurations for unfavourable outcome with a consistency of $87.2 \%$ could be explained by 63 combinations. The top configurations for unfavourable outcome are reported.

|  | age | eye | motor | verbal | pupils | oblt | mdls | hmt | sah | RC | UC | CONS |
| :---: | :---: | :--- | :--- | :--- | :--- | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 1 | $<45$ | - | localises <br> or follows <br> commands | $>$ single <br> words | both <br> reactive | - | no | - | no | 0.335 | 0.028 | 0.905 |
| 2 | $<45$ | any <br> response | - | $>$ single <br> words | both <br> reactive | no | no | - | - | 0.370 | 0.014 | 0.899 |
| 3 | $<45$ | any <br> response | localises <br> or follows <br> commands | - | - | no | no | no | - | 0.318 | 0.000 | 0.896 |
| 4 | $<45$ | any <br> response | localises <br> or follows <br> commands | - | both <br> reactive | no | no | - | - | 0.414 | 0.020 | 0.885 |
| 5 | $<45$ | - | withdrawal <br> or less | - | both <br> reactive | - | no | no | yes | 0.086 | 0.015 | 0.806 |
| 6 | - | any <br> response | withdrawal <br> or less | $>$ single <br> words | - | no | no | no | no | 0.315 | 0.079 | 0.872 |

Table 6.4. Top 6 Configurations for Favourable Outcome Based on the 9-Var Model

### 6.8 Analysis of the 7-Var Model

Since variable importance ranking of hmt and sah are the lowest in the rankings of our classification tree, we removed these two variables to evaluate the resulting configurations without them. The AUC of the 7-var model is 0.811 ( $95 \%$ CI: 0.8136-0.8334 (DeLong)). At DeLong's p-value 9.474E-04 compared with the original model, the ROC curves of the two models were significantly different. It was found that $57.2 \%$ of the cases with favourable outcome with a consistency of $85.7 \%$ could be explained by 9 combinations. From these 9 configurations in Table 6.6, we report on the top 6 that have the highest RC and UC.

With a raw coverage of 0.48 , the configuration of row 1 in Table 6 explains the highest number of favourable outcomes covered by the total model ( 2740 cases), capturing the configuration "patients (below 45), with motor (localizes OR follows commands) AND pupils (both reactive) AND mdls (no) AND oblt (no)." This means that regardless of the value of eye and verbal, with $86 \%$ consistency, any configuration that matches row 1 results in favourable outcome. On the other hand, $44.5 \%$ of the cases of unfavourable outcome with a consistency of $83 \%$ could be explained by 20 combinations. Due to space limitation, we only report the top 8 configurations in Table 6.7 below.

|  | age | eye | motor | verbal | pupils | oblt | mdls | hmt | sah | RC | UC | CONS |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 1 | - | no response | withdrawal or less | Incomp. sounds or no response | no re- sponse/ unilateral | - | - | - | yes | 0.125 | 0.009 | 0.901 |
| 2 | - | no re- sponse | withdrawal or less | Incomp. sounds or no response | - | yes | - | yes | yes | 0.087 | 0.018 | 0.889 |
| 3 | - | - | withdrawal or less | Incomp. sounds or no response | - | - | yes | yes | no | 0.078 | 0.024 | 0.857 |
| 4 | <45 | no response | withdrawal or less | - | - | no | no | - | - | 0.059 | 0.001 | 0.921 |
| 5 | < 45 | - | withdrawal or less | Incomp. sounds or no response | no re- sponse/ unilateral | - | no | no | yes | 0.030 | 0.004 | 0.848 |
| 6 | <45 | any response | - | Incomp. sounds or no response | no re- <br> sponse/ unilateral | no | - | yes | - | 0.029 | 0.002 | 0.964 |

Table 6.5. Top 6 Configurations for Unfavourable Outcome Based on the 9-Var Model

With a raw coverage of 0.2 , the configuration of row 1 in Table 6.7 explains the highest number of unfavourable outcomes covered by the total model; capturing the configuration "patients who are 45 and above with eye = (no response) AND verbal = (incomprehensible sounds or no response)."

### 6.9 Predicting Outcome with QCA

To evaluate the usefulness of the 7-variable QCA model, we compared its ability to predict the TBI outcome with that of a simple binary logistic regression (Logit) model:
$P($ TBI outcome $)=\operatorname{Logit}\left(\beta_{0}+\beta_{1} *\right.$ age $+\beta_{2} *$ eye $+\beta_{3} *$ motor $+\beta_{4} *$ verbal $+\beta_{5} *$ pupils + $\beta_{6} *$ oblt $+\beta_{7}{ }^{*}$ mdls )

|  | age | eye | motor | verbal | pupils | oblt | mdls | RC | UC | CONS |
| :---: | :---: | :--- | :--- | :--- | :--- | :---: | :---: | :---: | :---: | :---: |
| 1 | $<45$ | - | localises <br> OR follows <br> commands | - | both <br> reactive | no | no | 0.483 | 0.058 | 0.860 |
| 2 | $<45$ | - | localises <br> OR follows <br> commands | at least <br> single <br> words | both <br> reactive | - | no | 0.394 | 0.037 | 0.893 |
| 3 | $<45$ | any <br> response | - | at least <br> single <br> words | both <br> reactive | no | - | 0.384 | 0.025 | 0.897 |
| 4 | $<45$ | any <br> response | localises <br> OR follows <br> commands | - | - | no | no | 0.423 | 0.003 | 0.883 |
| 5 | $<45$ | any <br> response | localises <br> OR follows <br> commands | at least <br> single <br> words | - | no | - | 0.364 | $2.3 \mathrm{E}-4$ | 0.899 |
| 6 | $<45$ | no re- <br> sponse | withdrawal <br> or less | at least <br> single <br> words | - | no | no | 0.003 | 0.003 | 0.736 |

TABLE 6.6. Top 6 Configurations for Favourable Outcome Based on the 7-Var Model
where $P$ is the predicted probability of TBI outcome based on the assumption of linear relationship between the variables. The purpose of using this simple model for comparison is to show the difference between the results of a conventional additive model with that of QCA. The two models are based on very different assumptions. The linear logistic regression model assigns a weight to all independent variables and is additive in nature. The QCA model takes patterns of interactions between variables into account and outputs multiple combinations. We compared the predictive power of the two models based on the number of true positives and false negatives they predict as well as their overall prediction accuracy. The results are shown in Table 6.8. Precision reports the percentage of correct predictions that the model makes. Recall reports the fraction of positive predictions that are truly positive. Accuracy of the model is the percentage of all true predictions from the number of predictions the model makes.

Precision: $T P /(T P+F P)$; Recall: $T P /(T P+F N)$; Accuracy: $(T P+T N) /((T P+$ $T N)+(F P+F N))$. Abbreviations $T P, F P, T N, F N$ respectively refer to True Positive, False Positive, True Negative and False Negative.

|  | age | eye | motor | verbal | pupils | oblt | mdls | RC | UC | CONS |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| 1 | $\begin{aligned} & 45 \text { and } \\ & \text { above } \end{aligned}$ | no response | - | incomp. <br> sounds <br> or no response | - | - | - | 0.200 | 0.069 | 0.832 |
| 2 | $\begin{aligned} & 45 \text { and } \\ & \text { above } \end{aligned}$ | - | - | incomp. <br> sounds <br> or no response | no re- <br> sponse/ unilateral | no | - | 0.059 | 0.009 | 0.858 |
| 3 | $\begin{aligned} & 45 \text { and } \\ & \text { above } \end{aligned}$ | - | - | incomp. <br> sounds <br> or no response | - | no | yes | 0.034 | 0.006 | 0.840 |
| 4 | - | no re- sponse | withdrawal or less | incomp. <br> sounds <br> or no response | no re- sponse/ unilateral | - | - | 0.239 | 0.043 | 0.857 |
| 5 | - | no re- sponse | withdrawal or less | incomp. <br> sounds <br> or no response | - | - | yes | 0.161 | 0.020 | 0.873 |
| 6 | $\begin{aligned} & 45 \text { and } \\ & \text { above } \end{aligned}$ | no re- sponse | - | - | no re- <br> sponse/ unilateral | no | no | 0.035 | 0.001 | 0.860 |
| 7 | 45 and above | no response | - | - | both reactive | yes | no | 0.018 | 0.001 | 0.836 |
| 8 | 45 and above | any response | withdrawal or less | - | both reactive | - | yes | 0.009 | 0.002 | 0.896 |

Table 6.7. Top 8 Configurations for Unfavourable Outcome Based on the 7-Var Model

One main difference between the two models is that the Logit model generates one model for the whole dataset, but the QCA only explains patterns in a fraction of the dataset.

| Model $^{2}$ | Precision | Recall | True Positive <br> Rate | False Positive <br> Rate |
| :--- | :--- | :--- | :--- | :--- |
| 7-var QCA favourable* | 0.86 | 0.85 | 0.858 | 0.133 |
| 7-var <br> unfavourable* | 0.83 | 0.83 | 0.830 | 0.169 |
| 7-var Logit favourable | 0.75 | 0.86 | 0.881 | 0.447 |
| 7-var Logit unfavourable | 0.73 | 0.57 | 0.553 | 0.317 |

${ }^{2} \mathrm{~N}=6945$ (Favourable outcome: 4182 Cases, Unfavourable outcome: 2763 Cases)

* For QCA (Favourable outcome: 2790 Cases, Unfavourable outcome: 1483 Cases)

TABLE 6.8. Predicting favourable outcome in TBI in CRASH dataset.

If precision and recall for the two models are calculated based on the number of cases that they claim to explain, the QCA model benefits from higher accuracy:

- Accuracy of the QCA model on the fraction of the dataset explained by the 7-var model: $(2394+1232) /(2790+1483)=0.848$
- Accuracy of the Logit model : $(3684+1527) /(4182+2763)=0.750$

However, when we evaluate the predictive power of each model on the full dataset, the Logit model demonstrates a better precision and recall than the 7-var QCA model in predicting outcome, but suffers from higher false positive rates for both cases of unfavourable and favourable outcome. These results highlight the advantages of using QCA particularly when variables that affect outcome positively do not necessarily have reverse effect when they are removed, hence enabling us to highlight the possible asymmetries in the way individual variables can influence the outcome through their participation in configurations.

For the 9-var model, the cases of favourable outcome that QCA did not cover totals 215 different combinations, and for cases of unfavourable outcome that number is 174 . For the 7 -var model, the numbers are 79 and 59 respectively. Some of these non-covered cases are single occurrences of the configuration of variables that could not be factored with other configurations.

### 6.10 Discussion

Our study demonstrated a different approach to evaluating predictors of clinical outcome in TBI. With methods of QCA we established multiple configurations for admission variables that are predictive of favourable versus unfavourable outcome. Most of the findings are intuitive, young age (<45), good neurological condition and lack of CT abnormalities are in keeping with favourable outcome. Whereas older age, poor neurological condition and CT findings such as mass effect or traumatic subarachnoid bleed are suggestive of an unfavourable outcome. These results are in line with previous studies; however an unexpected finding was that on formal variable importance ranking using RPART age fell behind the GCS components
as well as pupillary response. This is further traceable in several of the configurations (1-3 Table 6.5) for unfavourable outcome where age does not appear. A further finding in our study is the dichotomisation values for admission variables which we established using a binary decision tree algorithm (RPART). Binary adaptation of clinical features is appealing to clinicians because it simplifies patient assessment particularly in the emergency setting. We have demonstrated that collapsing multi-level variables into binary does not impact model performance when maintaining the full set or most of variables present in the original model (Table 6.2). Consequently, a binary model can potentially inform a simplified assessment protocol without substantial loss of clinical information. A translational value of our findings is that the configurations of admission variables, backed by the raw coverage, unique coverage and consistency parameters, can be regarded as "typical" patient scenarios that are strongly predictive of a clinical outcome.

## Chapter 7

## Conclusion

Machine learning models have gained considerable traction in safety critical and sensitive domains, such as healthcare, finance, transportation, and legal policy, owing to their potential to improve decision-making, enhance efficiency, and deliver innovative solutions. However, this rapid adoption has raised social and legal concerns regarding the trustworthiness and fairness of these models (GDPR, European Commission 2016). As a result, a growing field of research has emerged, focusing on the development of techniques to produce verifiable and humanly-understandable explanations for the internal decision-making rationale and outputs of these models (Ribeiro, Singh and Guestrin 2018; Ribeiro, Singh and Guestrin 2016; Lundberg and Lee 2017; Shih, Choi and Darwiche 2018; Ignatiev, Pereira et al. 2018; Ignatiev, Narodytska and Marques-Silva 2019a; Simonyan, Vedaldi and Zisserman 2013; Shrikumar, Greenside and Kundaje 2017; Koh and Liang 2017).

In this thesis we contribute to this line of research. We observed that many of the existing explanation methods do not consider background knowledge when generating explanations. In the context of ML explanations, background knowledge may include information about the relationships between input variables and the preferences of the end users of the model for explanations. Focusing on the role of background knowledge on enhancing the quality of explanations, we made a few contributions.

First, we focused on the notion of sufficient reasons and showed both theoretically and empirically that the size of sufficient reasons may be shortened by incorporating background knowledge as domain constraints into the process of generating them. In particular, in Chapter 3 and Chapter 4 we generalised certain aspects of the state of the art in formal explanation methods (Shih, Choi and Darwiche 2018) and (Ignatiev, Narodytska and Marques-Silva

2019a) for the Boolean and multi-value case by incorporating domain constraints, and showed that we can improve explanations in terms of their size. Indeed, sufficient reasons that take domain constraints into account may become more parsimonious. In addition, we remarked in Chapter 4 that one can achieve further generality by preferring non-constraint-subsumed prime implicants. We acknowledged that the two objectives - generality vs. parsimony in terms of size - may be in tension.

Then, we focused explanations to be generated from a pre-specified "context" for an explanation, enhancing explanations in terms of relevancy to the end users. Limiting the number of explanations or focusing explanations to a user-specified context is an important parallel to taking background knowledge into account. It addresses the practical applicability of the ML models as it helps to explain the model's behaviour with reference to an acceptable explanation framework from the perspective of the end users. In Chapter 5, by using irredundant prime-covers, we showed that we can narrow down explanations to a specific context. This helps to reduce the number of possible explanations while keeping explanations coherent with respect to a pre-specified model. Further, we investigated the close connection between explanations and causality and formalised some concepts and notions from the social science literature. We illustrated the usefulness of these formalised notions for making causal arguments over some canonical examples from the causality literature, and then, evaluated the performance of our proposed approach in dealing with some important subtleties discussed in the causality analysis literature. The pursuit of a technique to produce explanations based on a model that can be regarded as causal is an important endeavour. Perhaps other, and more sophisticated models of causality could be investigated in future work.

Finally, knowledge of an "original dataset" is yet another form of background knowledge that can be used to enhance the quality of explanations. We demonstrated that explanations can be backed up (and perhaps be made more trustworthy) by using some quantitative support metrics. Inspired by methods used in social science literature, in Chapter 6 we used irredundant primecovers as causal models, and measured the consistency and coverage of each explanation over the original dataset that the ML model was trained on. We provided the details of a real life application in medical domains, thus evaluated our method empirically.

In summary, ignoring background knowledge when explaining ML models, can have considerable consequences. It can lead to producing sub-optimal explanations, i.e. those that are unnecessarily detailed and hard to interpret. It may also lead to providing too many competing explanations (or seemingly random or unrelated explanations) and thus, imposing unnecessary cognitive load on the end users. These issues may restrict the practical applicability the explanation methods. Incorporating background knowledge, as we show how to do in this thesis, may help to produce explanations that are more relevant to the end users and to what they already know about the problem domain.

## Reference for Important Notations and Definitions

The purpose of this section is to provide the notations and definitions that are used in Chapter 3 and Chapter 4 in one location for easy reference. The first section includes the notations and definitions for finding explanations for Binary classifiers, and the second section extends the notations to the case of the multi-value classifiers.

## Reference for Chapter 3

$X$ : Boolean variables.
$\mathbf{X}$ : Bold capital letters denote the set of $n$ Boolean variables, i.e., $\mathbf{X}=$ $\left\{X_{1}, X_{2}, \cdots, X_{n}\right\}$.
Boolean formulas $\varphi$ : The set of Boolean formulas is generated from $\mathbf{X}$, the constants $\top$ (true) and $\perp$ (false), and the logical operations $\wedge$ (conjunction), $\vee$ (disjunction), $\neg$ (negation), $\rightarrow$ (conditional) and $\leftrightarrow$ (bi-conditional).
literal : Variables $X$ and their negations $\neg X$
Term $t$ : A conjunction of literals with no literal repeated. The empty-conjunction is also denoted $T$. The size of a term $t$ is the number of literals that occur in it.
instance (over $\boldsymbol{X}$ ) : An element of $\{0,1\}^{n}$, denoted $\mathbf{x}$ (intuitively, it is an instantiation of the variables $\mathbf{X}$ ). An instance $\mathbf{x}$ satisfies a formula $\varphi$ if $\varphi$ evaluates to true when the variables in $\varphi$ are assigned truth-values according to $\mathbf{x}$. The set of instances that satisfy the formula $\varphi$ is denoted [ $\varphi$ ], and is called the set represented by $\varphi$, i.e., a set $C$ of instances is represented by $\varphi$ if $C=[\varphi]$.

Logical entailment : For a Boolean formula $\varphi$ and an instance $\mathbf{x}$, define $\mathbf{x} \models \varphi$, read $\boldsymbol{x}$ satisfies $\varphi$, inductively as usual:

- $\mathbf{x} \models X_{i}$ if $\mathbf{x}_{i}=1$
- $\mathbf{x} \vDash \neg \varphi$ if $\mathbf{x} \nLeftarrow \varphi$,
- $\mathbf{x} \models \varphi_{1} \wedge \varphi_{2}$ if $\mathbf{x} \models \varphi_{i}$ for all $i=1,2$,
- $\mathbf{x} \models \varphi_{1} \vee \varphi_{2}$ if $\mathbf{x} \models \varphi_{i}$ for some $i=1,2$.

If $\Phi$ is a set of Boolean formulas, write $\Phi \models \phi$ if every assignment satisfying all the formulas in $\Phi$ satisfies $\phi$. In this case we say that $\Phi$ logically implies $\phi$, and that $\phi$ logically follows from $\Phi$. In case $\Phi$ is a singleton, we may write $\phi$ instead of the more precise $\{\phi\}$. A set $\Phi$ of formulas is consistent if $\Phi \not \vDash$ false, i.e., if there is some $\mathbf{x}$ that satisfies every formula in $\Phi$.
Logical equivalence : If $[\varphi]=[\psi]$ then we say that $\varphi, \psi$ are logically equivalent, intuitively, they mean the same thing.
Subsumption between terms : For terms $s, t$, we say that $s$ subsumes $t$ if $[t] \subseteq[s]$, i.e., if every instance that satisfies $t$ also satisfies $s$. If $[t] \subset[s]$ then we say that $s$ properly subsumes $t$; depending on the context, we also describe this by saying that $s$ is more general or more parsimonious than $t$, or $s$ is more succinct than $t$ (note that $s$ is smaller than $t$ ).

Partial Boolean function A partial Boolean function $F\left(\right.$ over $\boldsymbol{X}$ ) is a function $\{0,1\}^{n} \rightarrow\{0,1, *\}$. For $i \in\{0,1, *\}$ define $F^{i}$ to be the set $F^{-1}(i)$. The instances in $F^{1}, F^{0}, F^{*}$ are called, respectively, the positive, negative, undefined instances of $F$.

Total Boolean function If the set $F^{*}$ is empty, then $F$ is a total Boolean function.
If $[\varphi]=F^{1}$ we say that the formula $\varphi$ represents the total Boolean function $F$.

Implicant A term $t$ is an implicant of $F$ if $[t] \subseteq F^{1} \cup F^{*}$;
Prime implicant an implicant $t$ is prime if no other implicant of $F$ subsumes $t$. Intuitively, $t$ is prime if removing any literal from $t$ results in a term that is no longer an implicant. This generalises the notion of implicant and prime implicant from total Boolean functions, defined in (Quine 1952a; Shih, Choi and Darwiche 2018; Darwiche and Hirth 2020), to partial Boolean functions, as described in (McCluskey 1956a; Coudert 1994).

Set $P I(F)$ the set of all prime implicants of $F$.
Constraint $C$ : a set of instances over $\mathbf{X}$.
Constraint-equivalent : terms $t, s$ are $C$-equivalent (or simply, constraint-equivalent when the constraint is understood), if $C \cap[s]=C \cap[t]$. For instance, if $C$ is represented by $\left(X_{1} \vee X_{2}\right) \wedge \neg\left(X_{1} \wedge X_{2}\right)$ then $t=\neg X_{1}$ is $C$-equivalent to $s=X_{2}$, and thus $s$ and $t$ may be identified as the same reason in the presence of $C$.
Constraint-subsumption : For two terms $t, s$ say that $s$ is constraint-subsumed by $t$ if $[s] \cap C \subseteq$ $[t] \cap C$.

## Reference for Chapter 4

$X$ : Capital letters denote multi-value variables. We may use subscripts, e.g. $X_{i}$ denotes the $i^{\text {th }}$ variable.
$\mathbf{X}$ : Bold capital letters denote a finite set of multi-value variables $\left\{X_{1}, X_{2}, \cdots, X_{n}\right\}$ or $\{X, Y, \ldots, Z\}$.
$x$ : Lower case letters denote (nominal) variable values. We may also use numbers for variable values.
$D_{X}$ : Domain of variable $X$ that contains the set of values that the variable can take. In the running examples, we use two multi-value variables $X$ and $Y$ where $D_{X}=\left\{x_{1}, x_{2}, x_{3}\right\}$ and $D_{Y}=\left\{y_{1}, y_{2}, y_{3}\right\}$.
Feature space $\mathcal{U}$ : The product of the domains of the variables, i.e., $\mathcal{U}=\prod_{X \in \mathbf{X}} D_{X}$. Elements of $\mathcal{U}$ are called assignments, typically denoted $\alpha$. We may also use functional notation, i.e., $\alpha\left(X_{i}\right) \in D_{X_{i}}$.
MV-formula $\Delta$ : a formula generated from atomic predicates of the form $X=x$ where $X \in \mathbf{X}$ and $x \in D_{X}$, the constants $\top$ (true) and $\perp$ (false), and the logical operations $\wedge($ conjunction $), \vee($ disjunction $), \neg($ negation $), \rightarrow($ conditional $)$ and $\leftrightarrow$ (bi-conditional).
MV-instance $\alpha$ : An MV-instance (aka instance, aka assignment) $\alpha$ is an element of the universe $\mathcal{U}$. We may write it in functional notation, i.e, $\alpha$ is a function such that $\alpha(X) \in D_{X}$ for every $X \in \mathbf{X}$.

Logical entailment : For a MV-formula $\Delta$ and an assignment $\alpha$, define $\alpha \models \Delta$ (read $\alpha$ satisfies $\Delta$ or $\alpha$ is a model of $\Delta$ ), inductively as usual:

- $\alpha \models \mathrm{T}$,
- $\alpha \models(X=x)$ if $\alpha(X)=x$.
- $\alpha \models\left(\Delta_{1} \wedge \Delta_{2}\right)$ if $\alpha \models \Delta_{i}$ for all $i=1,2$,
- $\alpha \models \neg \Delta_{1}$ if it is not the case that $\alpha \models \Delta_{1}$.

A formula $\Delta$ is consistent if there is some assignment that satisfies $\Delta$, otherwise it is inconsistent. A set $\Phi$ of formulas is consistent if there is some assignment $\delta$ that satisfies every formula in $\Phi$.

A formula $\Delta_{i}$ logically implies another formula $\Delta_{j}$, written $\Delta_{i} \models \Delta_{j}$, iff every assignment satisfying $\Delta_{i}$ satisfies $\Delta_{j}$. If we also have $\Delta_{j} \models \Delta_{i}$ then the two formulas are logically equivalent, written $\Delta_{i} \equiv \Delta_{j}$.

We may write $[\Delta]$ for the set of assignments that satisfy $\Delta$.
A formula $\Delta$ is trivial if neither $\Delta$ nor $\neg \Delta$ is consistent.
MV-literal : A non-trivial MV-formula that mentions a single variable.
Let $D_{X}=\left\{x_{1}, x_{2}, x_{3}\right\}$. The following are literals:

- $X=x_{1}$,
- $X \neq x_{1}$.
- $X=x_{1} \vee X=x_{3}$,
- $X \neq x_{1} \wedge X \neq x_{2}$.

The following are not literals, as they are trivial:

- $X=x_{1} \vee X=x_{2} \vee X=x_{3}$,
- $X=x_{1} \wedge X \neq x_{2}$.

Intuitively, even-though a literal is an arbitrary formula, it just determines a strict subset of $D_{X}$. For instance if $D_{X}=\{0,1\}$ semantically there are 2 literals (which can be written in many different ways). E.g. $(X=1)$, $(X \neq 0)$ and $(X=1 \wedge X=1)$ are semantically the same. We emphasise that although having disjunctions in literals may seem unnatural, semantically, terms can be expressed as conjunctions of negations of atoms. Simple MV-literal : A MV-literal that specifies a single value, i.e., the cardinality of set $S$ is 1 .

MV-Term $\tau$ : A conjunction of MV-literals over distinct variables. Let $D_{X}=\left\{x_{1}, x_{2}, x_{3}\right\}$ and $D_{Y}=\left\{y_{1}, y_{2}, y_{3}\right\}$. The following is an MV-term: $\left(X=x_{1} \vee X=\right.$ $\left.x_{2}\right) \wedge\left(Y \neq y_{1}\right)$.
Simple MV-term : An MV-term is simple if all of its MV-literals are simple. The following MV-terms are simple:

- $X=x_{1} \wedge Y=y_{1}$,
- $X=x_{2} \wedge Y=y_{1}$.

The following MV-terms are not simple:

- $X \neq x_{1} \wedge Y=y_{1}$,
- $X=x_{2} \wedge\left(Y=y_{1} \vee Y=y_{3}\right)$.

Subsumption of MV-terms : Logical entailment applied to MV-terms is called subsumption.
That is, an MV-term $\tau_{i}$ subsumes MV-term $\tau_{j}$ if $\tau_{j} \models \tau_{i}$. If we also have $\tau_{i} \not \equiv \tau_{j}$, then $\tau_{i}$ strictly subsumes $\tau_{j}$. For example, the MV-term $X=x_{1} \wedge\left(Y=y_{1} \vee Y=y_{3}\right)$ is strictly subsumed by the MV-terms $X \neq x_{3} \wedge\left(Y=y_{1} \vee Y=y_{3}\right)$ and $X=x_{1}$

Moreover, when logical entailment is applied to literals we use general instead of "subsumes", i.e., if $l_{j} \models l_{i}$ we say that $l_{i}$ is more general than $l_{j}$.
$\Delta(\alpha)=1($ resp. $=0):$ the formula $\Delta$ evaluates to 1 (resp. 0 ) on instance $\alpha$.
Implicant : An MV-term $\tau$ is an implicant of MV-formula $\Delta$ if $\tau \models \Delta$.
Prime Implicant: An MV-term $\tau$ is a prime implicant of $\Delta$ if $\tau$ is an implicant of $\Delta$ that is not strictly subsumed by another implicant of $\Delta$.
$P I(\Delta)$ : the set of all prime implicants of $\Delta$.
MV-constraint $\kappa$ : an MV-formula representing a set of instances. Specifically, $\kappa$ captures background knowledge.

Constraint subsumption : Let $\kappa$ be an MV-constraint. For two terms $\tau_{i}, \tau_{j}$ we say that $\tau_{j}$ is $\kappa$ subsumed by $\tau_{i}$ if $\tau_{j} \wedge \kappa \models \tau_{i} \wedge \kappa$. If also have that $\tau_{i} \wedge \kappa \not \vDash \tau_{j} \wedge \kappa$ we say that $\tau_{j}$ is strictly $\kappa$-subsumed by $\tau_{i}$. We can think of this as inducing an order on terms: $\tau_{j} \leqslant \tau_{i}$ if $\tau_{j}$ is $\kappa$-subsumed by $\tau_{i}$.

Constraint equivalence Two terms $\tau_{i}, \tau_{j}$ that are not logically equivalent may still be logically equivalent modulo $\kappa$, i.e., each $\kappa$-subsumes the other. Note that in this case $[\kappa] \cap\left[\tau_{i}\right]=[\kappa] \cap\left[\tau_{j}\right]$.

## Bibliography

Acuña, Vicente et al. (2012). 'Algorithms and complexity of enumerating minimal precursor sets in genome-wide metabolic networks’. In: Bioinformatics 28.19, pp. 2474-2483.

Adadi, Amina and Mohammed Berrada (2018). 'Peeking inside the black-box: A survey on Explainable Artificial Intelligence (XAI)'. In: IEEE Access 6, pp. 52138-52160.
Agarwal, Sushant et al. (2021). 'Towards the unification and robustness of perturbation and gradient based explanations'. In: International Conference on Machine Learning. PMLR, pp. 110-119.

Amarú, Luca et al. (2014). 'Data compression via logic synthesis'. In: ASP-DAC. IEEE, pp. 628-633.
Audemard, Gilles, Frédéric Koriche and Pierre Marquis (2020). 'On tractable XAI queries based on compiled representations'. In: $K R$.
Baumgartner, Michael (2009). 'Inferring causal complexity'. In: Sociological methods \& research 38.1, pp. 71-101.

- (2013). ‘A regularity theoretic approach to actual causation'. In: Erkenntnis 78.1, pp. 85109.
- (2015). 'Parsimony and causality'. In: Quality \& Quantity 49.2, pp. 839-856.
- (2022). 'Qualitative Comparative Analysis and robust sufficiency'. In: Quality \& quantity 56.4, pp. 1939-1963.

Baumgartner, Michael and Christoph Falk (2018). 'Boolean difference-making: a modern regularity theory of causation'. In.

Beebee, Helen, Christopher Hitchcock and Peter Menzies (2009). The Oxford handbook of causation. Oxford University Press.
Bochman, Alexander (2018). 'Actual Causality in a Logical Setting.' In: IJCAI, pp. 17301736.

Brayton, Robert K, Gary D Hachtel et al. (1984). Logic minimization algorithms for VLSI synthesis. Vol. 2. Springer Science \& Business Media.

Brayton, Robert K, Patrick C McGeer et al. (1993). 'A new exact minimizer for two-level logic synthesis'. In: Logic Synthesis and Optimization, pp. 1-31.
Brayton, Robert K and Fabio Somenzi (1989). 'An exact minimizer for boolean relations'. In: ICCAD.

Breiman, Leo, Jerome Friedman et al. (1984). Classification and regression trees. CRC press.
Breiman, Leo, Jerome H Friedman et al. (2017). Classification and regression trees. Routledge.

Brewka, Gerhard, Ilkka Niemelä and Mirosław Truszczyński (2008). ‘Nonmonotonic reasoning'. In: Foundations of Artificial Intelligence 3, pp. 239-284.

Bryant, Randal E (1992). 'Symbolic boolean manipulation with ordered binary-decision diagrams'. In: ACM Computing Surveys (CSUR) 24.3, pp. 293-318.

Choi, Arthur, Weijia Shi et al. (2019). 'Compiling neural networks into tractable Boolean circuits'. In: AAAI VNN.

Choi, Arthur, Andy Shih et al. (2020). 'On Symbolically Encoding the Behavior of Random Forests'. In: CoRR abs/2007.01493.

Collaborators, MRC Crash Trial et al. (2008a). 'Predicting outcome after traumatic brain injury: practical prognostic models based on large cohort of international patients'. In: Bmj 336.7641, pp. 425-429.

- (2008b). 'Predicting outcome after traumatic brain injury: practical prognostic models based on large cohort of international patients'. In: BMJ 336.7641.

Coudert, Olivier (1994). ‘Two-level logic minimization: an overview'. In: Integration 17.2.

- (1995). ‘Doing Two-Level Logic Minimization 100 Times Faster.' In: SODA, pp. 112-121. Coudert, Olivier and Jean Christophe Madre (1992). 'Implicit and Incremental Computation of Primes and Essential Primes of Boolean Functions.' In.
- (1993). 'Fault tree analysis: $10^{20}$ prime implicants and beyond'. In: ARMS.

Coudert, Olivier and Tsutomu Sasao (2002). 'Logic Synthesis and Verification’. In: ed. by Soha Hassoun and Tsutomu Sasao. Springer US. Chap. 1, pp. 1-28.

Darwiche, Adnan (2011). 'SDD: A new canonical representation of propositional knowledge bases'. In: Twenty-Second International Joint Conference on Artificial Intelligence.

- (2020). ‘Three Modern Roles for Logic in AI'. In: PODS.

Darwiche, Adnan and Auguste Hirth (2020). 'On The Reasons Behind Decisions'. In: ECAI.

- (2022). 'On the (Complete) Reasons Behind Decisions'. In: Journal of Logic, Language and Information, pp. 1-26.

Darwiche, Adnan and Pierre Marquis (2002). ‘A knowledge compilation map’. In: Journal of Artif. Intell. Research 17, pp. 229-264.

De Kleer, Johan, Alan K Mackworth and Raymond Reiter (1990). 'Characterizing diagnoses'. In: International Workshop on Expert Systems in Engineering. Springer, pp. 1-15.

- (1992). ‘Characterizing diagnoses and systems'. In: Artif. Intell. 56.2-3.

Deutch, Daniel and Nave Frost (2018). ‘CEC: Constraints based explanation for classifications'. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, pp. 1879-1882.

Dua, Dheeru and Casey Graff (2017). UCI Machine Learning Repository. URL: http : //archive.ics.uci.edu/ml.
Duşa, Adrian and Alrik Thiem (2015). 'Enhancing the Minimization of Boolean and Multivalue Output Functions with e QMC’. In: The Journal of Mathematical Sociology 39.2, pp. 92-108.

GDPR, European Commission (2016). Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation).

Ghorbani, Amirata, Abubakar Abid and James Zou (2019). ‘Interpretation of neural networks is fragile'. In: AAAI. Vol. 33, pp. 3681-3688.

Ghosh, Abhijit, Srinivas Devadas and A Richard Newton (1991). 'Test generation and verification for highly sequential circuits'. In: IEEE transactions on computer-aided design of integrated circuits and systems $10.5, \mathrm{pp} .652-667$.

Golson, Steve (1993). ‘One-hot state machine design for FPGAs'. In: Proc. 3rd Annual PLD Design Conference \& Exhibit. Vol. 1. 3.

Gordon, Robert A (1968). 'Issues in multiple regression'. In: American journal of Sociology 73.5, pp. 592-616.

Gorji, Niku and Sasha Rubin (2022). 'Sufficient Reasons for Classifier Decisions in the Presence of Domain Constraints'. In: AAAI, February.
Gorji, Niku, Zsolt Zador and Simon Poon (2017). 'A configurational analysis of risk patterns for predicting the outcome after traumatic brain injury'. In: AMIA annual symposium proceedings. Vol. 2017. American Medical Informatics Association, p. 780.

Graßhoff, Gerd and Michael May (2001). ‘Causal regularities’. In: Current issues in causation, pp. 85-114.
Halpern, Joseph Y and Judea Pearl (2005a). ‘Causes and explanations: A structural-model approach. Part I: Causes'. In: The British journal for the philosophy of science 4, pp. 843887.

- (2005b). 'Causes and Explanations: A Structural-Model Approach. Part II: Explanations’. In: The British Journal for the Philosophy of Science 56.4, pp. 889-911.

Harman, Gilbert (1968). ‘Knowledge, inference, and explanation'. In: American Philosophical Quarterly 5.3, pp. 164-173.
Hawley, Carol et al. (2017). 'Traumatic brain injuries in older adults-6 years of data for one UK trauma centre: retrospective analysis of prospectively collected data'. In: Emergency medicine journal 34.8, pp. 509-516.
Hitchcock, Christopher (2007). 'Prevention, preemption, and the principle of sufficient reason'. In: The Philosophical Review 116.4, pp. 495-532.
Hopkins, Mark and Judea Pearl (2003). ‘Clarifying the usage of structural models for commonsense causal reasoning'. In: pp. 83-89.

Huang, Xuanxiang et al. (2021). ‘On efficiently explaining graph-based classifiers'. In: arXiv preprint arXiv:2106.01350.
Hukkelhoven, Chantal WPM et al. (2005). 'Predicting outcome after traumatic brain injury: development and validation of a prognostic score based on admission characteristics'. In: Journal of neurotrauma 22.10, pp. 1025-1039.

Ignatiev, Alexey, Nina Narodytska, Nicholas Asher et al. (2020). ‘From contrastive to abductive explanations and back again'. In: International Conference of the Italian Association for Artificial Intelligence. Springer, pp. 335-355.

Ignatiev, Alexey, Nina Narodytska and Joao Marques-Silva (2019a). 'Abduction-based explanations for machine learning models'. In: AAAI.

- (2019b). ‘On validating, repairing and refining heuristic ML explanations’. In: arXiv preprint arXiv:1907.02509.
Ignatiev, Alexey, Filipe Pereira et al. (2018). 'A SAT-based approach to learn explainable decision sets'. In: IJCAR. Springer.
Izza, Yacine, Alexey Ignatiev and Joao Marques-Silva (2020). 'On explaining decision trees'. In: arXiv preprint arXiv:2010.11034.

Izza, Yacine and Joao Marques-Silva (2022). 'On Computing Relevant Features for Explaining NBCs'. In: arXiv preprint arXiv:2207.04748.

Karp, Richard M (1972). 'Reducibility among combinatorial problems'. In: Complexity of computer computations. Springer, pp. 85-103.

Katz, Garrett et al. (2016). 'Imitation learning as cause-effect reasoning'. In: International Conference on Artificial General Intelligence. Springer, pp. 64-73.

Kay, Judy (2000). 'Stereotypes, student models and scrutability’. In: International Conference on Intelligent Tutoring Systems. Springer, pp. 19-30.
Kay, Judy and Bob Kummerfeld (2006). 'Scrutability, user control and privacy for distributed personalization'. In: Proceedings of the CHI2006 Workshop on Privacy-Enhanced Personalization, pp. 21-22.
Kisa, Doga et al. (2014). 'Probabilistic sentential decision diagrams: Learning with massive logical constraints'. In: ICML LTPM.

Koh, Pang Wei and Percy Liang (2017). 'Understanding black-box predictions via influence functions'. In: International conference on machine learning. PMLR, pp. 1885-1894.
Lakkaraju, Himabindu et al. (2019). 'Faithful and customizable explanations of black box models'. In: AAAI.

Langlois, Jean A and Richard W Sattin (2005). ‘Traumatic brain injury in the United States: Research and programs of the Centers for Disease Control and Prevention (CDC)-Preface'. In: Journal of Head Trauma Rehabilitation 20.3, pp. 187-188.
Lewis, David (1974). ‘Causation’. In: The journal of philosophy 70.17, pp. 556-567.
Linardatos, Pantelis, Vasilis Papastefanopoulos and Sotiris Kotsiantis (2020). 'Explainable AI: A review of machine learning interpretability methods'. In: Entropy 23.1, p. 18.

Lingsma, Hester F et al. (2010). 'Early prognosis in traumatic brain injury: from prophecies to predictions'. In: The Lancet Neurology 9.5, pp. 543-554.

Lundberg, Scott M and Su-In Lee (2017). 'A unified approach to interpreting model predictions'. In: Adv Neural Inf Process Syst. Pp. 4765-4774.

Maas, Andrew IR et al. (2010). 'IMPACT recommendations for improving the design and analysis of clinical trials in moderate to severe traumatic brain injury'. In: Neurotherapeutics 7.1, pp. 127-134.

Mackie, John L (1965). ‘Causes and conditions’. In: American philosophical quarterly 2.4, pp. 245-264.

Mahoney, James, Gary Goertz and Charles C Ragin (2013). ‘Causal models and counterfactuals'. In: Handbook of causal analysis for social research. Springer, pp. 75-90.

Majdan, Marek et al. (2017). 'Outcome prediction after traumatic brain injury: comparison of the performance of routinely used severity scores and multivariable prognostic models'. In: Journal of neurosciences in rural practice 8.01, pp. 020-029.

Marmarou, Anthony et al. (2007). 'Prognostic value of the Glasgow Coma Scale and pupil reactivity in traumatic brain injury assessed pre-hospital and on enrollment: an IMPACT analysis'. In: Journal of neurotrauma 24.2, pp. 270-280.

Marques-Silva, Joao and Alexey Ignatiev (2022). ‘Delivering Trustworthy AI through Formal XAI'. In: Proceedings of the AAAI Conference on Artificial Intelligence.

Marquis, Pierre (1991). 'Extending abduction from propositional to first-order logic'. In: Fundamentals of Artificial Intelligence Research.

McCain, Norman, Hudson Turner et al. (1997). ‘Causal theories of action and change'. In: AAAI/IAAI, pp. 460-465.

McCluskey, Edward J (1956a). 'Minimization of Boolean functions'. In: Bell Syst. 35.6, pp. 1417-1444.

- (1956b). 'Minimization of Boolean functions'. In: The Bell System Technical Journal 35.6, pp. 1417-1444.

McDermott, Michael (1995). 'Redundant causation'. In: British Journal for the Philosophy of Science, pp. 523-544.

McGeer, Patrick C et al. (1993). ‘ESPRESSO-SIGNATURE: A new exact minimizer for logic functions'. In: VLSI.

Menzies, Peter and Helen Beebee (2001). ‘Counterfactual theories of causation'. In.
Miller, Tim (2018). 'Contrastive explanation: A structural-model approach', In: arXiv preprint arXiv:1811.03163.

- (2019). 'Explanation in artificial intelligence: Insights from the social sciences'. In: Artificial Intelligence 267, pp. 1-38.

Mothilal, Ramaravind K, Amit Sharma and Chenhao Tan (2020). 'Explaining machine learning classifiers through diverse counterfactual explanations'. In: Proceedings of the 2020 conference on fairness, accountability, and transparency, pp. 607-617.

Murray, Gordon D et al. (2007). 'Multivariable prognostic analysis in traumatic brain injury: results from the IMPACT study'. In: Journal of neurotrauma 24.2, pp. 329-337.

Nelson, Raymond J (1955). 'Weak simplest normal truth functions'. In: The Journal of Symbolic Logic 20.3.
Paul, Laurie A (2004). 'Aspect causation'. In: The Journal of philosophy 97.4, pp. 235-256.
Pearl, Judea (2002). 'Reasoning with cause and effect'. In: AI Magazine 23.1, pp. 95-95.

- (2009). Causality. Cambridge university press.

Peng, Yun and James A Reggia (1990). Abductive inference models for diagnostic problemsolving. Springer Science \& Business Media.

Polikar, Robi (2012). 'Ensemble learning'. In: Ensemble machine learning. Springer, pp. 134.

Quine, Willard V (1952a). 'The problem of simplifying truth functions'. In: Am Math Mon 59.8.

Quine, Willard V (1952b). 'The problem of simplifying truth functions'. In: The American mathematical monthly 59.8, pp. 521-531.

Ragin, Charles C (2000). Fuzzy-set social science. University of Chicago Press.

- (2009a). Redesigning social inquiry: Fuzzy sets and beyond. University of Chicago Press.
- (2009b). Redesigning social inquiry: Fuzzy sets and beyond. University of Chicago Press.
- (2014a). The comparative method: Moving beyond qualitative and quantitative strategies. Univ of California Press.
- (2014b). The comparative method: Moving beyond qualitative and quantitative strategies. Univ of California Press.
Ragin, Charles C, Kriss A Drass and Sean Davey (2006). ‘Fuzzy-set/qualitative comparative analysis 2.0'. In: Tucson, Arizona: Department of Sociology, University of Arizona 23.6, pp. 1949-1955.

Reiter, Raymond (1987). 'A theory of diagnosis from first principles'. In: Artif. Intell. 32.1. Ribeiro, Marco Tulio, Sameer Singh and Carlos Guestrin (2016). "Why should I trust you?" Explaining the predictions of any classifier'. In: SIGKDD.

- (2018). 'Anchors: High-Precision Model-Agnostic Explanations'. In: AAAI.

Rihoux, Benot and Charles C Ragin (2008). Configurational comparative methods: Qualitative comparative analysis (QCA) and related techniques. Vol. 51. Sage Publications.

Roozenbeek, Bob, Andrew IR Maas and David K Menon (2013). ‘Changing patterns in the epidemiology of traumatic brain injury'. In: Nature Reviews Neurology 9.4, pp. 231-236.
Sasao, Tsutomu (1978). 'An application of multiple-valued logic to a design of programmable logic arrays'. In: Proceedings of the eighth international symposium on Multiple-valued logic, pp. 65-72.

Shannon, Claude E (1949). 'The synthesis of two-terminal switching circuits'. In: The Bell System Technical Journal 28.1, pp. 59-98.

Shih, Andy, Arthur Choi and Adnan Darwiche (2018). 'A symbolic approach to explaining bayesian network classifiers'. In: IJCAI.

Shrikumar, Avanti, Peyton Greenside and Anshul Kundaje (2017). 'Learning important features through propagating activation differences'. In: International conference on machine learning. PMLR, pp. 3145-3153.

Shrotri, Aditya A et al. (2022). 'Constraint-Driven Explanations for Black-Box ML Models'. In: Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 36. 8, pp. 83048314.

Simonyan, Karen, Andrea Vedaldi and Andrew Zisserman (2013). 'Deep inside convolutional networks: Visualising image classification models and saliency maps'. In: CoRR abs/1312.6034.

Slack, Dylan et al. (2020). 'Fooling lime and shap: Adversarial attacks on post hoc explanation methods'. In: Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society, pp. 180-186.
Sokol, Kacper and Peter Flach (2019). 'Desiderata for interpretability: explaining decision tree predictions with counterfactuals'. In: Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 33. 01, pp. 10035-10036.

- (2020). 'Explainability fact sheets: a framework for systematic assessment of explainable approaches'. In: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, pp. 56-67.
Spirtes, Peter (2010). 'Introduction to causal inference'. In: Journal of Machine Learning Research 11.May, pp. 1643-1662.

Steyerberg, Ewout W et al. (2008a). 'Predicting outcome after traumatic brain injury: development and international validation of prognostic scores based on admission characteristics'. In: PLoS medicine 5.8.

- (2008b). 'Predicting outcome after traumatic brain injury: development and international validation of prognostic scores based on admission characteristics'. In: PLoS medicine 5.8, e165.

Sundararajan, Mukund, Ankur Taly and Qiqi Yan (2017). ‘Axiomatic attribution for deep networks'. In: International conference on machine learning. PMLR, pp. 3319-3328.

Tagliaferri, Fernanda et al. (2006). 'A systematic review of brain injury epidemiology in Europe'. In: Acta neurochirurgica 148.3, pp. 255-268.
Therneau, Terry M, Beth Atkinson, Brian Ripley et al. (2010). 'rpart: Recursive partitioning'. In: R package version 3.3.8.

Thiem, Alrik and Adrian Dusa (2013). 'QCA: A Package for Qualitative Comparative Analysis.' In: R J. 5.1, p. 87.

Van Bouwel, Jeroen and Erik Weber (2002). 'Remote causes, bad explanations?' In: Journal for the Theory of Social Behaviour 32.4, pp. 437-449.
Verwer, Sicco and Yingqian Zhang (2019). 'Learning optimal classification trees using a binary linear program formulation'. In: AAAI.

Wachter, Sandra, Brent Mittelstadt and Luciano Floridi (2017). 'Why a right to explanation of automated decision-making does not exist in the general data protection regulation'. In: International Data Privacy Law 7.2, pp. 76-99.
Weslake, Brad (2015). 'A partial theory of actual causation'. In.
Wright, Richard W (1985). ‘Causation in tort law'. In: Calif. L. Rev. 73, p. 1735.
Yordanov, Boyan et al. (2016). 'A method to identify and analyze biological programs through automated reasoning'. In: NPJ Syst. Biol. Appl. 2.1.
Zador, Zsolt, Matthew Sperrin and Andrew T King (2016a). 'Predictors of outcome in traumatic brain injury: new insight using receiver operating curve indices and Bayesian network analysis'. In: PLoS one 11.7.

- (2016b). 'Predictors of outcome in traumatic brain injury: new insight using receiver operating curve indices and Bayesian network analysis'. In: PLoS one 11.7, e0158762.


[^0]:    ${ }^{1}$ Continuous data can be discretised, and discrete/categorical data can be binarized (Breiman, J. Friedman et al. 1984).
    ${ }^{2}$ Categorical variables are: Age(1-7), Eye(1-4) Motor(1-6), Verbal(1-5), Pupils(1-3). Boolean variables are: $E C, P H, O B, S A, M D, H M$.

[^1]:    ${ }^{3}$ In Chapter 4, by treating CRASH as a multi-value classifier, we show that the second reason should be returned.

[^2]:    ${ }^{1}$ This example is loosely inspired by the toy example given in (Yordanov et al. 2016).

[^3]:    ${ }^{1}$ INUS is an acronym of "Insufficient but Non-redundant part of an Unnecessary yet Sufficient condition"; NESS is an acronym of "a Necessary Element in a Sufficient Set of conditions".

