# Characterizing Formulas using Post's Lattice

**MSc Thesis** *(Afstudeerscriptie)*

written by


**Arunavo Ganguly**
(born October 4th, 2000 in Kolkata, India)


under the supervision of **Dr. Balder Ten Cate** and **Dr. Nick Bezhanishvili**, and submitted to the Examinations Board in partial fulfillment of the requirements for the degree of

**MSc in Logic**

at the *Universiteit van Amsterdam.*

**Date of the public defense:**
*August 28, 2024*

**Members of the Thesis Committee:**
Dr. Maria Aloni (Chair)
Dr. Balder Ten Cate (Co-supervisor)
Dr. Nick Bezhanishvili (Co-supervisor)
Dr. Alexandru Baltag
Dr. Victor Dalmau


INSTITUTE FOR LOGIC, LANGUAGE AND COMPUTATION

# Contents

# Acknowledgement

*"I usually solve problems by letting them devour me" -Franz Kafka, (Letter to Max Brod)*

This perfectly ordinary thesis wouldn't have been possible without the presence of some perfectly extra-ordinary people. I would like to take a 'brief' moment to convey my gratitude towards them, and all the 'brief' interactions I enjoyed during my 'brief' stay in Amsterdam.

First and foremost I would like to thank my supervisors Balder and Nick, without whom this manuscript wouldn't have come into being. Balder's brilliant insight into the subject matter and of the intersections between Computer Science, Logic and Mathematical methods helped me discover new ways of looking into problem solving. Many a times I would propose an outlandish idea, and it was only through his intellectual prowess that it took a shape of a well-formed result. I thank you for the guidance in academia and beyond, I hope we keep working together in the future.

On the other hand Nick is the first person to introduce me to the landscape of dualities. His passion for Logic in Mathematics and vice-versa was something that inspired me from the very first week of the MSL class, which was one of the best courses I ever did. If it weren't due to his efforts we wouldn't have met the Russian folks, which is by far still the most interesting academic talk I had. I thank you and I hope we continue to work in the future.

It would be criminal not to mention my delightfully quirky group of friends, namely Alyssa, who never shies away from theorizing things; Davide, the person with over-the-top pranks; Micheal, the biggest fan of David Lynch (or maybe not); Sarah, the original MoL student and Swapnil, the consistently inconsistent person. I would like to give them my heartfelt thanks for embracing my eccentricity and tolerating at my sub-par sense of humor.

I would like to thank my mother Rupa and my uncle Manas for their unwavering support throughout these years and always encouraging me to follow my heart, despite my humble background. Finally, I would like to thank my girlfriend Mayra, for the companionship and support during these two years. The world, as described by her, was complex, structured and impossible to ignore. Dissecting and putting together the world, as painted by her, made me appreciate the intricacies of the entities that surround me. Thank you for your all the conversations, and believing in me when I couldn't. I hope to continue these discussions into the future.

Last but not the least I would like to thank the city of Amsterdam, that provided me with a challenge, and also the tools to tackle it. Throughout these two years I have biked and walked through the streets of this city, trying to solve the next problem. This city taught me a meaning of a lot of things in life, and

I would be forever thankful for that.

- সবাইকে ধন্যবাদ

# Chapter 1

# Introduction

Computational Learning Theory (CLT) emerged as a prominent area of study in the latter half of the past century, one of the sub-disciplines studied under this area is Exact Learning with membership queries. This sub-discipline operates under two central assumptions: First, each object gives rise to a particular set of examples/properties it possesses, called positive examples, and a set of examples/properties it doesn't possess, called negative examples (see section 2.4). Second, each object can be uniquely identified by a subset of its positive and negative examples (see thm. 2.4.6). These sets are referred to as *unique characterizations* with respect to the target concept (see def. 2.4.5).

Investigations have been made to find the upper-bounds of these unique characterizations (see works [GK95], [RS94], [SDHK91]). In particular, cases where the object of study are Boolean functions, over a specified set of variables, have also been studied (see works [ABCS92], [BI95]). In [ABCS92], it is proven that if our objects of study consists of *all* the possible Boolean functions, over a specified set of variables, then every unique characterization must be exponential in size. The natural response to this negative result is to restrict ourselves to some particular set of Boolean formulas for finding 'better' upper-bounds. In the works [ABCS92] and [BI95] many such restrictions have been explored. This motivates the first question we address in the thesis: (1) *Precisely which sets of propositional formulas, over a specified set of variables, have 'better' upper-bounds?* Through this question what we are asking for is a complete characterization of the formulas.

The case where our objects of study are modal formulas, over a specified set of variables, have also been investigated. In a recent work by ten Cate and Koudijs [tCK24], it has been shown that if our objects are *all* the possible modal formulas over a specified set of variables, then none of the unique characterizations can be finite. This makes finite characterizations more appealing for the second question we address in our thesis: (2) *Precisely which sets of modal formulas formed from* $\{\Box, \Diamond\}$ *and set of propositional variables* $Pr$*, over a specified set of variables, have finite upper-bounds?* In the work [tCK24], it is already established that some important classes of modal formulas do have finite characterizations.

## Motivation and Outline of the Thesis

In this thesis we propose the study of propositional and modal clones to produce characterizations answering (1) and (2) (see def 3.1.10 and 4.1.11). As a consequence, the thesis splits off into two parts: propositional and modal. Clone Theory has been studied extensively in Universal Algebra (see [Lau06]), additional it has been employed in branches of Theoretical Computer Science like CSP. One of the prominent results in clone Theory is due to E. Post, who in his work [Pos41], characterized all the size 2 clones. Victor Dalmau, in his work [Dal99], provided a complete classification of boolean functions based on learnability under certain cryptographic assumptions. This serves as the initial motivation of this thesis. We follow Dalmau's proof closely to provide three complete characterizations regarding different upper-bounds of unique characterizations, namely theorems 3.2.26, 3.3.22.1 and 3.3.26. However there are two key differences from Dalmau's original proof.

First, we use fragments of propositional and modal logic instead of clones (see def. 3.1.10 and 4.1.11). In definitions 3.1.10 and 4.1.11, we have developed the motivation as of why fragments can be identified with clones. This well-known correspondence between fragments and clones facilitates the translation of many results from the theory of clones to fragments (for eg. theorem 3.1.19). The advantage we get by using fragments is that we can generalise it easily to the modal case, whereas it will be difficult to do so solely using clone theory.

Second, Dalmau used pwm-reductions, which were introduced by D. Angluin in [Ang87]. We develop a weaker version of the reductions, namely PC-reductions (see section 2.5). Building upon the three results mentioned above, we also establish three results pertaining to exact learnability, namely theorems 3.2.28, 3.3.31 and 3.3.32. Interestingly, we get back Dalmau's original result in [Dal99], minus the cryptographic assumptions, in the form of 3.2.28. Hence, the propositional part of the thesis is a partial extension of Dalmau's work [Dal99]. To close off this part, we discuss the decidibility of the characterization problems.

Similarly, the modal part of the thesis is an extension of ten Cate and Koudijs work in [tCK24]. We uplift the results from the propositional part and establish one crucial theorem 4.2.22. Finally, we are able to produce the characterizations in theorem 4.2.24 and 4.2.25, leading up to our main result 4.2.26. The results mentioned are again classification results of three particular sub-lattices of the lattice of all modal fragments, based upon whether a particular fragment admits finite characterization. Additionally, we discuss exact learnability in 4.2.31, 4.2.32 and 4.2.33. We also take into account the decidibility of these problems.

## Structure of the Thesis

In Chapter 2 we start off by introducing all the necessary mathematical structures. It is in this chapter that we introduce PC-reductions(see section 2.5) and state some results (see 2.5.3, 2.5.2, 2.5.1) on how it preserves unique characterizations.

In Chapter 3, i.e.the propositional part of the thesis firstly introduces fragments of propositional logic, followed by the correspondence of Post's Lattice with the lattice of all propositional fragments. Subsequently we closely follow Dalmau's proof to establish our first main result 3.2.26 and its learnability counterpart. Following that we provide an analysis of the propositional fragment generated by $x \oplus y \oplus z$

(see result 3.3.7.1 and 3.3.7.2). We provide the next main results 3.3.22.1, 3.3.26. We also discuss the decidibility of the unique characterization problems and some characterizations for the notions of exact learnability.

Chapter 4, i.e. the modal part, follows the same blueprint. Due to the lack of representation of the lattice of all modal fragments, we focus on three important sub-lattices of it. In the result 4.1.17, we establish that all of them are isomorphic to the Post's Lattice and hence to each other. The result 4.2.22, states that every propositional concept class is PC-reducible to some particular modal concept classes. The results pertaining to unique characterization for this part are 4.2.24, 4.2.25 and 4.2.26. We refine these results to establish their learnability counterpart 4.2.31, 4.2.32 and 4.2.33. The decidibility of the unique characterization results are again discussed.

In the conclusion we discuss the various open problems and the scope of future work.

# Chapter 2

# Preliminaries

Our goal is to view fragments of Propositional and Modal logic as concept classes, and provide classifications of them based on unique characterizations. The purpose of this chapter is to formally introduce the Mathematical structures and the relevant framework that will aid our goal.

To start with, we introduce the mathematical structures necessary for our characterizations. In the first section we gloss over various notions from order theory, building up to splitting pairs (see def. 2.1.7) and lattices (see def. 2.1.8 and 2.1.9). The following section is dedicated to the theory of Clones. After defining what clones are (see def. 2.2.4), we briefly point out the recent development of this field of study, and introduce Post's Lattice. To end the section, we look at some of the splitting pairs of the Post's Lattice that we will be using in the thesis.

The final section is used to lay down the framework of Computational Learning Theory (or CLT). We discuss various notions of Unique Characterisations (see def. 2.4.5, 2.4.7, 2.4.8, 2.4.10 and 2.4.9) and Learning (see def. 2.4.12, 2.4.13, 2.4.14, 2.4.15). We also provide some connections between them. Roughly, we discuss the following implications.

1. Exactly Learnable with membership queries $\Rightarrow$ Polynomial Sized unique characterization (see thm. 2.4.16).
2. Exactly Learnable dependent on size $\Rightarrow$ Upper bound dependent on size (see thm. 2.4.17 i.).
3. Exactly Learnable dependent on polynomial size $\Rightarrow$ Polynomial upper bound dependent on size (see thm. 2.4.17 ii.).
4. Effectively Learnable with membership queries $\Rightarrow$ Finite Characterizations (see thm. 2.4.17 iii.).
5. Computable Finite Characterizations with countably many concepts $\Rightarrow$ Effectively Learnable with membership queries (see thm. 2.4.18).

The formal introduction of Propositional and Modal fragments have been delayed to their respective chapters. Lastly, we introduce an important tool, namely PC-reductions which will be instrumental in our characterizations.

## 2.1 Partial Order and Lattices

Partial Orders are relatively simple mathematical structures, they are defined as follows:

**Definition 2.1.1** (Partial Orders)**.** *A partial ordering is a pair $\mathscr{P} = (P, \leq)$ where $P$ is a set and $\leq$ is a binary relation on $P$ which is:*

- (*Reflexive*) : $\forall x \in P(x \leq x)$
- (*Anti-symmetric*) : $\forall x \forall y[(x \leq y \wedge y \leq x) \to x = y]$
- (*Transitive*) : $\forall x \forall y \forall x[(x \leq y \wedge y \leq z) \to x \leq z]$

There are various concepts that can be developed from partial orderings, two of them are supremum and infimum.

**Definition 2.1.2** (Infimum and Supremum)**.** *Let $(P, \leq)$ be any partial order and let $S \subseteq P$. We call $e$ a lower bound on $S$ if $e \leq s$ for every $s \in S$. Furthermore, if there exists an $e_*$ s.t. $e_*$ is a lower bound of $S$ and for any other lower bound $e$ of $S$, $e \leq e_*$, then we call $e_*$ the infimum of $S$; written as $inf(S)$. An analogous definition applies to supremum.*

Of our particular interest are anti-chains. For developing anti-chains we need to introduce the concept of *incomparable* elements.

**Definition 2.1.3** (Incomparable elements)**.** *Let $\mathscr{P} = (P, \leq)$ be a partial ordering and let $a, b \in P$. We say that $a, b$ are incomparable if neither $a \leq b$ nor $b \leq a$ holds.*

The name 'partial ordering' comes from the existence of such incomparable elements in the set. Chains and anti-chains are important notions within partial Orderings.

**Definition 2.1.4** (Chain)**.** *Let $\mathscr{P} = (P, \leq)$ be a partial ordering. We call a set $C \subseteq P$ an anti-chain of $\mathscr{P}$ if for any $a, b \in C$, $a$ and $b$ are not incomparable. In other words, for every $a, b \in C$ either $a \leq b$ or $b \leq a$.*

**Definition 2.1.5** (Anti-chain)**.** *Let $\mathscr{P} = (P, \leq)$ be a partial ordering. We call a set $C \subseteq P$ an anti-chain of $\mathscr{P}$ if for any $a, b \in C$, $a$ and $b$ are incomparable. Furthermore, $C$ is a maximal anti-chain if,*

1. *$C$ is an anti-chain, and*
2. *for any $c \notin C$ there is a $c' \in C$ s.t. either $c \leq c'$ or $c' \leq c$.*

Anti-chains also give rise to splitting pairs. Intuitively, splitting pairs can be thought of as generators of the partial ordering. To establish the claim formally, we first define *upsets* and *downsets*.

**Definition 2.1.6** (Upsets and Downsets)**.** *Given a partial order $(P, \leq)$, a set $F \subseteq P$ is called an upset if for every $a, a'$ with $a \in F$ and $a \leq a'$, we have that $a' \in F$. Similarly, $D \subseteq P$ is a downset if for every $a, a'$ with $a \in D$ and $a' \leq a$, we have that $a' \in D$.*

Given a set $S$, we can talk about upset generated by the set. We define the upset, generated by a subset $S$ of a partial order $\mathscr{P} = (P, \leq)$ as:

$$\uparrow S = \{s \in P : \exists c \in S[c \leq s]\}$$

Similarly, downset generated by $S$ can be defined as:

$$\downarrow S = \{s \in P : \exists c \in S[s \leq c]\}$$

**Definition 2.1.7** (Splitting Pairs)**.** *Let $\mathscr{P} = (P, \leq)$ be a partial ordering. We call a pair of sets $(A, B)$ a splitting pair if $A \cup B$ is an anti-chain and $P = \uparrow A \cup \downarrow B$.*

Splitting pair will be a handy tool while discussing Post's Lattice. Lattices are yet another kind of mathematical structure which will be helpful in our study. We can introduce the structure in two possible ways, using order theory or algebra, both of which are equivalent.

**Definition 2.1.8** (Order Theoretic)**.** *A lattice $\mathcal{L} = (L, \leq)$, in the order theoretic sense, is a partial order s.t. for any elements $a, b$ the following two elements are well defined:*

1. $a + b := sup\{c \in L : a \leq c \wedge b \leq c\}$
2. $a \cdot b := inf\{c \in L : c \leq a \wedge c \leq b\}$

*where sup and $inf$ means supremum and infimum of a given set.*

**Definition 2.1.9** (Algebraic)**.** *A lattice $\mathcal{L} = (L, +, \cdot)$, in the order theoretic sense, is an ordered triple, where $L$ is a set and $+, \cdot$ are commutative and associative binary operations on $A$. Futhermore, both of them satisfy the following condition for any $a, b \in L$:*

1. $(a \cdot a) = a = (a + a)$
2. $[a + (a \cdot b)] = a = [a \cdot (a + b)]$

The two definitions are indeed equivalent. The order theoretic definition provides a blueprint of how to go from it to the algebraic definition. On the other hand we can do a similar thing, start from the algebraic definition and go to the order theoretic one. In a lattice $\mathcal{L} = (L, +, \cdot)$, defined in the algebraic sense, for any two elements $a, b \in L$ define

$$a \leq b \iff a + b = b$$

or equivalently $a \leq b$ if and only if $a \cdot b = a$. What we have defined on $L$ is a *partial ordering.*
The following can be easily verified.

**Proposition 2.1.10.** *The relation $\leq$ induces a partial ordering on $L$.* $\square$

We refrain from settling on a particular definition of lattice, while the order theoretic one provides more intuition, the algebraic definition is easier to use. Hence, we will be oscillating between the two definitions, but the reader can figure out the definition so used from the context.

Sometimes in a lattice there are *maximal* and *minimal* elements. *Sub-lattices* is also a concept that will be of importance to us. We define all the terms mentioned above in the following way:

**Definition 2.1.11.** *Let $t, b$ be two elements from a lattice $\mathcal{L} = (L, +, \cdot)$. We say $t$ is the top element if for any $a \in L$, we have $a \cdot t = a$ and $a + t = t$. Similarly, we say $b$ is the bottom element if for any $a \in L$ we have that $b + a = a$ and $b \cdot a = b$.*

**Definition 2.1.12.** *Given a lattice $\mathcal{L} = (L, +, \cdot)$ a sub-lattice $\mathcal{L}' = (L', +, \cdot)$ is a lattice with the following properties: firstly $L' \subseteq L$, and if $a, b \in L'$ then $a \cdot b \in L'$ and $a + b \in L'$.*

It is standard in the literature to denote the top and bottom elements of a lattice $\mathcal{L}$ by the symbols $\top_{\mathcal{L}}$ and $\perp_{\mathcal{L}}$ respectively. We will be cautious about dropping the subscript since both the symbols mentioned above also refers to standard terminologies in logic. If top and bottom elements exist in a lattice, then we call it a lattice with *end points*. The intuition behind using the terms 'top' and 'bottom' is understood better by the partial ordering counter-part of the definition.

So far we have established lattices as a special kind of partial ordering, hence concepts like anti-chains and dualities are also applicable to them. In the coming section we will harness these tools to examine in depth the structure of Post Lattice.

## 2.2 The Theory of Clones and Post's Lattice

Let us denote by $\mathbb{N}$ the set of all natural numbers, then we define $\mathbb{N}^+ = \{1, 2, 3 \dots\}$, i.e. the set of all natural numbers excluding 0. A *finitary operation* $f$, over a given set $A$, is a function of the form $f : A^n \to A$ where $n \in \mathbb{N}^+$. We define an *algebra* as:

**Definition 2.2.1** (Algebra)**.** *An algebra is an ordered pair $\mathcal{A} = (A, F)$, where $A$ is a set and $F$ is a set of finitary operations over $A$.*

For defining clones, we need the help of compositions and projections.

**Definition 2.2.2** (Projections)**.** *Let $n, k \in \mathbb{N}^+$ and $k \leq n$. Given a set $A$, by the projection map $\pi^n_k : A^n \to A$ we mean the function $\pi^n_k : (x_1, \dots x_n) \mapsto x_k$, i.e the function that maps every $n$ tuple to the $k^{th}$ element of that tuple.*

**Definition 2.2.3** (Compositions)**.** *Let $f_1, \dots f_n$ be a collection of $m$-ary functions over a set $A$, and let $f$ be a $n$-ary function over the same set. By composition of $f$ over $f_1 \dots f_n$ we mean a $m$-ary function defined in the following way:*

$$f(f_1, \dots, f_n)(x_1, \dots, x_m) := f(f_1(x_1, \dots x_m), \dots, f_n(x_1, \dots, x_m)$$

In the definition above, $f_1, \ldots, f_n$ were assumed to be the same arity, but that does not cause us to loose generality. Indeed we can use the projection map and the above definition to 'increase' the arity of any function. For an example, consider the bi-variate function $f(x, y)$. We can make it into a tri-variate function in the following way:

$$f'(x, y, z) = f(\pi_1^3(x, y, z), \pi_2^3(x, y, z))$$

'z' in the above example acts like a place holder. A similar treatment can be applied to a finite set of functions $g_1, \ldots, g_n$. Pick the function with the greatest arity and 'increase' the arity of all the others, to make all of their arity uniform.

Let $\mathcal{A} = (A, F)$ be an algebra and $n \in \mathbb{N}^+$, by $O_n^A$ we denote the class of *all* functions from $A^n$ to $A$ (there might be functions that are not in $F$). Finally, we can define the set of all finitary functions as

$$O^A = \bigcup_{n \in \mathbb{N}^+} O_n^A$$

Roughly speaking, clones are composition-closed sets of functions that contain projections. More formally, clones are defined as:

**Definition 2.2.4** (Clones). *For a set $A$, a clone is a set $C \subseteq O^A$, satisfying the two given conditions:*

1. *All the projections are in $C$, i.e. for any $n, k \in \mathbb{N}^+$ and $k \leq n$, $\pi_n^k$ is in $C$.*
2. *$C$ is closed under compositions, i.e. for any $f_1, \ldots, f_n \in (O_m^A \cap C)$ and $f \in (O_n^A \cap C)$, we have that $f(f_1, \ldots, f_n)$ is also in $C$.*

Lattices are often used to arrange mathematical objects of interest. For our purpose, we will be looking at the *lattice of all possible clones* over a given a domain $A$. Let $\mathscr{L}$ denote the set of all clones of the algebra $\mathcal{A} = (A, F)$. Given two clones $C_1, C_2$ of $\mathcal{A}$, it is easy to verify that $C_1 \cap C_2$ is also a clone of $\mathcal{A}$. In fact for any arbitrary set of clones $\{C_i : i \in I\}$, $\bigcap_{i \in I} C_i$ also forms a clone of $\mathcal{A}$. Given a set $T$ of finitary functions over $A$, we define

$$Cl(T) := \bigcap \{C \in \mathscr{L} : T \subseteq C \text{ and } C \text{ is a clone}\}$$

We call $Cl(T)$ the clone generated by $T$, which is the smallest clone containing $T$. For every clone $C, C'$ of an algebra $\mathcal{A}$, define $\sqcup(C, C') = Cl(C \cup C')$, this definition helps us to state the following result

**Theorem 2.2.5.** *For any algebra $\mathcal{A} = (A, F)$, the set $(\mathscr{L}, \sqcup, \cap)$ forms a lattice, where $\mathscr{L}$ is set of all clones of $\mathcal{A}$.*

*Proof.* Fix an algebra $\mathcal{A}$ and let $\mathcal{L}$ be set of all its clones. $\cap$ and $\delta$ are by definition commutative and associative operations on $\mathcal{L}$. We just need to verify that they follow the given two conditions of the lattice definition.

But notice that for any clone $C$, $C \cap C = C$ and $Cl(C \cup C) = Cl(C) = C$ (since $C$ is already a clone of $\mathcal{A}$). Now let $C, D$ be arbitrary clones, as $C \subseteq Cl(C \cup D)$ we can infer that

$$C \cap Cl(C \cup D) = C$$

On the other hand $Cl(C \cup (C \cap D))$ can be written as $Cl(C \cap (C \cup D))$ using De Morgan's Laws. However $C \cap (C \cup D) = C$ so,

$$Cl(C \cap (C \cup D)) = Cl(C) = C$$

since $C$ is already a clone. As $C, D$ were arbitrary, this concludes our proof. $\square$

Clone Theory as a discipline investigates algebras irrespective of their signature. Hence, for any Algebra $\mathcal{A} = (A, F)$ the purpose of clone theory is to study the domain of the algebra, namely $A$. One of the areas of study, following the theorem above, is the following: Given a set $A$, identify the lattice of all of its clones $\mathcal{L}_A$.

The following section is dedicated to the case when $|A| = 2$, which forms the heart of our thesis.

## 2.3 The Post Lattice

Whilst there is ongoing research for the cases $|A| \geq \aleph_0$ (see [GP07] and the references within), vast majority of the research focuses on the cases where $A$ is finite. Even in the finite case, the situation gets too complex even for lower cardinalities. In addition to the sheer size of the lattices when $|A| \geq 3$, there are results that state the structure of them is much intricate (see [BUL93]). In the case of $|A| \geq 4$, it has been shown that every finite product of sub-lattices of $\mathcal{L}_A$ is a sub-lattice of $\mathcal{L}_A$ (see [Bul94]).

Fortunately, the landscape when $|A| = 2$, also known as the boolean case, is a lot clearer to us. The lattice corresponding to all the clones of $|A| = 2$ (known as Boolean clones), though infinite, has been classified fully. The breakthrough work was done by Emil Post in 1941 book [Pos41], though he first announced his results in 1920s. The lattice of all Boolean clones came to be known as Post's Lattice.

Formally, Post's Lattice is the lattice of all the clones of the algebra $\mathcal{B} = (\{0, 1\}, \varnothing)$. Each of the elements in the lattice, i.e. each Boolean clone, is generated by a finite set of *boolean function.*

**Definition 2.3.1.** *A function $f$ is called a boolean function if $dom(f) = \{0, 1\}^n$ for some $n \in \mathbb{N}^+$ and $ran(f) = \{0, 1\}$.*

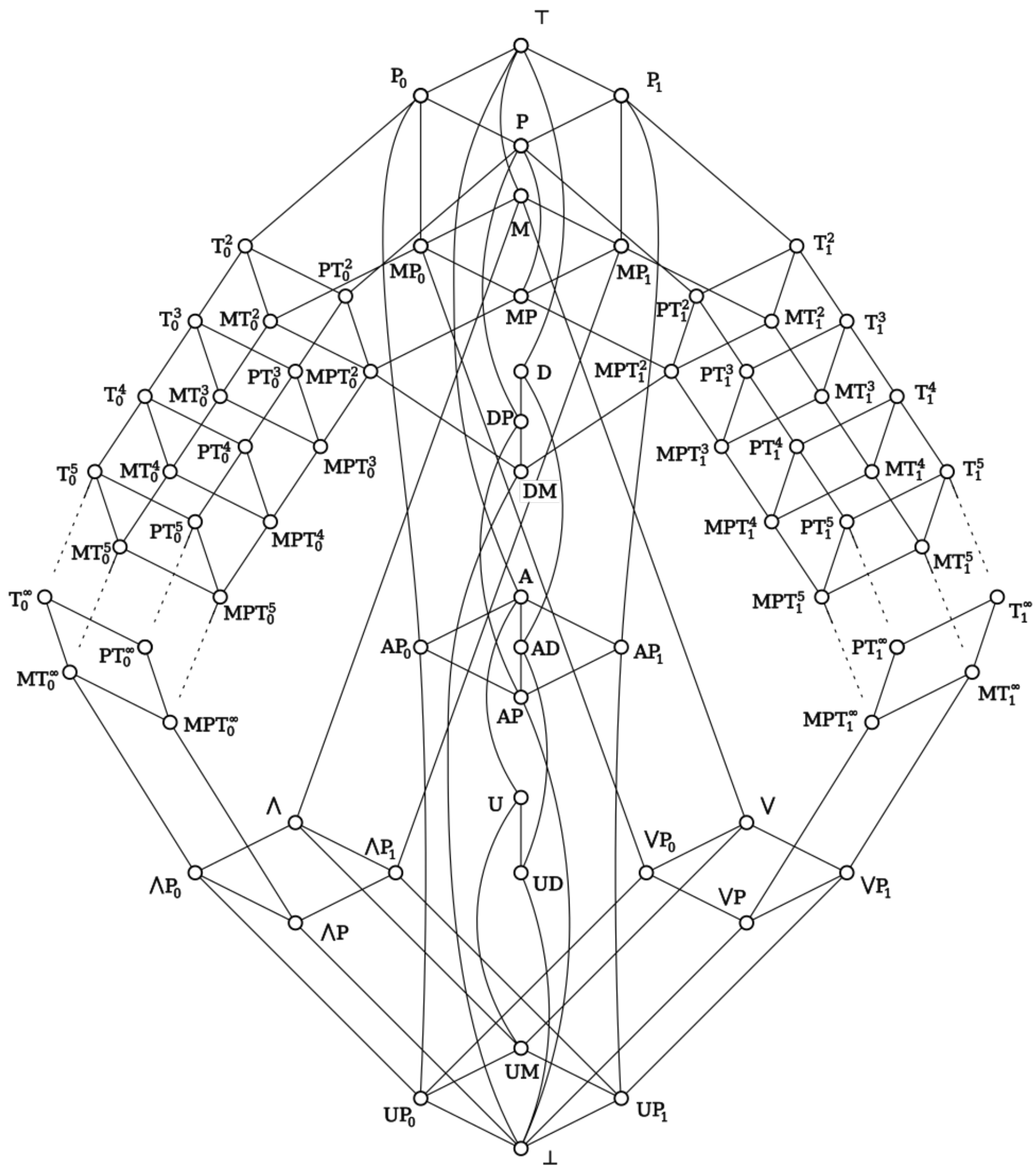


Figure 2.1: The Post Lattice

| Clones | Generating Formulas |
|---|---|
| U | $\neg, \bot$ |
| AP | $x \oplus y \oplus z$ |
| A | $x \leftrightarrow y, \top$ |
| DM | $(x \wedge y) \vee (y \wedge z) \vee (x \wedge z)$ |
| | $\wedge, \top, \bot$ |
| $\mathsf{V}$ | $\vee, \top, \bot$ |
| $\mathrm{MPT}_0^\infty$ | $x \wedge (y \vee z)$ |
| $\mathrm{MPT}_1^\infty$ | $x \vee (y \wedge z)$ |
| $\mathrm{PT}_0^\infty$ | $x \wedge (y \to z)$ |
| $\mathrm{PT}_1^\infty$ | $x \vee (y \oplus z)$ |
| M | $\wedge, \vee, \top, \bot$ |

Boolean functions can be expressed using propositional formulas. Every propositional truth table can be thought of as a 'boolean connective', indeed formally every truth table over $n$ variables is a function from $\{0,1\}^n \to \{0,1\}$. Our claim follows from one of the classic theorems in Propositional Logic.

**Theorem 2.3.2** (see [Men09])**.** *There is a propositional formula corresponding to every truth table.* $\square$

Following definition 2.3.3 of generating clone given below, we are justified in thinking that every clone is generated by a set of *propositional formulas* (to be more precise, every clone is generated by the boolean connectives corresponding to the truth tables of a given set of propositional formulas).

**Definition 2.3.3.** *Let $\mathcal{C}$ be a boolean clone. A finite set $C \subseteq \mathcal{C}$ is called it's generating set if*

$$\mathcal{C} = \bigcap \{c \in \mathscr{L}_{\mathcal{B}} : C \subseteq c\}$$

*where $\mathscr{L}_{\mathcal{B}}$ is the set of all boolean clones.*

If $C$ is the generating set of a clone $\mathcal{C}$, then we write $\mathcal{C} = Cl(C)$. The table above denotes some of the clones in Post's Lattice. These clones would be crucial in our thesis later on. Another important thing about the presented list is that some combinations of them produce splitting pairs in the Post Lattice. We introduce some splitting to us in the subsequent chapters.

$$\mathcal{S}_1 := (\{\mathrm{DM}, \mathrm{MPT}_0^\infty, \mathrm{MPT}_1^\infty\}, \{, \mathsf{V}, \mathrm{A}\})$$
$$\mathcal{S}_2 := (\{\mathrm{M}, \mathrm{PT}_0^\infty, \mathrm{PT}_1^\infty\}, \{, \mathsf{V}, \mathrm{U}\})$$
$$\mathcal{S}_3 := (\{\mathrm{P}_0, \mathsf{V}\mathrm{P}_1\}, \{\mathrm{MP}, \mathrm{U}\})$$
$$\mathcal{S}_4 := (\{\mathrm{AP}, \mathrm{MPT}_0^\infty, \mathrm{MPT}_1^\infty\}, \{, \mathsf{V}, \mathrm{U}\})$$

The final remark we want to make before moving forward is that the problem of deciding whether a particular boolean connective belong to a clone is decidable.

**Theorem 2.3.4** (see [Vol09])**.** *Given a boolean connective $f$ and a finite class of boolean connectives $C$, it is decidable whether $f \in Cl(C)$ or not.* $\square$

## 2.4 Computational Learning Theory and Concept Classes

We now focus on developing the framework we work with in this thesis, namely Computational Learning Theory. CLT is a relatively recent branch of study in Artificial Intelligence and focuses on a computer's ability to 'learn' from examples. The questions we examine falls under the umbrella of CLT, in this section we introduce the reader to the type of inquiry we are aiming towards and how it binds to Algebra and Logic.

CLT is based upon the premise that *concepts* can be distinguished using *examples*. We start off by defining what concept classes are.

**Definition 2.4.1** (Concept classes)**.** *A concept class is an ordered triple $\mathcal{C} = (C, E, \lambda)$ where:*

- *$C$ is a set, which we call the concept space.*
- *$E$ is another set which we call the example space.*
- *$\lambda : C \to \mathcal{P}(E)$ is a function that maps each concept to a set of examples.*

In addition to examples we often use labelled examples. Since we distinguish concepts based on examples, we emphasis on *equivalent concepts*.

**Definition 2.4.2** (Equivalent concepts)**.** *Let $\mathcal{C} = (C, E, \lambda)$ be a concept class. We call $c, c' \in C$ equivalent, denoted as $c \cong c'$, if $\lambda(c) = \lambda(c')$.*

**Definition 2.4.3** (Labelled examples)**.** *Let $\mathcal{C} = (C, E, \lambda)$ be a concept class. The set of labelled examples $E^l$ of $\mathcal{C}$ is the set $E^l = E \times \{+, -\}$.*

**Definition 2.4.4.** *Suppose $\mathcal{C} = (C, E, \lambda)$ is a concept class and let $T \subseteq E^l$. We say a concept $c \in C$ fits $T$ if the following two conditions holds:*

- $(t, +) \in T \Rightarrow t \in \lambda(c)$
- $(t, -) \in T \Rightarrow t \notin \lambda(c)$

We often call labelled examples of the form $(t, +)$ as *positive examples* and $(t, -)$ as *negative examples*. It is easy to see that two non-equivalent concepts are distinguishable if there exists a labelled example that distinguishes them. In other words we have that $c \ncong c'$, for some concepts $c, c'$ in a concept class $\mathcal{C} = (C, E, \lambda)$, if there is some example $t$ s.t. $t \in \lambda(c)$ and $t \notin \lambda(c')$. As evident in the previous discussion, distinction between various concepts within a concept class is dictated by examples. Therefore it is worthwhile to look for a representative set of examples given a concept $c$.

**Definition 2.4.5** (Unique characterization)**.** *Suppose $\mathcal{C} = (C, E, \lambda)$ is a concept class, we say that a set $T \subseteq E^l$ uniquely characterizes a concept $c \in C$ if*

- *$c$ fits $T$ and*
- *for any concept $c'$ that fits $T$ we have that $c \cong c'$.*

It is easy to see that every concept $c$ of any given concept class $\mathcal{C} = (C, E, \lambda)$ has at-least one unique characterization; we just consider the set $\lambda(c)$.

**Theorem 2.4.6.** *For every concept $c$, of any given concept class $\mathcal{C} = (C, E, \lambda)$, there exists at-least one unique characterization.*

The interesting part is, as the reader might guess, there might be multiple unique characterizations. In the following paragraph will illustrate two examples.

Consider the concept class $\mathcal{C}(\mathbb{N}) = (\mathbb{N}, \mathcal{O}, \lambda_{\mathbb{N}})$, where $\mathbb{N}$ is the set of all natural numbers, $\mathcal{O}$ is the set of all examples of the form '$< n$', where $n$ is again a natural number. Finally, $\lambda_{\mathbb{N}}$ is defined the following way:

$$\text{`} < n \text{'} \in \lambda_{\mathbb{N}}(s) \iff s < r$$

Notice that, following the previous theorem, for each concept $c$, there is an unique characterization of (countably) infinite many examples. But again every natural number, $n$ say, has a unique characterization of size 2! Consider this set $\{(\text{`} < n+1 \text{'}, +), (\text{`} < n-1 \text{'}, -)\}$. Hence we are interested in such 'better' unique characterizations.

We introduce some general notions of unique characterizations that we will adapt for our purposes in the later chapters.

**Definition 2.4.7** (Finite Characterizations)**.** *We a concept class $\mathcal{C} = (C, E, \lambda)$ has finite characterization if for every concept $c \in C$ we have an unique characterization $T_c$ of $c$ s.t. $T_c$ is finite.*

In the definitions below, $|E|$ will denote the size of the example space $E$, but in a liberal sense. The reader should think of $|E|$ as map from $E$ to $\mathcal{N}$ s.t. for every example space $E, E'$

$$E \subseteq E' \Rightarrow |E| \leq |E'|$$

On top of that, the notions we introduce are about *collection of concept classes*, instead of a particular concept class. The choice is justified as follows: Let $\mathcal{C}$ be a finite concept class (i.e. the concept space and the example space is finite). In this case $\mathcal{C}$ will always have a polynomial bounding the size of unique characterizations. Hence, the study of one (finite) concept class regarding polynomial bounds is trivial.

**Definition 2.4.8** (Polynomial sized Unique characterization)**.** *Let $\mathcal{X}$ be a collection of concept classes. We say $\mathcal{X}$ has Polynomial sized unique characterization (or PSUC) if there exists a polynomial $p(x, y)$ s.t. for every concept class $\mathcal{C} = (C, E, \lambda)$ in the collection $\mathcal{X}$ and every concept $c \in C$, there is an unique characterization $T_c$ of the concept $c$ s.t. $|T_c| \leq p(|c|, |E|)$.*

**Definition 2.4.9** (Upper bound dependent on concept size)**.** *Let $\mathcal{X}$ be a collection of concept classes. We say $\mathcal{X}$ has an upper dependent only on concept size if There exists a function $f(x)$ s.t. for every concept class $\mathcal{C} = (C, E, \lambda)$ in the collection $\mathcal{X}$ and every concept $c \in C$, there is an unique characterization $T_c$ of the concept $c$ s.t. $|T_c| \leq f(|c|)$.*

**Definition 2.4.10** (Polynomial upper bound dependent on concept size)**.** *Let $\mathcal{X}$ be a collection of concept classes. We say $\mathcal{X}$ has a polynomial upper dependent only on concept size if there exists a polynomial $p(x)$ s.t. for every concept class $\mathcal{C} = (C, E, \lambda)$ in the collection $\mathcal{X}$ and every concept $c \in C$, there is an unique characterization $T_c$ of the concept $c$ s.t. $|T_c| \leq p(|c|)$.*

The definition 2.4.9 and 2.4.10 will be adapted in the form of 'upper-bound dependent on variables' and 'polynomial upper-bound dependent on variables' in the chapter where we analyse Propositional Logic. To define the notions of learnability that we use, we need the concept of membership queries.

**Definition 2.4.11** (Membership Queries)**.** *Let $c$ be a concept from the concept class $\mathcal{C} = (C, E, \lambda)$. A membership query $\mathsf{MQ}_c(x)$ is an oracle that takes in any example $e$ from $E$ and labelled it according to $c$, i.e. $\mathsf{MQ}_c(e) = +$ if and only if $e \in \lambda(c)$, otherwise $\mathsf{MQ}_c(e) = -$.*

**Definition 2.4.12** (Exactly Learnable with membership queries)**.** *Let $\mathcal{X}$ be a collection of concept classes. We say $\mathcal{X}$ is exactly learnable with membership queries if there exists an algorithm $\mathsf{Alg}(x)$ and a polynomial $p(x, y)$ s.t. for every concept class $\mathcal{C} = (C, E, \lambda) \in \mathcal{X}$, and for every concept $c \in C$, if the algorithm has access to the membership query $\mathsf{MQ}_c(x)$, then the algorithm $\mathsf{Alg}(|E|)$, on input $|E|$, halts with at-most $p(|c|, |E|)$ many steps and returns a concept $c'$ s.t. $c \cong c'$.*

**Definition 2.4.13** (Exactly Learnable dependent on size)**.** *Let $\mathcal{X}$ be a collection of concept classes. Fix a concept $d$ from any concept class $\mathcal{C} = (C, E, \lambda) \in \mathcal{X}$. We say $\mathcal{X}$ is exactly learnable dependent on size if there exists an algorithm $\mathsf{Alg}(x)$ and a function $f(x)$ s.t. for every concept class $\mathcal{C} = (C, E, \lambda) \in \mathcal{X}$ and concept $c$ of $\mathcal{C}$, if the algorithm has access to the membership query $\mathsf{MQ}_c(x)$, then the algorithm $\mathsf{Alg}(|E|)$, on input $|E|$, eventually halts and returns a concept $c'$ s.t. $c \cong c'$, and for every concept class $\mathcal{C} = (C, E, \lambda) \in \mathcal{X}$ with $d \in C$, if the algorithm has access to the membership query $\mathsf{MQ}_d(x)$, then the algorithm $\mathsf{Alg}(|E|)$, on input $|E|$, halts with at-most $f(|d|)$ many steps and returns a concept $d'$ s.t. $d \cong d'$,*

**Definition 2.4.14** (Exactly Learnable dependent on polynomial size)**.** *Let $\mathcal{X}$ be a collection of concept classes. Fix a concept $d$ from any concept class $\mathcal{C} = (C, E, \lambda) \in \mathcal{X}$. We say $\mathcal{X}$ is exactly learnable with polynomial dependent on size if there exists an algorithm $\mathsf{Alg}(x)$ and a function $p(x)$ s.t. for every concept class $\mathcal{C} = (C, E, \lambda) \in \mathcal{X}$ and concept $c$ of $\mathcal{C}$, if the algorithm has access to the membership query $\mathsf{MQ}_c(x)$, then the algorithm $\mathsf{Alg}(|E|)$, on input $|E|$, eventually halts and returns a concept $c'$ s.t. $c \cong c'$, and for every concept class $\mathcal{C} = (C, E, \lambda) \in \mathcal{X}$ with $d \in C$, if the algorithm has access to the membership query $\mathsf{MQ}_d(x)$, then the algorithm $\mathsf{Alg}(|E|)$, on input $|E|$, halts with at-most $p(|d|)$ many steps and returns a concept $d'$ s.t. $d \cong d'$.*

**Definition 2.4.15** (Effectively Learnable with membership queries)**.** *We say a concept class $\mathcal{C} = (C, E, \lambda)$ is effectively learnable with membership queries if there exists an algorithm $\mathsf{Alg}(x)$ s.t. for every concept $c \in C$, if the algorithm has access to the membership query $\mathsf{MQ}_c(x)$, then the algorithm $\mathsf{Alg}(|E|)$, on input $|E|$, halts and returns a concept $c'$ s.t. $c \cong c'$.*

We now establish how the notions of unique characterization(s) and exact learnability we developed are interconnected.

**Theorem 2.4.16.** *For every collection of concept classes $\mathcal{X}$, if $\mathcal{X}$ is exactly learnable with membership queries then it has polynomial sized unique characterization.*

*Proof.* Suppose that $\mathcal{X}$ is exactly learnable with membership queries. therefore there is an algorithm $\mathsf{Alg}(x)$ and polynomial $p(x)$ that witness it. We define the following set:

$$\mathcal{K}_c = \{(e, \mathsf{MQ}_c(e)) : \mathsf{Alg}(|E|) \text{ invokes } \mathsf{MQ}_c(e) \text{ while running on input } |E|\}$$

Now the algorithm $\mathsf{Alg}(x)$ inquires the membership query $\mathsf{MQ}_c(x)$, for any $c$, at most $p(|c|, |E|)$ many times. Hence, $|\mathcal{K}_c| \leq p(|c|, |E|)$. It is easy to verify that $\mathcal{K}_c$ is an unique characterization. □

The same holds for other notions as we will state now. We skip the proofs of these theorems since they are essentially the same.

**Theorem 2.4.17.** *For every collection of concept classes $\mathcal{X}$, and concept class $\mathcal{C}$, the following holds,*

*i. If $\mathcal{X}$ is exactly learnable dependent on size then $\mathcal{X}$ has upper bound dependent on concept size.*

*ii. If $\mathcal{X}$ is exactly learnable polynomial dependent on size then $\mathcal{X}$ has polynomial upper bound dependent on concept size.*

*iii. If $\mathcal{C}$ is effectively learnable then $\mathcal{C}$ has finite characterization.* □

In chapter 3, we will prove that for propositional fragments the converse of *i.* and *ii.* for theorem 2.4.17 holds. It is also true that the converse of *iii.* in the above theorem holds under some special restrictions.

**Theorem 2.4.18.** *Let $\mathcal{C} = (C, E, \lambda)$ be a concept class where $|C| \leq |\mathbb{N}|$ and the concept class has finite characterization. Furthermore, for each $c \in C$, there is an computable finite characterization of $c$. Under these assumptions, $\mathcal{C}$ is effectively learnable.*

*Proof.* Let $\mathcal{C} = (C, E, \lambda)$ satisfying all the constraints in the theorem. Enumerate all the concepts of $C$ as follows:

$$c_1, c_2, \ldots c_n, \ldots$$

Now there should exist an algorithm, $\mathsf{Alg}_1(x)$ say, which on input $c_i$ should output an unique characterization $T_{c_i}$ of $c_i$. Now we define an algorithm $\mathsf{Alg}$ as follows.

$\mathsf{Alg}$ goes through each of the concept $c_i$ and simulates the working of $\mathsf{Alg}_1(c_i)$ and obtains the unique characterization $T_{c_i}$. Following that, for each $(t, q) \in T_{c_i}$ asks its accessible membership query whether $\mathsf{MQ}_c(t) = q$ or not. If above constraint is satisfied for every $(t, q) \in T_{c_i}$, then the algorithm $\mathsf{Alg}$ halts, otherwise it moves onto $c_{i+1}$. It is evident that for some $j$, $c_j = c$ and the machine will eventually halt. □

We conclude this chapter here, the following chapter will be focused on introducing an important tool, namely PC-reductions, that will crucial for our analysis. The subsequent chapters will formally introduce Propositional and Modal Logic as concept classes, and that is where our analysis will take place.

## 2.5 PC Reductions

Inspired from [AK95], in this section we introduce a specific type of reduction which will be crucial for our study. The main aim of these reductions, which we will call PC reductions, is to preserve unique characterization, modulo polynomial blowup. Upon a closer look, the reader might recognize that these reductions are indeed a special case of pwm-reductions, as used in [AK95, Dal99]. The reason we propose an alternate reduction is due to the scope of this thesis, PC reductions preserve less things (unlike pwm-reductions which preserve learnability with membership queries) and hence are easier to define between concept classes.

Before defining the reduction formally, we need some definitions. Let $\mathcal{C} = (C, E, \lambda)$ be a concept class then by $\mathcal{N}_\mathcal{C}$ we mean the following set:

$$(c, t) \in \mathcal{N}_\mathcal{C} \iff c \text{ is a concept in } C,\ t \text{ is a labelled example s.t. } c \text{ fits } \{t\}.$$

The next idea is that of a *critical set* of a concept class. Let $\mathcal{C} = (C, E, \lambda)$ be a concept class, we say $\mathsf{Cr}(\mathcal{C}) \subseteq E$ is a critical set if for any concepts $c, c' \in C$ we have an $e \in \mathsf{Cr}(\mathcal{C})$ distinguishing them, i.e $e \notin \lambda(c) \cap \lambda(c')$ for any concepts $c, c'$.

Now, let $\mathcal{C}_1 = (C_1, E_1, \lambda_1)$ and $\mathcal{C}_2 = (C_2, E_2, \lambda_2)$ be two concept classes. We say $\mathcal{C}_1$ is pc-reducible to $\mathcal{C}_2$, written as $\mathcal{C}_1 \leq_{pc} \mathcal{C}_2$, if there exists $f, g$ and $p$ such that

1. $f : C_1 \to C_2$ is a function mapping concepts to concepts and $h : E_1 \rightharpoonup E_2$ be a partial function with $\mathsf{Cr}(\mathcal{C}_1) \subseteq dom(h)$, and for any concept $c \in C_1$ and any example $e \in E_1$, if $e$ is a positive (negative) example of $c$ then $h(e)$ is a positive (negative) example of $f(c)$.

2. $g : \mathcal{N}_{\mathcal{C}_2} \rightharpoonup \mathcal{P}(E_2^l)$ is a partial map, mapping concepts and their fitting examples to a subset of labelled examples s.t. for every $(c, t) \in \mathcal{N}_{\mathcal{C}_2}$ if $c \in ran(f)$ then $(c, t) \in dom(g)$. Additionally $g$ should satisfy that for any $c \in C_2$ and for $t \in E_2^l$, if $(f(c), t) \in \mathcal{L}_{\mathcal{N}_2}$ then $c$ should fit $g(f(c), t)$.

3. $p$ is a non-decreasing polynomial in two variables s.t. for any $(c, t) \in \mathcal{L}_{\mathcal{C}_2}$, the size of $g(c, t)$ is bounded by $p(|c|, |E_2|)$, i.e $|g(c, t)| \leq p(|c|, |E_2|)$.

4. If $c \in C_1$ be such that $c$ fit $g(f(k), t)$, for some $(f(k), t) \in \mathcal{L}_{\mathcal{C}_2}$, then $f(c)$ fit $t$.

Sometimes our $h$ will be entire functions i.e. $dom(h) = E_1$, in those cases we do not explicitly mention the existence of critical set. To see that PC Reductions indeed preserve polynomial sized unique characterization, we first fully state what it means to have Polynomial Sized Unique Characterization (PSUC for short). Remember that by $\mathbb{C}(\mathcal{L}_O[Prop])$ we denote a collection of concept classes instead of one particular instance. Our generalization would extend that concept to all concept classes.

We say a collection (or class) of concept classes $\mathcal{X}$ has polynomial sized unique characterization if there is a non-decreasing polynomial $p(x, y)$ s.t. for any $\mathcal{C} = (C, E, \lambda) \in \mathcal{X}$ and any concept $c$ of $\mathcal{C}$, we have that $c$ has a unique characterization with at-most $p(|c|, |E|)$ many examples. Similarly, we say a collection of concept classes $\mathcal{X}$ has finite sized unique characterization if for each concept class $\mathcal{C} \in \mathcal{X}$ and each $c \in \mathcal{C}$ there is a finite sized unique characterization for $c$. If $\mathcal{X} = \{\mathcal{C}_1\}$ then we just say $\{\mathcal{C}_1\}$ has PSUC (and accordingly, finite characterization).

In the previous definition $|c|$ stands for 'size of the concept' and $|E|$ stands for 'size of the example space'. We do not formally define what 'size' and 'length' should be, it because of the abstract nature of concept classes. But for our purposes, if we fix the set of formulas $O$ and list of propositional variables $Prop$ then for $\mathbb{C}(\mathcal{L}_O[Prop])$ 'size of the concept' is the connectives used in the formula and 'size of the example space' is $|Prop|$.

Now we begin to uncover the properties of PC reductions. The next proposition is esay to infer.

**Proposition 2.5.1.** *Let $\mathcal{C} = (C, E, \lambda)$ and $\mathcal{D} = (D, E, \lambda)$ be two concept classes s.t. $C \subseteq D$. It follows that $\mathcal{C} \leq_{pc} \mathcal{D}$.*

*Proof.* Our choice of functions are the following. Take $f$ to be the identity map from $C$ to $D$. Furthermore, $g((d,t)) = t$ whenever $(d,t) \in \mathcal{N}_{\mathcal{C}}$. Finally, $h$ again the identity map and $p(x) = 1$, the constant polynomial. The reader can verify our choices of $f, g, h$ and $p$ is a PC-reduction from $\mathcal{C}$ to $\mathcal{D}$. $\square$

Secondly, PC reductions are supposed to preserve polynomial sized unique characterization. Instead of proving the above, we prove something stronger.

**Theorem 2.5.2.** *Let $\mathcal{X}, \mathcal{Y}$ be two classes consisting of concept classes. Fix positive integers $k, k', m$ and a polynomial $p(x,y)$. Suppose for each $\mathcal{C}_1 = (C_1, E_1, \lambda_1) \in \mathcal{X}$, there is a $\mathcal{C}_2 = (C_2, E_2, \lambda_2) \in \mathcal{Y}$ s.t. there are functions $f, g, h$ satisfying the following:*

1. *$f, g, h, p$ witnesses a PC reduction from $\mathcal{C}_1$ to $\mathcal{C}_2$.*
2. *$|E_2| \leq m|E_1|$*
3. *for any concepts $c_1 \in C_1$ and $|f(c_1)| \leq k|c_1|$*

*then if $\mathcal{Y}$ has polynomial sized unique characterization then so does $\mathcal{X}$.*

*Proof.* Fix $k, k', m$. Pick any concept class $\mathcal{C}_1 = (C_1, E_1, \lambda_1) \in \mathcal{X}$, and assume there is $\mathcal{C}_2 = (C_2, E_2, \lambda_2)$ and $f, g, h, p$ which satisfies the above properties. Let $q$ witness $\mathcal{X}$'s polynomial sized unique characterization.

Pick a concept $c \in C_1$, and let $T$ be a polynomial sized unique characterization of $f(c)$. We know $|T| \leq q(|f(c)|, |E_2|)$, . Consider the set

$$S = \bigcup \{g(f(c), t) \mid t \in T\}$$

We claim $S$ uniquely characterizes $c$. It is evident from definition of PC reduction itself that $c$ fits $S$, since $c$ fits each of $g(f(c), t)$. Now, suppose $c' \in C_1$ also fits $S$, which again from the definition gives us $f(c')$ fits $T$. Since $T$ was an unique characterization, we can conclude $f(c) \cong f(c')$, now if $c \ncong c'$ then without the loss of generality we can assume that there is an example $e$ s.t. $c$ fits $e$ but $c'$ does not. Therefore, $h(e)$ is a positive example of $f(c)$ but a negative example of $f(c')$ which is contradiction, since they are equivalent. Hence, we conclude $c \cong c'$.

We have proven $S$ is a unique characterization, now we put a bound to it's size. Notice that for each $t \in T$,

$$g(f(c), t) \leq p(|f(c)|, |E_2|) \leq p(k|c| + k'|E_1|, m|E_1|)$$

Now, as

$$|T| \leq q(|f(c)|, |E_2|) \leq p(k|c| + k'|E_1|, m|E_1|)$$

using the union bound lemma we can see that $|S| \leq p(k|c| + k'|E_1|, m|E_1|) \cdot q(k|c| + k'|E_1|, m|E_1|)$ which is a bi variate polynomial in $c, E_1$. Since $\mathcal{C}_1, c$ was arbitrary, we can conclude that $\mathcal{X}$ has polynomial sized unique characterization property, whenever the conditions in the theorems are satisfied. $\square$

With minor changes in the original proof we can establish the two corollaries given below. Suppose $\mathcal{X} = \{\mathcal{C}_1\}$ and $\mathcal{Y} = \{\mathcal{C}_2\}$, then using the above theorem we have the next two corollaries:

**Corollary 2.5.2.1.** *Let $\mathcal{C}_1 = (C_1, E_1, \lambda_1)$, $\mathcal{C}_2 = (C_2, E_2, \lambda_2)$ be two concept classes. Suppose there are $k, m \in \mathbb{N}$ and a non-decreasing polynomial $p(x, y)$ s.t.*

1. *$|e_2| \leq k|e_1|$ for any $e_1 \in E_1$ and $e_2 \in E_2$*
2. *for any concepts $c_1 \in C_1$ and $c_2 \in C_2$, we have $|c_2| \leq m|c_1|$*
3. *There are $f, g, h$ s.t. $\mathcal{C}_1 \leq_{pc} \mathcal{C}_2$ and $f, g, h, p$ witnesses the reduction.*

*then if $\mathcal{C}_2$ has polynomial sized unique characterization then so does $\mathcal{C}_1$.* $\square$

**Corollary 2.5.2.2.** *Let $\mathcal{C}_1 = (C_1, E_1, \lambda_1)$, $\mathcal{C}_2 = (C_2, E_2, \lambda_2)$ be two concept classes. Suppose there is constant $p > 0$ s.t. there are $f, g, h$ . $\mathcal{C}_1 \leq_{pc} \mathcal{C}_2$ and $f, g, h, p$ witnesses the reduction. Then if $\mathcal{C}_2$ has finite unique characterization then so does $\mathcal{C}_1$.* $\square$

As it is the case with the last corollary, even the general case requires less constraints if we are aiming to preserve finiteness.

**Corollary 2.5.2.3.** *Let $\mathcal{X}, \mathcal{Y}$ be two classes consisting of concept classes. Suppose there is a constant $p > 0$ s.t. for each $\mathcal{C}_1 = (C_1, E_1, \lambda_1) \in \mathcal{X}$ there is a $\mathcal{C}_2 = (C_2, E_2, \lambda_2) \in \mathcal{Y}$ satisfying the following: There are $f, g, h$ s.t. $\mathcal{C}_1 \leq_{pc} \mathcal{C}_2$ and $f, g, h, p$ witnesses the reduction. It follows that if $\mathcal{Y}$ has polynomial sized unique characterization then so does $\mathcal{X}$.* $\square$

**Theorem 2.5.3.** *Let $\mathcal{X}, \mathcal{Y}$ be two collection of concept classes. Fix positive integers $k, m$ and a constant polynomial $p(x)$. Suppose for each $\mathcal{C} \in \mathcal{X}$, there is a $\mathcal{C}' \in \mathcal{Y}$ s.t. there are functions $f, g, h$ satisfying the following:*

1. *$f, g, h, p(x)$ witnesses a PC reduction from $\mathcal{C}$ to $\mathcal{C}'$.*
2. *for every concept $c \in \mathcal{C}$ we have $|f(c)| \leq k|c|$*

*then if $\mathcal{Y}$ has an upper bound dependent on concept size, then so does $\mathcal{X}$.*

*Proof.* The proof is almost the same as in theorem 2.5.2. Let $\alpha$ be the function witnessing concept size dependent upper-bound on $\mathcal{Y}$ and assume the antecedent of our theorem. Fix a concept $c$ of $\mathcal{C}$, and consider its image $f(c)$ in $\mathcal{C}'$.Suppose $T$ is an unique characterization of $f(c)$ s.t. $|T| \leq \alpha(|f(c)|)$. Consider the set

$$S = \bigcup\{g(f(c), t) \mid t \in T\}$$

We claim $S$ uniquely characterizes $c$. It is evident from definition of PC reduction itself that $c$ fits $S$, since $c$ fits each of $g(f(c), t)$. Now, suppose $d \in \mathcal{C}$ also fits $S$, which again from the definition gives us $f(d)$ fits $T$. Since $T$ was an unique characterization, we can conclude $f(c) \cong f(d)$, now if $c \ncong d$ then without the loss of generality we can assume that there is an example $e$ s.t. $c$ fits $e$ but $d$ does not. Therefore, $h(e)$ is a positive example of $f(c)$ but a negative example of $f(d)$ which is contradiction, since they are equivalent. Hence, we conclude $c \cong d$.

To put a bound on the size of $S$, we see that

$$|S| \leq p|T| \leq p \times \alpha(|f(c)|) \leq p \times \alpha(k|(c)|)$$

Therefore $|S|$ is again bounded by a function of $|c|$. □

In the subsequent sections we will use PC reductions on various collections of concept classes, it will be a crucial tool in establishing some of the results in this thesis.

# Chapter 3

# Case Study: The Propositional Fragments

As we remarked earlier, our goal is to view fragments of propositional and modal Logic as concept classes and provide classifications upon various notions unique characterizations. In this chapter we tackle the task for the propositional case. To start with, we provide a very brief introduction of propositional Logic, covering all the essential concepts needed to define fragments of this logic (see def. 3.1.10). Following that, through a sequence of results, we establish that the lattice of *all* propositional fragments is isomorphic to the Post's Lattice (see thm. 3.1.19).

The rest of the chapter is inspired from, and is an extension of V. Dalmau's work in [Dal99]. We establish three characterizations with respect to unique characterizations, namely theorem 3.2.26, 3.3.22.1 and 3.3.26 , and three characterizations with respect to exact learnability, namely theorem 3.2.27, 3.3.31 3.3.32. We provide two tables (Table 3.1 and Table 3.2) listing all of our findings. In this table we look at fragments of propositional logic parameterized by a set of propositional formulas $Pr$. The contents to the left of the table is linked by an 'if-and-only-if' condition on the right of the table. Furthermore, we establish that each of these problems are decidable, i.e there is an algorithm that takes as input a finite set of propositional formulas and outputs whether the fragment formed by that set has any of these properties or not.

The first table mentions three different notions of unique characterizations. The notions differ from each based upon the function witnessing the upper bound. 'Polynomial sized unique characterization' or PSUC is witnessed by a bi-variate polynomial dependent on size of concept and the specified set of propositional variables, $p(|varphi|, |Prop|)$ say (see def. 2.4.8). On the other hand 'unique characterization based on variables' is witnesses by an uni-variate function dependent on variables of a given formula, $f(|vars(\varphi)|)$ say, which is not necessarily a polynomial (see def. 2.4.9). Finally, 'polynomial unique characterization based on variables' is the case where the witnessing function, $f(x)$ say, for the case of 'unique characterization based on variables' is an uni-variate polynomial (see def. 2.4.10).

For the latter table we look into three notions of exact learnability. These three notions now differ due to the admitted running time of the algorithm. A fragment admitting 'Exactly learnable with membership

| $\mathbb{C}(\mathcal{L}_{Pr})$ admits ... | $Cl(Pr)$ is a subset of |
|---|---|
| Polynomial Sized unique characterization | $(i)$ $Cl(\wedge, \bot, \top)$ <br> $(ii)$ $Cl(\vee, \bot, \top)$ or <br> $(iii)$ $Cl(\leftrightarrow, \bot)$. |
| Unique characterization based on variables | $(i)$ $Cl(\wedge, \vee, \bot, \top)$ or <br> $(ii)$ $Cl(\neg, \bot)$. |
| Polynomial unique characterization based on variables | $(i)$ $Cl(\wedge, \bot, \top)$ <br> $(ii)$ $Cl(\vee, \top, \bot)$ or <br> $(iii)$ $Cl(\neg, \bot)$. |

Table 3.1: Table for Unique characterization

queries' would be witnessed by an algorithm halting in polynomial time depending on size of the formula and the specified set of propositional variables, $p(|\varphi|, |Prop|)$ say (see def. 2.4.12). 'Exactly learnable dependent on variables' is more complicated. If a fragment admits this property then all the formulas from this fragment are effectively learnable (see def. 2.4.13) but some of the formulas are learnt in time $< f(|vars(\varphi)|)$, where $f(x)$ is a non-decreasing function. 'Exactly learnable dependent on variables' is the same as in the previous case, the only difference being $f(x) = p(x)$ for some polynomial $p(x)$ (see def. 2.4.14).

The bulk of this chapter is dedicated to establishing various PC reductions and analysis of particular fragments. For the first result, theorem 3.2.26, we follow V. Dalmau's proof blueprint. For the positive results we analyse the fragments formed by the sets $\{x \wedge y, \top, \bot\}$, $\{x \wedge y, \top, \bot\}$ and $\{x \oplus y, \top\}$. For the negative results we resort to PC-reductions.

The second result, namely theorem 3.3.22.1, follows the same proof strategy. For positive results we analyse the fragments formed by $\{x \wedge y, x \vee y, \top, \bot\}$ and by $\{\neg x, \bot\}$. For the negative results we firstly analyse the fragment $\{x \oplus y \oplus z\}$ and resort to using PC-reductions.

The last result, namely theorem 3.3.26, is a refinement of the theorem 3.3.22.1 and 3.3.26. The results pertaining to learnability are established by providing explicit algorithms and using theorems 2.4.17 and 2.4.16. The reader is encouraged to read the concerned sections for detailed proofs.

## 3.1 Propositional Logic and Propositional Fragments

We introduce the notion of propositional fragments and look into it's correspondence with Post's Lattice.

**Definition 3.1.1.** *The full propositional logic over a set of propositional variables Prop, denoted as* $\mathcal{L}_{Prop}$*,*

| $\mathbb{C}(\mathcal{L}_{Pr})$ admits .... | $Cl(Pr)$ is a subset of |
|---|---|
| Exactly Learnable with membership queries | $(i)$ $Cl(\wedge, \perp, \top)$ $(ii)$ $Cl(\vee, \perp, \top)$ or $(iii)$ $Cl(\leftrightarrow, \perp)$. |
| Exactly Learnable dependent on variables | $(i)$ $Cl(\wedge, \vee, \perp, \top)$ or $(ii)$ $Cl(\neg, \perp)$. |
| Exactly Learnable polynomially dependent on variables | $(i)$ $Cl(\wedge, \perp, \top)$ $(ii)$ $Cl(\vee, \top, \perp)$ or $(iii)$ $Cl(\neg, \perp)$. |

Table 3.2: Table for Exact Learnability

*is generated recursively by,*

$$\varphi ::= \ p \mid \neg\varphi \mid p \wedge p \mid p \vee p \mid \top \mid \ \perp$$

*where $p \in$Prop.*

There is no restriction on Prop to be finite (or infinite). Notice that we omitted some functions like $\rightarrow$, this can be defined from other functions such as $p \rightarrow q = \neg p \vee q$.

**Definition 3.1.2.** *Given a propositional formula $\varphi$, $vars(\varphi)$ denote the variables occurring in that formula.*

**Definition 3.1.3** (Variable assignment)**.** *A function $\mu : Prop \rightarrow \{0, 1\}$ is called a variable assignment.*

**Definition 3.1.4** (Models of propositional logic)**.** *Let $\mathcal{L}_{Prop}$ denote the set of all propositional formulas over the set $Prop$. An model (of a propositional logic) is a function $\hat{\mu} : \mathcal{L}_{Prop} \rightarrow \{0, 1\}$, that follows the following conditions for any $\psi, \theta \in \mathcal{L}_{Prop}$.:*

$$\begin{aligned} \hat{\mu}(\psi \wedge \theta) = 1 &\iff \hat{\mu}(\psi) = 1 \textit{ and } \hat{\mu}(\theta) = 1 \\ \hat{\mu}(\psi \vee \theta) = 0 &\iff \hat{\mu}(\psi) = 0 \textit{ and } \hat{\mu}(\theta) = 0 \\ \hat{\mu}(\neg\psi) = 1 &\iff \hat{\mu}(\psi) = 0 \\ \hat{\mu}(\top) = 1 &\qquad \hat{\mu}(\perp) = 0 \end{aligned}$$

**Definition 3.1.5** (Satisfaction in propositional logic)**.** *Given a propositional model $\hat{\mu}$ and a propositional formula $\varphi$ over a set of variables Prop, we say $\hat{\mu} \models \varphi$ if and only $\hat{\mu}(\varphi) = 1$.*

Notice that every function $\mu : Prop \rightarrow \{0, 1\}$, can be extended to a model of propositional logic $\hat{\mu}$ and vice-versa. We will often abuse notation and state variable assignments as models, the reader can easily infer our intentions from the context.

**Proposition 3.1.6** (see [Men09])**.** *For any propositional formula $\varphi(p_1, \ldots, p_n)$(over a set of variables Prop) with all of its variables listed, we have the that for any models $\hat{\mu}, \hat{\nu}$ over Prop,*

$$\hat{\mu}(\varphi) = \hat{\nu}(\varphi) \iff \hat{\mu}(vars(\varphi)) = \hat{\nu}(vars(\varphi))$$

The previous lemma establishes that satisfaction of every formula $\varphi$ is dependent only on $vars(\varphi)$. This fact will be crucial in the subsequent definition. Remember that $A^B$ stands *for all the possible functions from $B$ to $A$*, and for any function $f$, $(f \restriction S)$ denotes the restriction of $f$ to the subset $S$.

**Definition 3.1.7** (Boolean functions defined by formulas)**.** *For every propositional formula $\varphi \in \mathcal{L}_{Prop}$, the function defined by $\varphi$, denoted by $f_\varphi$, is a function $f_\varphi : \{0,1\}^{vars(\varphi)} \to \{0,1\}$ s.t. for each $\mu \in \{0,1\}^{Prop}$, we have that $f_\varphi(\mu) = \hat{\mu}(\varphi)$.*

Following the completeness of Propositional Logic, every formula of propositional Logic can be identified with a function of the above form (see [Men09] chapter on propositional Logic for more details). This observation will be crucial in connecting Boolean Clones and propositional fragments, once we formally define fragments.

Notice that there is a natural equivalence relation on $\mathcal{L}_{Prop}$. Pick any two $\psi, \varphi \in \mathcal{L}_{Prop}$, we say $\psi \sim \varphi$ if for each propositional model $\hat{\mu}$,

$$\hat{\mu} \models \psi \iff \hat{\mu} \models \varphi$$

Following the result given below (see Prop 3.1.8), $\sim$ will partition the set $\mathcal{L}_{Prop}$. By $[\![\varphi]\!]_\sim$ we will denote the equivalence class to which $\varphi$ belongs.

**Proposition 3.1.8.** *The relation $\sim$ is an equivalence relation on $\mathcal{L}_{Prop}$.* □

**Definition 3.1.9** (Substitution of propositional formulas)**.** *Suppose $\psi_1, \ldots, \psi_n$ are propositional formulas and $\varphi(p_1, \ldots, p_n)$ is a propositional formula with all of its variables listed (in other words $vars(\varphi) = \{p_1, \ldots, p_n\}$). By $\varphi(\psi_1, \ldots, \psi_n)$ we denote the propositional formula obtained by replacing every $p_i$ by the formula $\psi_i$.*

**Definition 3.1.10** (Fragments of propositional logic)**.** *Fix a set of variables Prop and let $\mathcal{L}_{Prop}$ denote the set of all propositional formulas over Prop. We say a subset $\mathcal{F} \subseteq \mathcal{L}_{Prop}$ is a fragment if,*

*i. for each $p \in Prop$, $p \subseteq \mathcal{F}$.*

*ii. for every $\psi_1, \ldots, \psi_n \in \mathcal{F}$ and $\varphi(p_1, \ldots, p_n) \in \mathcal{F}$, we have $\varphi(\psi_1, \ldots, \psi_n) \subseteq \mathcal{F}$.*

*iii. If $\psi \in \mathcal{F}$ then $[\![\psi]\!]_\sim \in \mathcal{F}$.*

**Definition 3.1.11** (Generating set)**.** *Let $\mathcal{F}$ be propositional fragment, we say a set of propositional formulas $Pr$ generates $\mathcal{F}$, if $Pr \subseteq \mathcal{F}$ and for every fragment $\mathcal{F}'$ with $Pr \subseteq \mathcal{F}'$, we have $\mathcal{F} \subseteq \mathcal{F}'$. We write $\mathcal{F} := \mathcal{F}(Pr)$.*

Notice that fragments of propositional logic depends on the set of variables we start from. From now on till the end of this section, we talk about propositional fragment over an infinite set $IProp$, where

$$IProp = \{p_1, \dots, p_n, \dots\}$$

Much like Boolean clones, propositional fragments are also closed under arbitrary intersections. We define an operation $\sqcup$ in the following way:

$$\sqcup(\mathcal{F}_1, \mathcal{F}_2) := \bigcap\{\mathcal{F} : \mathcal{F}_1 \cup \mathcal{F}_2 \subseteq \mathcal{F}\}$$

where $\mathcal{F}_1, \mathcal{F}_2$ are propositional fragments. If we denote the class of *all* propositional fragments (over $IProp$) as $\mathscr{L}_{IProp}$, then the following triple forms a lattice $(\mathcal{L}_{IProp}, \cap, \Delta)$. The claims are rather easy to verify.

**Proposition 3.1.12.** *Let $I$ be an indexing set. If $\{\mathcal{F}_i : i \in I\}$ is a set of propositional fragments, then $\mathcal{F} = \bigcap\{\mathcal{F}_i : i \in I\}$ is again a propositional fragment.* □

**Proposition 3.1.13.** *The ordered triple $(\mathscr{L}_{IProp}, \cap, \Delta)$ forms a lattice.* □

Now we move onto the task of representing the lattice $(\mathscr{L}_{IProp}, \cap, \Delta)$. We do so by showing that the Post's Lattice isomorphic to the above lattice. Let $Pr$ be a finite set of propositional formulas, then define

$$Func(Pr) = \{f_\varphi : \varphi \in Pr\}$$

The correspondence we want to establish is the following:

**Theorem 3.1.14.** *Let $Pr$ be a set of finite propositional formulas over $IProp$, then the following equivalence holds*

$$\varphi \in \mathcal{F}(Pr) \iff f_\varphi \in Cl(Func(Pr))$$

This correspondence strongly on the correlation between substitution and function composition. In fact we can prove that substitution is, in a specific way, equivalent to composition. We will come back to theorem 3.1.14 after proving the following theorem:

**Theorem 3.1.15.** *Let $\psi_1, \dots, \psi_n$ be propositional formulas over the set $IProp$ and $f_{\psi_1}, \dots, f_{\psi_n}$ be their corresponding Boolean functions. Additionally, let $\varphi(p_1, \dots, p_n)$ is another propositional formula with all of its variables listed. Suppose $\theta = \varphi(\psi_1, \dots, \psi_n)$, then we have:*

$$f_\theta = f_\varphi(f_{\psi_1}, \dots, f_{\psi_n})$$

*Proof.* Suppose $vars(\phi_1) \neq vars(\phi_2)$ for some formulas $\phi_1$ and $\phi_2$. We can pad the formulas in the following way: Let $vars(\phi_1) \cup vars(\phi_2) = \{q_1, \dots, q_m\}$ and define

$$\phi_i' := \phi_i \wedge (\neg(q_1 \wedge \cdots \wedge q_m) \vee (q_1 \wedge \cdots \wedge q_m))$$

where $i \in \{1, 2\}$. Notice that $vars(\phi'_1) = vars(\phi'_2)$. From this argument, we can assume without the loss of generality

$$vars(\psi_1) = ... = vars(\psi_n)$$

Now our proof goes on by induction on the complexity of $\varphi$. Suppose $\varphi$ has complexity 0, it means that $\varphi = p$ for some $p \in Prop$ and $T_\varphi = p$. Hence it follows trivially that $T_\theta = T_\varphi(T_\psi)$ for any $\psi$. Our base case is done.

Now suppose our claim holds for any formula with complexity $< n$. Suppose $\varphi$ is of complexity $n$. We have three cases, but showing the results for $\neg, \wedge$ should suffice.

- **Case 1:** $(\varphi = \neg\phi)$

  In this case notice that $f_{\varphi(p_1,...,p_n)} = 1 - f_{\phi(p_1,...,p_n)}$. Since complexity of $\phi$ is less than $n$, using I.H. we get that $f_{\phi(\psi_1,...,\psi_n)} = f_\phi(f_{\psi_1}, \ldots, f_{\psi_n})$. Putting both facts together we get that

  $$T_{\varphi(\psi_1,...,\psi_n)} = 1 - f_\phi(f_{\psi_1}, \ldots, f_{\psi_n}) = f_\varphi(f_{\psi_1}, \ldots, f_{\psi_n})$$

- **Case 2:** $(\varphi = \phi \wedge \kappa)$

  In this case notice that $f_{\varphi(p_1,...,p_n)} = min(f_{\phi(p_1,...,p_n)}, f_{\kappa(p_1,...,p_n)})$. Since the complexity of both, $\phi$ and $\kappa$ are less than $n$, we can use I.H. to infer that $f_{\phi(\psi_1,...,\psi_n)} = f_\phi(f_{\psi_1}, \ldots, f_{\psi_n})$ and $f_{\kappa(\psi_1,...,\psi_n)} = f_\kappa(f_{\psi_1}, \ldots, f_{\psi_n})$. Using both the facts together we get,

  $$f_{\varphi(\psi_1,...,\psi_n)} = min(f_{\phi(\psi_1,...,\psi_n)}, f_{\kappa(\psi_1,...,\psi_n)}) = f_\varphi(f_{\psi_1}, \ldots, f_{\psi_n})$$

This closes our induction cases and we have proven the result. ☐

*Proof of theorem 3.1.14.* : Fix a set of formulas $Pr$. and consider the propositional fragment $\mathcal{F} = \mathcal{F}(Pr)$. Furthermore, notice that $Cl(Func(\mathcal{F})) = Cl(Func(Pr))$. We know that $IProp \subseteq \mathcal{F}$, and hence $f_p$, for any propositional variable $p$, is a projection map and in turn $f_p \in Cl(Func(\mathcal{F}))$. Now let $\pi^n_k$ be a projection map. The formula

$$\theta := (p_1 \vee \neg p_1) \wedge \cdots \wedge p_k \wedge \ldots (p_n \vee \neg p_n)$$

has $\pi^n_k$ as its truth table. Now, $\theta$ is equivalent to $p_k$ and hence $\theta \in \mathscr{F}$.

Now suppose we have established that for any $\varphi(p_1, \ldots, p_m), \psi_1, \ldots, \psi_m \in \mathcal{F}$, we have that $f_\varphi, f_{\psi_1}, \ldots, f_{\psi_m} \in Cl(Func(\mathcal{F}))$. By the property of clones and using the theorem above, we get $f_\varphi(f_{\psi_1}, \ldots, f_{\psi_m}) = f_{\varphi(\psi_1,...,\psi_n)} \in Cl(Func(\mathcal{F}))$. For the converse, let us assume that $f_\varphi, f_{\psi_1}, \ldots, f_{\psi_m} \in Cl(Func(\mathcal{F}))$, where $|vars(\varphi)| = m$, and $\varphi, \psi_1, \ldots, \psi_m \in \mathcal{F}$. It implies that $[\![\varphi]\!], [\![\psi_1]\!], \ldots, [\![\psi_m]\!] \subseteq \mathcal{F}$ , again by definition of fragments and the theorem above we get that, $\varphi(\psi_1, \ldots, \psi_m) \in Cl(Func(\mathcal{F}))$. ☐

Just like we talked about Boolean clones generated by propositional formulas, we can talk about propositional fragments generated by Boolean formulas. Let $F$ be any Boolean formulas. Define

$$Form(F) := \{\varphi : f_\varphi \in F\}$$

So we denote the fragment formed by $F$ by $\mathcal{F}(Form(F))$. The following is again easy to verify.

**Proposition 3.1.16.** *$\mathcal{F}(Form(F))$ is a propositional fragment for any finite set of Boolean formulas $F$.* $\square$

**Proposition 3.1.17.** *Let $Cl$ be a Boolean clone, then we have that $Form(Cl)$ forms a propositional fragment, i.e. $\mathcal{F}(Form(Cl)) = Form(Cl)$. Similarly, if $\mathcal{F}$ is a propositional fragment then $Cl(Func(\mathcal{F})) = Func(\mathcal{F})$.*

*Proof.* We prove the first one and leave out the latter, since they follow the same structure.

Let $Cl$ is a Boolean Clone it contains all the projections and hence we can conclude $IProp \subseteq Form(Cl)$. Again, it is straightforward to see $Form(Cl)$ is closed under substitution, since $Cl$ is closed under composition and using theorem 3.1.15. $\square$

We developed a way of generating a clone from a fragment and vice versa, now we ask the question 'Will we get back the same fragment, if we start with one, go to its clone and come back?' The answer is Yes.

**Theorem 3.1.18.** *For any propositional fragment $\mathcal{F}$, we have that $\mathcal{F} = \mathcal{F}(Form(Cl(Func(\mathcal{F}))))$. Similarly, for any Boolean clone $Cl$, we have that $Cl(Func(\mathcal{F}(Form(Cl)))) = Cl$.*

*Proof.* For any formula $\varphi$, use thm 3.1.14 to obtain

$$\varphi \in \mathcal{F}(\mathcal{F}) \iff f_\varphi \in Cl(Func(\mathcal{F}))$$

Notice that $\mathcal{F}(\mathcal{F}) = \mathcal{F}$. Again by previous theorem, we get that $\varphi \in \mathcal{F}(Form(Cl(Func(\mathcal{F}))))$ if and only if $\varphi \in Form(Cl(Func(\mathcal{F}))$. Now by definition, it follows $\varphi \in Form(Cl(Func(\mathcal{F})))$ if and only if $f_\varphi \in Cl(Func(\mathcal{F}))$. We have this chain of equivalences:

$$\varphi \in \mathcal{F} \iff f_\varphi \in Cl(Func(\mathcal{F})) \iff \varphi \in Form(Cl(Func(\mathcal{F}))) \iff \mathcal{F}(Form(Cl(Func(\mathcal{F}))))$$

This closes our first claim.

The other claim is established in an analogous way. This closes the proof. $\square$

The above proof that the natural map from the collection of propositional fragments to the collection of Boolean clones, given by $\mathsf{h}(\mathcal{F}) = Cl(Func(\mathcal{F}))$ is a bijection.

**Theorem 3.1.19.** *$\mathsf{h}$ is an isomorphism between $(\mathscr{L}_{IProp}, \cap, \Delta)$ and the Post's Lattice.*

*Proof.* For showing that it is an bijection, notice that for any two fragments $\mathcal{F}_1, \mathcal{F}_2$, using proposition 3.1.17 we have:

$$Func(\mathcal{F}_1) = Cl(Func(\mathcal{F}_1)) = Cl(Func(\mathcal{F}_2)) = Func(\mathcal{F}_2)$$

Now by definition of $Func$ and fragments, $\mathcal{F}_1 = \mathcal{F}_2$. Now for every clone $Cl$, there is a pre-image of it, namely $\mathcal{F}(Func(Cl))$.

Now for showing that $\mathsf{h}$ is an isomorphism, it is enough to show that $\mathsf{h}$ preserves order, i.e if $\mathcal{F}_1 \subseteq \mathcal{F}_2$, then $\mathsf{h}(\mathcal{F}_1) \subseteq \mathsf{h}(\mathcal{F}_2)$ for every fragments $\mathcal{F}_1, \mathcal{F}_2$. But notice that if $\mathcal{F}_1 \subseteq \mathcal{F}_2$ then $Func(\mathcal{F}_1) \subseteq Func(\mathcal{F}_2)$. Again following proposition 3.1.17 and the chain of equalities given above, we get our result. $\square$

It follows that $(\mathscr{L}_{IProp}, \cap, \Delta)$ looks exactly like Post's lattice. Till now we have recognized many analogues between boolean clones and propositional fragments, the reason behind such a connection was propositional models had strong connections towards clones theory. In the subsequent section we move to Modal Logic and Modal Fragments, where we generalise the notion of fragments. Unlike the propositional case, we do not have functions as models for modal logic and that would make the investigations more challenging.

We are in a place to define propositional fragments as concept classes. Let $Pr$ be a set of propositional formulas, and $Prop$ be a set of *finite* propositional variables. By $\mathbb{C}(\mathcal{L}_{Pr}[Prop]) = (C, E, \lambda)$ we denote the concept class where

$$C = \{\varphi \in \mathcal{F}(Pr) : vars(\varphi) \subseteq Prop\}$$

i.e. $C$ consists of all the formulas from $\mathcal{F}_{Pr}$ which has variables from $Prop$. As our examples $E$ we use the set of all variable assignments from $Prop \to \{0, 1\}$. Finally, the function $\lambda$ maps each propositional formula to the set of all variable assignments that makes it true, i.e. for any $\varphi \in C$, we define

$$\lambda(\varphi) = \{t \in E : t \models \varphi\}$$

In this thesis we will often be interested in a collection of concept classes rather than one particular concept class. Fix a set of formulas (boolean or modal) and define $IPROP = \{p_1, p_2, \dots\}$. We denote

$$\mathbb{C}(\mathcal{L}_{Pr}) := \{\mathbb{C}(\mathcal{L}_{Pr}[Prop]) : Prop \subseteq_{fin} PROP\}$$

In addition, we would like to consider infinite propositional concept classes. For any finite set of propositional formulas, define the concept class $\mathbb{C}(\mathcal{L}_{Pr}[IProp]) = (C_I, E_I, \lambda_I)$ as follows:

$$\begin{aligned} C_I &= \{\varphi \in \mathcal{F}(Pr) : vars(\varphi) \subseteq Iprop\} \\ E_I &= \{0, 1\}^{IProp} \\ \lambda_I(\varphi) &= \{t \in E : t \models \varphi\} \end{aligned}$$

Finally, going forward we will abuse notation and write $Cl(Pr)$ to mean $Cl(Func(Pr))$ for any set of propositional formulas. Furthermore we often drop the variables in the subscript while writing propositional concept classes and clones i.e instead of writing $\mathbb{C}(\mathcal{L}_{x\wedge y}[Prop])$, we will simply write $\mathbb{C}(\mathcal{L}_{\wedge}[Prop])$ and instead of writing $Cl(\{x \wedge y\})$ we will write $Cl(\wedge)$.

## 3.2 Polynomial Sized Unique Characterization for Propositional Fragments

This part of our results is heavily influenced by V. Dalmau's work in [Dal99]. We produce a complete characterization for Boolean fragments, similar to the main work in the aforementioned paper. The main aim is to establish the following theorem,

**Theorem 3.2.1.** *For any set of propositional formulas $Pr$, the following the equivalent:*

- *There exists a polynomial $p(x, y)$ s.t. for every finite Prop, and every $\varphi \in \mathbb{C}(\mathcal{L}_{Pr}[Prop])$, there is a set of labelled examples $T$ that uniquely characterizes $\varphi$ with $|T| < p(|\varphi|, |Prop|)$*
- *$Cl(Pr)$ is a subset of either of the three (i) $Cl(\wedge, \bot, \top)$, (ii) $Cl(\vee, \bot, \top)$ or (iii) $Cl(\leftrightarrow, \bot)$.*

Similar to V. Dalmau's original work, the proof of the above theorem is by establishing reductions between concept classes. Unlike the original, we swap the pwm-reductions for PC-reductions. The formulas we choose as the range of the PC-reductions is also the exact same as used in V. Dalmau's proof. As stated in the introduction, this theorem is related to V.Dalmau's result in the sense that we obtain V. Dalmau's result, minus the cryptographic assumptions, as a direct corollary (see theorem 3.2.27).

The first step is to prove that the fragment $\mathbb{C}(\mathcal{L}_{\wedge,\vee,\neg}[Prop]) = \mathbb{C}(\mathcal{L}_{\wedge,\neg}[Prop])$ does not admit PSUC.

**Lemma 3.2.2.** *The fragment $\mathbb{C}(\mathcal{L}_{\wedge,\vee,\neg}[Prop])$ does not admit polynomial sized unique characterization.*

*Proof.* Suppose there is a polynomial $p(x, y)$ s.t. for every concept class $\mathbb{C}(\mathcal{L}_{\wedge,\vee,\neg}[Prop]) \in \mathbb{C}(\mathcal{L}_{\wedge,\vee,\neg})$ and concept $\varphi$ of $\mathcal{C}$ there is an unique characterization of size $< p(|\varphi|, |Prop|)$. Consider the formula

$$\Psi_{Prop} = \bigwedge_{p \in Prop} p$$

According to our assumptions $\Psi_{Prop}$ would *always* require $< p(|Prop|, |Prop|)$ many examples for unique characterization, irrespective of $|Prop|$. This is a clear contradiction to [ABCS92], since a polynomial bound on $Prop$ for $\Psi_{Prop}$ does not exist. □

### 3.2.1 Reduction from $\mathbb{C}(\mathcal{L}_{\wedge,\vee,\neg}[Prop])$ to $\mathcal{R}[Prop]$

We now immediately move onto the PC-reductions. For our reductions we will always use $p(x) = 1$ as the polynomial witnessing our reduction. For the $f, g$ and $h$ we will use the functions $\sigma_i, \tau_i, \gamma_i$ respectively.

The reduction is $\mathbb{C}(\mathcal{L}_{\wedge,\vee,\neg}[Prop]) \leq_{pc} \mathbb{C}(\mathcal{L}_{\wedge,\vee,\neg}[Prop'])$, where $Prop$ is a finite set of propositional variables and $Prop'$ is defined as follows.

$$Prop' = Prop \cup \{q_i : p_i \in Prop\}$$

Define $\sigma_1^*$ inductively from the concepts of $\mathbb{C}(\mathcal{L}_{\wedge,\vee,\neg}[Prop])$ to the ones in $\mathbb{C}(\mathcal{L}_{\wedge,\vee,\neg}[Prop])$ as follows:

$$\sigma_1^*(p_i) = p_i \qquad \sigma_1^*(\neg p_i) = q_i$$

$$\sigma_1^*(\varphi \wedge \psi) = \sigma_1^*(\varphi) \wedge \sigma_1^*(\psi) \qquad \sigma_1^*(\varphi \vee \psi) = \sigma_1^*(\varphi) \vee \sigma_1^*(\psi)$$

Now we define $\sigma_1$ from the concepts of $\mathbb{C}(\mathcal{L}_{\wedge,\vee,\neg}[Prop])$ to the ones in $\mathbb{C}(\mathcal{L}_{\wedge,\vee,\neg}[Prop])$

$$\sigma_1(\varphi) = (\sigma_1^*(\varphi) \vee \bigvee_{1 \leq i \leq n} (p_i \wedge q_i)) \wedge \bigwedge_{1 \leq i \leq n} (p_i \vee q_i)$$

Let $t$ be a variable assignment over $Prop'$. Notice that if $t(p_i) = t_(q_i) = 0$ for some $i$ then for every concept $\varphi$ of $\mathbb{C}(\mathcal{L}_{\wedge,\vee,\neg}[Prop])$ we have that $t \not\models \varphi$. Similarly, if for some $i$, $t(p_i) = t(q_i) = 1$ then for every $\varphi$ of $\mathbb{C}(\mathcal{L}_{\wedge,\vee,\neg}[Prop])$ it follows that $t \models \varphi$. It immediately follows:

**Proposition 3.2.3.** *Let $\varphi, \psi \in ran(\sigma_1)$, if $t \models \varphi$ and $t$ be a variable assignment over $Prop'$. If $\psi \not\models \varphi$, then $t(p_i) = 1 - t(q_i)$ for every $i$.* $\square$

Now to define a new concept class denoted by $\mathcal{R}[Prop] = (C_R, E_R, \lambda_R)$ where $C_R = ran(\sigma_1)$,

$$E_R = \{t : \text{ is a variable assignment over } Prop' \;\&\; t(p_i) = 1 - t(q_i) \text{ for every } i\}$$

and finally $\lambda_R(\varphi) = \{t \in E : t \models \varphi\}$. Finally define

$$\mathcal{R} = \{\mathcal{R}[Prop] : \text{ Prop is a finite set of propositional variables}\}$$

We claim the following:

**Proposition 3.2.4.** *For any set of propositional variables, Prop say, If the concept class $\mathbb{C}(\mathcal{L}_{\wedge,\vee})$ has PSUC then so does $\mathcal{R}$.*

*Proof.* Suppose $\varphi$ is a concept of $\mathcal{R}[Prop] \in \mathcal{R}$. It follows that $\varphi$ is also a concept of $\mathbb{C}(\mathcal{L}_{\wedge,\vee}[Prop'])$. Now according to our assumptions, $\varphi$ has an unique characterization, $T_\varphi$ say, w.r.t. $\mathbb{C}(\mathcal{L}_{\wedge,\vee}[Prop'])$. We define

$$T'_\varphi = \{t \in T_\varphi : t(p_i) = 1 - t(q_i)\}$$

We will prove that $T'_\varphi$ is an unique characterization of $\varphi$ w.r.t. $\mathcal{R}[Prop]$. Suppose $\psi, \psi'$ are concepts from $\mathcal{R}[Prop]$ and hence from $\mathbb{C}(\mathcal{L}_{\wedge,\vee}[Prop'])$. Therefore, there should be a $t \in T_\varphi$ that distinguishes them. But from proposition 3.2.3 it follows that $t \in T'_\varphi$. Hence, $T'_\varphi$ is an unique characterization of $\varphi$ w.r.t. $\mathcal{R}[Prop]$. Furthermore,

$$|T'_\varphi| \leq |T_\varphi| \leq 2|Prop|$$

This tells us $p(x) = 2x + 1$ witnesses the PSUC for $\mathcal{R}$. $\square$

The next observation is concerning the examples of $\mathcal{R}[Prop]$ and $\mathbb{C}(\mathcal{L}_{\wedge,\vee,\neg}[Prop])$. Let $t$ be an example of $\mathbb{C}(\mathcal{L}_{\wedge,\vee,\neg}[Prop])$, then define

$$t'(p_i) = t(p_i) \qquad t'(q_i) = 1 - t(p_i)$$

The correspondence is one-one. Notice that every example of the example space of $\mathcal{R}[Prop]$ is can from an example of the example space of $\mathbb{C}(\mathcal{L}_{\wedge,\vee,\neg}[Prop])$. This motivates the following theorem.

**Theorem 3.2.5.** *For any concept $\varphi$ of $\varphi$ of $\mathbb{C}(\mathcal{L}_{\wedge,\vee,\neg}[Prop])$ and any example $t$ of $\mathbb{C}(\mathcal{L}_{\wedge,\vee,\neg}[Prop])$ we have*

$$t \models \varphi \iff t' \models \sigma_1(\varphi)$$

*Proof.* We assume that $\varphi$ is a DNF. The proof is via induction the complexity of formulas. Notice that the base case follows trivially since $t(p_i) = t'(p_i)$.

For the case of $\varphi = \neg\psi$, it follows $\psi = p_i$ and the rest follows simply by definition of satisfaction. The same goes for $\varphi = \psi \wedge \theta$ and $\varphi = \psi \vee \theta$. $\square$

We now focus on providing the first PC-reduction, namely $\mathbb{C}(\mathcal{L}_{\wedge,\vee,\neg}[Prop]) \leq_{pc} \mathcal{R}[Prop]$. Let $\varphi$ be a concept of $\mathcal{R}[Prop]$ and $(t', p)$ a labelled example of $\mathcal{R}[Prop]$. We define

$$\tau_1(\varphi, (t', p)) = (t, p)$$

where $t'$ is the unique example of $\mathcal{R}[Prop]$ obtained from $t$. Again, for any example $t$ of $\mathcal{R}[Prop]$ we define $\gamma_1(t) = t'$. Notice that conditions $1, 2$ and $4$ of the PC-reductions are satisfied.

**Lemma 3.2.6.** *For every finite set of variables Prop,* $\mathbb{C}(\mathcal{L}_{\wedge,\vee,\neg}[Prop]) \leq_{pc} \mathcal{R}[Prop]$. ☐

**Lemma 3.2.7.** *If $\mathcal{R}$ has PSUC, then so does $\mathbb{C}(\mathcal{L}_{\wedge,\vee,\neg})$.*

*Proof.* We invoke theorem 2.5.2 with $k = 2, m = 2$ and $p(x) = 1$. ☐

**Theorem 3.2.8.** *The collection $\mathbb{C}(\mathcal{L}_{\wedge,\vee})$ does not have PSUC.*

*Proof.* Suppose it does, then proposition tells us $\mathcal{R}$ has PSUC and lemma 3.2.7 gives us $\mathbb{C}(\mathcal{L}_{\wedge,\vee,\neg})$ has PSUC. A contradiction. ☐

### 3.2.2 Positive Results and Reduction from $\mathbb{C}(\mathcal{L}_{\leftrightarrow,\top}[Prop])$ to $\mathbb{C}(\mathcal{L}_{\oplus,\top}[Prop])$

Let us define three particular propositional formulas

$$\begin{aligned} \mathsf{maj}(x, y, z) &= (x \wedge y) \vee (y \wedge z) \vee (x \wedge z) \\ \mathsf{AOR}(x, y, z) &= (x \wedge y) \vee z \\ \mathsf{OAND}(x, y, z) &= (x \vee y) \wedge z \end{aligned}$$

One can verify that $Cl(Func(\mathsf{maj})) = \mathrm{DM}$ ,$Cl(Func(\mathsf{OAND})) = \mathrm{MPT}_0^\infty$ ,$Cl(Func(\mathsf{AOR})) = \mathrm{MPT}_1^\infty$. In the subsequent parts of this section we will prove that each of the $\mathbb{C}(\mathcal{L}_{\mathsf{maj}}), \mathbb{C}(\mathcal{L}_{\mathsf{AOR}}), \mathbb{C}(\mathcal{L}_{\mathsf{OAND}})$ does not have PSUC. But before that, we will prove that a particular fragment has PSUC, namely $\mathbb{C}(\mathcal{L}_{\leftrightarrow,\perp})$. Firstly notice that,

**Theorem 3.2.9** (Anthony et al. [ABCS92])**.** *The class $\mathbb{C}(\mathcal{L}_{\oplus,\top})$ has PSUC, with respect to the polynomial $p(|Prop|) = |Prop| + 1$.* ☐

Our proof now depends on defining a PC reduction from a concept class $\mathbb{C}(\mathcal{L}_{\oplus,\top}[Prop])$ to the concept class $\mathbb{C}(\mathcal{L}_{\leftrightarrow,\perp}[Prop])$. In our proof, $Prop$ will be arbitrary and hence we can invoke theorem 2.5.2 yet again to get the result. We start again with defining a function $\sigma_3$, from the concepts of $\mathbb{C}(\mathcal{L}_{\leftrightarrow,\perp}[Prop])$ to the concepts of $\mathbb{C}(\mathcal{L}_{\oplus,\top}[Prop])$, for any arbitrary finite set of variables $Prop$. The definition is inductive.

$$\sigma_2(p) = p \qquad \sigma_2(\perp) = \top \oplus \top$$

$$\sigma_2(\phi \leftrightarrow \psi) = \sigma_2(\phi) \oplus \sigma_2(\psi) \oplus \top$$

**Proposition 3.2.10.** *Let $t$ be any variable assignment on Prop. For any concept $\varphi$ of $\mathbb{C}(\mathcal{L}_{\leftrightarrow,\perp}[Prop])$, we have*

$$t \models \varphi \iff t \models \sigma(\varphi)$$

*Proof.* The proof is by induction on the complexity of the formula $\varphi$. For propositional variables the result is trivial.

Suppose $t$ makes $\phi \leftrightarrow \psi$ true, then either both of $\phi, \psi$ are true or both of them are false. Apply the I.H., in the former case $\sigma_2(\phi), \sigma_2(\psi)$ both are true then $\sigma_2(\phi) \oplus \sigma_2(\psi) \oplus \top$ is true, a similar thing happens in the latter case. For the converse, let $t$ make $\sigma_2(\phi) \oplus \sigma_2(\psi) \oplus \top = \sigma_2(\phi \leftrightarrow \psi)$ true. Then by definition we have either both of $\sigma_2(\phi), \sigma_2(\psi)$ are true or both of $\sigma_2(\phi), \sigma_2(\psi)$ are false. In either of the cases, using I.H. we have that $\phi \leftrightarrow \psi$ is true. This closes the induction and our claim is proven. ☐

**Lemma 3.2.11.** *For any set Prop, we have that $\mathbb{C}(\mathcal{L}_{\leftrightarrow,\perp}[Prop]) \leq_{pc} \mathbb{C}(\mathcal{L}_{\oplus,\top}[Prop])$.*

*Proof.* Define $\gamma(t) = t$ and define $\tau$ is as follows, where $\varphi$ is a concept of $\mathbb{C}(\mathcal{L}_{\oplus,\top}[Prop])$ and $t$ a variable assignment over *Prop*.

$$\tau(\varphi, (t, +)) = (t, +)$$

$$\tau(\varphi, (t, -)) = (t, -)$$

Notice that, $\sigma_2, \tau_2, \gamma_2, p(x) = 1$ satisfies all the points of the PC reduction. ☐

**Theorem 3.2.12.** *The collection $\mathbb{C}(\mathcal{L}_{\leftrightarrow,\perp})$ has PSUC.*

*Proof.* We already know that the collection $\mathbb{C}(\mathcal{L}_{\oplus,\top})$ has PSUC. We invoke theorem 2.5.2, notice that $\mathbb{C}(\mathcal{L}_{\leftrightarrow,\perp}[Prop])$ is PC-reducible to $\mathbb{C}(\mathcal{L}_{\oplus,\top}[Prop])$ via $\sigma_1, \tau_1, \gamma_1$, having $k = 2, m = 2$ and $p(x) = 1$ as constants. ☐

Before moving onto the other reductions, we mention two other fragments that are of importance to us, namely $\mathbb{C}(\mathcal{L}_{\wedge,\top,\perp})$ and $\mathbb{C}(\mathcal{L}_{\vee,\top,\perp})$. As proven in [ABCS92], these two fragments has PSUC.

**Theorem 3.2.13** (Anthony et al. [ABCS92])**.** *The collections of concept classes $\mathbb{C}(\mathcal{L}_{\wedge,\top,\perp})$ and $\mathbb{C}(\mathcal{L}_{\vee,\top,\perp})$ has PSUC.* ☐

### 3.2.3 Reduction from $\mathbb{C}(\mathcal{L}_{\wedge,\vee}[Prop])$ to $\mathbb{C}(\mathcal{L}_{\mathsf{maj}}[Prop^*])$

Like in the previous section, we start from a finite set of propositional variables, *Prop* say, and put $Prop^* = Prop \cup \{x_0, x_1\}$. We define a map $\sigma_3^*$ from the concepts of $\mathbb{C}(\mathcal{L}_{\wedge,\vee}[Prop])$ to the concepts of $\mathbb{C}(\mathcal{L}_{\mathsf{maj}}[Prop^*])$ recursively.

$$\sigma_3^*(p) = p$$

$$\sigma_3^*(\phi \wedge \psi) = \mathsf{maj}(x_0, \sigma_3^*(\phi), \sigma_3^*(\psi))$$

$$\sigma_3^*(\phi \vee \psi) = \mathsf{maj}(x_1, \sigma_3^*(\phi), \sigma_3^*(\psi))$$

Using the above map $\sigma_3^*$ we define a map $\sigma_3$.

$$\sigma_3(\varphi) = \mathsf{maj}(\sigma_3^*(\varphi), x_0, x_1)$$

As a result of this definition, one result is immediate:

**Proposition 3.2.14.** *Let $\varphi$ be a formula in $\mathbb{C}(\mathcal{L}_{\wedge,\vee}[Prop])$, and $t$ a variable assignment on $Prop^*$ with $t(x_0) = 0 = 1 - t(x_1)$, then*

$$t \models \sigma_3(\varphi) \iff (t \restriction Prop) \models \varphi$$

*Proof.* Firstly notice that if a variable assignment $t$ on $Prop^*$ with $t(x_0) = 0 = 1 - t(x_1)$ makes $\sigma_3(\varphi)$ true then it is sufficient and necessary that $t$ makes $\sigma_3^*(\varphi)$ true. Now we proceed by induction on the complexity of $\varphi$.

If the complexity of $\varphi$ is 0, then $\varphi = p = \sigma_3^*(\varphi)$ for some prop variable in $Prop$. This case follows trivially.

Suppose our hypothesis holds for any $\varphi$ of complexity $< n$. Now, let $\varphi$ is of complexity $n$. Since $\varphi$ is in $\mathbb{C}(\mathcal{L}_{\wedge,\vee}[Prop])$, $\varphi$ is either of the form $\varphi = \phi \wedge \psi$ or $\varphi = \phi \vee \psi$. Now, suppose $t$ makes $\sigma_3^*(\varphi)$ true and $t(x_0) = 0 = 1 - t(x_1)$, and $\varphi = \phi \wedge \psi$, then it makes both $\sigma_3^*(\phi)$ and $\sigma_3^*(\psi)$ both true and using I.H. we conclude that $t \restriction Prop$ makes $\phi \wedge \psi$ true. For the converse, let $t \restriction Prop$ make both $\phi$ and $\psi$ true, and $t(x_0) = 0 = 1 - t(x_1)$, then using I.H. $\sigma_3^*(\phi)$, $\sigma_3^*(\psi)$ are both made true by $t$ and hence $t$ makes $\sigma_3^*(\varphi)$ true. Now if $\varphi = \phi \vee \psi$ then $t(x_1) = 1$ tells us $t$ making $\sigma_3^*(\varphi)$ true would entail either of $\sigma_3^*(\phi)$ or $\sigma_3^*(\psi)$ to be true, but this directly tells, using I.H., $\phi \vee \psi$ should be true. For the converse again let $t \restriction Prop$ make either $\phi \vee \psi$ true, and $t(x_0) = 0 = 1 - t(x_1)$, then according to our I.H., either $\sigma_3^*(\phi)$ or $\sigma_3^*(\psi)$ is made true by $t$, which translates to $t$ making $\sigma_3^*(\varphi)$ true. This closes the induction clause and our claim is proven. ☐

Now let us assume $t$ be any variable assignment on $Prop^*$ and let $\varphi$ be any member of $\mathbb{C}(\mathcal{L}_{\wedge,\vee}[Prop])$. Notice that if $t(x_0) = t(x_1) = 0$ then $\sigma_3(\varphi)$ is false and if $t(x_0) = t(x_1) = 1$ then $\zeta_2(\varphi)$ is true. The above theorem classifies the case when $t(x_0) = 0 = 1 - t(x_1)$. We want to look at the case when $t(x_0) = 1 = 1 - t(x_1)$. We will provide a theorem similar to theorem 1 but before that we need to learn how to *flip* a formula in $\mathbb{C}(\mathcal{L}_{\wedge,\vee}[Prop])$.

Let $Fl$ be a function from the concepts of $\mathbb{C}(\mathcal{L}_{\wedge,\vee}[Prop])$ defined recursively. Suppose $p, q$ is in $Prop$, then

$$Fl(p \wedge q) = p \vee q \qquad Fl(p \vee q) = p \wedge q$$

$$Fl(\phi \wedge \psi) = Fl(\phi) \vee Fl(\psi) \qquad Fl(\phi \vee \psi) = Fl(\phi) \wedge Fl(\psi)$$

**Proposition 3.2.15.** *Let $\varphi$ be a formula in $\mathbb{C}(\mathcal{L}_{\wedge,\vee}[Prop])$, and $t$ a variable assignment on $Prop^*$ with $t(x_0) = 1 = 1 - t(x_1)$,*

$$t \models \sigma_3(\varphi) \iff (t \restriction Prop) \models Fl(\varphi)$$

*Proof.* Notice that if a variable assignment $t$ on $Prop^*$ with $t(x_0) = 1 = 1 - t(x_1)$ makes $\sigma_3(\varphi)$ true then it is sufficient and necessary that $t$ makes $\sigma_3^*(\varphi)$ true. Now we proceed by induction on the complexity of $\varphi$.

The proof is again by induction the complexity of $\varphi$. The base case is trivial. We assume the I.H. holds for any formula of complexity $< n$.

If $\varphi = \phi \wedge \psi$ then $\sigma_3^*(\varphi) = \mathsf{maj}(x_0, \sigma_3^*(\phi), \sigma_3^*(\psi))$. Suppose $t$ is a variable assignment with $t(x_0) = 1 = 1 - t(x_1)$. Now $t$ makes $\sigma_3^*(\varphi)$ true iff $t$ makes either $\sigma_3^*(\phi)$ or $\sigma_3^*(\psi)$ true. But the latter condition implies $t \restriction Prop$ makes either $Fl(\phi)$ or $Fl(\psi)$ true by I.H., giving us $t \restriction Prop$ makes $Fl(\phi) \vee Fl(\psi) = F(\varphi)$ true. For the converse let $t \restriction Prop$ makes $Fl(\phi) \vee Fl(\psi)$ true then by I.H $t$ either makes $\sigma_3^*(\phi)$ or $\sigma_3^*(\psi)$ true then it follows that $t$ makes $\sigma_3^*(\varphi)$ true since $t(x_0) = 1$.

Similarly, let $\varphi = \phi \vee \psi$. Suppose $t$ is a variable assignment with $t(x_0) = 1 = 1 - t(x_1)$. Now $t$ makes $\sigma_3^*(\varphi)$ true iff $t$ makes $\sigma_3^*(\phi)$ and $\sigma_3^*(\psi)$ true. But the latter condition implies $t \restriction Prop$ makes $F(\phi)$ and $F(\psi)$ true by I.H., giving us $t \restriction Prop$ makes $F(\phi) \wedge F(\psi) = F(\varphi)$ true. For the converse let $t \restriction Prop$ makes $F(\phi) \wedge F(\psi)$ true then by I.H $t$ either both $\sigma_3^*(\phi)$ and $\sigma_3^*(\psi)$ true, then it follows that $t$ makes $\sigma_3^*(\varphi)$ true since $t(x_1) = 0$. This closes the induction and gives us our result. $\square$

For any variable assignment $t$ on $Prop^*$, define $t^*(x) = 1 - t(x)$ for each $x \in Prop^*$. One easy induction gives us the following result:

**Proposition 3.2.16.** *If $\varphi$ is a formula in $\mathbb{C}(\mathcal{L}_{\wedge,\vee})$ and $t$ a variable assignment on $Prop^*$, then*

$$t \models \varphi \iff t^* \not\models Fl(\varphi)$$

$\square$

We combine all the above theorem to provide the third reduction.

**Lemma 3.2.17.** *For any set Prop, we have that $\mathbb{C}(\mathcal{L}_{\wedge,\vee}[Prop]) \leq_{cp} \mathbb{C}(\mathcal{L}_{\mathsf{maj}}[Prop^*])$.*

*Proof.* Define for any variable assignment on $Prop$,

$$\gamma_3(t) = t \cup \{(x_0, 0), (x_1, 1)\}$$

The definition of $\tau_3$ is a little more involved. Let $x = \sigma_3(\varphi)$

$$\begin{aligned}
\tau_3(x, (t, +)) &= (t \restriction Prop, +) \quad \text{if } t(x_0) = 0 = 1 - t(x_1) \\
\tau_3(x, (t, +)) &= (t^* \restriction Prop, -) \quad \text{if } t(x_0) = 1 = 1 - t(x_1) \\
\tau_3(x, (t, -)) &= (t \restriction Prop, -) \quad \text{if } t(x_0) = 0 = 1 - t(x_1) \\
\tau_3(x, (t, -)) &= (t^* \restriction Prop, +) \quad \text{if } t(x_0) = 1 = 1 - t(x_1)
\end{aligned}$$

Finally, we map $(x, (t, +)), (x, (t, -)) \in \mathcal{N}_{\mathbb{C}(\mathcal{L}_{\mathsf{maj}}[Prop^*])}$ to $\{(t^* \restriction, +), (t^* \restriction, -)\}$ via $\tau_3$ whenever $t(x_0) = 1 = t(x_1)$ or $t(x_0) = 0 = t(x_1)$.

It is a routine verification to check $\sigma_3, \tau_3, \gamma_3, p(x) = 3$ witnesses the PC-reduction. $\square$

**Theorem 3.2.18.** $\mathbb{C}(\mathcal{L}_{\mathsf{maj}})$ *does not have PSUC.*

*Proof.* We already know $\mathbb{C}(\mathcal{L}_{\vee,\wedge})$ does not have PSUC due to theorem 3.2.8. We again invoke theorem 2.5.2, notice that $\mathbb{C}(\mathcal{L}_{\wedge,\vee}[Prop])$ is PC-reducible to $\mathbb{C}(\mathcal{L}_{\mathsf{maj}}[Prop^*])$ via $\sigma_3, \tau_3, \gamma_3$, having $k = 2, m = 2$ and $p(x) = 2$ as constants. So $\mathbb{C}(\mathcal{L}_{\mathsf{maj}})$ having PSUC means $\mathbb{C}(\mathcal{L}_{\wedge,\vee})$ has it as well, a contradiction. $\square$

### 3.2.4 Reduction from $\mathbb{C}(\mathcal{L}_{\wedge,\vee}[Prop])$ to $\mathbb{C}(\mathcal{L}_{\mathsf{OAND}}[Prop])$ and $\mathbb{C}(\mathcal{L}_{\mathsf{AOR}}[Prop])$

Now we focus on the dual fragments $\mathbb{C}(\mathcal{L}_{\mathsf{AOR}})$ and $\mathbb{C}(\mathcal{L}_{\mathsf{OAND}})$. The proof strategy of establishing $\mathbb{C}(\mathcal{L}_{\wedge,\vee}) \leq_{pc} \mathbb{C}(\mathcal{L}_{\mathsf{AOR}})$ is identical to that of proving $\mathbb{C}(\mathcal{L}_{\wedge,\vee}) \leq_{pc} \mathbb{C}(\mathcal{L}_{\mathsf{OAND}})$, hence we provide a detailed proof of one the reductions and only provide the blueprint for the other.

We proceed similarly as we did in $\mathbb{C}(\mathcal{L}_{\mathsf{maj}})$. We start from a finite set of propositional variables, $Prop$ say, and define $Prop^{\#} = Prop \cup \{x_0\}$. Let's define $\sigma_4^*$ from the concepts of : $\mathbb{C}(\mathcal{L}_{\wedge,\vee}[Prop])$ to the concepts of $\mathbb{C}(\mathcal{L}_{\mathsf{AOR}}[Prop^{\#}])$ recursively.

$$\begin{aligned}
\sigma_4^*(p) &= p \\
\sigma_4^*(\phi \wedge \psi) &= \mathsf{AOR}(\sigma_4^*(\phi), \sigma_4^*(\psi), x_0) \\
\sigma_4^*(\phi \vee \psi) &= \mathsf{AOR}(\sigma_4^*(\phi), \sigma_4^*(\phi), \sigma_4^*(\psi))
\end{aligned}$$

We define $\sigma_4$ using the function $\sigma_4^*$

$$\sigma_4(\varphi) = \mathsf{AOR}(\sigma_4^*(\varphi), \sigma_4^*(\varphi), x_0)$$

**Proposition 3.2.19.** *If $t$ is a variable assignment on $Prop^{\#}$ with $t(x_0) = 0$, and $\varphi$ is a concept of $\mathbb{C}(\mathcal{L}_{\wedge,\vee}[Prop])$, then*

$$t \upharpoonright Prop \models \varphi \iff t \models \sigma_4(\varphi)$$

*Proof.* From the previous definition it is sufficient and necessary that for $t$ to make $\sigma_4(\varphi)$ true, for some $\varphi$ in $\mathbb{C}(\mathcal{L}_{\wedge,\vee}[Prop])$, $t$ should make $\sigma_4^*(\varphi)$ true. The proof is now by induction on complexity of $\varphi$.

Suppose $t(x_0) = 0$ and $t$ is an assignment on $Prop^{\#}$. Now, if $\varphi = p = \sigma_4^*(p)$ then our theorem is trivially true.

Assume that for any formula of complexity $< n$ our theorem holds. Now, let $\varphi$ be a formula of complexity $n$. Let $t(x_0) = 0$ and $t$ is an assignment on $Prop^{\#}$. If $\varphi = \phi \wedge \psi$ then and $t \upharpoonright Prop$ makes both $\phi$ and $\psi$ true, then by I.H. $t$ makes $\sigma_4^*(\phi)$ and $\sigma_4^*(\psi)$ true. Then by definition $\sigma_4^*(\varphi)$ is made true by $t$. For the converse, if $\sigma_4^*(\varphi)$ is made true by $t$ then both $\sigma_4^*(\phi), \sigma_4^*(\psi)$ must be made true by it too and hence by I.H. $\phi, \psi$ should be made true by $t \upharpoonright Prop$, which is equivalent to $t \upharpoonright Prop$ making $\varphi$ true. Similarly, if $\varphi = \phi \vee \psi$, then $t$ making $\sigma_4^*(\varphi)$ true means either $\sigma_4^*(\phi)$ is made true by $t$ or $\sigma_4^*(\psi)$. Using I.H. we can conclude that $t \upharpoonright Prop$ makes $\phi$ true or $\psi$ true, which is equivalent to saying $t \upharpoonright Prop$ makes $\varphi$ true. For the converse, suppose $t \upharpoonright Prop$ makes $\varphi$ true, which implies it either makes $\phi$ or $\psi$ true. Using I.H. it implies $t$ makes either $\sigma_4^*(\phi)$ or $\sigma_4^*(\psi)$ true, which implies $t$ makes $\sigma_4^*(\varphi)$ true. This closes the induction clause and our claim is proven ☐

**Lemma 3.2.20.** *For every finite set of variables Prop, $\mathbb{C}(\mathcal{L}_{\wedge,\vee}[Prop]) \leq_{pc} \mathbb{C}(\mathcal{L}_{\mathsf{AOR}}[Prop^{\#}])$*

*Proof.* For any variable assignment $t$ on $Prop$, define

$$\gamma_4(t) = t \cup \{(x_0, 0)\}$$

The definition of $\tau_4$ is a little more involved. Let $x = \tau_4(\varphi)$.

$$\tau_4(x,(t,+)) = \{(t \restriction Prop, +)\} \quad \text{if } t(x_0) = 0$$
$$\tau_4(x,(t,-)) = \{(t \restriction Prop, -)\} \quad \text{if } t(x_0) = 0$$

Finally, if $(x,(t,+)) \in \mathcal{N}_{\mathbb{C}(\mathcal{L}_{\mathsf{AOR}}[Prop])}$ with $t(x_0) = 1$, then

$$\tau_4(x,(t,+)) = \{(t \restriction Prop, +), (t \restriction Prop, -)\}$$

It is again a routine verification to see that every condition of PC-reduction is satisfied by $\sigma_4, \tau_4, \gamma_4$ and $p(x) = 3$. ☐

**Theorem 3.2.21.** *The collection $\mathbb{C}(\mathcal{L}_{\mathsf{AOR}})$ does not have PSUC.*

*Proof.* We already know $\mathbb{C}(\mathcal{L}_{\vee,\wedge})$ does not have PSUC due to theorem 3.2.8. We again invoke theorem 2.5.2, notice that $\mathbb{C}(\mathcal{L}_{\wedge,\vee}[Prop])$ is PC-reducible to $\mathbb{C}(\mathcal{L}_{\mathsf{AOR}}[Prop^{\#}])$ via $\sigma_4, \tau_4, \gamma_4$, having $k = 2, m = 6$ and $p(x) = 2$ as constants. So $\mathbb{C}(\mathcal{L}_{\mathsf{AOR}})$ having PSUC means $\mathbb{C}(\mathcal{L}_{\wedge,\vee})$ has it as well, a contradiction. ☐

We are only left with the collection $\mathbb{C}(\mathcal{L}_{\mathsf{OAND}})$. Since it is the dual of $\mathbb{C}(\mathcal{L}_{\mathsf{AOR}})$, the proof of it's lack of PSUC follows almost the same proof trajectory. Due to the sake of brevity, we just motivate and state the theorems that will hold in this collection and leave it to the reader to reconstruct the proof on their own.

We start from a finite set of propositional variables, $Prop$ say, and define $Prop_{\#} = Prop \cup \{x_1\}$. As it was the case with all the other cases $\sigma_5^*$ from the concepts of $\mathbb{C}(\mathcal{L}_{\wedge,\vee}[Prop])$ to teh concept of $\mathbb{C}(\mathcal{L}_{\mathsf{OAND}}[Prop_{\#}])$ recursively.

$$\sigma_5^*(p) = p$$
$$\sigma_5^*(\phi \wedge \psi) = \mathsf{OAND}(\sigma_5^*(\phi), \sigma_5^*(\phi), \sigma_5^*(\psi))$$
$$\sigma_5^*(\phi \vee \psi) = \mathsf{OAND}(\sigma_5^*(\phi), \sigma_5^*(\psi), x_1)$$

We use the map $\sigma_5^*$, to define the map $\sigma_5$ as follows

$$\sigma_5(\varphi) = \mathsf{OAND}(\sigma_5^*(\varphi), \sigma_5^*(\varphi), x_1)$$

As the dual of the case with $\mathbb{C}(\mathcal{L}_{\mathsf{AOR}}[Prop^{\#}])$, for any variable assignment $t$ on $Prop_{\#}$, $t$ say, if $t(x_1) = 0$ then trivially $t \not\models \sigma_5(\varphi)$ for any concept $\varphi$ of $\mathbb{C}(\mathcal{L}_{\wedge,\vee}[Prop])$. Now we investigate case when $t(x_1) = 1$, which again is reminiscent of the proposition 3.2.19,

**Proposition 3.2.22.** *If $t$ is a variable assignment on $Prop_{\#}$ with $t(x_1) = 1$, and $\varphi$ is an element of $\mathbb{C}(\mathcal{L}_{\wedge,\vee}[Prop])$, then $t \restriction Prop$ makes $\varphi$ true iff $t$ makes $\sigma_5(\varphi)$ true.*

*Proof.* The proof is exactly the same as in 3.2.19, just change $x_0$ with $x_1$, $\sigma_4^*$ with $\sigma_5^*$, $\sigma_4$ with $\sigma_5$, and finally $t(x_0) = 0$ with $t(x_1) = 1$ in the proof. ☐

This smoothly transitions to the dual analogue of theorem 3.2.20, which is

**Lemma 3.2.23.** *For every finite set of variables Prop,* $\mathbb{C}(\mathcal{L}_{\wedge,\vee}[Prop]) \leq_{pc} \mathbb{C}(\mathcal{L}_{\mathsf{OAND}}[Prop_{\#}])$

*Proof.* For any variable assignment $t$ over $Prop$, define

$$\gamma_5(t) = t \cup \{(x_1, 1)\}$$

Let $x = \sigma_5(\varphi)$, then we define

$$\begin{aligned}\tau_5(x, (t, +)) &= (t \restriction Prop, +) \;\; \text{if } t(x_1) = 1\\ \tau_5(x, (t, -)) &= (t \restriction Prop, -) \;\; \text{if } t(x_1) = 1\end{aligned}$$

Finally, if $(x, (t, +)) \in \mathcal{N}_{\mathbb{C}(\mathcal{L}_{\mathsf{AOR}}[Prop])}$ with $t(x_1) = 0$, then $\tau_5(x, (t, -)) = \{(t \restriction, +), (t \restriction, -)\}$. One can verify that the conditions of PC reduction are satisfied. $\square$

**Theorem 3.2.24.** *The collection of classes* $\mathbb{C}(\mathcal{L}_{\mathsf{OAND}})$ *does not have PSUC.*

*Proof.* Suppose it did. Then we use $k = 2, m = 2$ and $p(x) = 3$ and invoke theorem 2.5.2 for the PC reduction from $\mathbb{C}(\mathcal{L}_{\wedge,\vee}[Prop])$ to $\mathbb{C}(\mathcal{L}_{\mathsf{OAND}}[Prop_{\#}])$. This will tell us that $\mathbb{C}(\mathcal{L}_{\wedge,\vee})$ has PSUC, a contradiction. $\square$

### 3.2.5 Main Result

We are almost ready to prove the main result of this section. We state the following result first.

**Lemma 3.2.25.** *If for some sets of propositional formulas $Pr, Pr'$, we have that if $\mathcal{F}(Pr) \subseteq \mathcal{F}(Pr')$ then $\mathbb{C}(\mathcal{L}_{Pr'})$ having PSUC implies $\mathbb{C}(\mathcal{L}_{Pr})$ has PSUC as well.*

*Proof.* Theorem 2.5.1 tells us $\mathbb{C}(\mathcal{L}_{Pr}[Prop]) \leq_{pc} \mathbb{C}(\mathcal{L}_{Pr'}[Prop])$. Now we can use $k = 2, m = 2$ and $p(x) = 2$ to invoke 2.5.2. $\square$

As a contraposition of the previous result, *not* having PSUC is *upwards closed.* Through our results [3.2.12], [3.2.13], [3.2.18], [3.2.21], [3.2.24] we have provided a classification of a splitting pair in the lattice of all propositional fragments. The splitting pair in the question is

$$\{\mathcal{F}(\wedge, \top, \bot),\ \mathcal{F}(\vee, \top, \bot),\ \mathcal{F}(\leftrightarrow, \top),\ \mathcal{F}(\mathsf{AOR}),\ \mathcal{F}(\mathsf{OAND}),\ \mathcal{F}(\mathsf{maj})\}$$

which corresponds to the anti-chain

$$\mathcal{S}_1 = \{\{\text{DM, MPT}_0^\infty, \text{MPT}_1^\infty\}, \{, \mathsf{V}, \text{A}\}\}$$

in Post's Lattice. Using lemma 3.2.25, our main result follows immediately

**Theorem 3.2.26.** *For any set of propositional formulas Pr, the following the equivalent:*

- *There exists a polynomial* $p(x,y)$ *s.t. for every finite Prop, and every* $\varphi \in \mathbb{C}(\mathcal{L}_{Pr}[Prop])$*, there is a set of labelled examples* $T$ *that uniquely characterizes* $\varphi$ *with* $|T| < p(|\varphi|, |Prop|)$

- $Cl(Func(Pr))$ *is a subset of either of the three* $(i)$ $Cl(\wedge, \bot, \top)$, $(ii)$ $Cl(\vee, \bot, \top)$ *or* $(iii)$ $Cl(\leftrightarrow, \bot)$

$\square$.

In fact this result can be modified to a certain extent, notice that lemma 3.2.25 also provides us an size bound for the unique characterizations, and every one of the fragments that have PSUC has linear size bound, in other words every unique characterization is bounded by $|Prop| + 1$. Now we state the following

**Corollary 3.2.26.1.** *For any set of propositional formulas* $Pr$ *exactly one of 1. or 2. holds.*

1. *For every finite Prop, and every concept* $\varphi$ *of* $\mathbb{C}(\mathcal{L}_{Pr}[Prop])$*, there is a set of labelled examples* $T$ *that uniquely characterizes* $\varphi$ *with* $|T| \leq |Prop| + 1$.
2. *The collection* $\mathbb{C}(\mathcal{L}_{Pr})$ *does not admit PSUC.* $\square$

We look back to the main source that encouraged us to talk about Unique Characterization, in other words we again analyze V.Dalmau's paper [Dal99], but from a learning theoretic point of view. Dalmau's main theorem in the paper mentioned above is as follows,

**Theorem 3.2.27** (V.Dalmau in [Dal99])**.** *For any set of boolean functions* $O$ *and a set of propositional variables Prop, precisely one of the following holds:*

*i.* $\mathbb{C}(\mathcal{L}_O[Prop])$ *is exactly learnable with* $|Prop| + 1$ *many membership queries.*

*ii.* $\mathbb{C}(\mathcal{L}_O[Prop])$ *is not polynomially predictable with membership queries under the assumption that any of the following three problems are intractable: testing quadratic residues modulo a composite, inverting RSA encryption, or factoring Blum integers.*

(If the reader wants to inquire more about the technical terms used in condition *ii.*, we would like to refer them to Angluin and Kharitonov's original work [AK95].) The result(s) we prove in this section is a slight variation to V.Dalmau's original work. Instead of working with various cryptographic assumptions, we will work with a notion which we will call *Learnability with Membership Queries*.

Our main objective is to establish an analogue of Dalmau's main theorem in terms of *learnability using polynomial queries*, but first we prove a theorem in the spirit of our result 3.2.26

**Theorem 3.2.28** (V. Dalmau in [Dal99])**.** *For any set of propositional formulas* $Pr$*, the following are equivalent*

*i.* $\mathbb{C}(\mathcal{L}_{Pr})$ *is learnable with membership queries.*

*ii.* $Cl(Pr)$ *is a subset of either of the three sets* $(i)\,Cl(\wedge, \top, \bot)$, $(ii)\,Cl(\vee, \top, \bot)$ *or* $(iii)\,Cl(\leftrightarrow, \bot)$.

The first step is to provide three algorithms that will learn $\mathbb{C}(\mathcal{L}_{\wedge,\top,\bot})$, $\mathbb{C}(\mathcal{L}_{\vee,\top,\bot})$ and $\mathbb{C}(\mathcal{L}_{\leftrightarrow,\bot})$ with polynomial queries. The algorithms will then in turn prove the *ii.* to *i.* part of our theorem. We remind the reader that $t_p^+$ and $t_p^-$ are special variable assignments s.t.

$$t_p^+(x) = 1 \iff x = p \quad \text{and} \quad t_p^-(x) = 0 \iff x = p$$

And **0** and **1** are variable assignments that assign all the propositional variables to 0 and 1 respectively.

**Data:** x is the number of propositional variables present in *Prop*
**if** ($\mathsf{MQ}(\mathbf{1}) = 0$) **then**
  Print $\bot$ and exit the program
**end**
$\varphi = \top$;
**for** $1 \leq i \leq x$ **do**
  **if** ($\mathsf{MQ}(t_{p_i}^-) = 0$) **then**
    $\varphi = \varphi \wedge p_i$
  **end**
**end**
Print $\varphi$

**Algorithm 1:** $\mathsf{Poly}(\wedge, \top, \bot)(x)$

**Data:** x is the number of propositional variables present in *Prop*
**if** ($\mathsf{MQ}(\mathbf{0}) = 1$ **then**
  Print $\top$ and exit the program
**end**
$\varphi = \bot$;
**for** $1 \leq i \leq x$ **do**
  **if** ($\mathsf{MQ}(t_{p_i}^+) = 1$) **then**
    $\varphi = \varphi \vee p_i$
  **end**
**end**
Print $\varphi$

**Algorithm 2:** $\mathsf{Poly}(\vee, \top, \bot)(x)$

**Lemma 3.2.29.** *For any set of propositional formulas Pr, if collection $Cl(Pr)$ is a subset of either (i) $Cl(\wedge, \top, \bot)$, (ii) $Cl(\vee, \top, \bot)$ or (iii) $Cl(\leftrightarrow, \top)$ then the collection $\mathbb{C}(\mathcal{L}_O)$ is learnable with membership queries.* □

*Proof of theorem 3.2.28.* After proving the *ii.* to *i.* direction, we prove the other direction. Again this proof will take us to the analysis of Post Lattice. Reconsider the splitting pair

$$\mathcal{S}_1 = \{\{\text{DM}, \text{MPT}_0^\infty, \text{MPT}_1^\infty\}, \{, \mathsf{V}, \text{A}\}\}$$

**Data:** x is the number of propositional variables present in $Prop$
$y \leftarrow \mathsf{MQ}(\mathbf{1})$;
Based on $y$ do either of the two cases

**Case 1:** y=1;
$\varphi = \bot \leftrightarrow \bot$ **for** $1 \leq i \leq x$ **do**
  **if** $\mathsf{MQ}(t^-_{p_i}) = 0$ **then**
    $\varphi = \varphi \leftrightarrow p_i$
  **end**
**end**
Print $\varphi$ and exit the program

**Case 2:** y=0;
$\varphi = \bot$ **for** $1 \leq i \leq x$ **do**
  **if** $\mathsf{MQ}(t^-_{p_i}) = 1$ **then**
    $\varphi = \varphi \leftrightarrow p_i$
  **end**
**end**
Print $\varphi$ and exit the program

**Algorithm 3:** $\mathsf{Poly}(\leftrightarrow, \bot)(x)$

Let us assume $Cl(Pr)$ is NOT a subset of $(i)$ $Cl(\wedge, \top, \bot)$, $(ii)$ $Cl(\vee, \top, \bot)$ and $(iii)$ $Cl(\leftrightarrow, \top)$ and $\mathbb{C}(\mathcal{L}_O)$ is learnable with polynomial membership queries. As it turns out, the former assumption says that $Cl(Pr)$ is a superset of $(iv)$ $Cl(\mathsf{maj})$, $(v)$ $Cl(\mathsf{OAND})$ or $(vi)$ $Cl(\mathsf{AOR})$ from the splitting pair we mentioned. According to our result 3.2.26, $\mathbb{C}(\mathcal{L}_{Pr})$ cannot have PSUC. On the other hand $\mathbb{C}(\mathcal{L}_{Pr})$ is learnable with polynomial membership queries implies $\mathbb{C}(\mathcal{L}_{Pr})$ has PSUC using theorem 2.4.16. hence a contradiction. So, $\mathbb{C}(\mathcal{L}_{Pr})$ is NOT learnable with membership queries. ☐

We now slightly refine the result we obtained just now. Notice that the algorithms we mentioned run in $|Prop| + 1$ computation steps.

**Corollary 3.2.29.1.** *For any set of propositional formulas $Pr$, for the collection $\mathbb{C}(\mathcal{L}_{Pr})$ precisely one of the following two conditions hold:*

*i.* $\mathbb{C}(\mathcal{L}_{Pr})$ *is learnable with polynomial membership queries that is witnessed by an algorithm that runs in $|Prop| + 1$ many computation steps.*

*ii.* $\mathbb{C}(\mathcal{L}_{Pr})$ *is NOT learnable with polynomial membership queries.*

*Proof.* Again consider the anti-chain $\mathcal{C}_1$, if $\mathbb{C}(\mathcal{L}_O)$ is a subset of $(i)$ $Cl(\wedge, \top, \bot)$, $(ii)$ $Cl(\vee, \top, \bot)$ and $(iii)$ $Cl(\leftrightarrow, \top)$, then our algorithms give us the condition $ii$. If $\mathbb{C}(\mathcal{L}_O)$ is NOT a subset of $(i)$ $Cl(\wedge, \top, \bot)$, $(ii)$ $Cl(\vee, \top, \bot)$ and $(iii)$ $Cl(\leftrightarrow, \top)$ then it is a superset of $(iv)$ $Cl(\mathsf{maj})$, $(v)$ $Cl(\mathsf{OAND})$ or $(vi)$ $Cl(\mathsf{AOR})$, but theorem 2.4.16 says they are not learnable. ☐

## 3.3 Upper bounds dependent on variables and Polynomial upper bounds dependent on variables

The results 3.2.26 and 3.2.26.1 that we unraveled in the first section, are in essence dichotomy theorems. On top of that, theorem 3.3.7.1 motivates us to once again to analyse the Post lattice and establish more characterizations in the spirit of result 3.2.26. Our dichotomies will be based on two different questions, namely:

1. Which (prop.) fragments have an upper-bound dependent on the variables ?
2. Which (prop.) fragments have a *polynomial* upper-bound dependent on the variables ?

The polynomial clause in the latter question means the following: A fragment has polynomial upper-bound dependent on the variables if the fragment has an upper-bound dependent on the variables, which is witness by a function, $\alpha$ say, where $\alpha(x) = p(x)$ where $p$ is a polynomial.

Based upon the two questions, we establish the following two dichotomies:

**Theorem 3.3.1.** *For any set of propositional formulas $Pr$ the following are equivalent,*

- *For every Prop, and every concept $\varphi$ of $\mathbb{C}(\mathcal{L}_{Pr}[Prop])$, there is an unique characterization $T_\varphi$ of $\varphi$ with $|T_\varphi| < 2^{(|vars(\varphi)|+1)}$*
- *$Cl(Pr)$ is a subset of either of the three (i) $Cl(\wedge, \vee, \bot, \top)$, (ii) $C(\neg, \bot)$.*

**Theorem 3.3.2.** *For any set of propositional formulas $Pr$ the following are equivalent,*

- *There exists a polynomial $p(x)$ s.t. for every Prop, and every concept $\varphi$ of $\mathbb{C}(\mathcal{L}_{Pr}[Prop])$, there is an unique characterization $T_\varphi$ that uniquely characterizes $\varphi$ with $|T_\varphi| < p(|vars(\varphi)|)$*
- *$Cl(Pr)$ is a subset of either of the three (i) $Cl(\wedge, \bot, \top)$, (ii) $Cl(\vee, \bot, \top)$ or (iii) $Cl(\neg, \top)$.*

PC reductions will again be vital in establishing these results. For the purpose of this section, we are interested in the fragments generated by the clones $\mathrm{PT}_1^\infty$, $\mathrm{PT}_0^\infty$ and AP. For the sake of simpler notation, we re-introduce the propositional formulas generating the fragments as follows:

$$\mathsf{AIMP}(x, y, z) = x \wedge (y \to z)$$
$$\mathsf{3XOR}(x, y, z) = x \oplus y \oplus z$$
$$\mathsf{OXOR}(x, y, z) = x \vee (y \oplus z)$$

### 3.3.1 Upper Bounds of the Fragment $\mathbb{C}(\mathcal{L}_{\mathsf{3XOR}})$

We start this section through the analysis of this particular fragment. The main property we will be concerned with is the following

**Definition 3.3.3.** *We say a collection of propositional concept classes $\mathcal{X}$ has as an upper bound dependent on variables if,*

1. *There exists a non-decreasing function $\alpha : \mathbb{N} \to \mathbb{N}$ s.t.*
2. *for every concept class $\mathcal{C} \in \mathcal{X}$ and every concept $\varphi$ of $\mathcal{C}$, there exists an unique characterization $T_\varphi$ of $\varphi$ s.t.*

$$|T_\varphi| < \alpha(|vars(\varphi)|)$$

In theorem 2.5.3 we can modify the meaning of 'size' for a concept. In particular for any formula $\varphi$, we can stipulate that $|vars(\varphi)|$ as the 'size' of $\varphi$. Hence we can adapt theorem 2.5.3 to preserve upper bounds on varibles.

One of the main aims of this section is to prove the following

**Theorem 3.3.4.** *There exists no upper bound dependent on variables for the collection $\mathbb{C}(\mathcal{L}_{\mathsf{3XOR}})$.*

We make the assumption, aiming for a contradiction, that $\mathbb{C}(\mathcal{L}_{\mathsf{3XOR}})$ has an upper-bound dependent on the variable, which is witnessed by $\alpha$. Let $\varphi \in \mathbb{C}(\mathcal{L}_{\mathsf{3XOR}}[Prop])$, now according to definition $\varphi \in \mathbb{C}(\mathcal{L}_{\mathsf{3XOR}}[Q])$ for any set $Q \supseteq Prop$. Furthermore, for every concept class $\mathbb{C}(\mathcal{L}_{\mathsf{3XOR}}[Q])$ containing $\varphi$, from our assumptions, there is an unique characterization of $\varphi$ size at-most $\alpha(vars(\varphi))$ w.r.t. $\mathbb{C}(\mathcal{L}_{\oplus,\top}[Q])$. In other words for every $\varphi$, there is an unique characterization $T_\varphi$ that is unaffected by the size of $Prop$ for any concept class that contains it.

Suppose we have fixed a concept $\varphi$ s.t. $\varphi \in \mathbb{C}(\mathcal{L}_{\mathsf{3XOR}}[Prop])$, and $T_\varphi$ is an unique characterization of $\varphi$. We assume $\varphi$ is of the following form:

$$\varphi := p_{i_1} \oplus \cdots \oplus p_{i_n}$$

where $n$ is odd. By $Tr(T_\varphi)$ we mean the following set:
For each propositional variable $p$ occurring in $\varphi$, we add $(t_p, +) \in Tr(T)$, where $t_p$ is the variable assignment that makes only $p$ true and other variables false. Now if $(t, -) \in T$ then $(t, -) \in Tr(T)$. Lastly, we consider the case when $t \models \varphi$ and $(t, +) \in T$. In this case we put $(t', -) \in Tr(T)$, where $t'(p) = 0$ for any $p$ occurring in $\varphi$ and $t(q) = t'(q)$ for any $q$ not occurring in $\varphi$. We claim the following:

**Lemma 3.3.5.** *If $T$ uniquely characterizes $\varphi$ w.r.t. $\mathbb{C}(\mathcal{L}_{\mathsf{3XOR}}[Prop])$, then so does $Tr(T_\varphi)$.*

*Proof.* Firstly for all the $(t_p, +) \in Tr(T_\varphi)$ implies any $\phi$ fitting $Tr(T_\varphi)$ must have all the propositional variables $p$ in $\varphi$ occurring it. Suppose $\phi$ fits $Tr(T_\varphi)$, we can assume $\phi = \varphi \oplus \psi$.

Now, if $p \not\cong \psi$, then there is a $(t, q) \in T$ that distinguishes between them, where $q \in \{+, -\}$. If $t \not\models \varphi$, then $t \in Tr(T_\varphi)$ and we are done. Otherwise $t \models \varphi$, which gives us $t \not\models \varphi \oplus \psi$. Then we have that $t \models \psi$ (as $t \models \varphi$ and if $t \not\models \psi$ would lead to $t \models \varphi \oplus \psi$), but this tells us $t' \models \psi$ (as none of the $p$ in $\varphi$ occur in $\psi$ and definition of $t'$) and $t' \not\models \varphi$. Hence, $(t', -) \in Tr(T_\varphi)$ differentiates $\phi$ and $\varphi$. So any concept fitting $Tr(T_\varphi)$ should be equivalent to $\varphi$. This closes our proof. $\square$

A bound on the size of $Tr(T_\varphi)$ can be easily given, it easy to see that

$$|T_\varphi| \leq |T| + |vars(\varphi)| \leq \alpha(vars(\varphi)) + |vars(\varphi)|$$

If we define $\beta(x) = \alpha(x) + x$ then $Tr(T_\varphi)$ is bounded by $\beta(vars(\varphi))$, which is again a function dependent (only) on variables. Furthermore our choice of $\varphi$ was arbitrary, so we can find $Tr(T_\varphi)$ for every $\varphi$ and every concept class of the form $\mathbb{C}(\mathcal{L}_{\oplus,\top}[Prop])$. As opposed to $T_\varphi$, the unique characterization $Tr(T_\varphi)$ gives us more information and hence it is easier to work with. One obvious thing is $Tr(T_\varphi)$ always has exactly $vars(\varphi)$ many positive examples and at-most $\alpha(vars(\varphi))$ many negative ones.

Now if we change our concept class from $\mathbb{C}(\mathcal{L}_{\mathsf{3XOR}}[Prop])$ to $\mathbb{C}(\mathcal{L}_{\mathsf{3XOR}}[Q])$, where $Q \supseteq Prop$, we still have $\varphi$ as a concept of $\mathbb{C}(\mathcal{L}_{\mathsf{3XOR}}[Q])$. From the previous assumptions, we get that $\mathbb{C}(\mathcal{L}_{\mathsf{3XOR}}[Q])$ also has an unique characterization, $T_\varphi^Q$ say, of size at-most $\alpha(vars(\varphi))$, and we define $Tr(T_\varphi^Q)$ as above. Finally the previous theorem should tell us $Tr(T_\varphi^Q)$ also uniquely characterize $\varphi$ (since the previous lemma does not make use of the concept class class $\mathbb{C}(\mathcal{L}_{\mathsf{3XOR}}[Prop])$).

We can take $Q$ to be arbitrarily large, and the larger it is, there are more formulas of the form

$$\Psi(q, r) = \varphi \oplus q \oplus r$$

where $q, r$ are propositional variables from $Q$ not occurring in $\varphi$ and $q \neq r$. Now, $Tr(T_\varphi^Q)$ is tasked with differentiating all such $\Psi(q, r)$ from $\varphi$. Notice that any positive example of $Tr(T_\varphi^Q)$ cannot distinguish between $\varphi$ and $\Psi(q, r)$ for any $q$, so the separation must be done by a negative example. Furthermore the size of $Tr(T_\varphi^Q)$ should be bound by $\beta(vars(\varphi))$.

Alternatively, every variable assignment $t$ on $Q$ naturally gives rise to a set $P(t) \subseteq Q$ s.t.

$$q \in P(t) \iff t(q) = 1$$

This kind of representation of variable assignments helps us to develop a set-theoretic formulation of our statements. Suppose a labelled assignment $(t, -)$ is in $Tr(T_\varphi^Q)$, upon fixing $q_0, r_0$ we see that $t$ distinguishes between $\varphi$ and $\Psi(q_0, r_0)$ if and only if precisely $t(q_0) = 1 - t(r_0)$. In other words, the value of $t$ disagree on $q_0$ and $r_0$, which is in turn equivalent to stating $|P(t) \cap \{q_0, r_0\}| = 1$. we claim the following which is obvious from the discussion we had.

**Lemma 3.3.6.** *For every $q, r \in Q$ s.t. $q, r$ does not occur in $\varphi$ and $q \neq r$, we have that*

1. *there exists a labelled variable assignment $(t, -) \in Tr(T_\varphi^Q)$ s.t.*
2. $|P(t) \cap \{q, r\}| = 1$ ☐

But the previous lemma gives way to a contradiction, as mentioned above the size of $Tr(T_\varphi^Q)$ only depends on $vars(\varphi)$ and not on the size of $Q$. The following theorem states the opposite

**Theorem 3.3.7** (see [(ht]). *Let $X$ be any set and $A \subseteq \mathcal{P}(X)$ be s.t. for any $x, y \in X$ there is some $Y \in A$ with*

$$|Y \cap \{x, y\}| = 1$$

*It follows that $|A| \geq log_2(|X|)$.* ☐

In particular, if we define a set

$$P(Tr(T_\varphi^Q)) = \{P(t) : t \in Tr(T_\varphi^Q)\}$$

and notice that $Q \setminus vars(\varphi)$ and $P(Tr(T_\varphi^Q))$ satisfies the antecedent of theorem 3.3.7 as $X$ and $A$ respectively. It easy to see if we take $Q$ to be large enough we can surpass the bound $\beta(vars(\varphi))$, hence a contradiction! We formally state what we have proven.

**Corollary 3.3.7.1.** *The collection $\mathbb{C}(\mathcal{L}_{\mathsf{3XOR}})$ does not admit have an upper bound on the variables. In-fact no formula $\varphi$ from the concept class $\mathbb{C}(\mathcal{L}_{\mathsf{3XOR}}[Prop])$ with $vars(\varphi) \subseteq Prop$, has an unique characterization dependent on variables.* ☐

**Corollary 3.3.7.2.** *The fragment $\mathbb{C}(\mathcal{L}_{\mathsf{3XOR}}[\mathsf{Inf}Prop])$ does not have finite characterization.*

*Proof.* Suppose it has finite characterization. Let $\varphi$ be a concept of $\mathbb{C}(\mathcal{L}_{\mathsf{3XOR}}[\mathsf{Inf}Prop])$, so it follows that $\varphi$ is also a concept of $\mathbb{C}(\mathcal{L}_{\mathsf{3XOR}}[Prop])$, where $vars(\varphi) \subseteq Prop$.

Suppose $T_\varphi$ is an unique characterization of $\varphi$ w.r.t. $\mathbb{C}(\mathcal{L}_{\mathsf{3XOR}}[\mathsf{Inf}Prop])$. Notice that

$$T'_\varphi = \{t \restriction Prop : t \in T_\varphi\}$$

is also an unique characterization of $\varphi$ w.r.t. $\mathbb{C}(\mathcal{L}_{\mathsf{3XOR}}[Prop])$. Now, notice that $|T_\varphi|$ is constant, and does not depend on $Prop$, which is a contradiction to 3.3.7.1. ☐

### 3.3.2 Positive results and reduction from $\mathbb{C}(\mathcal{L}_{\wedge,\to}[Prop])$ to $\mathbb{C}(\mathcal{L}_{\mathsf{AIMP}}[Prop^*])$

We start off by looking into the collection $\mathbb{C}(\mathcal{L}_{\vee,\wedge,\top,\bot})$. Let us develop the idea why it is sufficient to prove the following proposition to prove that $\mathbb{C}(\mathcal{L}_{\vee,\wedge,\top,\bot})$ does have an upper-bound dependent on variables.

**Proposition 3.3.8.** *For any set of variables Prop and any concept $\varphi$ of the concept class $\mathbb{C}(\mathcal{L}_{\vee,\wedge,\top,\bot})$, the following are equivalent*

- *$\varphi$ is uniquely characterized by $m$ positive examples and $n$ negative examples.*
- *$\varphi$ is equivalent to a DNF with $m$ terms and to a CNF with $n$ clauses.*

Indeed, given any concept $\varphi$ of $\mathbb{C}(\mathcal{L}_{\vee,\wedge,\top,\bot}[Prop])$, we can use the distributive laws and De Morgan Laws to transform $\varphi$ into a CNF and a DNF. Perhaps an example might help, suppose $\varphi = x \wedge (z \vee (y \wedge w))$. Clearly $\varphi$ is neither in CNF nor in DNF, but we can use the distributive laws to change it to a CNF, namely $\varphi \cong x \wedge (z \vee y) \wedge (z \vee w)$, and subsequently use De Morgan's Laws to change it to a DNF, namely $\varphi \cong (x \wedge z \wedge w) \vee (x \wedge y \wedge z \wedge w)$. Moreover, it can be seen that using distributive laws and De Morgan's Laws do not change the number of variables that we had in the original formula. Hence we can see that $\varphi$ can be characterized by an unique characterization that contains at-most $2^{|vars(\varphi)|}$ positive examples and at-most $2^{|vars(\varphi)|}$ negative examples. So in total, we can an unique characterization of size $2^{|vars(\varphi)|+1}$.

*Proof of proposition 3.3.8.* See [ABCS92] for the proof. ☐

After dealing with the collection $\mathbb{C}(\mathcal{L}_{\vee,\wedge,\top,\bot})$, we deal with the collection $\mathbb{C}(\mathcal{L}_{\neg,\bot})$. It is a much simpler fragment since the concepts of $\mathbb{C}(\mathcal{L}_{\neg,\bot}[Prop])$ can be precisely of the form $\neg p, p, \bot, \neg\bot$. One might infer from the simple nature of this fragment that it will allow much nicer upper bounds, and indeed it does. To prove it rigorously we introduce some notations. Consider the example space of the concept class $\mathbb{C}(\mathcal{L}_{\neg,\bot}[Prop])$, i.e the set of all variable assignments on *Prop*. We define the following the variable assignments $t_p^+$ and $t_p^-$,

$$t_p^+(x) = 1 \iff x = p \qquad t_p^-(x) = 0 \iff x = p$$

We define $\mathbf{0}$ to be the variable assignment that assigns every propositional variable of *Prop* to 0. Similarly, $\mathbf{1}$ is the variable assignment that assigns every propositional variable of *Prop* to 1

**Proposition 3.3.9.** *The collection $\mathbb{C}(\mathcal{L}_{\neg,\bot})$ has an upper bound dependent on variables. Moreover the function witnessing the upper-bound is a constant.*

*Proof.* Suppose $\varphi$ is a concept of $\mathbb{C}(\mathcal{L}_{\neg,\bot}[Prop])$ for some arbitrary set of variables *Prop*. We do a case analysis, firstly let $\varphi = p$. We claim that $T_\varphi = \{(t_p^+, +), (\mathbf{0}, -)\}$ uniquely characterizes $\varphi$, where $\mathbf{0}$ is the variable assignment that maps every propositional variable to 0. It is easy to see that $\bot, \neg\bot$ cannot fit $T_\varphi$, since we have both a positive and negative example in our set. Similarly any formula of the form $\neg q$ cannot fit $T_\varphi$, if it did then $\neg q$ should satisfy $\mathbf{0} \not\models \neg q$ which is a contradiction. Now finally if $q \neq p$ fits $T_\varphi$ then $t_p^+$ would be a negative example of $q$, so it cannot fit $T_\varphi$. Hence, $T_\varphi$ is an unique characterization.

If $\varphi = \neg p$ then we define $T_\varphi = \{(t_p^-, +), (\mathbf{1}, -)\}$ as the set uniquely characterizing $\varphi$, where $\mathbf{1}$ again is the variable assignment that maps every propositional variable to 1. It follows through a similar argument that $T_\varphi$ again uniquely characterizes $\varphi$.

For the case of $\varphi = \bot$, we nominate $T_\varphi = \{(\mathbf{1}, -), (\mathbf{0}, -)\}$ to uniquely characterize it. $\neg\bot$ cannot fit $T_\varphi$ negative examples so it cannot $T_\varphi$. Now formulas of the form $p$ also cannot fit $T_\varphi$ since $\mathbf{1} \models p$ and similarly formulas of the form $\neg p$ cannot fit $T_\varphi$ since $\mathbf{0} \models \neg p$. For the final case let $\varphi = \neg p$, through a similar argument as before we can establish that $T_\varphi = \{(\mathbf{1}, +), (\mathbf{0}, -)\}$ uniquely characterizes $\varphi$.

Lastly, notice that in each of the cases we have $|T_\varphi| \leq 2$, so there is a finite upper bound for the collection $\mathbb{C}(\mathcal{L}_{\neg,\bot})$. ☐

**Lemma 3.3.10.** *The class $\mathbb{C}(\mathcal{L}_{\wedge,\rightarrow})$ does not have an upper bound dependent on variables.*

*Proof.* Upon a glance at the Post's Lattice we can see that $Cl(\mathsf{3XOR}) \subseteq Cl(\rightarrow, \wedge)$. So we can conclude that the concepts of $\mathbb{C}(\mathcal{L}_{\mathsf{3XOR}}[Prop])$ is a subset of the concepts of $\mathbb{C}(\mathcal{L}_{\rightarrow,\wedge}[Prop])$. Hence, if $\mathbb{C}(\mathcal{L}_{\rightarrow,\wedge})$ has an upper bound dependent on the variables then so does $\mathbb{C}(\mathcal{L}_{\mathsf{3XOR}})$, a contradiction. Therefore, $\mathbb{C}(\mathcal{L}_{\rightarrow,\wedge})$ cannot have an upper bound dependent on variables. ☐

The strategy we employ is in line with the strategy in the first section - we show via PC reductions that $\mathbb{C}(\mathcal{L}_{\mathsf{AIMP}})$ does not have an upper bound dependent on variables, subsequently we establish that *not* having an upper bound dependent on variables is *upwards closed*, these two together give up the segment of Post's Lattice that would not not have an upper bound dependent on variables.

We now move onto the reductions. We remind the reader that $Prop_{\#} = Prop \cup \{x_0\}$, where $x_0$ does not belong to $Prop$. We again define a map $\sigma_6$ from the concepts of $\mathbb{C}(\mathcal{L}_{\mathsf{3XOR}}[Prop])$ to the concepts of $\mathbb{C}(\mathcal{L}_{\mathsf{OXOR}}[Prop_{\#}])$. For any propositional variable $p_i$, we have $\sigma_6(p_i) = p_i$ and for more complex formulas $\varphi = p_{i_1} \oplus \cdots \oplus p_{i_{2m+1}}$,

$$\sigma_6(\varphi) = x_0 \vee (p_{i_1} \oplus \cdots \oplus p_{i_{2m+1}})$$

We define a function $\tau_6$. Let $(c, (t, p))$ be an ordered pair where $c$ is a concept of $\mathbb{C}(\mathcal{L}_{\mathsf{OXOR}}[Prop_{\#}])$ and $(t, p)$ is a labelled example of $\mathbb{C}(\mathcal{L}_{\mathsf{OXOR}}[Prop_{\#}])$ s.t. $c$ fits $\{(t, l)\}$, we define

$$\tau_6(c, (t, p)) = \{(t \cup \{(x_0, 0)\}, +), (t \cup \{(x_0, 0)\}, -)\}$$

whenever $c$ is not in the range of $\sigma_6$. Now suppose $c$ is in the range of $\sigma_6$, we have that $\tau_6(c, (t, p)) = \{(t \restriction Prop, p)\}$ whenever $t(x_0) = 0$. Additionally if $c$ is in the range of $\sigma_6$, and $t(x_0) = 1$, then $t \models c$ and hence we define $\tau_6(c, (t, p)) = \{(\mathbf{0}, -)\}$ whenever $t(x_0) = 1$.

**Lemma 3.3.11.** *If $\varphi$ is a concept of $\mathbb{C}(\mathcal{L}_{\mathsf{3XOR}}[Prop])$, then $\varphi$ fits $\tau_6(\sigma_6(\varphi), (t, p))$.*

*Proof.* It is easy to see that $f(\varphi)$ fits $\tau_6(\sigma_6(\varphi), (t, p)) = \{(t, p)\}$ whenever $t(x_0) = 1$ (in this case $t = \mathbf{0}$ and $p = +$). Furthermore $\tau_6(\sigma_6(\varphi), (t, p)) = \{(t \restriction Prop, l)\}$ whenever $t(x_0) = 0$, but remember that $\sigma_6(\varphi) = x_0 \vee \varphi$. So $t \models x_0 \vee \varphi$ implies $t \models \varphi$, and hence $(t \restriction Prop) \models \varphi$. ☐

**Lemma 3.3.12.** *If $\phi, \psi$ fit $\tau_6(\sigma_6(\varphi), (t, p))$ then $\sigma_6(\phi), \sigma_6(\psi)$ fits $(t, p)$.*

*Proof.* Again our proof is on how $t$ assigns the variable $x_0$. If $t(x_0) = 1$ then for any $\phi, \psi$, $\sigma_6(\phi), \sigma_6(\psi)$ fits $\tau_6(\sigma_6(\varphi), (t, p))$ (here $p$ is $+$).

Now fix $t(x_0) = 0$ then for any $\psi, \phi, \varphi$ s.t. $\psi, \phi$ fits $\tau_6(\sigma_6(\varphi), (t, p))$ means $\phi$ and $\psi$ also fit $\{(t \restriction Prop), p\}$ and $\{(t \restriction Prop)\}$. Now, we get that $\sigma_6(\phi)$ fit $\tau_6(\sigma_6(\varphi), (t, l))$ from the following equivalence(s) for the case where $t(x_0) = 0$:

$$t \models \sigma_6(\phi) \iff t \models x_0 \vee \phi \iff t \models \phi \iff (t \restriction Prop) \models \phi$$

A similar series of equivalences follow for $\psi$, and hence $\sigma_6(\psi)$ fit $\tau_6(\sigma_6(\varphi), (t, p))$ ☐

We now provide a function $\gamma_6$ from the example space of $\mathbb{C}(\mathcal{L}_{\mathsf{3XOR}}[Prop])$ to the example space of $\mathbb{C}(\mathcal{L}_{\mathsf{OXOR}}[Prop^{\#}])$. We simply define

$$\gamma_6(t) = t \cup \{x_0, 0\}$$

It is a trivial exercise to prove the following

**Lemma 3.3.13.** *If $t$ is a positive (negative) example of $\varphi$ then $\gamma_6(\varphi)$ is a positive (negative) example of $\sigma_6(\varphi)$, where $\varphi$ is a concept of $\mathbb{C}(\mathcal{L}_{\mathsf{3XOR}}[Prop])$.* ☐

**Lemma 3.3.14.** *For any Prop, the concept class $\mathbb{C}(\mathcal{L}_{\mathsf{3XOR}}[Prop]) \leq_{pc} \mathbb{C}(\mathcal{L}_{\mathsf{OXOR}}[Prop_{\#}])$*

*Proof.* The functions witnessing the reduction are $\sigma_6, \tau_6, \gamma_6$. The polynomial we choose is a constant $p(x) = 2$. Lemmas 3.3.11, 3.3.12 and 3.3.13 tells us $\tau_6$ and $\gamma_6$ satisfy conditions 1,2 and 4 of our reduction, and $p$ trivially satisfies our claims. ☐

**Theorem 3.3.15.** *The collection $\mathbb{C}(\mathcal{L}_{\mathsf{OXOR}})$ does not have an upper bound dependent on variables.*

*Proof.* Notice that $|vars(\sigma_6(\varphi))| = |vars(\varphi)| + 1$ for any concept $\varphi$ of $\mathbb{C}(\mathcal{L}_{\mathsf{3XOR}}[Prop])$. We use this fact along with theorem 3.3.14 to invoke lemma 2.5.3. This would tell us that the collection $\mathbb{C}(\mathcal{L}_{\mathsf{3XOR}}[Prop])$ has an upper-bound dependent on variables, a contradiction. $\square$

### 3.3.3 Reduction from $\mathbb{C}(\mathcal{L}_{\to,\wedge})$ to $\mathbb{C}(\mathcal{L}_{\mathsf{AIMP}})$

After dissecting the collection $\mathbb{C}(\mathcal{L}_{\mathsf{OXOR}})$ for variable dependent upper bounds, we turn our attention to the collection $\mathbb{C}(\mathcal{L}_{\mathsf{AIMP}})$. We want to establish a similar result as in 3.3.15 for the collection $\mathbb{C}(\mathcal{L}_{\mathsf{AIMP}})$.

**Lemma 3.3.16.** *The collection $\mathbb{C}(\mathcal{L}_{\wedge,\to})$ does not admit an upper bound dependent on variables.*

*Proof.* Notice that the concept class $\mathbb{C}(\mathcal{L}_{\mathsf{3XOR}}[Prop])$ is a sub concept class of the concept class $\mathbb{C}(\mathcal{L}_{\wedge,\to}[Prop])$, for any $Prop$. Now, if $\mathbb{C}(\mathcal{L}_{\wedge,\to})$ has an upper bound dependent on variables then so does $\mathbb{C}(\mathcal{L}_{\mathsf{3XOR}})$, a contradiction. $\square$

We move forward with our definition of $\sigma_7^*$. For every propositional variable $p_i$, we define $\sigma_7^*(p_i) = p_i$. Now suppose for any concepts $\varphi, \psi$ of $\mathbb{C}(\mathcal{L}_{\wedge,\to}[Prop])$

$$\sigma_7^*(\varphi \wedge \psi) = \sigma_7^*(\varphi) \wedge (x_1 \to \sigma_7^*(\psi))$$
$$\sigma_7^*(\varphi \to \psi) = x_1 \wedge (\sigma_7^*(\varphi) \to \sigma_7^*(\psi))$$

We now define a function $\sigma_7$, with the aid of $\sigma_7^*$. Define

$$\sigma_7(\varphi) = x_1 \wedge (x_1 \to \sigma_7^*(\varphi))$$

**Proposition 3.3.17.** *Let $t$ be a variable assignment. If $t(x_1) = 1$ then, $t \models \sigma_7(\varphi)$ if and only if $t \models \sigma_7^*(\varphi)$. Additionally, if $t(x_1) = 0$ then $t \not\models \sigma_7(\varphi)$.* $\square$

**Lemma 3.3.18.** *Let $t$ be a variable assignment on $Prop_{\#}$. If $t(x_1) = 1$ then,*

$$t \models \varphi \iff (t \restriction Prop) \models \sigma_7^*(\varphi)$$

.

*Proof.* The proof is by induction on the complexity of formulas. The base case, where $\varphi$ is a propositional formula, $p_i$ say, we can see that whenever $t(x_1) = 1$, $\sigma_7(\varphi) = p_i$. It is easy to infer that our claim holds.

Assume our claim holds for any $\varphi$ of complexity $< n$. Now suppose $\varphi$ is a formula of complexity $n$, this means $\varphi$ is either of the two forms $\psi \to \theta$ or $\psi \wedge \theta$, where $\psi, \theta$ are of complexity $< n$. For the latter case, if $t(x_1) = 1$, then $\sigma_7^*(\psi \wedge \theta) \cong \sigma_7^*(\psi) \wedge \sigma_7^*(\theta)$. Now, using our Induction hypothesis we conclude that $t \models \sigma_7^*(\psi) \wedge \sigma_7^*(\theta)$ if and only if $t \models \psi \wedge \theta$.

We now look at the case when $\varphi = \psi \to \theta$, again if $t(x_1) = 1$ then following the definition $\sigma_7^*(\psi \to \theta) \cong \sigma_7^*(\psi) \to \sigma_7^*(\theta)$. Again using our Induction hypothesis we have $t \models \sigma_7^*(\psi) \to \sigma_7^*(\theta)$ if and only if $t \models \psi \to \theta$. This completes the proof. $\square$

We now define a function $\tau_7$ on an ordered pair $(c,(t,p))$ where $c$ is a concept of $\mathbb{C}(\mathcal{L}_{\mathsf{AIMP}}[Prop^{\#}])$ and $(t,p)$ is a labelled example of $\mathbb{C}(\mathcal{L}_{\mathsf{AIMP}}[Prop^{\#}])$. If $c$ does not belong to the range of $\sigma_7$ and if $c$ belong to the range of $\sigma$ and $t(x_1)=0$, then

$$\tau_7(c,(t,p)) = \{(t \upharpoonright Prop, +), (t \upharpoonright Prop, -)\}$$

On the other hand, if $c$ does belong to the range of $\sigma_7$ then

$$\tau_7(c,(t,p)) = \{(t \upharpoonright Prop, p)\}$$

whenever $t(x_1)=1$. Define a function $\gamma_7$ from the example space of $\mathbb{C}(\mathcal{L}_{\wedge,\rightarrow}[Prop])$ to the example space of $\mathbb{C}(\mathcal{L}_{\mathsf{AIMP}}[Prop^{\#}])$. For any example $t$ of $\mathbb{C}(\mathcal{L}_{\wedge,\rightarrow}[Prop])$ we define

$$\gamma_7(t) = t \cup \{x_1, 1\}$$

**Lemma 3.3.19.** *For any concept $\varphi$ and any example $t$ of $\mathbb{C}(\mathcal{L}_{\wedge,\rightarrow}[Prop])$, if $t$ is a positive (negative) example of $\varphi$ then $\gamma_7(t)$ is a positive (negative) example of $\varphi$.* □

**Lemma 3.3.20.** *For any concept $\varphi$ of $\mathbb{C}(\mathcal{L}_{\wedge,\rightarrow}[Prop])$, if $\psi$ fits $\tau_7(\gamma_7(\varphi),(t,p))$ then $\gamma_7(\psi)$ fits $(t,p)$, where $(t,p)$ is a labelled example of $\mathbb{C}(\mathcal{L}_{\mathsf{AIMP}}[Prop^{\#}])$.*

*Proof.* Fix any concept $\varphi$ of $\mathbb{C}(\mathcal{L}_{\wedge,\rightarrow}[Prop])$ and a labelled example $(t,p)$ of $\mathbb{C}(\mathcal{L}_{\mathsf{AIMP}}[Prop^{\#}])$. If $t(x_1)=0$ then $p=-$ and trivially, due to the way $\sigma_7$ is defined, $\sigma_7(\varphi)$ fits $(t,p)$. If $t(x_1)=1$ then lemma 3.3.18 settles our claim. □

**Theorem 3.3.21.** *For any Prop, $\mathbb{C}(\mathcal{L}_{\wedge,\rightarrow}[Prop]) \leq_{pc} \mathbb{C}(\mathcal{L}_{\mathsf{AIMP}}[Prop^{\#}])$*

*Proof.* The functions witnessing the reductions are as follows $\sigma_7, \tau_7, \gamma_7$ and we can choose the polynomial to be $p(x)=2$. Lemmas 3.3.19 and 3.3.20 tell us that condition 1, 2 and 4 of the reduction are satisfied and from the way $\tau_7$ is defined we get that condition 3 of the reduction is satisfied as well. □

**Corollary 3.3.21.1.** *The collection $\mathbb{C}(\mathcal{L}_{\mathsf{AIMP}})$ does not have an upper bound dependent on variables.*

*Proof.* Suppose the collection $\mathbb{C}(\mathcal{L}_{\mathsf{AIMP}})$ admit an upper bound dependent on variables. Notice that $|vars(\zeta_6(\varphi))| = |vars(\varphi)| + 1$, and hence using this fact along with 3.3.21, we invoke theorem 2.5.3 which tells us that the collection $\mathbb{C}(\mathcal{L}_{\wedge,\rightarrow})$ should have an upper bound dependent on variables, a contradiction. □

### 3.3.4 Main Results

Consider the following splitting pair of the Post's Lattice

$$\{\{\text{U, M }\}, \{\text{AP, PT}_0^\infty\text{, PT}_1^\infty\}\}$$

So for any set of boolean functions $O$, $Cl(O)$ is either $(i)$ a subset of U or M or $(ii)$ a superset of AP, $\text{PT}_0^\infty$, $\text{PT}_1^\infty$. Expressing it in the terms of concept classes, the concept space of every concept class is either $(i)$ a subset of the concepts of $\mathbb{C}(\mathcal{L}_{\neg,\bot}[Prop])$ or $\mathbb{C}(\mathcal{L}_{\wedge,\vee,\top,\bot}[Prop])$ or $(ii)$ a superset of the concepts of $\mathbb{C}(\mathcal{L}_{\mathsf{AIMP}}[Prop])$ or $\mathbb{C}(\mathcal{L}_{\mathsf{OXOR}}[Prop])$. We now state a lemma which will nicely transition into the main result

**Lemma 3.3.22.** *Let $\mathbb{C}(\mathcal{L}_{Pr})$ and $\mathbb{C}(\mathcal{L}_{Pr'})$ be two collections of concept classes s.t. $Cl(Pr) \subseteq Cl(Pr')$. Now, if $\mathbb{C}(\mathcal{L}_{Pr'})$ has an upper bound dependent on variables then so does $\mathbb{C}(\mathcal{L}_{Pr})$.*

*Proof.* Let $\varphi$ be a concept from a concept class $\mathbb{C}(\mathcal{L}_{Pr}[Prop])$ which in turn is from the collection $\mathbb{C}(\mathcal{L}_{Pr})$. We assume that $\mathbb{C}(\mathcal{L}_{Pr'})$ has a variable dependent upper bound witnessed by $\alpha$ and $Cl(Pr) \subseteq Cl(Pr')$.

Now $\varphi$ is also a concept of $\mathbb{C}(\mathcal{L}_{Pr'}[Prop])$ from our assumptions. Now, our assumptions also say that there is an unique characterization of $\varphi$ of size $\alpha(|vars(\varphi)|)$, $T_\varphi$ say, w.r.t. to $\mathbb{C}(\mathcal{L}_{Pr'}[Prop])$. But $T_\varphi$ also is an unique characterization w.r.t $\mathbb{C}(\mathcal{L}_{Pr}[Prop])$ (since the concepts of $\mathbb{C}(\mathcal{L}_{Pr}[Prop])$ are a subset of the concepts of $\mathbb{C}(\mathcal{L}_{Pr'}[Prop])$). As $\varphi$ was arbitrary we get that $\alpha$ also witnesses the variable dependent upper bound for $\mathbb{C}(\mathcal{L}_{Pr})$. $\square$

**Corollary 3.3.22.1.** *For any set of propositional formulas $Pr$ the following are equivalent,*

- *For every Prop, and every concept $\varphi$ of $\mathbb{C}(\mathcal{L}_{Pr}[Prop])$, there is an unique characterization $T_\varphi$ of $\varphi$ with $|T_\varphi| < 2^{(|vars(\varphi)|+1)}$*
- *$Cl(Pr)$ is a subset of either of the two $(i)$ $Cl(\wedge, \vee, \bot, \top)$, $(ii)$ $C(\neg, \bot)$.*

*Proof.* The direction from the second point to the first point is straightforward and is an easy application of the previous lemma. For the other direction, suppose $Cl(Pr)$ is not a subset of $Cl(\wedge, \vee, \bot, \top)$ nor of $Cl(\neg, \bot)$. By inspecting the Post's Lattice we get that $Cl(Pr)$ must be a superset of either of D, $\mathrm{PT}_0^\infty$ or $\mathrm{PT}_1^\infty$, if $\mathbb{C}(\mathcal{L}_{Pr})$ has an upper bound dependent on variables then so does one of $\mathbb{C}(\mathcal{L}_{\mathsf{3XOR}})$, $\mathbb{C}(\mathcal{L}_{\mathsf{OXOR}})$ or $\mathbb{C}(\mathcal{L}_{\mathsf{AIMP}})$, following the previous lemma. As it leads to a contradiction, $Cl(Pr)$ does not have an upper-bound dependent on variables. The contradiction gives us the desired direction of proof. $\square$

The next main result of this section, is in some sense, a refinement of the results 3.2.26 and 3.3.22.1. Our main goal is to base the dichotomy on *polynomial sized* upper-bounds dependent on variables, instead of just 'upper bounds'.

**Definition 3.3.23.** *A collection of boolean concept classes $\mathbb{C}(\mathcal{L}_{Pr})$ has polynomial sized upper bound dependent on variables if $\mathbb{C}(\mathcal{L}_{Pr})$ has an upper bound dependent on variables, and the function witnessing it, $\alpha$ say, is a uni-variate non-decreasing polynomial.*

The correspondence between polynomial upper bounds dependent on variables and PSUC is explained by the result below

**Lemma 3.3.24.** *If a collection $\mathbb{C}(\mathcal{L}_{Pr})$ has a polynomial sized upper bound dependent on variables, then $\mathbb{C}(\mathcal{L}_{Pr})$ has PSUC.*

*Proof.* Assume that $\mathbb{C}(\mathcal{L}_{Pr})$ has a polynomial sized upper bound dependent on variables, witnessed by a polynomial $\alpha$. Therefore for every concept $\varphi$ of a concept class $\mathbb{C}(\mathcal{L}_{Pr}[Prop])$ of the collection $\mathbb{C}(\mathcal{L}_{Pr})$, we have that there is an unique characterization $T_\varphi$ for $\varphi$ s.t. $|T_\varphi| \leq \alpha(|vars(\varphi)|)$. But notice that $|vars(\varphi)| \leq |Prop|$ so

$$|T_\varphi| \leq \alpha(|vars(\varphi)|) \leq \alpha(|Prop|)$$

Therefore, as $\varphi$ is arbitrary, we can use $\alpha$ as a witness for PSUC. Hence our claim is proven. $\square$

This lemma highlights one key point, if a collection $\mathbb{C}(\mathcal{L}_{Pr})$ has an polynomial sized upper bound dependent on variables then it must satisfy

(i) $Cl(Func(Pr)) \not\supseteq \mathrm{MPT}_0^\infty$

(ii) $Cl(Func(Pr)) \not\supseteq \mathrm{MPT}_1^\infty$

(iii) $Cl(Func(Pr)) \not\supseteq \mathrm{DM}$

Additionally since it is an upper bound dependent on variables, it should additionally satisfy:

(iv) $Cl(Func(Pr)) \not\supseteq \mathrm{AP}$

Consider the splitting pair

$$\{\{\mathrm{U}, , \mathsf{V}\}, \{\mathrm{MPT}_0^\infty, \mathrm{MPT}_0^\infty, \mathrm{AP}\}\}$$

where for each set of boolean functions $O$, $Cl(O)$ is either a subset of U,, $\mathsf{V}$, or a superset of $\mathrm{MPT}_0^\infty$, $\mathrm{MPT}_0^\infty$, AP. From conditions (i), (ii) and (iv) we have already inferred that if a collection $\mathbb{C}(\mathcal{L}_{Pr})$ has a polynomial sized upper bound dependent on variables then it is must be a subset of U,  or $\mathsf{V}$. Now, if a $\mathbb{C}(\mathcal{L}_{Pr})$ is such that $Cl(Func(Pr))$ is a subset of U,  or $\mathsf{V}$, then $Cl(Pr)$ must have a polynomial sized upper bound dependent on variables.

**Lemma 3.3.25.** *Let $\mathbb{C}(\mathcal{L}_{Pr})$ and $\mathbb{C}(\mathcal{L}_{Pr'})$ be two collections of concept classes s.t. $Cl(Func(Pr)) \subseteq Cl(Func(Pr'))$. Now, if $\mathbb{C}(\mathcal{L}_{Pr'})$ has a polynomial sized upper bound dependent on variables then so does $\mathbb{C}(\mathcal{L}_{Pr})$.*

*Proof.* The proof is exactly the same as in lemma 3.3.22, we just need to change $\alpha$ to a non-decreasing polynomial. ☐

In [ABCS92] it is shown that the collection $\mathbb{C}(\mathcal{L}_{\wedge,\top,\bot})$ and $\mathbb{C}(\mathcal{L}_{\vee,\top,\bot})$ has an upper bound $|vars(\varphi)|+1$, for every concept $\varphi$ in everyone of the concept classes in the collection. This fact along with lemma 3.3.25 gives us our other main result.

**Theorem 3.3.26.** *For any set of propositional formulas $Pr$, the following are equivalent,*

- *There exists a polynomial $p(x)$ s.t. for every Prop, and every concept $\varphi$ of $\mathbb{C}(\mathcal{L}_{Pr}[Prop])$, there is an unique characterization $T_\varphi$ that uniquely characterizes $\varphi$ with $|T_\varphi| < p(|vars(\varphi)|)$*
- *$Cl(Pr)$ is a subset of either of the three (i) $Cl(\wedge, \bot, \top)$, (ii) $Cl(\vee, \bot, \top)$ or (iii) $Cl(\neg, \top)$.*

Using the polynomial bounds of the collections $\mathbb{C}(\mathcal{L}_{\wedge,\top,\bot})$, $\mathbb{C}(\mathcal{L}_{\vee,\top,\bot})$ and $\mathbb{C}(\mathcal{L}_{\neg,\bot})$ we can modify our result to the following:

**Corollary 3.3.26.1.** *For any set of propositional formulas $Pr$ exactly one of 1. or 2. holds.*

1. *For every finite Prop, and every $\varphi \in \mathbb{C}(\mathcal{L}_{Pr}[Prop])$, there is a set of labelled examples $T$ that uniquely characterizes $\varphi$ with $|T| \leq |vars(\varphi)| + 1$.*

*2. The collection $\mathbb{C}(\mathcal{L}_{Pr})$ does not admit upper bounds dependent on variables.* $\square$

We look into another corollary, which will be important for the modal part of the thesis. Fix a set of infinite propositional variables $\mathsf{Inf}Prop = \{P_i : i \in \mathbb{N}\}$, we claim

**Corollary 3.3.26.2.** *The concept classes $\mathbb{C}(\mathcal{L}_{\mathsf{OXOR}}[\mathsf{Inf}Prop])$ and $\mathbb{C}(\mathcal{L}_{\mathsf{AIMP}}[\mathsf{Inf}Prop])$ doesn't have finite characterization.*

*Proof.* Suppose $\mathbb{C}(\mathcal{L}_{\mathsf{OXOR}}[\mathsf{Inf}Prop])$ has finite characterization. Now pick a concept $\varphi$ of $\mathbb{C}(\mathcal{L}_{\mathsf{3XOR}}[Prop])$ and take it's image $\sigma_6(\varphi)$. Now suppose $T_{\sigma_6(\varphi)}$ is an unique characterization of $\sigma_6(\varphi)$ w.r.t. $\mathbb{C}(\mathcal{L}_{\mathsf{OXOR}}[\mathsf{Inf}Prop])$. We again define

$$T^{Prop}_{\sigma_6(\varphi)} = \{(t \restriction Prop_{\#}, p)Prop : (t, p) \in T_{\sigma_6(\varphi)}\}$$

Notice that it is an unique characterization of $\sigma_6(\varphi)$ w.r.t. $\mathbb{C}(\mathcal{L}_{\mathsf{OXOR}}[Prop_{\#}])$. Now the size of $T^{Prop}_{\sigma_6(\varphi)}$ is independent of the set of propositional variables $Prop$. Using PC reductions we can get an unique characterization of $\varphi$, $T^P_\varphi rop$ say, w.r.t. $\mathbb{C}(\mathcal{L}_{\mathsf{OXOR}}[Prop])$ that is independent of the size of $Prop$. But this contradicts the result 3.3.7.1.

Similarly, following Post's Lattice $\varphi$ is a formula of $\mathbb{C}(\mathcal{L}_{\wedge,\to}[\mathsf{Inf}Prop])$. Suppose $\mathbb{C}(\mathcal{L}_{\mathsf{AIMP}}[\mathsf{Inf}Prop])$ has finite characterization, then following the above argument we again can derive a contradiction. $\square$

Now we look into the two other notions of learnability we introduced, namely learnable dependent on concept size and learnable dependent on polynomial concept size. Here by size, we will mean the variables used in the concept. We redefine notions for the reader.

**Definition 3.3.27.** *For any set of propositional formulas $Pr$, we say the collection $\mathbb{C}(\mathcal{L}_{Pr})$ is learnable with queries dependent on variables (LDoV for short) if for every concept class $\mathbb{C}(\mathcal{L}_{Pr}[Prop])$ and a concept $\varphi$ of that concept class, there exists an algorithm $\mathsf{Alg}(x)$ and a non-decreasing function $f(x)$ s.t. the following two conditions hold*

*i. If the algorithm $\mathsf{Alg}$ has access to the membership $\mathsf{MQ}_\varphi(x)$, then it halts after $f(|vars(\varphi)|)$ many computation steps and returns a concept $\varphi' \cong \varphi$ and*

*ii. for every other concept $\psi$ of $\mathbb{C}(\mathcal{L}_{Pr}[Prop])$, if the algorithm $\mathsf{Alg}(x)$ has access to the membership $\mathsf{MQ}_\psi(x)$, then the algorithm $\mathsf{Alg}$ halts and returns a concept $\psi' \cong \psi$.*

**Definition 3.3.28.** *For any set of propositional formulas $Pr$, we say the collection $\mathbb{C}(\mathcal{L}_{Pr})$ is learnable with polynomial queries dependent on variables (LPDoV for short) if for every concept class $\mathbb{C}(\mathcal{L}_{Pr}[Prop])$ and a concept $\varphi$ of that concept class, there exists an algorithm $\mathsf{Alg}(x)$ and a non-decreasing polynomial $p(x)$ s.t. the following two conditions hold*

*i. If the algorithm $\mathsf{Alg}$ has access to the membership $\mathsf{MQ}_\varphi(x)$, then it halts after $p(|vars(\varphi)|)$ many computation steps and returns a concept $\varphi' \cong \varphi$ and*

*ii. for every other concept $\psi$ of $\mathbb{C}(\mathcal{L}_{Pr}[Prop])$, if the algorithm $\mathsf{Alg}(x)$ has access to the membership $\mathsf{MQ}_\psi(x)$, then the algorithm $\mathsf{Alg}$ halts and returns a concept $\psi' \cong \psi$.*

**Lemma 3.3.29.** *For any set of propositional formulas $Pr$, if the collection $\mathbb{C}(\mathcal{L}_{Pr})$ has LDoV then $\mathbb{C}(\mathcal{L}_{Pr})$ has an upper bound dependent on variables.*

**Lemma 3.3.30.** *For any set of propositional formulas $Pr$, if the collection $\mathbb{C}(\mathcal{L}_{Pr})$ has LPDoV then $\mathbb{C}(\mathcal{L}_{Pr})$ has a polynomial upper bound dependent on variables.*

Both of the above lemmas are proven in the preliminaries, namely in the theorem 2.4.17.

The collections in question are $\mathbb{C}(\mathcal{L}_{\wedge,\vee,\top,\perp})$ and $\mathbb{C}(\mathcal{L}_{\neg,\perp})$, we provide algorithms for each case. Fix one particular $\varphi$ from the concept class $\mathbb{C}(\mathcal{L}_{\wedge,\vee,\top,\perp}[Prop])$. In the proof of proposition 3.3.8, we have illustrated a way of producing an unique characterization of $\varphi$ of size $2^{|vars(\varphi)|+1}$, we will call this unique characterization $T_\varphi$. We will assume the knowledge of this characterization in our algorithm. For the collection $\mathbb{C}(\mathcal{L}_{\neg,\perp})$, we will provide an algorithm for the case when $\varphi = p_i$ for the concept class $\mathbb{C}(\mathcal{L}_{\neg,\perp}[Prop])$. The reader can easily figure out the case for $\varphi = \neg p_i$.

```
k ← 0;
for (t, q) ∈ T_φ do
    if MQ(t) ≠ q then
        k ← 1;
        Exit the loop
    end
end
Based on the value of k, do the following cases:

Case 1: k = 1;
Print φ and exit the program

Case 2: k = 0;
ψ = ⊥;
for t in the example space of ℂ(L_{∧,∨,⊤,⊥}[Prop]) do
    if MQ(t) = 1 then
        θ = ⊤;
        for x ∈ Prop do
            if t(x) = 1 then
                θ = θ ∧ x
            end
        end
        ψ = ψ ∨ θ
    end
end
Print ψ and exit the program
```

**Algorithm 4:** $\mathsf{LDoV}_\varphi(\wedge,\vee,\top,\perp)$

$y_0 \leftarrow \mathsf{MQ}(\mathbf{0})$;
$y_1 \leftarrow \mathsf{MQ}(\mathbf{1})$;
**if** $y_0 = y_1 = +$ **then**
| Print $\top$ and exit the program
**end**
**if** $y_0 = y_1 = -$ **then**
| Print $\perp$ and exit the program
**end**
**if** $\mathsf{MQ}(t^+_{p_i}) = 1$ *and* $y_0 = 0$ **then**
| Print $p_i$ and exit the program
**end**
**for** $x \in Prop$ **do**
| **if** $\mathsf{MQ}(t^+_x) = 1$ *and* $y_0 = 0$ **then**
| | Print $x$ and exist the program
| **end**
| **if** $\mathsf{MQ}(t^-_x) = 1$ *and* $y_1 = 0$ **then**
| | Print $\neg x$ and exist the program
| **end**
**end**

**Algorithm 5:** $\mathsf{LDoV}_\varphi(\neg, \perp)$

**Theorem 3.3.31.** *For any set of propositional formulas Pr the following are equivalent:*

- $\mathbb{C}(\mathcal{L}_{Pr})$ *has LDoV.*
- $Cl(Pr)$ *is a subset of either* $(i)\,\mathbb{C}(\mathcal{L}_{\wedge,\vee,\top,\perp})$ *or* $(ii)\,\mathbb{C}(\mathcal{L}_{\neg,\perp})$.

*Proof.* ($\Leftarrow$) The given algorithms 1 and 2 provides us with the proof for this direction.
($\Rightarrow$) For this direction we analyse a particular anti-chain of the Post's Lattice, namely

$$\{\mathrm{U}, \mathrm{M}, \mathrm{D}, \mathrm{PT}^\infty_0, \mathrm{PT}^\infty_1\}$$

If we assume that $Cl(O)$ is neither a subset of $(i)\,\mathbb{C}(\mathcal{L}_{\wedge,\vee,\top,\perp})$ nor $(ii)\,\mathbb{C}(\mathcal{L}_{\neg,\perp})$ (which corresponds to M and U respectively), we get that $Cl(O)$ is a superset of either of D or $\mathrm{PT}^\infty_0$ or $\mathrm{PT}^\infty_1$. Now if $Cl(O)$ has LDoV then following lemma 3.3.29 $\mathbb{C}(\mathcal{L}_O)$ has an upper dependent on variables. This in turn means that $\mathbb{C}(\mathcal{L}_{\mathsf{AIMP}})$, $\mathbb{C}(\mathcal{L}_{\mathsf{OXOR}})$ or $\mathbb{C}(\mathcal{L}_{\mathsf{3XOR}})$ has an upper dependent on variables, contradicting our result 3.3.22.1. This means $Cl(O)$ cannot have LDoV, the contraposition of what we proved gives us the desired direction. $\square$

To close off this section we would like to prove our last result, providing a similar classification for LPDoV.

**Theorem 3.3.32.** *For any set of propositional formulas Pr the following are equivalent:*

- $\mathbb{C}(\mathcal{L}_{Pr})$ *has LPDoV.*

- $Cl(Pr)$ *is a subset of either* $(i)\,\mathbb{C}(\mathcal{L}_{\wedge,\top,\bot}), (ii)\,(i)\,\mathbb{C}(\mathcal{L}_{\vee,\top,\bot})$ *or* $(iii)\,\mathbb{C}(\mathcal{L}_{\neg,\bot})$.

*Proof.* ($\Leftarrow$) Algorithms 4 and 5 is enough for the cases of $\mathbb{C}(\mathcal{L}_{\wedge,\top,\bot})$ and $\mathbb{C}(\mathcal{L}_{\vee,\top,\bot})$. Algorithm 8 can be reused for this case too.
($\Rightarrow$) Again we analyse a particular anti-chain in the Post's Lattice, namely

$$\{\mathrm{U},\ ,\mathsf{V},\ \mathrm{MPT}_0^\infty,\ \mathrm{MPT}_0^\infty,\ \mathrm{AP}\}$$

If we assume that is neither a subset of $(i)\,\mathbb{C}(\mathcal{L}_{\vee,\top,\bot})$ nor $(ii)\,\mathbb{C}(\mathcal{L}_{\wedge,\top,\bot})$ nor $(iii)\,\mathbb{C}(\mathcal{L}_{\neg,\bot})$ (which corresponds to $\mathbf{V}$, and U respectively), we get that $Cl(Pr)$ is a superset of either of AP or $\mathrm{MPT}_0^\infty$ or $\mathrm{MPT}_1^\infty$. Now if $Cl(Pr)$ has LDoV then following lemma 3.3.30 $\mathbb{C}(\mathcal{L}_{Pr})$ has a polynomial upper dependent on variables. This in turn means that $\mathbb{C}(\mathcal{L}_{\vee,\top,\bot})$, $\mathbb{C}(\mathcal{L}_{\wedge,\top,\bot})$ or $\mathbb{C}(\mathcal{L}_{\neg,\bot})$ has an upper dependent on variables, contradicting our result 3.3.26. This means $Cl(Pr)$ cannot have LPDoV, the contraposition of what we proved gives us the desired direction. $\square$

## 3.4 On the Decidibility Aspect

Now we talk about the decidibility of our main results 3.2.26, 3.3.22.1 and 3.3.26. We claim that there exists algorithms which, given a set of propositional formulas $Pr$, decides

- whether $\mathcal{F}(Cl(Func(Pr)))$ has PSUC.

- whether $\mathcal{F}(Cl(Func(Pr)))$ has an upper-bound dependent on variables.

- whether $\mathcal{F}(Cl(Func(Pr)))$ has a polynomial sized upper-bound dependent on variables.

The algorithms we discuss are based on the *Uniform Clone Membership* problem. The complexity analysis of the problem can be found in [Vol09].

**Theorem 3.4.1** (Uniform Clone Membership)**.** *There exists a algorithm that accepts a set of boolean finite boolean functions $O$ and a particular connective $b$, and decides whether $b \in Cl(Pr)$ or not.*

*Proof.* The proof outline is to generate all the functions of arity that is same as that of $b$, that we can within $Cl(Pr)$. We check the variables in $b$ and generate all the functions possible via substitution, using $O$ and having the same variables as $b$. The final part is checking whether $b$ is equal to one of the generated functions. $\square$

For a detailed account of the proof, the reader is advised to check out the first chapter of N. Pippenger's book [Pip10]. Although the algorithm we have provided prove the decidability of the Uniform Clone Membership problem, it is not the best complexity wise. In [BS00], the authors discussed the complexity of various decidable problems In Abstract Algebra, one problem in particular was the *Variety Equivalence* problem. Now for any algebra $(A, F)$ the Clone Membership problem asks whether a particular function

$g \in F$ or not. On the other hand, the Variety Equivalence problem asks whether the variety generated by two algebras $(A, F)$ and $(A, G)$ of the same similarity type are equal or not. The algorithm they provide for the Variety Equivalence problem had better complexity than of the one described in the previous theorem. The key observation that links the Uniform Clone Membership problem and the Variety Equivalence problem for the case when $A = \{0, 1\}$ is the following observation:

$$g \in F \iff (\{0,1\}, F) \text{ and } (\{0,1\}, F \cup \{g\}) \text{ are equivalent as varieties}$$

For our purposes we will be building upon the Uniform Clone Membership algorithm, which we will denote by $\mathsf{CloMem}(x, Y)$. We additionally assume $\mathsf{CloMem}(x, Y)$ will return 1 if the function $x$ indeed belongs to the clone generated by $Y$, and returns 0 otherwise. We now provide 3 distinct algorithms The above

**Data:** X is a set of boolean functions
**if** $(\mathsf{CloMem}(\mathsf{maj}, X) = 1$ *or* $\mathsf{CloMem}(\mathsf{OAND}, X) = 1$ *or* $\mathsf{CloMem}(\mathsf{AOR}, X) = 1)$ **then**
| Print '$Cl(X)$ does not have PSUC.'
**else**
| Print '$Cl(X)$ does have PSUC.'
**end**

**Algorithm 6:** $\mathsf{AlgPSUC}(X)$

**Data:** X is a set of boolean functions
**if** $(\mathsf{CloMem}(\mathsf{3XOR}, X) = 1$ *or* $\mathsf{CloMem}(\mathsf{OXOR}, X) = 1$ *or* $\mathsf{CloMem}(\mathsf{AIMP}, X) = 1)$ **then**
| Print '$Cl(X)$ does not have PSUC.'
**else**
| Print '$Cl(X)$ does have PSUC.'
**end**

**Algorithm 7:** $\mathsf{AlgUpVar}(X)$

**Data:** X is a set of boolean functions
**if** $(\mathsf{CloMem}(\mathsf{OAND}, X) = 1$ *or* $\mathsf{CloMem}(\mathsf{AOR}, X) = 1$ *or* $\mathsf{CloMem}(\mathsf{3XOR}, x) = 1)$ **then**
| Print '$Cl(X)$ does not have PSUC.'
**else**
| Print '$Cl(X)$ does have PSUC.'
**end**

**Algorithm 8:** $\mathsf{AlgUpPolyVar}(X)$

three algorithms give us three decidability results, namely

**Theorem 3.4.2.** *It is decidable, given a set of propositional formulas $Pr$, whether the collection $\mathbb{C}(\mathcal{L}_{Pr})$ will admit PSUC or not.* □

**Theorem 3.4.3.** *It is decidable, given a set of propositional formulas $Pr$, whether the collection $\mathbb{C}(\mathcal{L}_{Pr})$ will have an upper bound dependent on variables or not.* $\square$

**Theorem 3.4.4.** *It is decidable, given a set of propositional formulas $Pr$, whether the collection $\mathbb{C}(\mathcal{L}_{Pr})$ will have a polynomial sized upper bound dependent on variables or not.* $\square$

# Chapter 4

# Case Study: The Modal Fragments

Leaving the propositional landscape behind, in this chapter, we focus on the modal counterpart of our thesis. In the light of theorem 3.1.19, we were able to able to justify propositional fragments as propositional clones. In this chapter, we firstly define modal fragments (see def. 4.1.11) and identify them as modal clones by using clones from Lindenbaum-Tarski algebra.

In contrast to the propositional case, we lack a representation of *all the possible modal fragments*, which makes the analysis much harder. However, due to theorem 2.2.5, we can be sure that all the modal clones form a lattice; therefore study of sub-lattices of the entire modal lattice is still feasible. We consider three different sub-lattices (see 4.1.15 and 4.1.16), which are isomorphic to each other, and provide a complete classification of them in terms of unique characterization and effective learnability .

This chapter is in-fact an extension of Cate and Koudijs' work in [tCK24].The results they prove along with our main result, thm 4.2.22, helps us to produce the characterizations 4.2.24,4.2.25 and 4.2.26. We list our findings as Tables 4.1 and 4.2 below. The $i$ in the tables below is one the three possible sets $\{\Diamond\}, \{\Box\}, \{\Box, \Diamond\}$ and depends on the sub-lattice considered, for e.g. if the sub-lattice considered is $\mathscr{M}_{\Diamond}$, then $i = \Diamond$ . (The description of the sub-lattices can be found in the following section). Similar to the propositional case, the left and the right of the table are linked by an if-and-only-if clause.

As opposed to Tables 3.1 though, Table 4.1 only focuses on finite characterization. The three distinct rows represent the sub-lattices of the entire lattice of modal fragments that we consider for each characterization. The same follows Table 4.2, where we consider effective learnability of the three distinct sub-lattices. The main focus of our analysis would be the modal logic $K$. We begin with the definitions of all of the notions, to the make the manuscript self-contained and familiarise the reader with the notations we use.

## 4.1 Modal Logic and sub-Lattices of the entire modal fragment lattice

**Definition 4.1.1.** *The full modal logic over a set of propositional variables Prop, denoted as $\mathcal{ML}_{Prop}$ is generated recursively by,*

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

| Sub-lattice Considered | $\mathbb{C}(\mathcal{L}_{i,Pr})$ has finite characterization iff $Cl(Pr)$ is a subset of |
|---|---|
| $\mathscr{M}_{\Box}$ | (i) $Cl(\wedge, \vee, \top, \bot)$ or (ii) $Cl(\neg, \bot)$ |
| $\mathscr{M}_{\Diamond}$ | (i) $Cl(\wedge, \vee, \top, \bot)$ or (ii) $Cl(\neg, \bot)$ |
| $\mathscr{M}_{\Box,\Diamond}$ | (i) $Cl(\wedge, \vee)$ or (ii) $Cl(\neg, \bot)$ |

Table 4.1: Table for finite characterization

| Sub-lattice Considered | $\mathbb{C}(\mathcal{L}_{i,Pr})$ is effectively learnable iff $Cl(Pr)$ is a subset of |
|---|---|
| $\mathscr{M}_{\Box}$ | (i) $Cl(\wedge, \vee, \top, \bot)$ or (ii) $Cl(\neg, \bot)$ |
| $\mathscr{M}_{\Diamond}$ | (i) $Cl(\wedge, \vee, \top, \bot)$ or (ii) $Cl(\neg, \bot)$ |
| $\mathscr{M}_{\Box,\Diamond}$ | (i) $Cl(\wedge, \vee)$ or (ii) $Cl(\neg, \bot)$ |

Table 4.2: Table for effectively learnability

*where $p \in$Prop.*

Notice that the $\Diamond$ modality can be expressed as $\Diamond p = \neg\Box\neg p$. Definition 3.1.2 hold analogously for the modal counterpart. Another thing we will be talking about is modal depth.

**Definition 4.1.2.** *Given a formula $\varphi \in \mathcal{ML}_{Prop}$, we define the formula $d(\varphi)$ in a recursive way. If $\varphi = p$ for some propositional variable $p \in Prop$, we put $d(\varphi) = 0$. Assume $\psi, \theta \in \mathcal{ML}_{Prop}$ and use*

$$\begin{aligned} d(\psi \vee \theta) &= max\{d(\psi), d(\theta)\} = d(\psi \wedge \theta) \\ d(\neg\psi) &= d(\psi) \\ d(\Box\psi) &= d(\psi) + 1 \end{aligned}$$

*The function $d$ is called the modal depth operator, and $d(\varphi)$ is referred to as the modal depth of $\varphi$.*

As it was mentioned in the propositional case, *Prop* can be both finite or infinite, we will mostly assume *Prop* to be infinite in this section. The main difference we notice is talking about the *semantics* of modal logic. Unlike propositional models, modal models are *Kripke models*.

**Definition 4.1.3** (Kripke models)**.** *A Kripke frame is an ordered pair $M_K = (W, R)$ where $W$ is a set of worlds and $R$ is a binary relation on $W$. A Kripke model is an ordered triple $M = (W, R, V)$ where $(W, R)$ is a Kripke frame and $V : Prop \to \mathcal{P}(W)$ is a function, assigning to each propositional variable, the set of worlds it is valid in.*

**Definition 4.1.4** (Semantics)**.** *Let $M = (W, R, V)$ be a Kripke model and let $\varphi \in \mathcal{ML}_{Prop}$, if $\varphi = p$ for some propositional variable $p \in Prop$, we say $M, s \models \varphi$ holds if $s \in V(p)$. Now, the relation $\models$ is defined recursively: Assume $\psi, \theta \in \mathcal{ML}_{Prop}$*

$$\begin{aligned} M, s \models \neg\psi &\iff M, s \not\models \psi \\ M, s \models \psi \wedge \theta &\iff M, s \models \psi \text{ and } M, s \models \theta \\ M, s \models \psi \vee \theta &\iff M, s \models \psi \text{ or } M, s \models \theta \\ M, s \models \Box\psi &\iff \forall w\, (sRw \Rightarrow M, w \models \psi) \end{aligned}$$

A closer look at the above semantics reveals that modal logic is just propositional logic with $\Box$ added to it. The underlying set $W$ behind every Kripke frame $M$, is called the set of all *possible worlds.* Sometimes we will also talk about *pointed Kripke models.*

**Definition 4.1.5** (Pointed Kripke models)**.** *A pointed Kripke model is a pair $(M, s)$, where $M = (W, R, V)$ is a Kripke model and $s \in W$. Furthermore for any $w \in W$, there is a path $w_1 R \dots R w_n$ s.t. $w_1 = s$ and $w_n = w$.*

Fix any formula $\varphi \in \mathcal{ML}_{Prop}$. We say that $\varphi$ is satisfiable if there exists a Kripke model $M = (W, R, V)$ and a world $s \in W$ s.t. $M, s \models \varphi$. Furthermore, we say $\varphi$ is valid if for any Kripke model $M = (W, R, V)$, there exists a $s \in W$ s.t. $M, s \models \varphi$. The next set of definitions and theorems would serve to illustrate how satisfiability is preserved between pointed Kripke models.

**Theorem 4.1.6** (see [BdRV01])**.** *Fix a set Prop and pick any* $\varphi \in \mathcal{L}_{Prop}$*. The following equivalence holds:* $\varphi$ *is satisfiable if and only if there is a pointed Kripke model* $(M, s)$ *s.t.* $M, s \models \varphi$ $\square$.

**Definition 4.1.7** (Bisimulation)**.** *Let* $M, s = (W, R, V), s$ *and* $N, w = (W', R', V'), w$ *be pointed Kripke models, We say* $M, s$ *and* $N, w$ *are bisimilar if there is a binary relation* $Z$ *on* $W \times W'$ *satisfying:*

*i.* $sZw$ *holds, and if* $s'Zw'$ *then* $s' \in V(p_i)$ *if and only if* $w' \in V'(p_i)$*, for every propositional variables* $p_i$.

*ii.* (*Back*) *If* $s'Zw'$ *and* $s'Rs''$ *holds then there exists* $w'' \in W'$ *s.t.* $s''Zw''$ *and* $w'R'w''$

*iii.* (*Forth*) *If* $s'Zw'$ *and* $w'R'w''$ *holds then there exists* $s'' \in W$ *s.t.* $s''Zw''$ *and* $w'Rw''$

**Definition 4.1.8** (n-Bisimulations)**.** *Let* $(M, s) = ((W, R, V), s)$ *and* $(N, w) = ((W', R', V'), w)$ *be pointed Kripke models, We say* $M, s$ *and* $N, w$ *are n-bisimilar if there is a chain of binary relations* $Z_0, \dots, Z_n$ *with* $Z_0 \subseteq Z_1 \subseteq \cdots \subseteq Z_n$ *on* $W \times W'$ *satisfying:*

1. $sZ_0w$ *holds, and if* $s'Z_nw'$ *then* $s' \in V(p_i)$ *if and only if* $w' \in V'(p_i)$*, for every propositional variables* $p_i$.
2. (*Back*) *If* $s'Z_iw'$ *and* $s'Rs''$ *holds then there exists* $w'' \in W'$ *s.t.* $s''Z_{i+1}w''$ *and* $w'R'w''$
3. (*Forth*) *If* $s'Z_iw'$ *and* $w'R'w''$ *holds then there exists* $s'' \in W$ *s.t.* $s''Z_{i+1}w''$ *and* $w'Rw''$

**Theorem 4.1.9** (see [BdRV01])**.** *Let* $(M, s)$ *and* $(N, w)$ *be pointed Kripke models s.t.*$(M, s)$ *is bisimilar to* $(N, s)$*. For every modal formula* $\varphi$*, the following holds:*

$$(M, s) \models \varphi \iff (N, s) \models \varphi$$

$\square$

**Theorem 4.1.10** (see [BdRV01])**.** *Let* $(M, s)$ *and* $(N, w)$ *be pointed Kripke models, then the following are equivalent:*

1. $M, s \models \varphi \iff (N, w) \models \varphi$ *for every modal formula of depth* $\leq n$.
2. $M, s$ *and* $N, w$ *are n-bisimlar.* $\square$

Based on the jargon we have developed so far, we can establish an equivalence relation $\sim_{\mathcal{ML}}$ on the set of all modal formulas $\mathcal{ML}_{Prop}$. As it was the case with propositional formulas, for any two formulas $\varphi, \psi \in \mathcal{ML}_{Prop}$ $\varphi \sim_{\mathcal{ML}} \psi$ if for any Kripke model $M = (W, R, V)$ and $s \in W$, we have the following equivalence:

$$M, s \models \varphi \iff M, s \models \psi$$

We will use $[\![\varphi]\!]_{\sim_{\mathcal{ML}}}$ to denote the equivalence class $\varphi$ belongs to. Additionally we can improve definition 3.1.9 to accommodate modal formulas. A simple change from 'propositional formulas' to 'modal formulas' in the aforementioned definition is enough. We still use the notation $\varphi(\psi_1, \dots, \psi_m)$ for denoting substitution of $\psi_1, \dots, \psi_m$ into $\varphi(p_1, \dots, p_m)$.

All the jargon we have developed so far, helps us to define modal clones. We define it analogously to the propositional fragments.

**Definition 4.1.11.** *Let $\mathcal{M} \subseteq \mathcal{ML}_{Prop}$, for a set of variables $Prop$. We say $\mathcal{M}$ is a modal fragment if it satisfies the following:*

1. *It contains all the propositional variables, i.e. $Prop \subseteq \mathcal{M}$.*
2. *$\mathcal{M}$ is closed under composition, i.e. for every collection $\varphi(p_1, \dots, p_m), \psi_1, \dots, \psi_m \in \mathcal{M}$, it follows that $\varphi(\psi_1, \dots, \psi_m) \in \mathcal{M}$.*
3. *For every $\varphi \in \mathcal{M}$ we have $[\![\varphi]\!]_{\sim_{\mathcal{ML}}} \in \mathcal{M}$.*

**Definition 4.1.12.** *A set $Md$ be a set of modal formulas is defined to be a generating set for a modal fragment $\mathcal{M}$, if $Md \subseteq \mathcal{M}$ and every modal fragment $\mathcal{M}'$ with $Md \subseteq \mathcal{M}'$ we have that $\mathcal{M} \subseteq \mathcal{M}'$. In such cases we denote $\mathcal{M}$ as $\mathcal{M}(Md)$.*

In the case of propositional logic we established through theorem 3.1.19 that propositional fragments can be viewed as propositional clones. The main observation behind that was the following: each propositional formula correspond to a boolean function. But what about the case of modal fragments? Can we be justified in calling them modal clones? The answer is yes!

Consider the Lindenbaum-Tarski algebra, $(A, +, \times, \ulcorner\ \urcorner, m)$ of modal Logic $K$ (see [BdRV01] chapter 7 for some details). Remember that $IProp = \{p_1, \dots p_n, \dots\}$ is a set of infinite propositional variables. $A$ is defined as follows:

$$A = \{[\![\varphi]\!]_{\sim_{\mathcal{ML}}} : \varphi \in \mathcal{ML}_{IProp}\}$$

The operations are defined as follows, for $\psi, \varphi \in \mathcal{ML}_{IProp}$:

$$\begin{aligned}
[\![\psi]\!]_{\sim_{\mathcal{ML}}} + [\![\varphi]\!]_{\sim_{\mathcal{ML}}} &= [\![\psi \vee \varphi]\!]_{\sim_{\mathcal{ML}}} \\
[\![\psi]\!]_{\sim_{\mathcal{ML}}} \cdot [\![\varphi]\!]_{\sim_{\mathcal{ML}}} &= [\![\psi \wedge \varphi]\!]_{\sim_{\mathcal{ML}}} \\
\ulcorner [\![\psi]\!]_{\sim_{\mathcal{ML}}} \urcorner &= [\![\neg\psi]\!]_{\sim_{\mathcal{ML}}} \\
m([\![\psi]\!]_{\sim_{\mathcal{ML}}}) &= [\![\Box\psi]\!]_{\sim_{\mathcal{ML}}}
\end{aligned}$$

Notice that the projection map can be defined in the following way,

$$\begin{aligned}
&\pi_n^k([\![\psi_1]\!]_{\sim_{\mathcal{ML}}}, \dots, [\![\psi_k]\!]_{\sim_{\mathcal{ML}}} [\![\psi_n]\!]_{\sim_{\mathcal{ML}}}) = \\
&\qquad ([\![\psi_1]\!]_{\sim_{\mathcal{ML}}} + \ulcorner [\![\psi_1]\!]_{\sim_{\mathcal{ML}}} \urcorner) \times \cdots \times [\![\psi_k]\!]_{\sim_{\mathcal{ML}}} \times \cdots \times ([\![\psi_n]\!]_{\sim_{\mathcal{ML}}} + \ulcorner [\![\psi_n]\!]_{\sim_{\mathcal{ML}}} \urcorner)
\end{aligned}$$

Now we can consider clones over the set $A$ as defined above. Notice that for each modal formula $\varphi \in \mathcal{ML}_{IProp}$, there is a corresponding operation/function on the set $A$. For example take the formula $(\neg p \vee q) \wedge r$, it defines the operation $(\ulcorner x \urcorner + y) \times z$ over the set $A$. For each formula $\varphi$, we use $f_\varphi$ to denote the function $\varphi$ corresponds to on the set $A$. We extend our definition of $Func$, let $Md$ be a set of modal formulas (over any set $Prop$), then

$$Func(Md) = \{f_\varphi : \varphi \in Md\}$$

Given a set of modal formulas $Md$, we can consider the clone over $A$ generated by $Func(Md)$, which we denote by $Cl_A(Func(Md))$. $Form$ can be extended in the following way. Given a set of functions $F$ over $A$, define

$$Form(F) = \{\varphi : f_\varphi \in F\}$$

The following is easy but tedious to verify, we leave it to the reader to verify the following.

**Proposition 4.1.13.** *For any set of modal formulas $Md$, we have that*

$$\mathcal{M}(Md) = \mathcal{M}(Form(Cl_A(Func(Md))))$$

□

The other direction, i.e. given a set of functions $F$ over $A$, whether holds

$$Cl_A(F) = Cl_A(Func(\mathcal{M}(Form(F))))$$

or holds or not is an open question. As the set $A$ is infinite, answering this question requires digressing into infinite clone theory, so we avoid it for the sake of our analysis. This discussion justifies that the study of modal fragments is just the study of modal clones.

We choose $\mathscr{M}$ as the denotation of all the modal fragments over a set of variables $Prop$. Any arbitrary set $\mathscr{M} = \{\mathcal{M}_i : i \in I\}$ of modal clones is also closed intersection. We abuse the notation and define a function

$$\sqcup(\mathcal{M}, \mathcal{M}') = \bigcap\{m \in \mathscr{M} : \mathcal{M} \cup \mathcal{M}' \subseteq m\}$$

where $\mathcal{M}, \mathcal{M}'$ are modal fragments. Again it is an easy verification task to establish the following is a lattice.

**Proposition 4.1.14.** *The ordered triple $(\mathscr{M}, \cap, \sqcup)$ forms a lattice.* □

We take some time to contrast the propositional and modal cases. We have a complete representation of all the propositional fragments in the form a lattice. This representation owes its existence to Post's Lattice, and the correspondence between them too is due to the semantics of propositional logic. Indeed, when viewed from the semantic side, composition of the set of formulas $\varphi(p_1, \ldots, p_m), \psi_1, \ldots, \psi_m$ to form $\varphi(\psi_1, \ldots, \psi_m)$, is equivalent to composing the *functions described by* $\varphi(p_1, \ldots, p_m), \psi_1, \ldots, \psi_m$ to form the function described by $\varphi(\psi_1, \ldots, \psi_m)$.

Turning to the modal logic and trying to follow the same manipulation, we immediately encounter two problems. Firstly, what does each modal formula *represent* semantically, i.e. what would each modal formula stand for in terms of pointed Kripke models, and how it behaves under formula composition. Secondly, we need to arrange the *representation* in terms of some mathematical structure (not necessarily a lattice). Solving these two problems will give us a clear picture of all the modal fragments, but it is a challenging task. Therefore, though we have established that $\mathscr{M}$ is a lattice, we cannot comment much more on it's structure.

Fortunately, the structure of Post's Lattice can employed further to generate interesting sub-lattices of the entire modal lattice. Consider the sets of formulas of the form $Pr_\Box = \{\Box p\} \cup \mathrm{Pr}$ where Pr is a set of propositional formulas, and $p$ is a propositional variable. We claim that the class of modal fragments

$$\mathscr{M}_\Box = \{\mathcal{M}(Pr_\Box) : Pr \text{ is a set of prop. formulas}\}$$

forms a sub-lattice of the lattice of all modal fragments.

**Theorem 4.1.15.** $(\mathscr{M}_\Box, \cap, \sqcup)$ *forms a sub-lattice.*

*Proof.* Firstly notice that for any set $Pr^\Box$, the fragment $\mathcal{M}(\mathrm{Pr}) = \mathcal{F}(Pr)$ is just a propositional fragment and hence there is a finite generating set $Pr_{fin} \subseteq Pr$, s.t. $\mathcal{M}(Pr) = \mathcal{M}(Pr_{fin})$. Now it is easy to see that $\mathcal{M}(Pr_\Box) = \mathcal{M}(Pr_{fin})$.

Now consider any two sets $Pr1_\Box$ and $Pr2_\Box$. Let Pr3 be the set s.t. $\mathcal{M}(Pr3) = \mathcal{M}(Pr1) \cap \mathcal{M}(Pr2)$. It is the case that $\mathcal{M}(Pr3_\Box) \subseteq \mathcal{M}(Pr1_\Box) \cap \mathcal{M}(Pr2_\Box)$. On the other hand it also the case that

$$\mathcal{M}(Pr3) \subseteq \mathcal{M}(Pr3_\Box) \subseteq \mathcal{M}(Pr1_\Box) \cap \mathcal{M}(Pr2_\Box)$$

and it is the case that $\Box p \in \mathcal{M}(Pr1_\Box) \cap \mathcal{M}(Pr2_\Box)$, and hence $\mathcal{M}(Pr1_\Box) \cap \mathcal{M}(Pr2_\Box)$ is a fragment that contains the set $Pr3_\Box$. It is trivial to see $\mathcal{M}(Pr3_\Box) \subseteq \mathcal{M}(Pr1_\Box) \cap \mathcal{M}(Pr2_\Box)$. Combining the two facts we get $\mathcal{M}(Pr3_\Box) = \mathcal{M}(Pr1_\Box) \cap \mathcal{M}(Pr2_\Box)$.

On the other hand, let Pr4 be the set such that $\mathcal{M}(Pr4) = \mathcal{M}(Pr1) \sqcup \mathcal{M}(Pr2)$. Again, we have that $\mathcal{M}(Pr4_\Box) \subseteq \mathcal{M}(Pr1_\Box) \sqcup \mathcal{M}(Pr2_\Box)$ again from definition. Now notice that $\mathcal{M}(Pr1_\Box) \subseteq \mathcal{M}(Pr4_\Box)$ and $\mathcal{M}(Pr2_\Box) \subseteq \mathcal{M}(Pr4_\Box)$ and hence

$$(\mathcal{M}(Pr1_\Box) \sqcup \mathcal{M}(Pr2_\Box) \subseteq \mathcal{M}(Pr4_\Box)$$

Combining them we have $\mathcal{M}(Pr1_\Box) \sqcup \mathcal{M}(Pr2_\Box) = \mathcal{M}(Pr4_\Box)$. This proves our result. ∎

Notice that we can similarly define $Pr_\Diamond$ and $Pr_{\Diamond,\Box}$. It is also the case that the sets $\mathscr{M}_{\Diamond,\Box}$ and $\mathscr{M}_\Diamond$ form a sub-lattice with respect to the functions $\cap$ and $\sqcup$. The proof is exactly the same as in the previous case.

**Theorem 4.1.16.** *The ordered triples* $(\mathscr{M}_{\Diamond,\Box}, \cap, \sqcup)$ *and* $(\mathscr{M}_\Diamond, \cap, \sqcup)$ *form a sub lattice of the lattice* $(\mathscr{M}, \cap, \sqcup)$. ∎

The rest of this chapter is dedicated towards analysing the sub-lattices $\mathscr{M}_\Box, \mathscr{M}_\Diamond$and $\mathscr{M}_{\Box,\Diamond}$. Before ending this section, we would like to make one crucial observation. Notice that from the proof of theorem 4.1.15, it follows that for every set of propositional formulas Pr1, Pr2

$$\mathcal{M}(Pr1_\Box) \subseteq \mathcal{M}(Pr2_\Box) \iff \mathcal{F}(Pr1) \subseteq \mathcal{F}(Pr2)$$

this correspondence tells us the sub-lattice $\mathscr{M}_\Box$[1] is isomorphic to the lattice of all propositional clones. The same can be said about the sub-lattices, we state the following theorem which the reader can easily verify

[1] We will abuse notation and call $\mathscr{M}_\Box$ a sub-lattice, the reader will be able to figure out from the context that we actually mean the triple $(\mathscr{M}_\Box, \cap, \sqcup)$. The same applies for $\mathscr{M}_{\Diamond,\Box}$ and $\mathscr{M}_\Diamond$

**Theorem 4.1.17.** *The sub-lattices $\mathscr{M}_{\Diamond,\Box}$, $\mathscr{M}_{\Diamond}$ and $\mathscr{M}_{\Box}$ are isomorphic to each other. In-fact they are isomorphic to the Post's lattice via the map*

$$\mathsf{Iso} : \mathcal{M}(Pr_i) = Cl(Func(Pr))$$

*where $Pr$ is a set of propositional formulas and $i \in \{\{\Diamond, \Box\}, \Diamond, \Box\}$.* $\Box$

The sub-lattices $\mathscr{M}_{\Box}, \mathscr{M}_{\Diamond}$ were considered in the works [BMS$^+$11] and [BSS$^+$07]. The scope of their analysis were different from unique characterization and learnability. In-fact they considered the complexity of reducing the various problems between the different fragments. Moreover [BSS$^+$07] considers some interesting applications of these complexity analysis in the real world scenarios.

## 4.2 Analysis of the sub-lattices $\mathscr{M}_{\Box}$, $\mathscr{M}_{\Diamond}$ and $\mathscr{M}_{\Diamond,\Box}$

This section builds up the work of Cate and Koudijs in [tCK24], based on the topics unique characterization and learnability of various modal fragments. As opposed to the full propositional fragment, the full modal fragment does not even have finite characterization. The main culprit of this negative result is the existence of formulas that allows us to manipulate the height of a pointed Kripke model. Let us define a formula $height_n$,

$$height_n := (\Box^{n+1} \bot) \wedge\ (\Diamond^n \top)$$

In [tCK24], it is established how $height_n$ prevents us from having finite characterization for *any* modal formula in the full modal fragment.

**Theorem 4.2.1** (Cate and Kouijs in [tCK24])**.** *No formula $\varphi$ in the concept class $\mathbb{C}(\mathcal{L}_{\Box,\neg,\wedge}[\varnothing])$ has finite characterization.*

*Proof.* This is an outline of the proof given in the original paper. Consider any formula $\varphi$ and any characterization $T_\varphi$ of it. If $T_\varphi$ is finite then we choose

$$n > \{|W| : ((W, R, V), s) \in T_\varphi\} \cup \{d(\varphi)\}$$

Now if there exists a $m \geq n$ s.t. $M_K, s \not\models \varphi$ and $M_K$ has a path of length $m$ starting from $s$, then we can see that $\varphi \vee height_m$ fits $T_\varphi$, but $\varphi \not\cong (\varphi \vee height_m)$. On the other hand if for all $m \geq n$ we have that $M_K, s \models \varphi$ whenever $K$ is a model that has a path of length $m$ starting from $s$, then $\varphi \wedge (\neg height_m)$ fits $T_\varphi$ but $\varphi \not\cong \varphi \wedge (\neg height_m)$. Hence, $T_\varphi$ cannot be finite. $\Box$

### 4.2.1 The positive results

The natural successor to this observation is to ask whether there's a modal fragment that admits finite characterization. The answer is a positive one, and it is already established in [tCK24] that every concept class $\mathbb{C}(\mathcal{L}_{\Diamond,\wedge,\vee,\top,\bot}[Prop])$ has finite characterization, whenever set $Prop$ is finite. The proof employed for this result is a remarkable one and spans over many papers, we try to provide an outline of the proof's structure so that the reader can appreciate it, leaving out all the excruciating details.

Similar to the cases of modal and propositional logic, one can talk about the fragments of *First Order Logic*[2]. One of the important fragments is the positive existential fragment, i.e. the set of all first order formulas that can be expressed using the connectives $\exists, \wedge, \vee, \top, \bot$, modulo equivalence, over a schema $S$. One can consequently talk about the concept class $\mathbb{C}(\mathcal{L}_{\exists,\wedge,\vee,\top,\bot}[S]) = (C, E, \lambda)$, where $C$ is the set of all positive existential formulas over the schema $S$, $E$ is set of all first order structures over the schema $S$ and finally $\lambda$ maps every formula of $C$ to the set of all structures that satisfies it.

One of the fundamental results in FOL is that homomorphisms (i.e. structure preserving maps) between two structures preserve the truth values of positive existential formulas.

**Proposition 4.2.2** (see [CK92])**.** *Let $(A, \boldsymbol{a})$ and $(B, \boldsymbol{b})$ be two structures over the schema $S$. If the structure $(A, \boldsymbol{a})$ is homomorphic to the structure $(B, \boldsymbol{b})$, then $(A, \boldsymbol{a}) \models \varphi$ implies $(B, \boldsymbol{b}) \models \varphi$, whenever $\varphi$ is a positive existential formula (over the schema $S$).* □

Homomorphisms allows us to talk about finite *dualty pairs*[3] on the class of all $S$ structures. A (finite) duality pair is a finite ordered pair $(F, D)$ s.t. for every $S$ structure $(A, \mathbf{a})$ either of the two happens:

i. there exists a $(F_i, \mathbf{f}) \in F$ s.t. $(F_i, \mathbf{f})$ is homomorphic to $(A, \mathbf{a})$, or

ii. there exists a $(D_i, \mathbf{d}) \in D$ s.t. $(A, \mathbf{a})$ is homomorphic to $(D_i, \mathbf{d})$

Duality pairs coupled with some properties gives way to unique characterization for the concept class $\mathbb{C}(\mathcal{L}_{\exists,\wedge,\vee,\top,\bot}[S])$.

**Proposition 4.2.3.** *For any schema $S$ and any concept $\varphi$ of $\mathbb{C}(\mathcal{L}_{\exists,\wedge,\vee,\top,\bot}[S])$, the following are equivalent:*

*i. There exists a finite duality pair $(F, D)$ s.t. for every positive example $(A, \mathbf{a})$ of $\varphi$, there exists a $(A, \mathbf{a})$ $(F_i, \boldsymbol{f}) \in F$ s.t. $(F_i, \boldsymbol{f})$ is homomorphic to $(A, \boldsymbol{a})$.*

*ii. The set $(F \times \{+\}) \cup (D \times \{-\})$ is an unique characterization of $\varphi$. In other words, $\varphi$ has a finite characterization.* □

Cate and Dalmau, in their work [tCD22], showed that a class of FO formulas, namely *c-connected formulas*, satisfy the first condition of proposition 4.2.3. In-fact they provide explicit *effective* constructions of the duality pairs; in other words, there is an algorithm that, given a c-connected formula $\varphi$, can construct $\varphi$'s duality pair $(F, D)$. As it turns out, the positive existential FO formulas form a subclass of c-connected formulas, and hence using proposition 4.2.3 we can conclude that $\mathbb{C}(\mathcal{L}_{\exists,\wedge,\vee,\top,\bot}[S])$ has finite characterization.

The last part is to establish a connection between the concept classes $\mathbb{C}(\mathcal{L}_{\exists,\wedge,\vee,\top,\bot}[S])$ and $\mathbb{C}(\mathcal{L}_{\Diamond,\wedge,\vee,\top,\bot}[Prop])$. Assume $Prop = \{p_1, \ldots, p_n\}$ is a finite set, and $S_{Prop} = \{R, P_1, \ldots, P_n\}$ is a finite signature, where $R$ is a binary relation and $P_1, \ldots, P_n$ are unary relations. Another fundamental jargon in the literature of Modal Logic is *standard translation* (see [BdRV01] for the definition), which is an injective map from

[2] For a detailed introduction to FOL, we advise the reader see [Men09].

[3] Duality pairs can be studied more generally as a pre-order over the example set, but we leave the details to make sake of brevity.

the set of all modal formulas to the set of FO formulas with one free variable (over a certain signature). If we are to restrict the standard translation, which we will denote by $ST_x(\varphi)$, to the concept space of $\mathbb{C}(\mathcal{L}_{\Diamond,\wedge,\vee,\top,\bot}[Prop])$, then the range of standard translation would be restricted to the concept space of $\mathbb{C}(\mathcal{L}_{\exists,\wedge,\vee,\top,\bot}[S_{Prop}])$. Another possible translation is from the class of all FO structures over $S_{Prop}$ to the class of all pointed Kripke models (over $Prop$). For any $S_{Prop}$ structure $(A, a)$, where $a$ is an element of the underlying set of $A$, we define the following pointed Kripke model $(M^A, a) = ((W, R, V), a)$. The set of worlds $W$ is same the underlying set of $A$, we also have that $R = R^A$ and finally

$$x \in P_i^A \iff x \in V(p_i)$$

We observe the following result.

**Proposition 4.2.4** (see [BdRV01]). *For any concept $\varphi$ of $\mathbb{C}(\mathcal{L}_{\Diamond,\wedge,\vee,\top,\bot}[Prop])$, we have the following equivalence*

$$M^A, a \models \varphi \iff A, a \models ST_x(\varphi)$$

□

**Theorem 4.2.5** (Cate and Koudijs in [tCK24]). *For any finite set of variable Prop, the concept class $\mathbb{C}(\mathcal{L}_{\Diamond,\wedge,\vee,\top,\bot}[Prop])$ has finite characterization.*

*Proof.* Consider any formula $\varphi$ of $\mathbb{C}(\mathcal{L}_{\Diamond,\wedge,\vee,\top,\bot}[Prop])$, and take the duality pair $(F, D)$ of $ST_x(\varphi)$. Now define $T^+ = \{((M^A, a), +) : (A, a) \in F\}$ and $T^- = \{((M^A, a), -) : (A, a) \in D\}$, we claim that $T^+ \cup T^-$ uniquely characterizes $\varphi$. Suppose $\psi$ be any formula s.t. $\varphi \not\cong \psi$, so by proposition 4.2.3 we get that there is some $(A, a) \in F \cup D$ s.t. $(A, a)$ distinguishes between $ST_x(\varphi)$ and $ST_x(\psi)$. Now by proposition 4.2.4, $(M^A, a)$ should differentiate between $\varphi$ and $\psi$. As $\psi$ was arbitrary, this proves $T^+ \cup T^-$ is a finite characterization. □

It is worthwhile to point out the *duality* that this proof produces. Suppose $\mathsf{C}$ be the category consisting of the following things as objects:

i. All modal formulas formed using $\{\Diamond, \wedge, \vee, \top, \bot\}$ (over the variable set $Prop$)

ii. all FO formulas formed using $\{\exists, \wedge, \vee, \top, \bot\}$ (over the set $S_{Prop}$)

The morphisms that exist are the standard translations between formulas, i.e $f : \varphi \to ST_x(\varphi)$.

On the other hand we have a category $\mathsf{D}$ which consists of the following as objects

i. Finite duality pairs $(F, D)$ of $S_{Prop}$ structures as objects.

ii. Finite pointed Kripke models $(G, H)$ (over $Prop$).

Now for every finite duality pair $(F, D)$ over the structures of $S_{Prop}$, there is a morphism $g \in \mathsf{D}$ s.t. the **domain** of $g$ is $(F, D)$ and the **codomain** of $g$ is $(G, H)$ where

$$G = \{(M^A, a) : (A, a) \in F\} \qquad H = \{(M^A, a) : (A, a) \in D\}$$

Now we will define a contravariant functor $\alpha : \mathsf{C}^{\mathsf{op}} \to \mathsf{D}$. For each FO formula $\psi$ in $\mathsf{C}$, we define $\alpha(\psi) = (F, D)$ where $(F, D)$ is the finite duality pair uniquely characterizing it, which is obtained using the methods laid out in [tCD22]. On the other hand, for any modal formula $\varphi \in \mathsf{C}$ we define $\alpha(\varphi) = (G, H)$ to be the (finite) pair of positive and negative examples characterizing it (we obtain $(G, H)$ by following the procedure discussed before theorem 4.2.5 and in the proof of the theorem). Fix any morphism $f \in \mathsf{C}$, it should be of the form $f : \varphi \to ST_x(\varphi)$, and by definition there is an unique morphism $g \in \mathsf{D}$ s.t. $g : \alpha(ST_x(\varphi)) \to \alpha(\varphi)$. We simply let $\alpha(f) = g$. It is easy to verify that $F$ is indeed a contravariant functor. The existence of the functor $\alpha$ provides a nice duality.

$$\begin{tikzcd}
\varphi \arrow[r, dashed, "\alpha"] \arrow[d, "f"'] & (G, H) \\
ST_x(\varphi) \arrow[r, dashed, "\alpha"] & (F, D) \arrow[u, "\alpha(f)"']
\end{tikzcd}$$

The positive result from theorem 4.2.5 can be transferred over to the fragment $\mathbb{C}(\mathcal{L}_{\Box,\wedge,\vee,\top,\bot}[Prop])$. In [tCK24], it is already established that the aforementioned concept class has finite unique characterization. The crux of the proof is the idea of *negation translation*. Fix a set $PExt = \{q_i : p_i \in Prop\}$, now for every concept $\varphi$ of $\mathbb{C}(\mathcal{L}_{\Box,\wedge,\vee,\top,\bot}[Prop])$, there is a formula $\psi$ of $\mathbb{C}(\mathcal{L}_{\Diamond,\wedge,\vee,\top,\bot}[PExt])$ s.t. if we change every occurrence of $q_i$ with $\neg p_i$ in $\psi$, then $\varphi$ and $\psi$ are equivalent. In other words, the formula $\neg\varphi \leftrightarrow (\psi[\neg p_1/q_1, \ldots, \neg p_n, q_n])$ is derivable (in $K$). We call $\psi$ the negation translation of $\varphi$, and denote it by $\varphi^-$.

We can also translate pointed Kripke models in a similar fashion. Suppose $(M, s) = ((W, R, V), s)$ is a pointed Kripke model, define $(M^-, s) = ((W, R, V^-), s)$ where $V^-(p_i) = W \setminus V(q_i)$ for any proposition $p_i$. One key observation, which can be proved by a simple induction, is the following.

**Lemma 4.2.6** (Cate and Koudijs in [tCK24]). *For any concept $\varphi$ of $\mathbb{C}(\mathcal{L}_{\Diamond,\wedge,\vee,\top,\bot}[Prop])$, and a pointed Kripke model $(M, s)$,*

$$(M, s) \models \varphi^- \iff (M^-, s) \models \varphi^-[\neg p_1/q_1, \ldots, \neg p_n, q_n]$$

□

**Theorem 4.2.7** (Cate and Koudijs in [tCK24]). *The concept class $\mathbb{C}(\mathcal{L}_{\Box,\wedge,\vee,\top,\bot}[Prop])$ admits finite unique characterization*

*Proof.* Consider any concept $\varphi$ of $\mathbb{C}(\mathcal{L}_{\Box,\wedge,\vee,\top,\bot}[Prop])$ and take it's negation translation $\varphi^-$. We already know from theorem 4.2.5 that $\varphi^-$ has a finite unique characterization, $T$ say. Now let define

$$U = \{((M^-, s)+) : ((M, s), +) \in T\} \cup \{((M^-, s)-) : ((M, s), -) \in T\}$$

Following the previous lemma $U$ should uniquely characterize $\varphi$. □

We would also like to point out how the previous theorem builds upon the previous duality, to give us a new duality. Firstly, we extend our category $\mathsf{C}$ to the category $\mathsf{CE}$, whose objects are:

i. All modal formulas formed using $\{\Box, \wedge, \vee, \top, \bot\}$ (over the variable set $Prop$)

ii. All modal formulas formed using $\{\Diamond, \wedge, \vee, \top, \bot\}$ (over the variable set $PExt$)

iii. all FO formulas formed using $\{\exists, \wedge, \vee, \top, \bot\}$ (over the set $S_{PExt}$)

Now for each formula $\varphi$ described in i., we have a morphism $f \in \mathsf{CE}$ s.t. $f : \varphi \to \varphi^-$. The other morphisms are standard translation(s) from the formulas described in ii. to the formulas described in iii. Finally we close the morphisms under compositionality.

Similarly, we extend the category $\mathsf{D}$ to $\mathsf{DE}$, whose objects will be

i. Finite duality pairs $(F, D)$ of $S_{PExt}$ structures as objects.

ii. Finite pointed Kripke models pairs $(G, H)$ (over $PExt$).

iii. Finite pointed Kripke models $(G, H)$ (over $Prop$).

We define a contravariant functor $\beta$ from $\mathsf{CE}$ to $\mathsf{DE}$. For every object $x$ in $\mathsf{CE}$ satisfying condition ii. or condition iii., we simply define $\beta(x) = \alpha(x)$, and for any object $\varphi \in \mathsf{CE}$ satisfying condition i. we have an object $\varphi^-$, which satisfies condition ii., using this fact we define $\beta(\varphi) = \alpha(\varphi^-)$. For every morpshim $f$ between objects satisfying condition ii. to objects satisfying condition iii., we again define $\beta(f) = \alpha(f)$. For morpshims $f$ between objects satisfying condition i. to objects satisfying condition ii., we define $\beta(f) = \mathsf{id}_{\alpha(dom(f))}$ and finally For morpshims $f$ between objects satisfying condition i. to objects satisfying condition iii. we know that there is an object satisfying condition ii. and a morphism $g$ s.t. the following diagram commutes

$$\begin{tikzcd} \varphi \arrow[r, "g"] \arrow[dr, "f"'] & \varphi^- \arrow[d, "ST_x"] \\ & ST_x(\varphi) \end{tikzcd}$$

We define $\beta(f) = \alpha(ST_x)$. The reader can verify that $\beta$ is actually a functor. We now provide the diagram of the duality

$$\begin{tikzcd} \varphi \arrow[r, "f"] \arrow[dr, "g\circ f"'] & \varphi^- \arrow[r, dashed, "\beta"'] \arrow[d, "g"] & (G^-, H^-) & (G, H) \arrow[l, "\beta(f)"'] \\ & ST_x(\varphi^-) \arrow[r, dashed, "\beta"] & (F, D) \arrow[u, "\beta(g)"] \arrow[ur, "\beta(g\circ f))"'] & \end{tikzcd}$$

We turn our attention to a relatively simpler concept class, namely $\mathbb{C}(\mathcal{L}_{\Diamond,\neg}[Prop])$. Notice that the concept classes $\mathbb{C}(\mathcal{L}_{\Diamond,\neg}[Prop])$ $\mathbb{C}(\mathcal{L}_{\Diamond,\Box,\neg}[Prop])$ are the same. Every concept $\varphi$ of the concept class $\mathbb{C}(\mathcal{L}_{\Diamond,\Box\neg}[Prop])$ is of the form

$$\varphi = Q_1 \dots Q_m q$$

$$0 \xrightarrow{R} 1 \xrightarrow{R} \dots \xrightarrow{R} n$$

Figure 4.1: The frame of $LChain_n[S]$

where $Q_i \in \{\Box, \Diamond\}$ and $q$ is either a propositional variables in *Prop* or the negation of some variable in *Prop* (we will call such $q's$ as *literals*). Will call the segment $Q_1, \dots, Q_m$, the *modal part* of the formula $\varphi$ and $q$ the *propositional part* of the formula $\varphi$.

We firstly introduce a special class of pointed Kripke models, for every $S \subseteq Prop$, called $SRP[S] = ((W, R, V), w)$, where $W = \{w\}$, $R = \{(w, w)\}$ and $w \in V(p_i)$ if and only if $w \in S$.

**Proposition 4.2.8.** *For every $Q_1, \dots, Q_m \in \{\Box, \Diamond\}$ and every variable $p$, we have*

$$SRP[S] \models p \iff SRP[S] \models Q_1, \dots, Q_n p$$

*Proof.* The proof is by induction on the number of modalities in-front of $p$. In the case the number of modalities is 0, it is trivial. We assume that our claim holds if the number of modalities is $n$.

Now consider the formula $Q_1Q_2 \dots Q_{n+1}p$. Assume $SRP[S] \models p$ then it follows that $SRP[S] \models Q_2 \dots Q_{n+1}p$ from I.H. We consider two cases now, for the first case let $Q_1 = \Diamond$, then by definition of satisfiability, $SRP[S] \models \Diamond Q_2 \dots Q_{n+1}p$ if and only if $SRP[S] \models Q_2 \dots Q_{n+1}p$. For the second case let $Q_1 = \Box$, then again by definition of satisfiability $SRP[S] \models \Diamond Q_2 \dots Q_{n+1}p$ if and only if $SRP[S] \models Q_2 \dots Q_{n+1}p$. So, we can conclude that $SRP[S] \models Q_2 \dots Q_{n+1}p$ if and only if $SRP[S] \models Q_1Q_2 \dots Q_{n+1}p$, which coupled with the implication provided by I.H. gives us $SRP[S] \models p$ implies $SRP[S] \models Q_1Q_2 \dots Q_{n+1}p$.

For the other direction, let $SRP[S] \models Q_1Q_2 \dots Q_{n+1}p$. We have already established in the last paragraph that $SRP[S] \models Q_2 \dots Q_{n+1}p$ if and only if $SRP[S] \models Q_1Q_2 \dots Q_{n+1}p$, and now we use I.H. to obtain $SRP[S] \models Q_2 \dots Q_{n+1}p$ implies $SRP[S] \models p$. Combining all of these implications together, we get that $SRP[S] \models Q_1Q_2 \dots Q_{n+1}p$ implies $SRP[S] \models p$. This concludes our proof. □

**Corollary 4.2.8.1.** *For every $Q_1, \dots, Q_m \in \{\Box, \Diamond\}$ and every variable $p$, we have*

$$SRP[S] \models \neg p \iff SRP[S] \models Q_1, \dots, Q_n \neg p$$

□

Another class of important pointed Kripke models is that of linear chains. For any $S \subseteq Prop$, we put $LChain_n[S] = ((W, R, V), 0)$ where $W = \{0, \dots, n\}$; $aRb$ holds if and only if $b = a + 1$ and finally, for any $p \in Prop$,

$$V(p) = \begin{cases} n & \text{if } p \in S \\ \varnothing & \text{otherwise} \end{cases}$$

Figure 4.1 provides a pictorial representation of the underlying frame of $LChain_n[S]$. Similarly, we can consider the class of frames called *loop chains* and *pointed chains* which we will denote by $Ochain_n[S]$

$$\begin{tikzcd}
0 \arrow[r, "R_O"] & 1 \arrow[r, "R_O"] & \dots \arrow[r, "R_O"] & n \arrow[loop below, "R_O"]
\end{tikzcd}$$

Figure 4.2: The frame of $OChain_n[S]$

$$\begin{tikzcd}
 & & & & n+1 \\
0 \arrow[r, "R_P"] & 1 \arrow[r, "R_P"] & \dots \arrow[r, "R_P"] & n \arrow[loop below, "R_P"] \arrow[ur, "R_P"] &
\end{tikzcd}$$

Figure 4.3: The frame of $PChain_n[S]$

and $PChain_n[S]$ respectively. The reader can consult the figures 4.2 and 4.3 for a better understanding. Formally, $Ochain_n[S] = ((W_O, R_O, V_O), 0)$ has as its set of worlds $W_O = \{0, 1, \dots, n\}$, and

$$aR_Ob \iff b = a + 1 \text{ or } a = b = n$$

The function $V_O$ is the same as in the case of $LChain_n[S]$. For the case of $Pchain_n[S] = ((W_P, R_P, V_P), 0)$, we have as its set of worlds $W_P = \{0, 1, \dots, n, n+1\}$, and

$$aR_Pb \iff b = a + 1 \text{ or } a = b = n$$

The function $V_P$ is the same as in the case of $LChain_n[S]$. One interesting property is $LChain_n[S]$, $OChain_n[S]$ and $PChain_n[S]$ are $n-$bisimilar.

**Proposition 4.2.9.** *For every $n$ and $S \subseteq Prop$, we have that $LChain_n[S]$, $OChain_n[S]$ and $PChain_n[S]$ are n-bisimilar.*

*Proof.* We provide a set of relations $Z_0, \dots, Z_n$ between $LChain_n[S]$ and $OChain_n[S]$ s.t. $Z_0 \subseteq \dots \subseteq Z_n$ and it satisfies all the conditions for being a n-bisimulation. We simply put $Z_0 = \{(0, 0)\}$, and recursively define $Z_i = Z_{i-1} \cup \{i, i\}$ for any $i \leq n$. Notice that it is already a n-bisimulation.

The same set of relations gives us a n-bisimulation between $LChain_n[S]$ and $PChain_n[S]$. $\square$

Since the formulas we are dealing with in the concept class of $\mathbb{C}(\mathcal{L}_{\Diamond,\Box,\neg}[Prop])$ are rather simple, we can define a weaker notion of n-bisimulation.

**Definition 4.2.10** (weak n-bisimulation)**.** *Two pointed Kripke models $(M, s) = ((W_M, R_M, V_M), s)$ and $(N, s') = ((W_N, R_N, V_N), s')$ are weakly n-bisimilar if there exists a set of relations $Z_0, \dots, Z_n$ between $W_M$ and $W_n$ s.t.*

*i.* $sZ_0s'$.

*ii. If $sZ_ns'$ then $s$ and $s'$ agree on propositional variables.*

*iii. If $wZ_iw'$ and $wR_Mu$, then there exists $v$ s.t. $w'R_Nv$ and $uZ_{i+1}v$ (where $i+1 \leq n$).*

*iv. If $wZ_iw'$ and $w'R_Nv$, then there exists $u$ s.t. $wR_Mu$ and $uZ_{i+1}v$ (where $i+1 \leq n$).*

The difference between this weak version and n-bismulation is the subset clause that we had between the relations $Z_0, \ldots Z_n$. But this weaker version is enough to make concepts of $\mathbb{C}(\mathcal{L}_{\Diamond,\Box,\neg}[Prop])$ invariant.

**Proposition 4.2.11.** *For every concept $\varphi$ of the concept class $\mathbb{C}(\mathcal{L}_{\Diamond,\Box\neg}[Prop])$ with $d(\varphi) = n$, condition i. given below implies condition ii.*

*i. $(M, s)$ and $(N, s)$ are weakly n-bisimilar pointed Kripke models.*

*ii. $(M, s) \models \varphi$ if and only if $(N, s) \models \varphi$.*

*Proof.* The proof is by induction on the complexity or depth of the formula (in this case, both of them coincide). If $\varphi$ is of depth 0, then by condition *ii* of our weak n- bisimulation, the result follows. Now assume our claim holds for formulas of depth $< n$.

Let $\varphi$ be a formula of depth $n$, then either $\varphi = \Box\psi$ or $\varphi = \Diamond\psi$, where $d(\psi) < n$. We analyse the former case first. Suppose $(M, s) \models \Box\psi$ and $(M, s)$ is weakly n-bisimilar to $(N, s')$ via the relations $Z_0, \ldots Z_n$. By definition of satisfaction, for every $w$ with $sR_Mw$, we have $(M, w) \models \psi$. But notice that by condition iv. of weak n-bisimulation for every $w'$ s.t. $s'R_Nw'$ there is a $w''$ s.t. $sR_Mw''$ and $w'Z_1w''$, and furthermore $(M, w'')$ and $(N, w')$ are weakly n-1-bisimilar via the relations $Z_1, \ldots Z_n$. We use the I.H. to conclude that $(N, w') \models \psi$, and since $w'$ was arbitrary we get that $(N, s) \models \Box\psi$. For the other direction we use a similar argument using condition iii.

We focus on the case where $\varphi = \Diamond\psi$. Suppose $(M, s) \models \Diamond\varphi$ and $(M, s)$ is weakly n-bisimilar to $(N, s')$ via $Z_0, \ldots Z_n$. By definition of satisfaction, there is a $w$ with $sR_MW$ s.t. $(M, w) \models \psi$. Using condition iii. we conclude that there must exist a $w'$ s.t. $s'Rw'$ and $wZ_1w'$. Now notice that $Z_1, \ldots Z_n$ witnesses a weak n-bisimulation between $(M, w)$ and $(N, w')$. Using I.H. we can conclude that $(N, w') \models \psi$ and hence $(N, s') \models \Diamond\psi$. ☐

**Lemma 4.2.12.** *For every $n$ and $S \subseteq Prop$, the pointed Kripke models $SRP[S]$ and $LChain_n[S]$ are weakly n-bisimilar.*

*Proof.* We assume that the underlying set for $SRP[S]$ is $\{0\}$. We define the relations $Z_0, \ldots, Z_n$ as follows: $Z_i = \{(0, i)\}$. It easily follows that $Z_0, \ldots, Z_n$ witnesses a weak n-bisimulation between $SRP[S]$ and $LChain_n[S]$. ☐

**Lemma 4.2.13.** *For every $m > n$ and $S \subseteq Prop$, we have that $LChain_n[\varnothing]$ is weakly n-bisimilar to $LChain_m[S]$.*

*Proof.* We again define the set of relations $Z_0, \ldots, Z_n$ as follows: $Z_i = \{(i, i)\}$. It is easy to verify that is a weak n-bisimulation. ☐

The previous results will aid us to answer questions on finite characterization of $\mathbb{C}(\mathcal{L}_{\Box,\neg,\bot}[Prop])$. Firstly we take into account the formulas of the form $\varphi_1 = Q_1, \ldots, Q_n p$, where $Q_i \in \{\Box, \Diamond\}$ and $p \in Prop$. Define the set

$$T^{Prop}_{\varphi_1} = \{(SRP[\varnothing], -), (SRP[\{p\}], +), (SRP[Prop], +)\}$$

Notice that if $\psi$ fits $(SRP[\{p\}], +)$ then the propositional part of $\psi$ is with $p$ or $\neg q$, where $p \neq q$, according to result 4.2.8. Similarly if $\psi$ fits $(SRP[Prop], +)$ then the propositional part of $\psi$ cannot be the negation of any propositional variables. Hence, every $\psi$ fitting $T^{Prop}_{\varphi_1}$ must have $p$ as its propositional part. To manipulate the modal part, we propose the set

$$T^{Mod}_{\varphi_1} = \{(LChain_n[\{p\}], +), (LChain_n[\varnothing], -)\}$$

Suposse $\psi$ fits $T^{Mod}_{\varphi_1} \cup T^{Prop}_{\varphi_1}$ and $d(\psi) > n$, so we can assume $\psi = Q'_1 \ldots Q'_n V_1 \ldots V_k p$ (since $\psi$ fits $T^{Prop}_{\varphi_1}$, we can assume the propositional part to be $p$). Now $V_1$ can either be $\Box$ or $\Diamond$. In the former case, $LChain_n[\varnothing]$ would be a positive example for $\psi$, a contradiction. In the latter case, $LChain_n[\{p\}]$ will be a negative example, again a contradiction. By lemma 4.2.12 and result 4.2.8, $\varphi_1$ fits $LChain_n[\{p\}]$ and $(LChain_n[\varnothing], -)$. So any $\psi$ fitting $T^{Mod}_{\varphi_1} \cup T^{Prop}_{\varphi_1}$ must have $d(\psi) \leq n$ and the propositional part as $p$. Infact, $T^{Mod}_{\varphi_1}$ gives us a better bound on $d(\psi)$, whenever $\psi$ fits $T^{Mod}_{\varphi_2}$. Now suppose $d(\psi) < n$ and $\psi$ fits $T^{Mod}_{\varphi_1} \cup T^{Prop}_{\varphi_1}$, we can assume again that $\psi = Q'_1 \ldots Q'_k p$, where $k < n$. By lemma 4.2.13, proposition 4.2.11 and the proposition 4.2.8, we conclude that $LChain_n[\varnothing]$ and $LChain_n[\{p\}]$ must be negative examples of $\psi$, a contradiction! Hence $d(\psi) = n$ and the propositional part of $\psi$ should be $p$, whenever $\psi$ fits $T^{Mod}_{\varphi_1} \cup T^{Prop}_{\varphi_1}$.

To finish our finite characterization, we define the following sets for each $Q_i$,

$$T^{Q_i}_{\varphi_1} = \begin{cases} \{(LChain_{i-1}[\varnothing], +)\} & \text{if } Q_i = \Box \\ \{(LChain_{i-1}[\varnothing], -)\} & \text{if } Q_i = \Diamond \end{cases}$$

Now suppose $\psi$ fits $T^{Mod}_{\varphi_1} \cup T^{Prop}_{\varphi_1} \cup T^{Q_1}_{\varphi_1} \cup \cdots \cup T^{Q_n}_{\varphi_1}$. From our previous discussion, since $\psi$ fits $T^{Mod}_{\varphi_1} \cup T^{Prop}_{\varphi_1}$, we can assume $\psi = Q'_1 \ldots Q'_n p$. Suppose $Q'_i \neq Q_i$ for some $i \leq n$, then it follows that $\psi$ does not fit $T^{Q_i}_{\varphi_1}$. It follows that $T^{Mod}_{\varphi_1} \cup T^{Prop}_{\varphi_1} \cup T^{Q_1}_{\varphi_1} \cup \cdots \cup T^{Q_n}_{\varphi_1}$ is an unique characterization of $\varphi_1$.

We analyse the case for the formulas of form $\varphi_2 = Q_1 \ldots Q_n \neg p$. Firstly, for every pointed Kripke model $(M, s) = ((W, R, V), s)$ over the set $Prop$, we define $(M, s)^* = ((W, R, V^*), s)$, where $V^*(p) = W \setminus V(p)$. We have this nice result

**Proposition 4.2.14.** *For any $n$, $S \subseteq Prop$ and a formula $\varphi$ of $\mathbb{C}(\mathcal{L}_{\Box,\neg}[Prop])$ with $d(\varphi) \leq n$, the following holds:*

$$LChain_n[S] \models \varphi \iff LChain_n[S]^* \models \neg\varphi$$

*Proof.* The proof is again by induction on the depth of the formula $\varphi$. If the depth is 0, then our claim follows straight from the way $LChain_n[S]^*$ is defined. Suppose our claim holds for any formula $\psi$ with $d(\psi) < n$.

Assume $\varphi = \Diamond\psi$ is a formula with $d(\varphi) = n$. Suppose $LChain_n[S] \models \varphi$, then by definition $(LChain_n[S], 1) \models \psi$.[4] Now since $(LChain_n[S], 1)$ is weakly n-1 bisimilar to $(LChain_{n-1}[S], 0)$, get that $LChain_{n-1}[S] \models \psi$. Using I.H. we get that $LChain_{n-1}[S]^* \models \psi$. Again, $(LChain_{n-1}[S]^*, 0)$ is weakly n-bisimilar to $(LChain_n[S]^*, 1)$ and finally we have $(LChain_n[S], 1)^* \models \psi$, which immediately gives us $LChain_n[S]^* \models \Diamond\psi$.

For the converse direction, just notice that $((M, s)^*)^* = (M, s)$, for every pointed Kripke model $(M, s)$. This closes the induction argument and completes our proof. □

Notice that $\neg\varphi_2$ is equivalent to a formula of the form $\varphi_1$. We already know that $\varphi_1$ has an unique characterization, $T_{\varphi_1}$ say. We define

$$T_{\varphi_2} = \{((M, s)^*, u) : ((M, s), u) \in T_{\varphi_1}\} \cup \{(SRP[Prop], -)\} \cup \{(SRP[\varnothing], +)\}$$

Since every $\psi$ that fits $T_{\varphi_2}$ should fit $(SRP[Prop], -)$, it follows that the propositional part of $\psi$ cannot be $\top$, similarly $T_{\varphi_2}$ should fit $(SRP[Prop], +)$, which means the propositional part of $\psi$ cannot be $\bot$. Therefore, if $\psi$ fits $T_{\varphi_2}$ then $\psi$ is a concept of $\mathbb{C}(\mathcal{L}_{\Box,\neg}[Prop])$. By the previous proposition it follows that $\{((M, s)^*, u) : ((M, s), u) \in T_{\varphi_1}\}$ distinguishes $\varphi_2$ from all the other concepts of $\mathbb{C}(\mathcal{L}_{\Box,\neg}[Prop])$, hence $T_{\varphi_2}$ is an unique characterization of $\varphi_2$.

We consider the formulas of the form $\varphi_3 = Q_1 \ldots Q_n\top$. Since we have the equivalence $\Box\top \leftrightarrow \top$, we can assume $Q_n = \Diamond$. Notice that every formula $\psi$ fitting the set

$$T^{Prop}_{\varphi_3} = \{(SRP[Prop], +), (SRP[\varnothing], +)\}$$

must have $\top$ as its propositional part, following result 4.2.8. Now for determining the modal part, we propose the set

$$T^{Mod}_{\varphi_3} = \{(PChain_n[\varnothing], +), (LChain_n[\varnothing], +)\}$$

Suppose a formula $\psi = Q'_1 \ldots Q'_n V_1 \ldots V_k\top$ fits $T^{Mod}_{\varphi_3}$ and $\psi$ is not equivalent to a formula $\psi'$ with $d(\psi') \leq n$. Therefore, atleast one of $V_i$ is $\Diamond$. Suppose $V_1 = \Diamond$, then $\psi$ cannot fit $(LChain_n[\varnothing], +)$. On the other hand let $j$ be the least index s.t. $V_j = \Diamond$ and for all $i < j$, $V_i = \Box$. We consider the pointed Kripke model $PChain_n[\varnothing]$, we consider the path $(n, \ldots n, n+1)$ of length $i-1$, where we loop around $n$ for $i-2$ and then take the path $n+1$. It is clear that $PChain_n[\varnothing], n \not\models V_1 \ldots V_k\top$ and we know that

$$PChain_n[\varnothing], 0 \models \psi \iff PChain_n[\varnothing], n \not\models V_1 \ldots V_k\top$$

hence $\psi$ cannot fit $(PChain_n[\varnothing], +)$. So we have that any $\psi$ fitting $T^{Mod}_{\varphi_3}$, must have $d(\psi) \leq n$. On the other hand if for some concept $\psi$ of $\mathbb{C}(\mathcal{L}_{\neg,\Box,\bot}[Prop])$ with the propositional part of $\psi$ being $\top$ and $d(\psi) \leq n$, it trivially follows that $PChain_n[\varnothing] \models \psi$. Since $n$ was arbitrary and $\varphi_3$ was not used in the previous description, we can extend our observation

[4]This is abusive notation what we actually mean by $(LChain_n[S], 1)$ is the pointed model $((W, R, V), 1)$ where $LChain_n[S] = ((W, R, V), 0)$. We will abuse notation for this particular proof, but the reader can figure it out from the context.

**Proposition 4.2.15.** *For every concept $\psi$ of $\mathbb{C}(\mathcal{L}_{\neg,\Box,\bot}[Prop])$ with the propositional part of $\psi$ being $\top$, and for every $m$, the following equivalence holds:*

$$PChain_m[\varnothing] \models \psi \;\&\; LChain_m[\varnothing] \models \psi \iff d(\psi) \leq m$$

$\square$

Keeping this result in mind we define the following sets

$$T^{Q_i}_{\varphi_3} = \begin{cases} \{(LChain_{i-1}[\varnothing], -), (PChain_{i-1}[\varnothing], +)\} & \text{if } Q_i = \Diamond \\ \{(LChain_{i-1}[\varnothing], +), (PChain_{i-1}[\varnothing], -)\} & \text{if } Q_i = \Box \end{cases}$$

Finally we claim that $T^{Prop}_{\varphi_3} \cup T^{Mod}_{\varphi_3} \cup T^{Q_1}_{\varphi_3} \cup \cdots \cup T^{Q_n}_{\varphi_3}$ is an unique characterization of $\varphi_3$. It follows from our discussion that every concept $\psi$ of $(\mathcal{L}_{\neg,\Box,\bot}[Prop])$ that fits $T^{Prop}_{\varphi_3} \cup T^{Mod}_{\varphi_3}$ must have $\top$ as its propositional part and $d(\psi) \leq n$. If $\psi$ fits $T^{Q_i}_{\varphi_3}$, by the previous proposition, it follows that $d(\psi) > i - 1$. Therefore, we can assume $\psi = Q'_1, \ldots, Q'_n \top$. Now it follows that if $\psi$ fits $T^{Q_i}_{\varphi_3}$, then $Q_i = Q'_i$. This gives us $\psi \cong \varphi_3$.

Our final analysis is of the formulas of the form $\varphi_4 = Q_1 \ldots Q_n \bot$. Due to the equivalence $\Diamond \bot \leftrightarrow \bot$, we can assume that $Q_n = \Box$. Notice that, any formula $\psi$ fitting

$$T^{Prop}_{\varphi_4} = \{(SRP[Prop], -), (SRP[\varnothing], -)\}$$

must have $\bot$ as its propositional part. To manipulate the modal part we propose the set

$$T^{Mod}_{\varphi_4} = \{(PChain_n[\varnothing], -), (LChain_n[\varnothing], -)\}$$

Any formula $\psi$ fitting $T^{Prop}_{\varphi_4} \cup T^{Mod}_{\varphi_4}$ must be of the form $\psi = Q'_1 \ldots Q'_n V_1 \ldots V_k \bot$. Let's assume, aiming for a contradiction, that $d(\psi) > n$. It follows that atleast one of $V_i$ is $\Box$, if $V_1 = \Diamond$ then $\psi$ would not fit $(LChain_n[\varnothing], -)$. On the other hand let $j$ be the least index s.t. $V_j = \Box$. We take the path $(n, \ldots, n, n{+}1)$ of length $i - 1$, where we loop around $n$ for $i - 2$ times then use $R$ to reach the node $n + 1$. As it turns out $\psi$ does not fit $(LChain_n[\varnothing], -)$, a contradiction in both cases. Lastly, we define a collection of sets

$$T^{Q_i}_{\varphi_4} = \begin{cases} \{(LChain_{i-1}[\varnothing], +), (PChain_{i-1}[\varnothing], -)\} & \text{if } Q_i = \Diamond \\ \{(LChain_{i-1}[\varnothing], -), (PChain_{i-1}[\varnothing], +)\} & \text{if } Q_i = \Box \end{cases}$$

We claim that $T^{Prop}_{\varphi_4} \cup T^{Mod}_{\varphi_4} \cup T^{Q_1}_{\varphi_4} \cup \cdots \cup T^{Q_n}_{\varphi_4}$ is an unique characterization of $\varphi_4$. It follows from our discussion that every concept $\psi$ of $(\mathcal{L}_{\neg,\Box,\bot}[Prop])$ that fits $T^{Prop}_{\varphi_4} \cup T^{Mod}_{\varphi_4}$ must have $\bot$ as its propositional part and $d(\psi) \leq n$. Now if a formula $\psi$ fits $T^{Prop}_{\varphi_4} \cup T^{Mod}_{\varphi_4} \cup T^{Q_1}_{\varphi_4} \cup \cdots \cup T^{Q_n}_{\varphi_4}$ we assume it to be of the form $\psi = Q'_1 \ldots Q'_n \bot$, notice that since $\psi$ fits $T^{Q_i}_{\varphi_4}$ as well we get that $Q_i = Q'_i$. Finally we conclude that $\varphi_4 \cong \psi$. This finishes our proof gives us the following result.

$$\begin{tikzcd}
 & & & & & n+1 \\
0(p) \arrow[r, "R"] & 1(\varnothing) \arrow[r, "R"] & 2(p) \arrow[r, "R"] & \dots \arrow[r, "R"] & n(\varnothing) \arrow[ur, "R"] \arrow[loop below, "R"] &
\end{tikzcd}$$

Figure 4.4: Diagram of $PChain_n[\{0,2\},\{p\}]$, here $V'(p) = \{0,2\}$

**Theorem 4.2.16.** *The concept class $\mathbb{C}(\mathcal{L}_{\neg,\Box,\bot}[Prop])$ admits finite characterization.* $\Box$

Infact this result can be extended further to Polynomial sized unique characterization.

**Corollary 4.2.16.1.** *The collection $\mathbb{C}(\mathcal{L}_{\neg,\Box,\bot})$ admits PSUC.* $\Box$

*Proof.* Notice that for any $Prop$, and any concept $\varphi$ of $\mathbb{C}(\mathcal{L}_{\neg,\Box,\bot}[Prop])$, there is an unique characterization, $T_\varphi$ say, of $\varphi$ s.t.

$$|T_\varphi| \leq 2d(\varphi) + 10$$

Now as $d(\varphi) \leq |\varphi|$, we can conclude that the polynomial $2|\varphi| + 10$ witnesses the PSUC of $\mathbb{C}(\mathcal{L}_{\neg,\Box,\bot})$. $\Box$

### 4.2.2 The negative results

The next concept class we focus on analysing the fragments $\mathbb{C}(\mathcal{L}_{\Diamond,\mathsf{3XOR}}[Prop]), \mathbb{C}(\mathcal{L}_{\Diamond,\mathsf{AIMP}}[Prop])$ and $\mathbb{C}(\mathcal{L}_{\Diamond,\mathsf{OXOR}}[Prop])$ and its $\Box$ counterparts. The difference between these concept classes from the others we have discussed so far is that they do *not* admit finite characterization. To prove this formally, we take the help of PC reductions. The main observation behind the proof is that the modal Logic $K$ is *not* locally tabular i.e. given a finite set of variables, there are only infinitely many non-equivalent formulas. For example the sets

$$\{p, \Diamond p, \Diamond\Diamond p, \dots \Diamond^i p \dots\} \qquad \{p, \Box p, \Box\Box p, \dots \Box^i p \dots\}$$

are infinite sets where no two formulas are equivalent. The idea is to replace/mimic each variable $p_i$ by the formula $\Diamond^i p$ or $\Box^i p$, for any formula of the concerned concept classes.

We firstly generalise our class of models $PChain_n[S]$. For every $n$, $S \subseteq Prop$ and $\mathbb{S} \subseteq \{0,1,\dots n\}$, we define a class of pointed Kripke models $PChain_n[\mathbb{S}, S]$. The underlying frame of $PChain_n[\mathbb{S}, S]$ is the same as $PChain_n[S]$, the only difference is the valuation function. Formally, assume that $PChain_n[S] = ((W,R,V),0)$, then $PChain_n[\mathbb{S}, S] = ((W,R,V'),0)$, where

$$i \in V'(p) \iff i \in \mathbb{S}$$

for every $p \in S$. Similarly we define the class $LChain_n[\mathbb{S}, S]$, the reader can check the figures 4.4 and 4.5 for a better intuition. The following result is immedaiate.

$$0(p) \xrightarrow{R} 1(\varnothing) \xrightarrow{R} 2(p) \xrightarrow{R} \dots \xrightarrow{R} n(\varnothing)$$

Figure 4.5: Diagram of $LChain_n[\{0,2\},\{p\}]$, here $V'(p) = \{0,2\}$

**Proposition 4.2.17.** *For every pointed Kripke frame $PChain_n[\mathbb{S}, S]$, we have the following equivalence*

$$PChain_n[\mathbb{S}, S] \models \Box^i p \iff i \in \mathbb{S} \ \& \ p \in S$$

$\Box$

**Proposition 4.2.18.** *For every pointed Kripke frame $LChain_n[\mathbb{S}, S]$, we have the following equivalence*

$$LChain_n[\mathbb{S}, S] \models \Box^i p \iff i \in \mathbb{S} \ \& \ p \in S$$

$\Box$

The next step is to define, for every set of propositional formulas $Pr$, two concept classes, namely $\mathsf{Rd}_\Diamond(Pr)$ and $\mathsf{Rd}_\Box(Pr)$. Remember that $\mathbb{C}(\mathcal{L}_{Pr}[\mathsf{Inf}Prop])$ is the concept class which has all the propositional formulas formed using the infinite set of variables $\mathsf{Inf}Prop$, as concepts. The example space is the set of all variable assignments over $\mathsf{Inf}Prop$. We define $\mathsf{Rd}_\Diamond(Pr) = (C, E, \lambda)$ as follows: For every concept $\varphi(p_{i_1}, \dots, p_{i_k})$ of $\mathbb{C}(\mathcal{L}_{Pr}[\mathsf{Inf}Prop])$ with it's free variables displayed we define,

$$C = \{\varphi(\Diamond^{i_1} p/p_{i_1}, \dots, \Diamond^{i_k} p/p_{i_k}) : \varphi(p_{i_1}, \dots p_{i_k}) \text{ is a concept of } \mathbb{C}(\mathcal{L}_{Pr}[\mathsf{Inf}Prop])\}$$

$E$ is again the set of all pointed Kripke models, and $\lambda(\psi) = \{(N, s) \in E : (N, s) \models \psi\}$.

The concept class $\mathsf{Rd}_\Box(Pr)$ is also defined similarly. The concepts of $\mathsf{Rd}_\Diamond(Pr)$ is the set

$$\{\varphi(\Box^{i_1} p/p_{i_1}, \dots, \Box^{i_k} p/p_{i_k}) : \varphi(p_{i_1}, \dots p_{i_k}) \text{ is a concept of } \mathbb{C}(\mathcal{L}_{Pr}[\mathsf{Inf}Prop])\}$$

The example space of $\mathsf{Rd}_\Box(Pr)$ is same as that of $\mathsf{Rd}_\Diamond(Pr)$, and the assignment function for each concept $\psi$ of $\mathsf{Rd}_\Box(Pr)$ is also the same as in $\mathsf{Rd}_\Diamond(Pr)$. It is easy of infer that $\mathsf{Rd}_\Diamond(Pr)$ and $\mathsf{Rd}_\Box(Pr)$ are sub concept classes of $\mathbb{C}(\mathcal{L}_{\Diamond,Pr}[\{p\}])$ and $\mathbb{C}(\mathcal{L}_{\Box,Pr}[\{p\}])$, respectively.

For every set of propositional formulas $Pr$, there is a natural bijection from the concepts of $\mathbb{C}(\mathcal{L}_{Pr}[\mathsf{Inf}Prop])$ to the concepts of $\mathsf{Rd}_\Diamond(Pr)$. We define the bijection, denoted by $\mathsf{NDia}_{Pr}$, as follows:

$$\mathsf{NDia}_{Pr}(\varphi(p_{i_1}, \dots, p_{i_n})) = \varphi(\Diamond^{i_1} p/p_{i_1}, \dots, \Diamond^{i_k} p/p_{i_k})$$

Analogous to $\mathsf{NDia}_{Pr}$, we define the function $\mathsf{NBox}_{Pr}$ from the concepts of $\mathbb{C}(\mathcal{L}_{Pr}[\mathsf{Inf}Prop])$ to the concepts of $\mathsf{Rd}_\Box(Pr)$,

$$\mathsf{NBox}_{Pr}(\varphi(p_{i_1}, \dots, p_{i_n})) = \varphi(\Box^{i_1} p/p_{i_1}, \dots, \Box^{i_k} p/p_{i_k})$$

which is again a bijection. Our original goal is to define PC reductions from the concept class $\mathbb{C}(\mathcal{L}_{Pr}[\mathsf{Inf}Prop])$ to the concept classes $\mathsf{Rd}_{\Box}(Pr)$, and $\mathsf{Rd}_{\Diamond}(Pr)$. Following that we want to define a function $\mathsf{TDia}_{Pr}$, on each pair $(c, t)$, where $c$ is a concept of $\mathsf{Rd}_{\Diamond}(Pr)$ and $t = (e, q)$ is a labelled example of $\mathsf{Rd}_{\Diamond}(Pr)$ that fits $c$. We define $\mathsf{TDia}_{Pr}$ in the following way: $\mathsf{TDia}_{Pr}(c, (e, q)) = \{(\alpha, q)\}$ where,

$$\alpha(p_m) = 1 \iff e \models \Diamond^m p$$

Similarly, we define $\mathsf{TBox}_{Pr}$, on a pair $(c', (e', q'))$. Here $c'$ is a concept of $\mathsf{Rd}_{\Box}(Pr)$ and $(e', q')$ is a labelled example of $\mathsf{Rd}_{\Box}(Pr)$ that fits $c$. Define $\mathsf{TBox}_{Pr}(c', (e', q')) = \{(\alpha', q)\}$ where,

$$\alpha'(p_m) = 1 \iff e' \models \Box^m p$$

Notice that $\mathsf{TBox}_{Pr}$ and $\mathsf{TDia}_{Pr}$ is invariant of the concepts ($c', c$ respectively) and just depend on the example of the labelled example pairs ($(e', q'), (e, q)$ respectively). Furthermore, since for every $Pr$, the example spaces of $\mathsf{Rd}_{Pr}$ are identical, we get that $\mathsf{TBox}_{Pr}(c, (e, q)) = \mathsf{TBox}_{Pr'}(c, (e, q))$ where $(e, q)$ is a labelled example of $\mathsf{Rd}_{\Box}(Pr)$ and $Cl(Pr') \subseteq Cl(Pr)$. We can say the same thing for $\mathsf{TDia}_{Pr}, \mathsf{TDia}_{Pr'}$. As a result of the previous definitions, we can prove the following claim:

**Lemma 4.2.19.** *For every set of propositional formulas $Pr$, and every concept $\varphi$ of $\mathbb{C}(\mathcal{L}_{Pr}[\mathsf{Inf}Prop])$ we have the following:*

*i. For any example $e$ of $\mathsf{Rd}_{\Diamond}(Pr)$, $\mathsf{NDia}_{Pr}(\varphi)$ fits $(e, q)$ if and only if $\varphi$ fits $\mathsf{TDia}_{Pr}(\varphi, (e, q))$.*

*ii. For any example $e$ of $\mathsf{Rd}_{\Box}(Pr)$, $\mathsf{NBox}_{Pr}(\varphi)$ fits $(e, q)$ if and only if $\varphi$ fits $\mathsf{TBox}_{Pr}(\varphi, (e, q))$.*

*Proof.* Notice that it is sufficient to prove the theorem for the set of propositional formulas $\{\neg, \wedge\}$. Since every set of boolean connectives $Pr$, any concept $\varphi$ of the concept class $\mathbb{C}(\mathcal{L}_{Pr}[\mathsf{Inf}Prop])$ can be expressed as a DNF formula, $\varphi'$ say. We get that $\mathsf{NDia}_{Pr}(\varphi)$ fits $(e, q)$ iff $\mathsf{NDia}_{\{\neg,\wedge\}}(\varphi')$ fits $(e, q)$, on the other hand $\varphi$ fits $\mathsf{TDia}_{Pr}(\varphi, (e, q))$ iff $\varphi'$ fits $\mathsf{TDia}_{\{\neg,\wedge\}}(\varphi', (e, q))$. A similar statement hold for $\mathsf{NBox}_{Pr}, \mathsf{NBox}_{Pr}$.

We focus on *i.* first and proceed by induction on the complexity of $\varphi$. Suppose that $(e, q)$ is a labelled example and $q$ is $1(0)$, additionally let $\varphi = p_k$ is a propositional variable i.e. a formula of complexity 0. Assuming $\mathsf{TDia}(e, q) = (\alpha, q)$, $\varphi$ fits $(\alpha, q)$ if and only if $\alpha(p_k) = 1(\alpha(p_k) = 0)$. By definition, $\alpha(p_k) = 1$ if and only if $e \models \Diamond^k p$. But notice that $\Diamond^k p = \mathsf{NDia}(p_k)$. This closes the base case.

Now assume for any formula of length $< n$, our claim holds. Let $\psi$ be any formula of length $n$. $\psi$ either of the form $\psi = \neg\theta$ or $\psi = \eta \wedge \theta$. For the former case, $\mathsf{NDia}(\psi)$ fits $(e, q)$ if and only if $\mathsf{NDia}_{\neg,\wedge}(\theta)$ fits $(e, 1-q)$. Using I.H. we get that $\mathsf{NDia}_{\{\neg,\wedge\}}(\theta)$ fits $(e, 1-q)$ if and only if $\theta$ fits $\mathsf{TDia}_{\{\neg,\wedge\}}(\theta, (e, 1-q))$. Notice that $\mathsf{TDia}_{\{\neg,\wedge\}}(\theta, (e, 1-q)) = (\{\alpha, 1-q\})$, if and only if $\mathsf{TDia}_{\{\neg,\wedge\}}(\neg\theta, (e, q)) = (\{\alpha, q\})$ and by the definition of satisfaction we have that $\theta$ fits $\mathsf{TDia}_{\{\neg,\wedge\}}(\theta, (e, 1-q))$ if and only if $\neg\theta = \psi$ fits $\mathsf{TDia}_{\{\neg,\wedge\}}(\neg\theta, (e, q))$. The chain of equivalences gives us our claim.

For the latter case, we see that $\mathsf{NDia}_{\{\neg,\wedge\}}(\psi)$ fits $(e, q)$ depends on the value of $q$. Firstly, we assume $q = 1$, then we have that the last condition happens iff $\mathsf{NDia}_{\{\neg,\wedge\}}(\theta)$ and $\mathsf{NDia}_{\{\neg,\wedge\}}(\eta)$ both fits $(e, q)$. Using I.H. we get the last condition happens iff $\theta$ fits $\mathsf{TDia}_{\{\neg,\wedge\}}(\theta, (e, q))$ and $\eta$ fits $\mathsf{TDia}_{\{\neg,\wedge\}}(\eta, (e, q))$. Again following the definition of the satisfaction relation, we conclude $\theta$ fits $\mathsf{TDia}_{\{\neg,\wedge\}}(\theta, (e, q))$ and $\eta$ fits

$\mathsf{TDia}_{\{\neg,\wedge\}}(\eta,(e,q))$ iff $\theta \wedge \eta = \psi$ fits $\mathsf{TDia}_{\{\neg,\wedge\}}(\eta \wedge \theta,(e,q))$. This closes the induction case and our claim is proven.

For the case of *ii.*, we reuse the above proof. The reader can verify that the above proof changes to a proof of *ii.*, just by changing $\Diamond$ to $\Box$ and the functions $\mathsf{NDia}_{\{\neg,\wedge\}}, \mathsf{TDia}_{\{\neg,\wedge\}}$ to $\mathsf{NBox}_{\{\neg,\wedge\}}, \mathsf{TBox}_{\{\neg,\wedge\}}$ respectively. $\Box$

The previous theorem shows us that $\mathsf{NDia}_{Pr}$ and $\mathsf{TDia}_{Pr}$ satisfy the second and fourth requirements of PC reduction from $\mathcal{C}(\mathcal{L}_{Pr}[\mathsf{Inf}Prop])$ to $\mathsf{Rd}_{\Diamond}(Pr)$. Similarly, the previous theorem again shows us that $\mathsf{NBox}_{Pr}$ and $\mathsf{TBox}_{Pr}$ satisfy the second and fourth requirements of PC reduction from $\mathcal{C}(\mathcal{L}_{Pr}[\mathsf{Inf}Prop])$ to $\mathsf{Rd}_{\Box}(Pr)$. in both the cases the constant polynomial $p(x) = 1$ satisfy the third condition of our reduction. We focus on the fourth condition.

**Definition 4.2.20.** *We call a variable assignment $\mu : \mathsf{Inf}Prop \to \{0,1\}$ eventually zero, if there exists an $i$ s.t. for every $j \leq i$, $\mu(p_j) = 0$.*

The interesting thing about eventually zero assignments is that they form a critical set of the for every concept class $\mathbb{C}(\mathcal{L}_{Pr}[\mathsf{Inf}Prop])$. In other words, if $\varphi$ and $\psi$ are two non-equivalent concepts of $\mathbb{C}(\mathcal{L}_{Pr}[\mathsf{Inf}Prop])$ then there is an eventually zero variable assignment, $t$ say, s.t. $t \not\models \varphi \leftrightarrow \psi$. Fortunately for us, we can represent eventually zero variable assignments as pointed Kripke models. We define a collection of functions $\mathsf{RDia}_{Pr}$ and $\mathsf{RBox}_{Pr}$, from the example set of $\mathbb{C}(\mathcal{L}_{Pr}[\mathsf{Inf}Prop])$ to the example set of $\mathsf{Rd}_{\Diamond}$ and $\mathsf{Rd}_{\Box}$ respectively. Let $t$ denote an eventually zero variable assignment, where $i$ witness the index s.t. for every $j \geq i$, $t(p_j) = 0$ let $\mathbb{S}$ be defined as follows: $\mathbb{S} = \{m : t(p_m) = 1\}$. Now we define,

$$\mathsf{RDia}_{Pr}(e) = LChain_i[\mathbb{S}, \{p\}]$$

$$\mathsf{RBox}_{Pr}(e) = PChain_i[\mathbb{S}, \{p\}]$$

The next result is immediate following propositions 4.2.17 and 4.2.18.

**Lemma 4.2.21.** *For every concept $\varphi$ and example $e$ of $\mathbb{C}(\mathcal{L}_{Pr}[\mathsf{Inf}Prop])$, we have the following*

*i.* $e \models \varphi$ *if and only if* $\mathsf{RDia}_{Pr}(e) \models \mathsf{NDia}_{Pr}(\varphi)$.

*ii.* $e \models \varphi$ *if and only if* $\mathsf{RBox}_{Pr}(e) \models \mathsf{NBox}_{Pr}(\varphi)$.

*Proof.* It is enough to prove the theorem for the set $\{\neg, \wedge\}$. Firstly, the definitions of $\mathsf{RBox}_{Pr}$ and $\mathsf{RDia}_{Pr}$ are independent of $Pr$. So, for any $e$, $\mathsf{RDia}_{Pr}(e) = \mathsf{RDia}_{\{\neg,\wedge\}}(e)$. Again, notice that $\mathsf{NDia}_{\{\neg,\wedge\}}$, is distributive i.e. $\mathsf{NDia}_{\{\neg,\wedge\}}(\psi \wedge \theta) = \mathsf{NDia}_{\{\neg,\wedge\}}(\psi) \wedge \mathsf{NDia}_{\{\neg,\wedge\}}(\theta)$, and $\mathsf{NDia}_{\{\neg,\wedge\}}(\neg\psi) = \neg\mathsf{NDia}_{\{\neg,\wedge\}}(\psi)$. We know what every concept class $\mathbb{C}(\mathcal{L}_{Pr}[\mathsf{Inf}Prop])$ is a sub concept class of $\mathbb{C}(\mathcal{L}_{\{\neg,\wedge\}}[\mathsf{Inf}Prop])$, and so every concept $\varphi$ of $\mathbb{C}(\mathcal{L}_{Pr}[\mathsf{Inf}Prop])$, is also a concept of $\mathbb{C}(\mathcal{L}_{\{\neg,\wedge\}}[\mathsf{Inf}Prop])$. Hence, from the definition, $\mathsf{NDia}_{Pr}(\varphi) = \mathsf{NDia}_{\{\neg,\wedge\}}(\varphi)$.

We start by proving $i$ and proceed via induction. The base case is simply given my the proposition 4.2.19. Now let our claim hold for any formula of complexity $< n$. Now assume that $\varphi$ is a formula of complexity $n$, therefore $\varphi$ is either of the form $\neg\psi$ or of the form $\psi \wedge \theta$.

For the former case, $e \models \varphi$ if and only if $e \not\models \psi$. Now the latter case happens if and only if $\mathsf{RDia}_{\{\neg,\wedge\}}(e) \not\models \mathsf{NDia}_{\{\neg,\wedge\}}(\psi)$, using I.H. Now the latter case happens if and only if $\mathsf{RDia}_{\{\neg,\wedge\}}(e) \models \neg\mathsf{NDia}_{\{\neg,\wedge\}}(\psi)$. But notice that $\neg\mathsf{NDia}_{\{\neg,\wedge\}}(\psi) = \mathsf{NDia}_{\{\neg,\wedge\}}(\neg\psi)$.

For the latter case, $e \models \varphi$ if and only if $e \models \psi$ and $e \models \theta$. Using I.H. the latter statement holds iff $\mathsf{RDia}_{\{\neg,\wedge\}}(e) \models \mathsf{NDia}_{\{\neg,\wedge\}}(\psi)$ and $\mathsf{RDia}_{\{\neg,\wedge\}}(e) \models \mathsf{NDia}_{\{\neg,\wedge\}}(\theta)$. Now notice that $\mathsf{NDia}_{\{\neg,\wedge\}}(\psi) \wedge \mathsf{NDia}_{\{\neg,\wedge\}}(\theta) = \mathsf{NDia}_{\{\neg,\wedge\}}(\psi\wedge\theta)$. Therefore, $\mathsf{RDia}_{\{\neg,\wedge\}}(e) \models \mathsf{NDia}_{\{\neg,\wedge\}}(\psi)$ and $\mathsf{RDia}_{\{\neg,\wedge\}}(e) \models \mathsf{NDia}_{\{\neg,\wedge\}}(\theta)$ iff $\mathsf{RDia}_{\{\neg,\wedge\}}(e) \models \mathsf{NDia}_{\{\neg,\wedge\}}(\psi \wedge \theta)$. This closes the induction case and our claim is proven.

The proof for *ii.* is analogous. Again, just change $\mathsf{NDia}$ to $\mathsf{NBox}$ and $\mathsf{RDia}$ to $\mathsf{RBox}$. ☐

**Theorem 4.2.22.** *For every set of propositional formulas $Pr$, the concept class $\mathbb{C}(\mathcal{L}_{Pr}[\mathsf{Inf}Prop])$ is PC reducible to the concept class $\mathsf{Rd}_\Diamond(Pr)$, and to the concept class $\mathsf{Rd}_\Box(Pr)$*

*Proof.* For the reduction from $\mathbb{C}(\mathcal{L}_{Pr}[\mathsf{Inf}Prop])$ to $\mathsf{Rd}_\Diamond(Pr)$, we use the maps $\mathsf{NDia}_{Pr}, \mathsf{TDia}_{Pr}, \mathsf{RDia}_{Pr}$ and the function $p(x) = 1$. Lemma 4.2.19 tells us condition 2 and 4 of the reduction is satisfied by our chosen maps, and lemma 4.2.21 tells us condition 1 of the reduction holds as well. $p(x) = 1$ trivially satisfies condition 3.

For the other reduction from $\mathbb{C}(\mathcal{L}_{Pr}[\mathsf{Inf}Prop])$ to $\mathsf{Rd}_\Box(Pr)$, we use the maps $\mathsf{NBox}_{Pr}, \mathsf{TBox}_{Pr}, \mathsf{RBox}_{Pr}$ and the function $p(x) = 1$. The previously mentioned lemmas work here as well, and so does $p(x) = 1$. ☐

**Theorem 4.2.23.** *For any set of propositional formulas $Pr$, if $\mathbb{C}(\mathcal{L}_{Pr}[\mathsf{Inf}Prop])$ does not have finite characterization, then $\mathbb{C}(\mathcal{L}_{\Diamond,Pr}[\{p\}])$ and $\mathbb{C}(\mathcal{L}_{\Box,Pr}[\{p\}])$ does not have finite characterization as well.*

*Proof.* The proof is trivial, suppose $\mathbb{C}(\mathcal{L}_{\Diamond,Pr}[\{p\}])$ has finite characterization then $\mathsf{Rd}_\Diamond(Pr)$ has finite characterization as well. Now from the previous theorem and using theorem 3.3.7.2 it implies $\mathbb{C}(\mathcal{L}_{Pr}[\mathsf{Inf}Prop])$ have finite characterization. The contraposition is the statement of the theorem.

The same argument works for $\mathbb{C}(\mathcal{L}_{\Box,Pr}[\{p\}])$ as well, just change $\mathsf{Rd}_\Diamond(Pr)$ to $\mathsf{Rd}_\Box(Pr)$. ☐

**Corollary 4.2.23.1.** *For every set of variables $Prop \neq \varnothing$, the concept classes $\mathbb{C}(\mathcal{L}_{\Diamond,\mathsf{3XOR}}[Prop])$, $\mathbb{C}(\mathcal{L}_{\Diamond,\mathsf{AIMP}}[Prop])$ and $\mathbb{C}(\mathcal{L}_{\Diamond,\mathsf{OXOR}}[Prop])$ does not admit finite characterizations.*

*Proof.* Directly follows from the results 3.3.7.2 and 3.3.26.2 and the previous theorem. ☐

### 4.2.3 The Main Results

We are now equipped with all the necessary tools to provide the main results of this section. The first group of results provide complete characterizations of all the modal concept classes of the form $\mathbb{C}(\mathcal{L}_{\Diamond,Pr}[Prop])$ and $\mathbb{C}(\mathcal{L}_{\Diamond,Pr}[Prop])$, based on finite characterization. We exploit the structure of the lattices $\mathscr{M}_\Box$ and $\mathscr{M}_\Diamond$, remember that they are both isomorphic to Post's Lattice.

**Theorem 4.2.24.** *For any set of propositional formulas $Pr$, the following are equivalent:*

*i. The concept class $\mathbb{C}(\mathcal{L}_{\Diamond,Pr}[Prop])$ has finite characterization, where $Prop$ is finite and non-empty.*

*ii. $Cl(Func(Pr))$ is a subset of (i) $Cl(\wedge,\vee,\top,\bot)$ or (ii) $Cl(\neg,\bot)$*

The above theorem is indeed a full classification of the sub-lattice $\mathscr{M}_\Diamond$. To see this, suppose $\mathcal{M}(S)$ denote any fragment from the sub-lattice $\mathscr{M}_\Diamond$. It is the case that $\mathcal{M}(S) = \mathcal{M}(\{\Diamond\} \cup \mathrm{Pr})$, where Pr is a set of propositional formulas and $Cl(\mathrm{Pr})$ should correspond to a clone in the Post Lattice (following theorem 4.1.16). Hence the above theorem is truly a complete classification of the sub-lattice $\mathscr{M}_\Diamond$.

This is

*Proof of theorem 4.2.24:* Our proof strategy is the same as the propositional case: We carefully pick out a splitting pair in the Post Lattice and analyse each of the fragments corresponding to the lattice. The splitting pair we choose is the following:

$$\{\{\mathrm{M}, \mathrm{U}\}, \{\mathrm{PT}_0^\infty, \mathrm{PT}_1^\infty, \mathrm{AP}\}\}$$

First suppose $Cl(Func(Pr))$ is a subset of $\mathrm{M} = Cl(\wedge, \vee, \top, \bot)$ or $\mathrm{U} = Cl(\neg, \bot)$, which means $\mathbb{C}(\mathcal{L}_{\Diamond,Pr}[Prop])$ is a sub concept class of either $\mathbb{C}(\mathcal{L}_{\Diamond,\wedge,\vee,\top,\bot}[Prop])$ or $\mathbb{C}(\mathcal{L}_{\Diamond,\neg,\bot}[Prop])$. From the theorems 4.2.16 and 4.2.5 and 2.5.1 it follows that $\mathbb{C}(\mathcal{L}_{\Diamond,Pr}[Prop])$ has finite characterization.

On the other hand if $Cl(Func(Pr))$ is a superset of $\mathrm{PT}_0^\infty = Cl(\mathsf{AIMP}), \mathrm{PT}_1^\infty = Cl(\mathsf{OXOR}), \mathrm{AP} = Cl(\mathsf{3XOR})$, then $\mathbb{C}(\mathcal{L}_{\Diamond,\mathsf{AIMP}}[Prop])$, $\mathbb{C}(\mathcal{L}_{\Diamond,\mathsf{AIMP}}[Prop])$ or $\mathbb{C}(\mathcal{L}_{\Diamond,\mathsf{AIMP}}[Prop])$ is a sub concept class of $\mathbb{C}(\mathcal{L}_{\Diamond,Pr}[Prop])$. Now from corollary 4.2.23.1, we can infer that in any of the cases $\mathbb{C}(\mathcal{L}_{\Diamond,Pr}[Prop])$ does not have finite characterization. This closes the proof. $\square$

We state a similar result for the fragment $\mathscr{M}_\Box$. The proof is essentially the same, we just need to change every $\Diamond$ to $\Box$ in the above proof.

**Theorem 4.2.25.** *For any set of propositional formulas $Pr$, the following are equivalent:*

*i. The concept class $\mathbb{C}(\mathcal{L}_{\Box,Pr}[Prop])$ has finite characterization, where Prop is finite and non-empty.*

*ii. $Cl(Func(Pr))$ is a subset of (i) $Cl(\wedge, \vee, \top, \bot)$ or (ii) $Cl(\neg, \bot)$* $\square$

In the spirit of theorems 4.2.24 and 4.2.25, we can provide a classification of the sub-lattice $\mathscr{M}_{\Diamond,\Box}$. As we described earlier, the sub-lattice $\mathscr{M}_{\Diamond,\Box}$ is again isomorphic to the Post's lattice and hence our proof strategy is effectively the same. Our proof again depends on Cate and Koudijs' work in [tCK24], where they prove that the fragment $\mathbb{C}(\mathcal{L}_{\Diamond,\Box,\wedge,\vee}[Prop])$.

**Theorem 4.2.26.** *For any set of propositional formulas $Pr$, the following are equivalent:*

*i. The concept class $\mathbb{C}(\mathcal{L}_{\Diamond,\Box,Pr}[Prop])$ has finite characterization, where Prop is finite and non-empty.*

*ii. $Cl(Func(Pr))$ is a subset of (i) $Cl(\wedge, \vee)$ or (ii) $Cl(\neg, \bot)$* $\square$

*Proof.* We consider the splitting pair

$$\{\{\mathrm{MP}, \mathrm{U}\}, \{\mathrm{P}_0, \mathsf{V}\mathrm{P}_1, \mathrm{AP}\}\}$$

First suppose $Cl(Func(Pr))$ is a subset of $\mathrm{MP} = Cl(\wedge, \vee)$ or $\mathrm{U} = Cl(\neg, \bot)$, which means $\mathbb{C}(\mathcal{L}_{\Diamond,Pr}[Prop])$ is a sub concept class of either $\mathbb{C}(\mathcal{L}_{\Diamond,\Box,\wedge,\vee,}[Prop])$ or $\mathbb{C}(\mathcal{L}_{\Diamond,\Box,\neg,\bot}[Prop])$. From Cate and Koudijs's work in [tCK24], theorem 4.2.16 and 2.5.1 it follows that $\mathbb{C}(\mathcal{L}_{\Diamond,\Box,Pr}[Prop])$ has finite characterization.

On the other hand if $Cl(Func(Pr))$ is a superset of $P_0 = Cl(\wedge, \bot), \vee P_1 = Cl(\vee, \top), \mathrm{AP} = Cl(\mathsf{3XOR})$, then $\mathbb{C}(\mathcal{L}_{\Diamond,\wedge,\bot}[Prop])$, $\mathbb{C}(\mathcal{L}_{\Diamond,\vee,\top}[Prop])$ or $\mathbb{C}(\mathcal{L}_{\Diamond,\mathsf{3XOR}}[Prop])$ is a sub concept class of $\mathbb{C}(\mathcal{L}_{\Diamond,Pr}[Prop])$. Now from corollary 4.2.23.1 and results 2.5.1 and 2.5.2.2, we can infer that in any of the cases $\mathbb{C}(\mathcal{L}_{\Diamond,Pr}[Prop])$ does not have finite characterization. This closes the proof. ☐

The next group of results is focused on dedicability. We would like to remind the reader that for a set of propositional formulas Pr, $\mathcal{F}(\mathrm{Pr})$ denotes the propositional formulas described by the elements of Pr. Notice that given two sets of formulas of the form $\{\Diamond p\} \cup \mathrm{Pr}_1$ and $\{\Diamond p\} \cup \mathrm{Pr}_2$ where $\mathrm{Pr}_1$ and $\mathrm{Pr}_2$ are propositional, the following holds

$$\mathcal{M}(\{\Diamond p\} \cup \mathrm{Pr}_1) \subseteq \mathcal{M}(\{\Diamond p\} \cup \mathrm{Pr}_2) \iff Cl(\mathcal{F}(\{\Diamond p\} \cup \mathrm{Pr}_1)) \subseteq Cl(\mathcal{F}(\{\Diamond p\} \cup \mathrm{Pr}_2))$$

Since the right hand side of the above equivalence is decidable, it is also decidable whether or not $\{\Diamond p\} \cup \mathrm{Pr}_1 \subseteq \{\Diamond p\} \cup \mathrm{Pr}_2$. We build upon this result to have the following theorem

**Theorem 4.2.27.** *Given a set of formulas of the form $\{\Diamond p\} \cup Pr$, where Pr is a set of propositional formulas, it is decidable whether or not $\mathcal{M}(\{\Diamond p\} \cup Pr)$ has finite characterization.*

*Proof.* We describe an algorithm that takes in an input $\{\Diamond p\} \cup \mathrm{Pr}$ and outputs yes or no depending on whether or not $\mathcal{M}(\{\Diamond p\} \cup \mathrm{Pr})$ has unique characterization. The algorithm simply asks whether $Cl(\mathcal{F}(\mathrm{Pr})) \subseteq Cl(\wedge, \vee, \top, \bot)$ and $Cl(\mathcal{F}(\mathrm{Pr})) \subseteq Cl(\neg, \bot)$. In the case both of the two queries is no, the algorithm returns no, otherwise it returns yes. ☐

We can provide a similar analysis of the formulas of the form $\{\Box p\} \cup \mathrm{Pr}$ and of the form $\{\Box p, \Diamond p\} \cup \mathrm{Pr}$. This would in turn give us the following decidability results:

**Theorem 4.2.28.** *Given a set of formulas of the form $\{\Box p\} \cup Pr$, where Pr is a set of propositional formulas, it is decidable whether or not $\mathcal{M}(\{\Box p\} \cup Pr)$ has finite characterization.* ☐

**Theorem 4.2.29.** *Given a set of formulas of the form $\{\Box p, \Diamond p\} \cup Pr$, where Pr is propositional, it is decidable whether or not $\mathcal{M}(\{\Box p, \Diamond p\} \cup Pr)$ has finite characterization.* ☐

Following the results on unique characterizations, we want to focus on the learnability aspect. Since we are concerned with finite characterizations we focus on effectively learnability.

Following the results 2.4.18 and 2.4.17, it is evident that eventual learnability and finite characterizations coincide. It is easy to see that for every finite *Prop*, the concept class $\mathbb{C}(\mathcal{L}_{\Diamond,\Box,Pr}[Prop])$ has countably many concepts. We would like to point out to the reader that in the paper [tCD22], it is proven that the concept class $\mathbb{C}(\mathcal{L}_{\exists,\wedge,\vee,\top,\bot}[S])$ has *effective* finite characterization (for any signature $S$), via $\mathsf{Alg}_1(x)$ say. Additionally, the translation from a $S$-structure $(A, a)$ to a pointed Kripke model $(M^A, a)$ is also effective, i.e. there is an algorithm that can return $(M^A, a)$ upon the input $(A, a)$, let's assume this is done by $\mathsf{Alg}_2(x)$. Therefore, for each concept $\varphi$ of $\mathbb{C}(\mathcal{L}_{\Diamond,\wedge,\vee,\bot,\top}[Prop])$ we first change it to a formula $\varphi'$ of $\mathbb{C}(\mathcal{L}_{\exists,\wedge,\vee,\bot,\top}[S_{Prop}])$. Now we get the unique characterization $T_{\varphi'}$ of $\varphi'$ using $\mathsf{Alg}_1(\varphi')$ and then emulate $\mathsf{Alg}_2(A, a)$ for each $(A, a) \in T_{\varphi'}$. Therefore, we have an effective unique characterization for $\mathbb{C}(\mathcal{L}_{\Diamond,\wedge,\vee,\bot,\top}[Prop])$.

Moreover, the proof of obtaining an unique characterization for $\mathbb{C}(\mathcal{L}_{\Diamond,\Box,\neg,\bot,\top}[Prop])$ in theorem 4.2.16 can be done by an algorithm. Hence we have that $\mathbb{C}(\mathcal{L}_{\Diamond,\neg,\bot}[Prop])$ has effective finite characterization. On the other hand if we consider any other concept class of the form $\mathbb{C}(\mathcal{L}_{\Diamond,Pr}[Prop])$, where $Pr$ is a set of propositional formulas and $Cl(Pr)$ is not contained in $Cl(\{\Diamond,\wedge,\vee,\top,\bot\})$ nor $Cl(\{\Diamond,\neg,\top,\bot\})$ then $\mathbb{C}(\mathcal{L}_{\Diamond,Pr}[Prop])$ does not have finite characterization (following theorem 4.2.24). We use the theorems 2.4.17 and 2.4.18 to reach the following result.

**Theorem 4.2.30.** *Let $Pr$ be a set of propositional formulas, then the following equivalence holds:*

$$\mathbb{C}(\mathcal{L}_{\Diamond,Pr}[Prop]) \text{ is effectively learnable} \iff \mathbb{C}(\mathcal{L}_{\Diamond,Pr}[Prop]) \text{ has finite characterization}$$

*Proof.* Following the discussion above and theorems 2.4.17 and 2.4.18, we immediately get our result. ☐

**Theorem 4.2.31.** *For a set of propositional formulas $Pr$, the following are equivalent:*

*i. The concept class $\mathbb{C}(\mathcal{L}_{\Diamond,pr}[Prop])$ is effectively learnable, where Prop is finite and non-empty.*

*ii. $Cl(Func(Pr))$ is a subset of (i) $Cl(\wedge,\vee,\top,\bot)$ or (ii) $Cl(\neg,\bot)$* ☐

Let $\varphi$ be a concept of the concept class of $\mathbb{C}(\mathcal{L}_{\Box,Pr}[Prop])$, where $Pr$ is a set of propositional formulas. Remember that we can construct an unique characterization of $\varphi$ from the unique characterization of its negation translation $\varphi^-$. Now remember that $\varphi^-$ is a concept of the concept class $\mathbb{C}(\mathcal{L}_{\Diamond,Pr}[PExt])$, which has effective characterization and we can effectively translate each element of the unique characterization of $\varphi^-$ to form an unique characterization of $\varphi$. Hence, $\mathbb{C}(\mathcal{L}_{\Box,Pr}[Prop])$ has effective characterization, so it is learnable. This allows us to extend the previous result.

**Theorem 4.2.32.** *For a set of propositional formulas $Pr$, the following are equivalent:*

*i. The concept class $\mathbb{C}(\mathcal{L}_{\Box,Pr}[Prop])$ is effectively learnable, where Prop is finite and non-empty.*

*ii. $Cl(Func(Pr))$ is a subset of (i) $Cl(\wedge,\vee,\top,\bot)$ or (ii) $Cl(\neg,\bot)$* ☐

*Proof.* The proof is basically the same as in theorem 4.2.31. We again just change $\Diamond$ to $\Box$ and we are done. ☐

In addition to showing that $\mathbb{C}(\mathcal{L}_{\Diamond,\Box,\wedge,\vee}[Prop])$ has finite characterization, Cate and Koudijs in their work [tCK24] showed that the concept class $\mathbb{C}(\mathcal{L}_{\Diamond,\Box,\wedge,\vee}[Prop])$ has effective finite characterization. We build upon this result to provide the last result of this chapter.

**Theorem 4.2.33.** *For a set of propositional formulas $Pr$, the following are equivalent:*

*i. The concept class $\mathbb{C}(\mathcal{L}_{\Diamond,\Box,Pr}[Prop])$ is effectively learnable, where Prop is finite and non-empty.*

*ii. $Cl(Func(Pr))$ is a subset of (i) $Cl(\wedge,\vee)$ or (ii) $Cl(\neg,\bot)$* ☐

# Chapter 5

# Conclusion and Future Work

To summarize, we provided six classifications for the propositional case. Three of those classifications were about unique characterizations based on various upper-bound we considered; the other other were the exact learnability notions stemming from those characterizations.

Similarly, in the modal case we provide six more classifications. Unlike the propositional case, in these characterizations we consider three sub-lattices of the lattice all the possible modal fragments. We classify the three sublattices based on finite characterizations, giving us our first three results. The latter three results are characterizing these lattices based on effective learnability.

While providing the characterizations we have noticed that the results pertaining to the propositional cases are in-fact stronger. Indeed the propositional classifications take into account all the fragments, whereas the modal classifications are restricted to a particular sub-lattice of $\mathscr{M}$. This gives way to the first problem we encounter

1. Provide a complete classification of $\mathscr{M}$, i.e. the lattice of all modal fragments.

This is a formidable task since the modal fragments are not investigated upon, and the lattice of infinite clones has a complex structure.

The next question comes from the modal expressivity problem. Given a formula $\varphi$ and a set of modal formulas $Md$, we say $\varphi$ is expressible in $\mathcal{M}(Md)$ if $\varphi \in \mathcal{M}$. The modal expressivity problem asks: Is there an algorithm that takes as input $\varphi$ and the set of modal formulas $Md$ and return whether $\varphi$ is expressible in $\mathcal{M}(Md)$ or not? M.F. Ratsa proved that in the case of the logic S4, the answer is no (see [Rat89] in Russian). B. Ten Cate's work [tC09] proves that it is undecidable for K4 as well. So our next question is:

2. What is the answer to modal expressivity problem in the case of modal Logic $K$?

In the case of propositional logic, the answer was given by the fact that given a set of propositional variables, there can only be finitely many non-equivalent formulas of it. The case where this property doesn't hold is yet to be investigated.

Out last set of questions is aimed at connecting our study to yet another branch of algebra, namely varieties. The questions in this part is more speculative/non-rigorous due to the lack of literature connecting these topics. The connection between hypervarieties and clones have been somewhat investigated in the works [Tay73] and [Wis95].

3. What is the connection between varieties and the theory of clones?

4. If such a connection is established, can it be used to understand more about the fragments of modal logic?

The existence of these connections would help us to use the pre-exisiting literature on varieties to understand the structure of modal clones further.

# Bibliography


[ABCS92] Martin Anthony, Graham R. Brightwell, David A. Cohen, and John Shawe-Taylor. On exact specification by examples. In David Haussler, editor, *Proceedings of the Fifth Annual ACM Conference on Computational Learning Theory, COLT 1992, Pittsburgh, PA, USA, July 27-29, 1992*, pages 311–318. ACM, 1992.

[AK95] Dana Angluin and Michael Kharitonov. When won't membership queries help? *J. Comput. Syst. Sci.*, 50(2):336–355, 1995.

[Ang87] Dana Angluin. Queries and concept learning. *Mach. Learn.*, 2(4):319–342, 1987.

[BdRV01] Patrick Blackburn, Maarten de Rijke, and Yde Venema. *Modal Logic*, volume 53 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2001.

[BI95] Jan C. Bioch and Toshihide Ibaraki. Complexity of identification and dualization of positive boolean functions. *Inf. Comput.*, 123(1):50–63, 1995.

[BMS$^+$11] Michael Bauland, Martin Mundhenk, Thomas Schneider, Henning Schnoor, Ilka Schnoor, and Heribert Vollmer. The tractability of model checking for ltl: The good, the bad, and the ugly fragments. *ACM Trans. Comput. Logic*, 12(2), jan 2011.

[BS00] Clifford Bergman and Giora Slutzki. Complexity of some problems concerning varieties and quasi-varieties of algebras. *SIAM Journal on Computing*, 30(2):359–382, 2000.

[BSS$^+$07] Michael Bauland, Thomas Schneider, Henning Schnoor, Ilka Schnoor, and Heribert Vollmer. The complexity of generalized satisfiability for linear temporal logic. In Helmut Seidl, editor, *Foundations of Software Science and Computational Structures*, pages 48–62, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[BUL93] A. A. BULATOV. Identities in the lattices of closed classes. *Discrete Mathematics and Applications*, 3(6):601–610, 1993.

[Bul94] Andrei Arnol'dovich Bulatov. Finite sublattices in the lattice of clones. *Algebra and Logic*, 33(5):287–306, 1994.

[CK92] C. C. Chang and H. Jerome Keisler. Model theory. *Studia Logica*, 51(1):154–155, 1992.

[Dal99] Víctor Dalmau. Boolean formulas are hard to learn for most gate bases. In Osamu Watanabe and Takashi Yokomori, editors, *Algorithmic Learning Theory, 10th International Conference, ALT '99, Tokyo, Japan, December 6-8, 1999, Proceedings*, volume 1720 of *Lecture Notes in Computer Science*, pages 301–312. Springer, 1999.

[GK95] Sally A. Goldman and Michael J. Kearns. On the complexity of teaching. *J. Comput. Syst. Sci.*, 50(1):20–31, 1995.

[GP07] Martin Goldstern and Michael Pinsker. A survey of clones on infinite sets. *Algebra universalis*, 59, 01 2007.

[(ht] Arun (https://math.stackexchange.com/users/1252876/arun). What is the least number of subsets that you need to separate all the 2-element subsets of a given (finite) set. Mathematics Stack Exchange.

[Lau06] Dietlinde Lau. *Function algebras on finite sets: Basic course on many-valued logic and clone theory.* Springer Science & Business Media, 2006.

[Men09] Elliott Mendelson. *Introduction to mathematical logic, proposition 1.5.* CRC press, 2009.

[Pip10] Nicholas Pippenger. *Theories of Computability (2010).* Cambridge University Press, 2010.

[Pos41] Emil L. Post. *The Two-Valued Iterative Systems of Mathematical Logic. (AM-5).* Princeton University Press, 1941.

[Rat89] M.F. Ratsa. Undecidability of the problem of functional expressibility in the modal logic s4. *Soviet Mathematics. Doklady*, 27, 1989.

[RS94] Kathleen Romanik and Carl Smith. Testing geometric objects. *Comput. Geom.*, 4:157–176, 1994.

[SDHK91] Steven Salzberg, Arthur L. Delcher, David G. Heath, and Simon Kasif. Learning with a helpful teacher. In John Mylopoulos and Raymond Reiter, editors, *Proceedings of the 12th International Joint Conference on Artificial Intelligence. Sydney, Australia, August 24-30, 1991*, pages 705–711. Morgan Kaufmann, 1991.

[Tay73] Walter Taylor. Characterizing mal'cev conditions. *algebra universalis*, 1973.

[tC09] Balder ten Cate. A note on the expressibility problem for modal logics and star-free regular expressions. *Inf. Process. Lett.*, 109(10):509–513, 2009.

[tCD22] Balder ten Cate and Victor Dalmau. Conjunctive queries: Unique characterizations and exact learnability. *ACM Trans. Database Syst.*, 47(4):14:1–14:41, 2022.

[tCK24] Balder ten Cate and Raoul Koudijs. Characterising modal formulas with examples. *ACM Trans. Comput. Log.*, 25(2):12:1–12:27, 2024.

[Vol09] Heribert Vollmer. The complexity of deciding if a boolean function can be computed by circuits over a restricted basis. *Theory of Computing Systems*, 44:82–90, 01 2009.

[Wis95] S.L. Wismath. Clone and hypervariety correspondence. *Algebra Universalis*, 1995.