Topological Models, Computer Science Semantics and Applications of Logic in AI

Yiannis Kiouvrekis

National Technical University of Athens

School of Applied Mathematical and Physics Science

Department of Mathematics

Key words and phrases. Mathematical Logic; AI; Topological Models; Institution Theory; Abstract Model Theory; Categorical Abstract Model Theory; Argumentation Logic; Gorgias;

Three-person doctoral committee

(1) Petros Stefaneas, Assistant Professor, Department of Mathematics, National Technical University of Athens (NTUA) (supervisor)

(2) Sergey Sudoplatov Prof. Technical Un. of Novosibirsk, Russia

(3) Alekos Arvanitakis, Assistant Professor, Department of Mathematics, National Technical University of Athens (NTUA)

Στην μάνα μου Αθανασία, στον πατέρα μου Σταύρο, στην κόρη μου Δήμητρα–Ελπινίκη, στον γιό μου Σταύρο, στην σύζυγο μου Γιώτα, στην γιαγιά μου Ελπινίκη, στο Χρήστο και την Μιμίκα για την υπομονή, επιμονή και στήριξη που έδειξαν όλα αυτά τα χρόνια.

Contents

Preface

This doctoral thesis is a work of research into a broad spectrum of diverse fields with symbolic logic serving as the focal point. We begin this introduction in reverse order so that the interdisciplinary nature of the thesis becomes clearer. The third and final part of the thesis deals with Explainable Artificial Intelligence (XAI); specifically, it deals with symbolic artificial intelligence. We created an early form of an information system which could produce explanations in the framework of statistical hypothesis testing. While our model could be applied primarily on classic artificial intelligence algorithms we chose to conduct our proof of concept on the field of hypothesis testing, as hypothesis tests are the most commonly applied statistical methods in medical research. Therefore, it is very useful in order to both minimize errors in interpreting statistical results and improve the ways of interpreting those results. The first part of this thesis deals with theory behind practice. We examine how we could expand the expressing capabilities of already existing logical systems by syntactically expanding each logical system and, furthermore, creating new semantics by utilizing semantic topologies. Finally, the second part of the thesis deals with the concept of generic constructions, a tool of mathematical logic which provides both syntactic and semantic constructions.

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my advisor Prof. Petros Stefaneas for his continuous support of my research, for his motivation, enthusiasm, and immense knowledge. His guidance helped me throughout my research and the writing of this thesis. I could not have wished for a better advisor, mentor and colleague for my Ph.D study.
In addition, I would like to thank my other research collaborators:

- Prof. Sergey Sudoplatov, for offering me the opportunity of being part of his group and for guiding me on diverse exciting projects on model theory.
- Prof. Angelika Kokkinaki for her guidance in the field of information systems and for the confidence she showed in my abilities.
- Nikos Asimakis, for the interesting discussions we have had in the last three years.

Last but not least, I would like to thank my family: my parents Stavros and Athanasia, for giving birth to me at the first place and for their support throughout my life.
My wife Yiota, who has been extremely supportive of me throughout this entire process.
My children, Dimitra-Elpiniki and Stavros for the wonderful breaks of joy and laughter.

Introduction

In Chapter 1 we present the fundamental concepts used in this work. We present the basic concepts of category theory, topology, ultraproducts, as well as an extensive introduction to institutional theory.

In Chapter 2 titled Aristotelian Institution-Independent Square we introduce the square of opposition, the concept of the Rhombus of Opposition and we examine basic cases of morphology change of the squares of opposition inside and between logical systems. In the third section of the second chapter, we use the concept of the Galois Connection to show the equilibrium that one can create between the standard square of sentences and the internal semantics of Boolean connectives, using them at a meta-level. Finally, we introduce the concept of a dual square that can give us not only squares for propositions but also squares for sets of sentences. Since quite a few logical systems do not have internal connectives, it is not useful to talk about proposition graphs, but about sets of models and of sets of sentences.

In Chapter 3 titled Topology, Topological Semantic and topological Ultraproducts via/for Institution-independent model theory, we introduce the concept of topological semantics at the level of abstract model theory provided by the institution-independent framework. Our abstract topological logic framework provides a method for systematic topological semantics extensions of logical systems from computer science and logic. Furthermore, it equips us with several appropriate theoretical model tools for proving semantic completeness on arbitrary Institutions via topological approach. We also extend the institution-independent method of ultraproducts to topological semantics and prove a fundamental preservation result for abstract topo-modal satisfaction. Furthermore, we prove a fundamental preservation result for abstract topo-modal satisfaction.

In Chapter 4 we study and compare basic characteristics for definable sets in generic structures. We introduce calculi for (type)-definable sets allowing for comparing their cardinalities. Using these calculi we characterize the possibility to construct a generic structure of a given generative class.

In Chapter 5 we present an early version of a decision-making ”eco”system. We refer to it as an “eco”system because it is primarily based on mathematical logic and combines concepts and principles from the fields of statistics, decision theory, artificial intelligence and modelling of human behavior. The primary goal of the proposed approach is to address errors that occur resulting from the misuse of statistical methods. In practice, such errors often occur either owing to the use of inappropriate statistical methods or wrong interpretations of results. The proposed approach relies on the LPwNF (Logic Programming without Negation as Failure) framework of non-monotonic reasoning as provided by Gorgias. The proposed system enables automatic selection of the appropriate statistical method, based on the characteristics of the problem and the sample. The expected impact could be twofold: it can enhance the use of statistical systems like R and, combined with a Java-based interface to Gorgias, make non-monotonic reasoning easy to use in the proposed context.

Chapter 1

Prerequisite

1.1. Category Theory

We begin this chapter with an example from literature. Readers claim that books should be read in their original language in order to grasp subtle nuances or word-play. Mathematics faces a similar concern; the language in which a theory is written is of major significance concerning the expressing capabilities of the theory itself. The choice of language becomes even more problematic when dealing with Meta-mathematics.

Category theory is a relatively new area of meta-mathematics that comes in addition to the standard set theoretic approach to provide greater mathematical expression.Category theory is a relatively recent development in meta-mathematics [**52**]; it supplements classic set theory and offers additional tools for expressing mathematical concepts. It is a language and, therefore, it is highly abstract; however, it can describe mathematical entities in a succinct and economical manner. To make this clear to the reader, we will present an example: F. William Lawvere wrote a paper [**43**] in which he showed how to describe many of the classical paradoxes and incompleteness theorems in an universal way.

Theorem 1.1.1 (Lawvere's fixed-point theorem [**43**])**.** *In a cartesian closed category, if there is a point-surjective map $\phi : A \to B^A$ then every morphism $f : B \to B$ has a fixed point $s : 1 \to B \quad f(s) = s$.*

This theorem implied that by satisfying certain conditions in a cartesian closed category paradoxical phenomena can occur.

Theorem 1.1.2 (Diagonal Theorem [76]). *If Y is a set and there exists a set T and a function $f : T \times T \to Y$ such that all functions $g : T \to Y$ are representable by f (there exists a $t \in T$ such that $g(*) = f(*, t)$) then all functions $\psi : Y \to Y$ have a fixed point.*

We know, however, that for any set Y with two or more elements, there exist functions$Y \to Y$ with no fixed points. If we look at this theorem, then we will find that we can very easily come to obvious contradictions and paradoxes. It is here that we get into trouble, ignoring the category theory which is necessary as our language. Some instances of Diagonal Theorems [76] are the following:

- Godel's First Incompleteness Theorem;
- Godel-Rosser's Incompleteness Theorem;
- Tarski's Theorem
- Parikh Sentences
- The Recursion Theorem
- Rice's Theorem
- Von Neumann's Self-reproducing Machines

The following is the definition of a category:

Definition 1.1.3 (Category). A category $\mathbb{C}$ consists of the following :

- Objects : $A, B, C, \ldots$
- Arrows or Morphisms : $f, g, h, \ldots$

that satisfy the following axioms :

(1) for every arrow f, there exist objects

$$A = \mathrm{dom(f)}, \quad \mathrm{B} = \mathrm{codom(f)}$$

called the domain and the codomain of f, and we write

$$f : A \to B$$

(2) Let $f : A \to B$ and $g : B \to C$ such that codom(f) = dom(g), then there exists an arrow/morphism h

$$h = g \circ f : A \to C$$

called the composite of f and g.

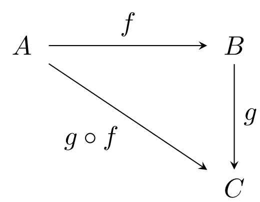


Figure 1. The scheme of arrow composition.

(3) For every object A, there exists an arrow

$$1_A : A \to A$$

called the identity arrow of A

(4) For all $f : A \to B,\ g : B \to C,\ h : C \to D$

$$h \circ (g \circ f) = (h \circ g) \circ f$$

(5) and for all $f : A \to B$

$$f \circ 1_A = f = 1_B \circ f$$

A category is everything that can satisfy the definition above. Let us illustrate this concept by two elementary and fundamental examples.

Example 1.1.4 (Sets). The first example concerns the category of sets. The objects of this category are the sets and the morphisms are the functions between the sets – where these are defined. It is obvious that the identity arrow or identity morphism is the identity function of the set onto itself. The triangle in Figure 2 represents the usual synthesis of functions.

Example 1.1.5 (Poset). A slightly more complex category is that of partial order sets – **Poset**. A space is called partially ordered when we can define a partial order in this space. What special feature does this category have? It is one of the most typical examples of categories and it will help us understand why the language of

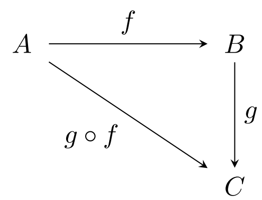


Figure 2

the category theory is what we will be using in the rest of the chapter. Morphisms, namely the arrows, are functions that preserve the structure.

Another concept in category theory that plays a fundamental role is that of the functor. The concept of functor is a generalization of the concept of function between classes of structures; it is morphism between categories.

Definition 1.1.6 (Functor)**.** Let $\mathbb{C}, \mathbb{D}$ be two categories, then a functor F with domain $\mathbb{C}$ and codomain $\mathbb{D}$ consists of two suitably related functions:

- the object function F, which assigns an object $F(X)$ of $\mathbb{D}$ to each object X of $\mathbb{C}$;
- the arrow function F which assigns the $F(f) : F(X) \to F(Y)$ of $\mathbb{D}$ to each morphism $f : X \to Y$ of $\mathbb{C}$ such that $F(1_X) = 1_{F(X)}$ and:

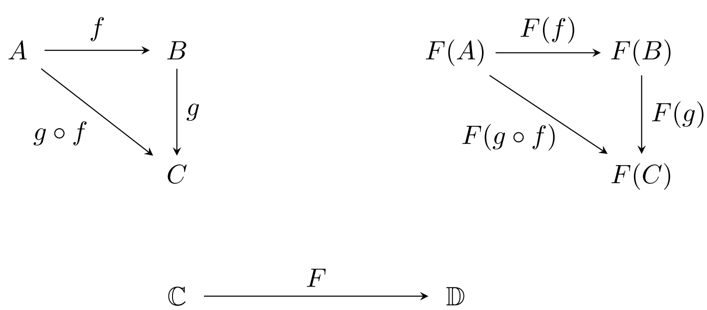


Figure 3. The fundamental schemes of Functors.

Example 1.1.7 (Power Set Functor)**.** One of the most simple examples is the power set functor $\mathcal{P} : \mathbf{Set} \to \mathbf{Set}$. The object function assigns its powerset $\mathcal{P}(X)$ for each set X and the morphism function assigns the $\mathcal{P}(f) : \mathcal{P}(X) \to \mathcal{P}(Y)$ to each $f : X \to Y$. This sends every subset $Z \subseteq X$ to its image $F(Z) \subseteq Y$.

The notion of functor came up in algebraic topology where geometric properties are described by means of algebraic invariants.

Example 1.1.8. The singular homology in a given dimension n assigns an abelian group $H_n(X)$ to each topological space X, the n-th homology group of the space X. Additionally, it assigns a corresponding homomorphism $H_n(f) : H_n(X) \mapsto H_n(Y)$ of groups to each continuous map $f : X \mapsto Y$ of spaces, and this in such a way that H_n acts as a functor $H_n : \boldsymbol{Top} \mapsto \boldsymbol{Ab}$.

Example 1.1.9. Functors also arise naturally in algebra. Let K be a commutative ring, then the set of all non-singular $n \times n$ matrices with entries in K is the usual general linear group $GL_n(K)$. For each homomorphism $f : K \mapsto K'$ of rings it produces a homomorphism $GL_n(f) : GL_n(K) \mapsto GL_n(K')$ of groups. This defines a functor $GL_n : \boldsymbol{CRng} \mapsto \boldsymbol{Grp}$ for each $n \in \mathbb{N}^*$.

Category theory is an algebra of functions and as such it has incorporated many concepts whose presentation goes beyond the aims of this chapter. The following two definitions, of the initial and the final object, are two easy examples, but they illustrate the different reading that category theory offers.

Definition 1.1.10 (Initial object)**.** An object $\mathcal{I}$ is initial in a category $\mathbb{C}$ if for every object X in $\mathbb{C}$ there exists a unique arrow in $\mathbb{C}$ from it to X. Initial objects are unique up to isomorphism.

Definition 1.1.11 (Terminal object)**.** An object is terminal (or final) $\mathcal{T}$ in a category $\mathbb{C}$ if for every object X in $\mathbb{C}$ there exists a unique arrow in $\mathbb{C}$ from X to it. Terminal objects are unique up to isomorphism.

Example 1.1.12. For example, the initial object in a preorder is the least element, if it exists; the terminal object is the top. In $\boldsymbol{Set}$ the initial object is the empty set $\emptyset$, while the terminal object is the singleton set.

The notions of pullback, pushout, product or co-product come up very often in mathematics and logic.

Definition 1.1.13 (Pullback)**.** In any category $\mathbb{C}$, given arrows f, g with cod(f) = cod(g) the pullback of f and g consists of arrows such that $f \circ p_1 = g \circ p_2$ and is universal with this property. That is, given any $z_1 : Z \mapsto A$ and $z_2 : Z \mapsto B$ with

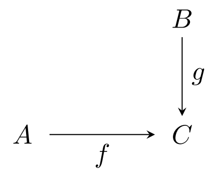


Figure 4

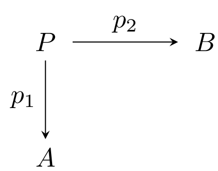


Figure 5

$f \circ z_1 = g \circ z_2$, there exists a unique $z : Z \mapsto P$ with $z_1 = p_1 \circ u$ and $z_2 = p_2 \circ u$. The following diagram describes the situation:

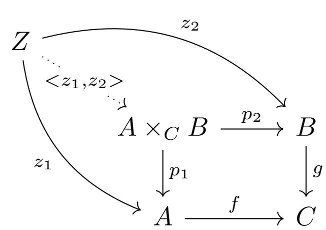


Figure 6

Definition 1.1.14 (Pushout)**.** In any category $\mathbb{C}$, given arrows f, g with dom(f) = dom(g) the pullback of f and g consists of arrows such that $p_1 \circ f = p_2 \circ g$ and is

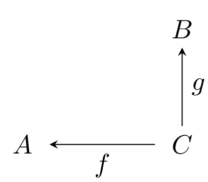


Figure 7

universal with this property. That is, given any $z_1 : A \mapsto Z$ and $z_2 : B \mapsto Z$ with $z_1 \circ f = z_2 \circ g$, there exists a unique $z : P \mapsto Z$ with $z_1 = u \circ p_1$ and $z_2 = u \circ p_2$. The following diagram describes the situation:

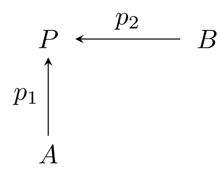


Figure 8

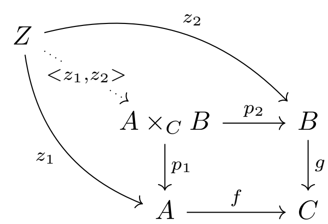


Figure 9

Definition 1.1.15 (Product)**.** Let $\mathbb{C}$ be category and X, Y be objects of $\mathbb{C}$. Then a product for the objects X and Y is an object $\mathcal{O}$ in $\mathbb{C}$ with "projection" arrows $\pi_1 : \mathcal{O} \to X, \pi_2 : \mathcal{O} \to Y$ and arrows $f_1 : S \to X, f_2 : S \to Y$ such that for any object S there is a unique arrow $u : S \to \mathcal{O}$ such that the following diagram commutes:

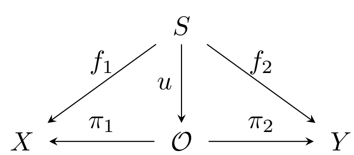


Figure 10. The fundamental scheme for the product

Example 1.1.16. In ***Set*** the usual Cartesian product treated as the set $X \times Y$ of pairs, i.e. $X \times Y = \{< x, y >= \{x, \{x, y\}\} \mid x \in X \ \& \ y \in Y\}$, together with the obvious projections $\pi_1 (< x, y >) = x \ \& \ \pi_2 (< x, y >) = y$, form a product.

Definition 1.1.17 (Co-product)**.** Let $\mathbb{C}$ be category and X, Y be objects of $\mathbb{C}$. Then a coproduct for the objects X and Y is an object $\mathcal{O}$ in $\mathbb{C}$ with "injection" arrows $\iota_1 : X \to \mathcal{O}, \iota_2 : Y \to \mathcal{O}$ such that for any object S and arrows $f_1 : X \to S, f_2 : Y \to S$ there is a unique arrow $v : \mathcal{O} \to S$ such that the following diagram commutes:

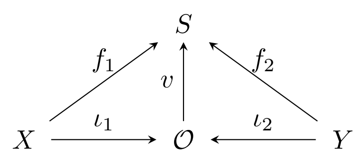


Figure 11. The fundamental scheme for the coproduct

Example 1.1.18. In ***Set*** let X, Y be sets, let $X \oplus Y$ be the disjoint union of the sets, i.e the set with members $< x, 0 >$ for $x \in X$ and $< y, 1 >$ for $y \in Y$. Let the injection arrow $\iota_1 : X \mapsto X \oplus Y$ be the function $\iota(x) =< x, 0 >$ and respectively $\iota_2 : Y \mapsto X \oplus Y$ be the function $\iota_2(y) =< y, 1 >$. Then the $X \oplus Y$ is a coproduct of X and Y.

Definition 1.1.19 (Diagram)**.** Let $\mathbb{I}, \mathbb{C}$ be two categories, such that $\mathbb{I}$ is a small category. Then by a diagram in a category $\mathbb{C}$, we denote a functor

$$\mathcal{X} : \mathcal{I} \mapsto \mathbb{C} \tag{1.1}$$

and we say that $\mathbb{I}$ is the index category of $\mathcal{X}$ and that $\mathcal{X}$ is an $\mathbb{I}$-diagram.

Definition 1.1.20 (Colimit)**.** Let $\mathcal{X}$ be an I-diagram in the category $\mathbb{C}$. Then the colimit of the functor $\mathcal{X}$ is an object

$$\text{colim}_I \mathcal{X} \tag{1.2}$$

of $\mathbb{C}$ that satisfies the following properties:

(1) For every object $i \in I$, there is a morphism in $\mathbb{C}$

$$f_i : \mathcal{X}(i) \to \text{colim}_I \mathcal{X} \tag{1.3}$$

and for every morphism $\phi : i \to i'$ in I holds $f_i = f_{i'} \circ \mathcal{X}(\phi)$, i.e. the following commutes

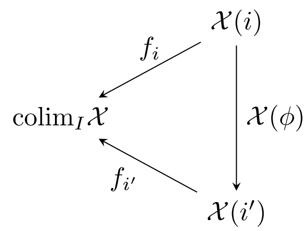


Figure 12

(2) Given an object Y in $\mathbb{C}$ and, for every object $i \in I$, a morphism in $\mathbb{C}$, $f_i : \mathcal{X}(i) \to Y$ such that, for every morphism $\phi : i \to i' \in I$ $g_i = g_{i'} \circ \mathcal{X}(\phi)$, there exists a unique morphism in $\mathbb{C}$

$$f : \text{colim}_I\mathcal{X} \to Y \tag{1.4}$$

such that for all objects $i \in I$ $g_i = f \circ g_i$

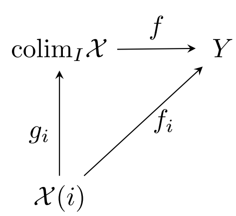


Figure 13

According to (i) and (ii) the colimit is well-defined up to canonical isomorphism.

Definition 1.1.21 (Natural Transformation)**.** Let $\mathbb{C}, \mathbb{B}$ be two categories and S, T be two functors $S : \mathbb{C} \mapsto \mathbb{D}$, $T : \mathbb{C} \mapsto \mathbb{D}$. A natural transformation $\tau : S \Rightarrow T$ is an action which assigns an arrow $\tau_c : S(c) \to T(c)$ of the category $\mathbb{B}$ to each object $c \in \mathbb{C}$ such that for every morphism $f : c \to c'$ in $\mathbb{C}$ the diagram in Figure 14 is commutative.

Therefore a natural transformation is a set of morphisms translating (mapping) the picture S to the picture T with all squares and parallelograms being commutative. Moreover, a natural transformation is often called morphism of functors [**42**].

A natural transformation is often called *morphisms of functors*.

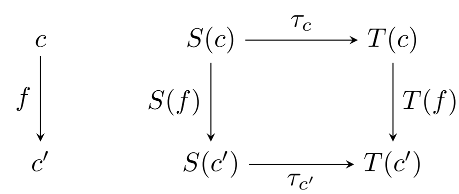


Figure 14. Fundamental Square of Natural Transformation

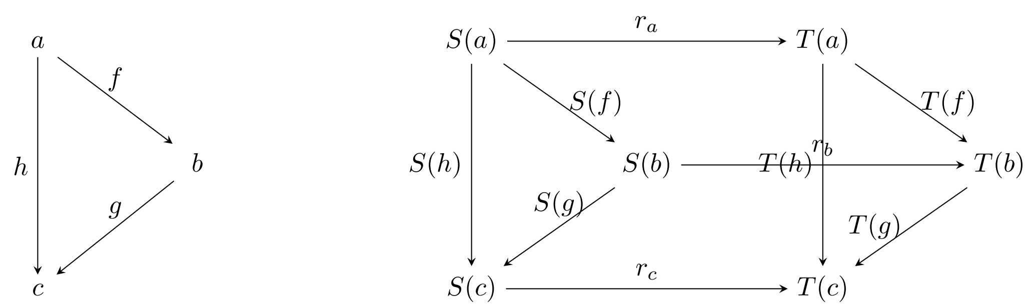


Figure 15

Example 1.1.22. The determinant is a natural transformation. Let $\det_K M$ be the determinant of the $n \times n$ matrix M with entries in the commutative ring K. If K^* denotes the group of units of K, then M is non-singular when $\det_K M$ is a unit, and $\det_K$ is a morphism $GL_n K \mapsto K^*$ of groups. Since the determinant is defined by the same formula for all rings K, each morphism $f : K \mapsto K'$ of commutative rings gives a commutative diagram, i.e.

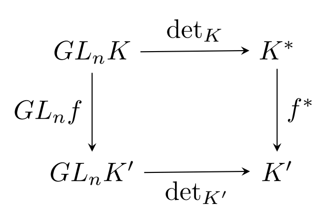


Figure 16

This means that the map $\det : GL_n \mapsto (\ )^*$ is natural between the functors $\boldsymbol{CRng} \to \boldsymbol{Grp}$.

Example 1.1.23. Let Id be the identity functor on $\boldsymbol{Set}$ and Δ the diagonal functor which assigns the (X, X) to each set X and the (f, f) to each function f. Then there is a natural transformation

$$\delta : \mathrm{Id} \mapsto \Delta \tag{1.5}$$

such that $\delta_X : X \mapsto X \times X$ with $x \mapsto (x, x)$.

This concludes the section presenting the basic concepts of category theory. We refer the reader to the bibliography at the end of the thesis for further study on category theory.

1.2. Preliminaries from Topology

Topology is a well defined mathematical discipline and a major branch of mathematics with enormous influence on many other branches of mathematics. The theory of topological spaces [46] can appear in an introductory analysis lesson, a calculus lesson, a mathematical logic lesson, a real or a functional analysis, etc. Our aim is to present the basic definitions and basic results of topological space theory in order to understand our work in topological semantics.

Definition 1.2.1 (Topology)**.** Let X be a set and τ be a family of subsets of X. Then τ is called a topology on X if :

(1) $\emptyset \in \tau$;

(2) $X \in \tau$;

(3) For any family $\{A_i\}_{i \in I}$ of subsets of X the arbitrary union $\bigcup_{i \in I} A_i$ belongs to τ;

(4) For any finite family $A_1, \ldots, A_n$ of subsets of X the intersection $\bigcap_{i=1}^{n} A_i$ belongs to τ;

Definition 1.2.2. Suppose that (X, τ) is a topological space and R is an equivalent relation on X. Let X/R denote the set of R-equivalence classes. If we define the function f from X to X/R by $f(x) = \overline{x}$, where $\overline{x}$ is the equivalent class of $x \in X$, and if we define a subset U of X/R to be open if $f^{-1}(U)$ is open to X, then we obtain a topology called quotient topology.

The ultraproduct construction has a long history. The beginning dates to the early 20th century with the work of T. Skolem who built non standard models of arithmetic. The breakthrough however came when J. Los published his fundamental theorem of ultraproducts; after that the fundamental work of H.J. Keisler and S. Shelah came to establish the theory of ultraproducts as a separate field in mathematical logic.

Proposition 1.2.3. *Let (X, τ) be a topological space, let R be an equivalence relation on X, and suppose that X/R is the quotient topology, then the map*

$$f : (X, \tau) \mapsto (X/R, \tau') \tag{1.6}$$

is continuous.

Definition 1.2.4 (Filter)**.** Let X be a set then a filter on $\mathcal{P}$ is a collection $\mathcal{F}$ of subsets of X satisfying :

(1) $X \in \mathcal{F}$;

(2) If $A \subseteq B$ and $A \in \mathcal{F}$ then $B \in \mathcal{F}$;

(3) If $A, B \in \mathcal{F}$ then $A \wedge B \in \mathcal{F}$;

A filter $\mathcal{F}$ is called proper if $\emptyset \notin \mathcal{F}$; A ultrafilter on X is a proper filter on X that is not contained in any other proper filter on X.

Example 1.2.5 (Power set lattices)**.** In power set lattices, the ultrafilters are precisely the prime ones. Filters $\mathcal{F}$ are prime on I such that, if for every $A, B \subseteq I$, if $A \cup B \in \mathcal{F}$, then either $A \in \mathcal{F}$ or $B \in \mathcal{F}$.

Definition 1.2.6 (Finite Intersection Property - FIP)**.** If $\mathcal{B}$ is a family of subsets of a set X, $\mathcal{B}$ is said to satisfy the Finite Intersection Property – FIP – if any finite intersection of elements of $\mathcal{B}$ is non empty.

Let L be an alphabet of finitary relation and function symbols, the L-structure $\mathcal{X}$ consists of an underlying set X and an interpretation of the relation and function symbols in the standard way.

Definition 1.2.7 (Direct Product)**.** Let $\langle X_i \ : \ i \in I \rangle$ be a family of L-structures, then $\mathcal{F}$ be a filter over I, then the direct product denotes as

$$\prod_{i \in I} X_i$$

and the ith coordinate if an element a denotes as $a(i)$.

Definition 1.2.8 (Reduct Product)**.** Let $\mathcal{F}$ be a filter over I and $a, b \in \prod_{i \in I} X_i$, then we define the equivalence relation

$$a \sim_{\mathcal{F}} b \Leftrightarrow \{i \in I \ : \ a(i) = b(i)\}$$

Also, for every $f \in \prod_{i \in I} X_i$ we denote $f/\mathcal{F} := \{b \ : \ f \sim_{\mathcal{F}} b\}$ as the equivalence class and $\prod_{\mathcal{F}} X_i = \prod_{i \in I} X_i/\mathcal{F} = \left\{ f/\mathcal{F} \ : \ f \in \prod_{i \in I} X_i \right\}$ as the reduct product.

Let $\{X_i \ in \in I\}$ be a collection of relational structures and let $\mathcal{F}$ be an ultrafilter on I. Then we can use the same language L to make statements about $\prod_{i \in I} X_i/\mathcal{F}$ as we have to describe properties for the structures from which the reduct product is built.

Ultraproducts from a categorical view. Let $\mathbb{C}$ be a category with small products, denoted as $\prod_{i \in I} A_i$, and let $\{A_i\}_{i \in I}$ be a family of objects, then each filter $\mathcal{F}$ over the set of indices I defines a functor $A_{\mathcal{F}} : \mathcal{F} \mapsto \mathbb{C}$ such that

$$A_{\mathcal{F}}(J \subseteq J') = \prod_{i \in J'} A_i \mapsto \prod_{i \in J} A_i$$

where $p_{J',J}$ is the canonical projection. Then a filtered product of $\{A_i\}_{i \in I}$ modulo $\mathcal{F}$ is a colimit $\mu : A_{\mathcal{F}} \Rightarrow \prod_F A_i$ of the functor A_F. Filtered products, when they exist, are unique up to isomorphism. If $\mathcal{F}$ is an ultrafilter, then the filtered product modulo $\mathcal{F}$ is called an ultraproduct.
The filtered product construction from classical model theory has been probably defined categorically for the first time in [10] and has been used in some abstract model theoretic works, such as [7].

Let $\mathcal{F}$ be a filter over X and $Y \subseteq X$. The reduction $\mathcal{F}$ to Y is denoted as $\mathcal{F}|_Y$ and is defined as

$$\mathcal{F}|_Y = \{Y \cap B \mid B \in \mathcal{F}\}$$

We say that a class of filters $\mathbb{F}$ is closed under reductions if and only in $\mathcal{F}|_X \in \mathbb{F}$ for every $\mathcal{F} \in \mathbb{F}$ and $X \in \mathcal{F}$.

Proposition 1.2.9. *Let $\mathcal{F}$ be a filter over I and $\{A_i\}_{i \in I}$ a family of objects in a category $\mathbb{C}$ with small products and direct colimits. For each $J \in \mathcal{F}$. the filtered product $\prod_{\mathcal{F}|_J} A_i$ and $\prod_{\mathcal{F}} A_i$ are isomorphic.*

Definition 1.2.10 (Preservation of Categorical Filtered Product)**.** Consider a functor $G : \mathbb{C}' \mapsto \mathbb{C}$ and F a filter over a set I. Then G preserves the filtered product $\mu' : B_F \Rightarrow \prod_{\mathcal{F}} B_i$,i.e. $\mu' G : B_{\mathcal{F}}; G \Rightarrow \prod_{\mathcal{F}} G(B_i)$ is also a filtered product in $\mathbb{C}$ of $\{G(B_i)\}_{i\in I}$. For any class $\mathbb{F}$ of filters, we say a functor preserves $\mathbb{F}$-filtered products if it preserves all filtered products modulo $\mathcal{F}$ for all $\mathcal{F} \in \mathbb{F}$.

Definition 1.2.11 (Lifting of Categorical Filtered Products)**.** Let $\mathbb{F}$ be a class of filters closed under reductions. A functor $G : \mathbb{C}' \to \mathbb{C}$ lifts $\mathbb{F}$-filtered products when for each $\mathcal{F} \in \mathbb{F}$, and each filtered product $\mu : A_{\mathcal{F}} \Rightarrow \prod_{\mathcal{F}} A_i$ (where $\{A_i\}_{i\in I}$ is a family of objects in $\mathbb{C}$), for each $B \in \mathbb{C}'$ such that $G(B) = \prod_{\mathcal{F}} A_i$ there exists $J \in \mathcal{F}$ and $\{B_i\}_{i\in I}$ a family of objects in $\mathbb{C}'$ such that $G(B_i) = A_i$ for each $i \in J$ and such that there exists a filtered product $\mu' : B_{\mathcal{F}|_J} \Rightarrow B$ such that $\mu'_{J'} = \mu_{J'}$ for each $J' \in \mathcal{F}|_J$.

1.3. Institution Theory

Institution theory [**24, 11**] is an important trend in so-called Universal Logic [**7**]. It is a categorical abstract model theory which formalizes the notion of logical systems, including syntax, semantics and the satisfaction relation between them. One of the many achievements of Institution theory has been to provide a conceptually elegant and unifying definition of the nature of logical systems [**24**]. It provides a complete form of abstract model theory, including signature morphisms, model reducts, mappings between logics noted as Institution-independent model theory. From the Universal Logic view, Institution independent model theory signfies the development of model theory in the very abstract setting of arbitrary institutions, which provides an efficient framework for doing model theory by translation [**17**] or borrowing via a mapping theory (homomorphisms) between institutions.

1.3.1. Definition of Institution.
The purpose of this section is to provide the definition of fundamental concepts of institution theory that will be subsequently applied in our study of the "square of oppositions".

Definition 1.3.1 (Institutions)**.** An *institution* $\mathcal{I} = \left(\mathbb{S}ig^{\mathcal{I}}, \boldsymbol{Sen}^{\mathcal{I}}, \boldsymbol{Mod}^{\mathcal{I}}, \models^{\mathcal{I}}\right)$ consists of:

(1) a category $\mathbb{S}ig^{\mathcal{I}}$, the objects of which are called *signatures*,

(2) a functor $\boldsymbol{Sen}^{\mathcal{I}} : \mathbb{S}ig^{\mathcal{I}} \to \mathbb{S}et$ such that it assigns a set the elements of which are called *sentences* over each signature,

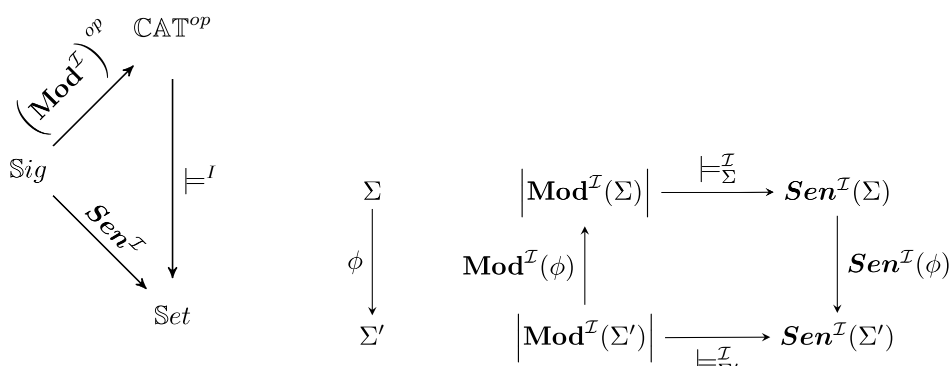
(3) a functor $\mathbf{Mod}^{\mathcal{I}} : \left(\mathbb{S}ig^{\mathcal{I}}\right)^{op} \to \mathbb{CAT}$ giving a category the objects of which are called Σ-*models* and the arrows of which are called Σ-*morphisms* for each signature Σ, and

(4) a relation $\models^{\mathcal{I}}_{\Sigma} \subseteq \left|Mod^{\mathcal{I}}(\Sigma)\right| \times \boldsymbol{Sen}^{\mathcal{I}}(\Sigma)$ for each $\Sigma \in \left|\mathbb{S}ig^{\mathcal{I}}\right|$, called Σ-*satisfaction*

such that for each morphism $\phi : \Sigma \to \Sigma'$ in $\mathbb{S}ig^{\mathcal{I}}$, the *satisfaction condition*

$$M' \models^{\mathcal{I}}_{\Sigma'} \boldsymbol{Sen}^{\mathcal{I}}(\phi)(\rho) \;\text{ iff }\; \mathbf{Mod}^{\mathcal{I}}(\phi)(M') \models^{\mathcal{I}}_{\Sigma} \rho$$

holds for each $M' \in \left|\mathbf{Mod}^{\mathcal{I}}\right|$ and $\rho \in \boldsymbol{Sen}^{\mathcal{I}}(\Sigma)$.



(a) Institution fundamental triangle

(b) Institution fundamental square

Figure 17. Institution fundamental schemes

In the following we will examine one of the most useful examples of institution, that of standard Propositional Logic (**PL**).

Example 1.3.2 (Propositional Logic)**.** The institution **PL** of Propositional Logic is defined in the following way:

- The category $\mathbb{S}ig^{\boldsymbol{PL}}$ has sets of propositional variables as objects and the arrows represent the functions between them.
- A signature morphism σ is a mapping between the propositional variables.

- The functor $\boldsymbol{Sen}^{\boldsymbol{PL}}$ maps every signature Σ to $\boldsymbol{Sen}^{\boldsymbol{PL}}(\Sigma)$ which consists of propositional formulas of propositional variables from Σ and connectives for conjunction, disjunction, implication and negation.
- The $\boldsymbol{Sen}^{\boldsymbol{PL}}(\sigma)$ is the extension of σ to all formulas.
- Models of Σ are truth valuations, i.e. mappings from Σ into the standard Boolean algebra $Bool = \{0, 1\}$.
- A model morphism between Σ-models M and M' exists if and only if for all $p \in \Sigma,\ M(p) \leq M'(p)$.
- Given $\sigma : \Sigma_1 \to \Sigma_2$ and a Σ_2-model $M_2 : \Sigma_2 \to Bool$, then the reduct $M_2 \upharpoonright_\sigma$ is the composition $M_2 \circ \sigma$.
- $M \models^{\boldsymbol{PL}}_{\Sigma} \phi$ if and only if ϕ evaluates 1 under the standard extension of M to all formulas.

Example 1.3.3 (Temporal Logic)**.** The following example is a simplified version of Temporal Logic (**TL**). The following is its formalization in the theory of institutions.

- The signatures $\mathbb{S}ig^{TL}$ consist of sets of actions;
- The models $\mathbf{Mod}^{TL}(\Sigma)$ consist of sets of runs, which are finite or infinite sequences of (sets of) actions;
- The sentences $\boldsymbol{Sen}(\Sigma)$ consist of sets of sentences which are built up from atomic sentences p using the standard propositional and temporal connectives;
- a satisfaction relation $M \models^{TL}_{\Sigma} \phi$ holds if and only if ϕ holds at the beginning of every run in M.

This section deals with examples of institutions. Next to propositional logic, the most standard example of institution is First Order Logic **FOL**.

Example 1.3.4 (First Order Logic)**.** A *signature* in **FOL** is a triple (S, F, P) and consists of :

- S; the set of *sort* symbols, for example $S = \{N, Z\}$ where N is denoting the natural numbers ($\mathbb{N}$) and Z the integers ($\mathbb{Z}$);
- $F = \{F_{w \to s} \mid w \in S^*, s \in S\}$; a family of sets of *operation* symbols such that $F_{w \to s}$ denotes the set of operations with arity w and sort s for example $F_{NN \to N} = \{+\}, F_{ZZ \to Z} = \{+, -\}$;

- $P = \{P_w \mid w \in S^*\}$; a family of set relation symbols where P_w denotes the set of relations with arity w. For example $P_{NN} = P_{ZZ} = \{\leq\}$ or $P_w = \emptyset$.

The models of **FOL** are related with the sort symbols in a natural way and the sentences are the standard expansion of **PL**. Furthermore, the institution of **PL** can be seen as a sub-institution of **FOL** obtained by restricting the signatures to those with an empty set of sort symbols.

Example 1.3.5 (Weak Propositional Logic)**.** The weak propositional logic (**WPL**) designates a variation of Propositional Logic in which the sentences are the same as in **PL**, but the models are valuations $M : \boldsymbol{Sen}(P) \mapsto \{0, 1\}$ with the standard truth table semantics of all Boolean connectives except negation, i.e.:

- $M(\phi \wedge \psi) = 1$ if and only if $M(\phi) = M(\psi) = 1$;
- $M(\phi \vee \psi) = 0$ if and only if $M(\phi) = M(\psi) = 0$;
- if $M(\phi) = 1$ then $M(\neg\phi) = 0$;

Example 1.3.6 (Modal First Order Logic)**.** The last example is the standard Modal First Order Logic (**MFOL**) with modalities $\Box, \Diamond$ and Kripke semantics. The **MFOL** *signatures* are sextuples (S, S_0, F, F_0, P, P_0), where (S, F, P) is the signature of **FOL** and (S_0, F_0, P_0) is a sub-signature of (S, F, P) of rigid symbols. An **MFOL** model (W, R), called a Kripke model, consists of

- a family $W = \{W_i\}_{i \in I_W}$ of possible worlds, which are models in **FOL**;
- a binary relation $R \subseteq I_W \times I_W$ between the possible worlds such that the following sharing constraint holds: $\forall i, j \in I_W$ we have that $W_i^x = W_j^x$ for each rigid x.

The satisfaction of **MFOL** sentences by the Kripke models is defined in the following way:

$$(W, R) \models \phi \;\Leftrightarrow\; (W, R) \models^i \phi \;\forall i \in I_W \tag{1.7}$$

where $(W, R) \models^i \phi$ is defined by induction:

- $(W, R) \models^i \phi$ if and only if $W_i \models^{\boldsymbol{FOL}} \phi$ for all atom ϕ and each $i \in I_W$;
- $(W, R) \models^i \phi \wedge \psi$ if and only if $W_i \models^{\boldsymbol{FOL}} \phi$ and $W_i \models^{\boldsymbol{FOL}} \psi$;
- $(W, R) \models^i \Box\phi$ if and only if $W_i \models^k \phi$ for $< i, k > \in R$;

- $(W, R) \models^i \forall X\phi$ if and only if $(W', R) \models^i \phi$ for all expansion (W', R) of (W, R) to a Kripke model;
- $\Diamond\phi$ is an abbreviation of $\neg\Box\neg\phi$;

Like in **PL** we get the institution of Modal Propositional Logic (**MPL**) as a sub-institution of **MFOL** defined by the signatures with an empty set of sort symbols and an empty set of rigid relation symbols.

1.3.2. Morphisms and Comorphisms. One of the basic notion in the Institution Theory is the morphism of two institutions. The concept of institution morphism is formalizing the mapping from a more complex to a simpler institution.

Definition 1.3.7 (Institution Morphism)**.** Let $\mathcal{I}$ and $\mathcal{I}'$ be two institutions, then an **institution morphism** $\Phi : \mathcal{I} \to \mathcal{I}'$ consists of:

(1) a functor $\Phi : \mathbb{S}ig \to \mathbb{S}ig'$ translating $\mathcal{I}$-signatures to $\mathcal{I}'$-signatures;

(2) a natural transformation $\alpha : \Phi; \boldsymbol{Sen'} \Rightarrow \boldsymbol{Sen}$, as a sentence translation $\alpha_\Sigma : \boldsymbol{Sen}^{\mathcal{I}'}(\Phi(\Sigma)) \to \boldsymbol{Sen}^{\mathcal{I}}(\Sigma)$;

(3) a natural transformation $\beta : \mathbf{Mod} \Rightarrow \Phi^{op}; \mathbf{Mod'}$, as a model translation $\beta_\Sigma : Mod^{\mathcal{I}}(\Sigma) \to \mathbf{Mod}^{\mathcal{I}'}(\Phi(\Sigma))$;

such that the following satisfaction condition holds

$$m \models^{\mathcal{I}}_{\Sigma} \alpha_\Sigma(e') \quad \text{iff} \quad \beta_\Sigma(m) \models^{\mathcal{I}'}_{\Phi(\Sigma)} e' \tag{1.8}$$

for any Σ-model m from $\mathcal{I}$ and any $\Phi(\sigma)$-sentence e' from $\mathcal{I}'$

Figures 19a and 19b show a representation of the natural transformations α_Σ and β_Σ. The institution morphisms are suitable to formalize "forgetful" maps between more complex and simpler institutions.

Example 1.3.8. A first example is a morphism from **FOL** to **PL**. This morphism maps any first-order logic signature to its set of sentences, because each sentence can be regarded as a propositional variable in a propositional logic signature. Also the $\alpha_\Sigma(\rho)$ is just the first-order sentence ρ and $\beta_\Sigma(M)$ is the propositional logic model consisting of all Σ-sentences that are true in M.

$$\begin{array}{ccc}
\boldsymbol{Sen}^{\mathcal{I}'}(\Phi(\Sigma)) & \xrightarrow{\alpha_\Sigma} & \boldsymbol{Sen}^{\mathcal{I}}(\Sigma) \\
{\scriptstyle \boldsymbol{Sen}^{\mathcal{I}'}(\Phi(\phi))}\downarrow & & \downarrow{\scriptstyle \boldsymbol{Sen}^{\mathcal{I}}(\phi)} \\
\boldsymbol{Sen}^{\mathcal{I}'}(\Phi(\Sigma')) & \xrightarrow[\alpha_{\Sigma'}]{} & \boldsymbol{Sen}^{\mathcal{I}}(\Sigma')
\end{array}$$

(a) First Morphism square

$$\begin{array}{ccc}
\mathbf{Mod}^{\mathcal{I}}(\Sigma') & \xrightarrow{\beta_{\Sigma'}} & \mathbf{Mod}^{\mathcal{I}'}(\Phi(\Sigma')) \\
{\scriptstyle \mathbf{Mod}^{\mathcal{I}}(\phi)}\downarrow & & \downarrow{\scriptstyle \mathbf{Mod}^{\mathcal{I}'}(\Phi(\phi))} \\
\mathbf{Mod}^{\mathcal{I}}(\Sigma) & \xrightarrow[\beta_{\Sigma}]{} & \mathbf{Mod}^{\mathcal{I}'}(\Phi(\Sigma))
\end{array}$$

(b) Second Morphism Square

Figure 18. Fundamental Schemes

Example 1.3.9 (The morphism between **FOL** and **MFOL**.)**.** Regarding the definition of these two institutions we can define the morphism $\Phi : \boldsymbol{FOL} \mapsto \boldsymbol{MFOL}$ which maps the $\boldsymbol{FOL} - (S, F, P)$ signature to the $\boldsymbol{MFOL} - (S, S, F, F, P, P)$ signature, such that the natural transformation α erases the modalities from the sentences.

By reversing the translation of the signatures we get the concept of comorphism of institutions which is formalizing the embedding of a simpler institution into a more complex one.

Definition 1.3.10 (Institution Comorphism)**.** Let $\mathcal{I}'$ and $\mathcal{I}$ be institutions, then an **institution comorphism** $\Phi : \mathcal{I}' \to \mathcal{I}$ consists of:

(1) a functor $\Phi : \mathbb{S}ig' \to \mathbb{S}ig$ translating $\mathcal{I}'$ signatures to $\mathcal{I}$ signatures;

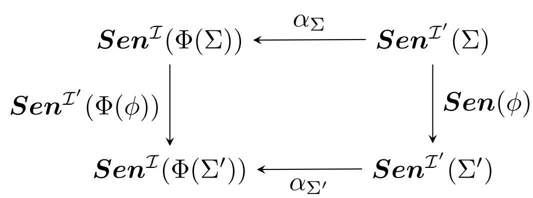
(2) a natural transformation $\alpha : \boldsymbol{Sen}' \Rightarrow \Phi; \boldsymbol{Sen}$, as a sentence translation $\alpha_\Sigma : \boldsymbol{Sen}^{\mathcal{I}'}(\Sigma) \to \boldsymbol{Sen}^{\mathcal{I}}(\Phi(\Sigma))$;

(3) a natural transformation $\beta : \Phi^{op}; \mathbf{Mod}' \Rightarrow \mathbf{Mod}$, as a model translation/reduct $\beta_\Sigma : \mathbf{Mod}^{\mathcal{I}}(\Phi(\Sigma)) \to \mathbf{Mod}^{\mathcal{I}'}(\Sigma)$;

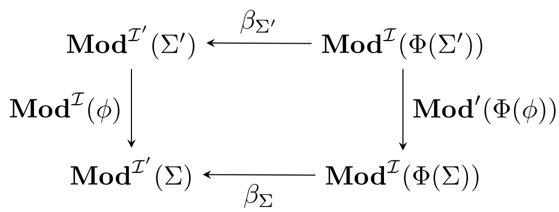
such that the following satisfaction condition holds

$$m \models^{\mathcal{I}}_{\Phi(\Sigma)} \alpha_\Sigma(e) \quad \text{iff} \quad \beta_\Sigma(m) \models^{\mathcal{I}'}_{\Sigma} e \tag{1.9}$$

for any $\Phi(\Sigma)$-model m from $\mathcal{I}$ and any Σ-sentence e from $\mathcal{I}'$.



(a) First Comorphism square



(b) Second Comorphism square

Figure 19. Fundamentals Schemes

Example 1.3.11. Reversing Example 1.3.8, the embedding of **PL** into **FOL** can be regarded as a comorphism that interprets any set of propositional variables as a first-order signature that has only relation symbols of arity zero.

1.3.3. Galois Connection. Let Σ be a signature in an arbitrary institution $\mathcal{I} = \left(\mathbb{S}ig^{\mathcal{I}}, \boldsymbol{Sen}^{\mathcal{I}}, \boldsymbol{Mod}^{\mathcal{I}}, \models^{\mathcal{I}}\right)$. If E is set of sentences, then the **models** of E are the a set of models such that $M \models \phi$ for every sentence in E. Moreover, the **theory** of a class of models $\mathbb{M}$ is the a set of sentences ϕ such that for every model M in this class holds $M \models \phi$. In formal language, this is expressed as follows:

- for every set of Σ-sentences E, we have

$$E^* = \{M \in \mathbf{Mod}(\Sigma) \mid M \models_{\Sigma} \phi \ \forall \phi \in E\}$$

- for every class $\mathbb{M}$ of Σ-models, we have

$$\mathbb{M}^* = \{\phi \in \boldsymbol{Sen}(\Sigma) \mid M \models \phi \ \forall M \in \mathbb{M}\}$$

Remark 1.3.12. For any sentence ϕ and a model M, we denote $\{\phi\}^* = \phi^*$ and $\{M\}^* = M^*$

It is evident that the previous definition implies two functions $(-)^*$, which are known as the *Galois Connection*. We continue with the following definitions, which are, as will be made clear, the basic tools employed in our demonstration. As noted in [**24**], a *specification* provides a mathematical theory of the behaviour of a program and if a theory consists of all the sentences that are true under that behaviour then it is important to define the fundamental properties of theories over an arbitrary institution.

Definition 1.3.13. Let $\mathcal{I}$ be a fixed but arbitrary institution. Then

(1) A Σ-*presentation* is a pair $\langle \Sigma, E \rangle$, where Σ is a signature and E is collection of Σ-sentences.

(2) A Σ-model M *satisfies* a presentation $\langle \Sigma, E \rangle$ if it satisfies each sentence in E; we write $M \models E$ in this case.

(3) Given a collection E of Σ-sentences, let E^* be the collection of all Σ-models that satisfy each sentence in E.

(4) Given a collection M of Σ-models, let M^* be the collection of all Σ-sentences that are satisfied by each model in M; also let M^* denote $\langle \Sigma, M^* \rangle$ called the *theory* of M.

(5) The *closure* of a collection E of Σ-sentences is E^{**}, denoted by $E^\bullet$

(6) A collection E of Σ-sentences is *closed* if and only if $E = E^\bullet$.

(7) A Σ-*theory* is a presentation $\langle \Sigma, E \rangle$ such that E is closed.

(8) The Σ-theory *presented* by a presentation $\langle \Sigma, E \rangle$ is $\langle \Sigma, E^\bullet \rangle$.

(9) A Σ-sentence e is *semantically entailed* by a collection E of Σ-sentences, writen $E \models e$, if and only if $e \in E^\bullet$

Lemma 1.3.14. *The two functions denoted* $< * >$ *in the previous paragraph form what is known as the* ***Galois connection*** *(see* [**31**]*), in that they satisfy the following properties, for any collections* E, E' *of* Σ*-sentences and collections* $\mathbb{M}, \mathbb{M}'$ *of* Σ*-models:*

(1) $E \subseteq E' \Rightarrow E'^* \subseteq E^*$.

(2) $\mathbb{M} \subseteq \mathbb{M}' \Rightarrow \mathbb{M}'^* \subset \mathbb{M}^*$.

(3) $E \subseteq E^{**}$.

(4) $\mathbb{M} \subseteq \mathbb{M}^{**}$.

(5) $E^* = E^{***}$.

(6) $\mathbb{M}^* = \mathbb{M}^{***}$.

(7) *There is a dual (i.e. inclusion reversing) isomorphism between the closed collections of sentences and the closed collections of models. This isomorphism maps unions to intersections and intersections to unions.*

(a) $\bigcap_n E_n^* = \left(\bigcup_n E_n\right)^*$

(b) $\left(\bigcap_n E_n^*\right)^{**} = \left(\bigcup_n E_n\right)^*$

(c) $\left(\bigcup_n E_n^{**}\right)^* = \bigcap_n E_n^*$

(d) $\left(\bigcup_n E_n^{**}\right)^* = \left(\bigcup_n E_n\right)^*$

(e) $\left(\bigcap_n E_n^{**}\right)^* = \left(\bigcup_n E_n^*\right)^{**}$

There are also dual identities to (a)-(e) for collections of models.

Proof. We procced to prove the (1), the (2) and the 7 (a). Let E be a set of Σ-sentences and

$$E^* = \{M \in \mathbf{Mod}(\Sigma) \mid \forall \phi \in E \ M \models \phi\}$$

a collection of models. Let $E_1 \subseteq E_2 \subseteq |Sen(\Sigma)|$ then we take two collections of set of models, E_1^* and E_2^* from the Galois Connection.

$$
\begin{aligned}
M \in E_2^* \Rightarrow \\
(\forall \phi \in E_2) \ M \models \phi \Rightarrow \\
(\forall \phi \in E_1) \ M \models \phi \Rightarrow \\
M \in E_1^* \Rightarrow \\
E_2^* \subseteq E_1^*
\end{aligned}
\tag{1.10}
$$

For the (2), if $M \subseteq M' \subseteq |\mathbf{Mod}(\Sigma)|$ then

$$\begin{aligned} \phi \in M'^* \Rightarrow \\ (\forall m \in M')\ m \models^{\mathcal{I}}_{\Sigma} \phi \Rightarrow \\ (\forall m \in M)\ m \models^{\mathcal{I}}_{\Sigma} \phi \Rightarrow \\ \phi \in M \Rightarrow \\ M'^* \subseteq M^* \end{aligned} \tag{1.11}$$

And for the conjunction, if $M \in E_1^* \cap E_2^*$ then

$$\begin{aligned} M \in E_1^* \ \&\ M \in E_2^* \Leftrightarrow \\ (\forall \phi \in E_1)\ M \models^{\mathcal{I}}_{\Sigma} \phi \ \&\ (\forall \phi \in E_2)\ M \models^{\mathcal{I}}_{\Sigma} \phi \Leftrightarrow \\ (\forall \phi \in (E_1 \cup E_2))\ M \models^{\mathcal{I}}_{\Sigma} \phi \ \Leftrightarrow \\ M \in (E_1 \cup E_2)^* \end{aligned} \tag{1.12}$$

Thus we conclude that $\phi^* \cap \psi^* = (\phi \cup \psi)^*$ □

1.3.4. Internal Logic. The semantics of the Boolean connectors can be formally defined in institutions [17] also by using the Galois connection as defined above.

Definition 1.3.15 (Internal Boolean Connectives)**.** Let Σ be a signature in an institution then :

- the Σ-sentence ϕ is a (semantic) negation of ψ when $\phi^* = |\mathbf{Mod}(\Sigma)| \setminus \psi^*$
- the Σ-sentence ϕ is the (semantic) conjuction of the Σ-sentence ψ_1 and ψ_2 when $\phi^* = \psi_1^* \cap \psi_2^*$

Remark 1.3.16. The Boolean connectives, such as disjunction $\vee$, implication $\Rightarrow$, equivalence $\Leftrightarrow$, etc can be derived as usually from negations and conjunctions.

Remark 1.3.17. The semantic conjunction, negation, implication etc. are unique only to semantic equivalence, which means that from this point of view sentences satisfied by the same models are indistinguishable. This observation will be very useful in the continuation of our work.

The approach to defining quantifiers in institution theory is based on the fundamental approach by assimilating the evaluation of variables along with the extension of models as signature extensions.

Naturally, in a certain institution one can talk about and define the valuation of variables X in M models as functions from X to some underlying set of model M. However, this is not the case when we want to achieve the maximum level of abstraction. We can do this by integrating the set of variables X with the symbol extensions, so that we can obtain this construction by adding them to the respective signatures. We may note that for any Σ-model M, there is a canonical one-to-one correspondence between the valuations $X \to M$ and the $\Sigma + X$-models M' such that their reducts to Σ are just M. This construction, according to Diaconescu [17], implies that variables can become part of the signatures, which breaks with the habit of traditional approaches to logic of keeping variables separated from the language. Hence at the level of institution independent model theory a variable of a signature Σ is just a signature morphism $\chi : \Sigma \to \Sigma'$. Given this, the reader can now understand that the following definitions come naturally based on our framework.

Definition 1.3.18 (Internal Quantifiers)**.** For any signature morphism $\chi : \Sigma \to \Sigma'$ in an arbitrary institution

- a Σ-sentence ϕ is a (semantic) *existential χ-quantification* of a χ-sentence ψ when $\phi^* = (\psi^*) \restriction_\chi$; in this case we write ϕ as $\exists\chi\psi$.
- a Σ-sentence ϕ is a (semantic) *universal χ-quantification* of a χ-sentence ϕ when $\phi^* = |\mathbf{Mod}(\Sigma)| \setminus (|\mathbf{Mod}(\Sigma')| \setminus \psi^*) \restriction_\chi$; in this case we write ϕ as $\forall\chi\psi$

1.3.5. Institutions with proofs. Approaching the definition of proof from the perspective of syntax is fundamental in mathematics. So what is proof? As Diaconescu notes in [17] it is a one-way move from a set E to a set E' of sentences, called E proves form E' and meaning that E' is established true on the basis of E being established true. There can be several different ways to prove E' from E.

Definition 1.3.19 (Proof System)**.** A Proof System $(\mathbb{S}ign, Sen, Pf)$ is a triple whose elements are

- a category of signatures $\mathbb{S}ign$

- a functor $Sen : \mathbb{S}ign \to \mathbb{S}et$ called sentence functor
- and a functor $Pf : \mathbb{S}ign \to \mathbb{CAT}$ called proof functor which the category of the $\Sigma-$proofs is giving for each signature Σ.

such that

(1) $\boldsymbol{Sen};\mathcal{P};(-)^{op}$ is a sub-functor of Pf, and

(2) the inclusion $\mathcal{P}\left(\boldsymbol{Sen}(\Sigma)\right)^{op} \hookrightarrow Pf(\Sigma)$ is broad and preserves finite products of disjoint sets of sentences for each signature Σ, where $\mathcal{P}\mathbb{S}et \to \mathbb{CAT}$ is the power-set functor.

Remark 1.3.20. The inclusion $\mathcal{P}\left(\boldsymbol{Sen}(\Sigma)\right)^{op} \hookrightarrow Pf(\Sigma)$ means that $Pf(\Sigma)$ has subsets of $\boldsymbol{Sen}(\Sigma)$ as objects, the preservation of products of which implies that there are distinguished monotonicity proofs $\supseteq_{\Gamma,E}\colon \Gamma \to E$ whenever $E \subseteq \Gamma$ which is preserved by signature morphisms, i.e. $\phi\left(\supseteq_{\Gamma,E}\right) = \supseteq_{\phi(\Gamma),\phi(E)}$ and that proofs $\Gamma \to E_1 \uplus E_2$ are in one-one natural correspondence with pairs of proofs $\langle \Gamma \to E_1, \Gamma \to E_2\rangle$ [17].

Example 1.3.21 (PL)**.** The set proof rules of propositional logic **PL**.

$$
\begin{array}{ll}
\text{A1} & \emptyset \vdash \phi \Rightarrow (\psi \Rightarrow \phi) \\
\text{A2} & \emptyset \vdash (\phi \Rightarrow (\psi \Rightarrow \chi)) \Rightarrow ((\phi \Rightarrow \psi) \Rightarrow (\phi \Rightarrow \chi)) \\
\text{A3} & \emptyset \vdash (\neg\psi \Rightarrow \neg\phi) \Rightarrow ((\neg\psi \Rightarrow \psi) \Rightarrow \psi) \\
\text{MP} & \{\phi, \phi \Rightarrow \psi\} \vdash \psi
\end{array}
$$

Proposition 1.3.22 (Entailment institution)**.** *Each proof system* $(\mathbb{S}ig, \boldsymbol{Sen}, Pf)$ *determines an institution* $\mathcal{I} = \left(\mathbb{S}ig^{\mathcal{I}}, \boldsymbol{Sen}^{\mathcal{I}}, \boldsymbol{Mod}^{\mathcal{I}}, \models^{\mathcal{I}}\right)$ *called the entailment institution of the proof system where for each signature* $\Sigma \in |\mathbb{S}ig|$

- *the entailment* Σ*-models are pairs* (ψ, E') *where* $\psi : \Sigma \to \Sigma'$ *is a signature morphism and* E' *is a* Σ'*-theory*
- *a* Σ*-model homomorphism* $\phi : (\psi : \Sigma' \to (\Sigma', E')) \to (\psi' : \Sigma \to (\Sigma'', E''))$ *is just a theory morphism* $\phi : (\Sigma, E) \to (\Sigma', E'')$ *such that* $\psi; \phi = \psi'$
- *a* Σ*-model* (ψ, E') *satisfies a* Σ*-sentence* ρ *iff* $\psi(\rho) \in E'$
- *model reducts are obtained just by composition to the left.*

We will conclude this introduction with the following theorem [17].

Theorem 1.3.23. *Any entailment institution is sound and complete*

This theorem holds particular significance, being the link between institution-independent model theory and topology, two areas virtually unrelated to each other. In the next chapter we will examine how to use existing models of a logical system to produce new topological semantics. This construct will enable us to explore the properties of logical systems through the use of topological tools. We will be able to explore the concept of completeness of a logical system by comparing two topologies, namely syntatic topology and semantic topology.

Chapter 2

Aristotelian Institution-Independent Square

Abstract. In recent decades, the research interest in the square of opposition has increased. New interpretations and extensions, including non-Aristotelian ones, have been proposed. The need to integrate these extensions into a universal theory leads us to abstract categorical model theory (theory of institutions) [38]. In the first section, we introduce the square of opposition; the second section introduces the concept of rhombus of opposition and examines basic cases of morphological change of the squares of opposition inside and between logical systems. In the third section, we use the concept of the *Galois Connection* to show the equilibrium that one can create between the standard square of sentences and the internal semantics of Boolean connectives, using them at a meta-level. Finally, we introduce the concept of a *dual square* that can give us not only squares for propositions but also squares for sets of sentences. Since quite a few logical systems do not have internal connectives, it is not useful to talk about proposition graphs, but about sets of models and of sets of sentences. Therefore, we can now write basic relationships, such as contradiction.

2.1. Aristotelian Relations

During the second half of the twentieth century, the research in the square of oppositionwas revived. First, Augustin Sesmat [53] and Robert Blanché [9] extended independently the square of opposition to a logical hexagon which includes the relationships of six statements. This was followed by an extension to a "logical cube", that paved the way to the development of a series of n-dimensional objects called logical bi-simplexes of dimension n [47].

The second line of research was developed in the past twenty years through Jean-Yves Beziau's attempts to find an intuitive basis for paraconsistent negation, which is the O-corner of the square of opposition [12].

The latter author posed the question how to compare all versions of the square of opposition and, if possible, their different illustrations by various configurations. This required moving to a wider framework, where the different versions of the theory within different logics could be compared. To this effect, the author appealed to abstract categorical model theory and specifically to the theory of institutions [37].

The concept of the institution was introduced by Joseph Goguen and Rod Burstall in the late 1970s, to deal with the vast variety of logical systems developed and used in computer science. The concept tries to capture the essence of the concept of "logical system" [24]. Informally speaking, an institution is a mathematical structure for "logical systems", based on the concept of satisfaction between sentences and models.

In the first section, we introduce the concept of the square of opposition. In the second section, we expose fundamental concepts from category theory and institution theory that are necessary for our study. The third section introduces the concept of the rhombus of opposition and examines certain aspects of the configurational change of the squares of opposition inside and between logical systems.

In the fourth section, we use the concept of the Galois connection, which is a useful generalization of correspondence between subgroups and subfields that are studied in Galois' theory, to show the equilibrium that one can establish between the standard square of opposition (of sentences) and the internal semantics of Boolean connectives at a meta-level.

Finally, we introduce the concept of the dual square that can give us not only squares for propositions but also squares for sets of sentences. Since quite a few logical systems do not have internal connectives, it is not useful to talk about proposition graphs, but sets of models and sets of sentences. In this way, we can now write basic relationships, such as contradiction.

2.2. Squares of Opposition

The theory of opposition was developed by Aristotle in *De Interpretatione* 6–7, 17 b 17–26 and *Prior Analytics* I.2, 25 a 1–25 to describe the relations between the four basic categorical judgements. During the Middle Ages, Aristotle's theory was represented by a square diagram. This was done by altering the semantics of the O form. During the 19th — 20th centuries, it assumed two major reinterpretations: a) within the context of the algebra of logic (see Figure 1) (Boole, Venn and others),

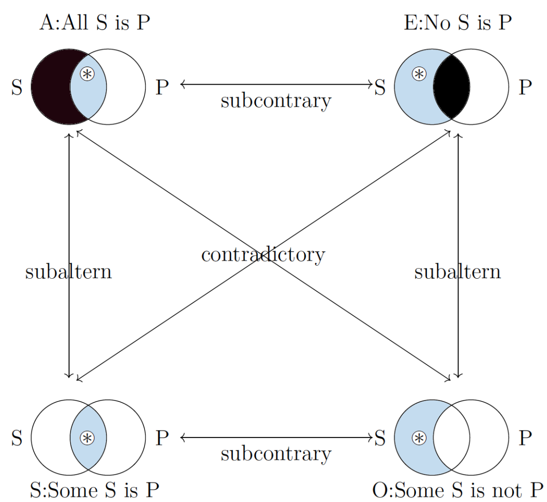


Figure 1. Representation of the Square of Opposition in algebra of logic (Boole, Venn, and others).

b) within the second-order predicate logic, by using the newly introduced concept of quantified variables by Frege (see Figures 2 and 3). Within these interpretations, the shape of the ”square” remains unaffected.

Beginning with Nicolai A. Vasiliev (1880 – 1940), the traditional “square” loses its original square shape for the first time; it is transformed into ”triangle.” This was through a new alteration of semantics of the O form, based on Aristotelian concepts that had been overlooked in the Aristotelian tradition of logic, notably the concept of indefinite judgement.

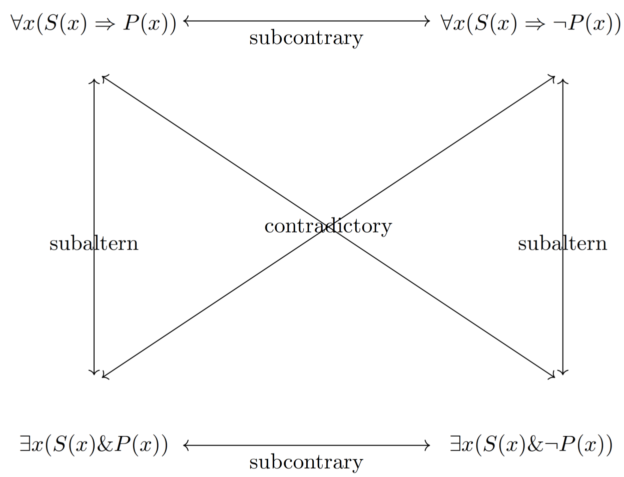


Figure 2. Representation of the square of opposition in the functional tradition of logic (Frege and others).

During the 20th century, new transformations of the "square" into various shapes appear, i.e. into "hexagon" [9], or "cube" (Figure 3), by altering the semantics and establishing relationships between truth-values. The new objects admit various interpretations in terms of traditional logic, quantification theory, modal logic, order theory, or paraconsistent logic.

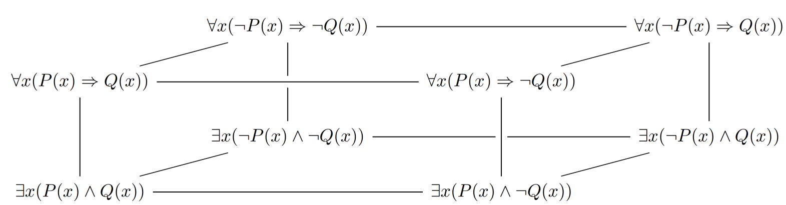


Figure 3. Cube of opposition of quantified statements

However, a question arises: how do all these configurations relate, often realized within different logics? Can we describe these transformations in logical terms? What changes and what remains invariant in these transformations?

To examine these questions, we appeal to the concepts of the theory of institutions, introduced by Goguen and Burstall [24]. The theory of institutions has the advantage of not being exclusive to any specific logical system. Moreover, its high level of abstraction allows for the accommodation of not only classical, but also non-classical logical systems. A structure-preserving mapping, called morphism of institutions, is defined by Goguen and Burstall [4] and it operates as a projection from a more complex institution into a simpler one. By reversing this operation, we get the concept of comorphism of institutions, which embeds a simple institution into more complex ones.

Using morphisms and co-morphisms of abstract logical systems, we will study the transformation of the configurations, traditionally called "square of opposition" into entities of different shape, taking into account the changes in semantics of the underlying logical systems. We will try to study the generation of new entities (diagrams) out of old ones with categorical tools, as well as by encoding/embedding simple diagrams (squares) into entities of higher complexity (polygons or 3D objects) and vice versa. In other words, in the context of universal algebra we will study the following question: how a change in semantics might generate different outcomes (of various shapes) of the so-called "square of opposition".

2.3. Institutional Square of Opposition

2.3.1. The Aristotelian relations of judgements. The square of opposition is commonly known as a diagram for which many extensions have been proposed in the second half of the twentieth century. However, most of them are discussed at an informal level.

Using the formalism introduced by Hans Smessaert and Lorenz Demey [**53**], we generalize these concepts to the level of the theory of institutions. It should be emphasized that the theory of institutions guarantees that the following definitions apply to all logical systems under consideration. For the sake of convenience, we assume that the logical systems contain the classical connectives as a syntactic method of constructing sentences. The authors cited above define Aristotelian Geometry as a logical system, which has the links of denial, conjunction and implication.

Definition 2.3.1 (Aristotelian Relation of Contradictoriness)**.** Let $\mathcal{I} = \left(\mathbb{S}ig^{\mathcal{I}}, \boldsymbol{Sen}^{\mathcal{I}}, \boldsymbol{Mod}^{\mathcal{I}}, \models^{\mathcal{I}}\right)$ be an arbitrary institution and $\Sigma \in \mathbb{S}ig^{\mathcal{I}}$ and ϕ, ψ be propositions in $\boldsymbol{Sen}(\Sigma)$. Then the propositions ϕ, ψ are called contradictory, if the truth of one implies the falsity of the other, and conversely.

$$\models^{\mathcal{I}}_{\Sigma} (\phi \Rightarrow \neg\psi) \wedge (\neg\psi \Rightarrow \phi) \tag{2.1}$$

$$\models^{\mathcal{I}}_{\Sigma} (\neg\phi \vee \neg\psi) \wedge (\psi \vee \phi) \tag{2.2}$$

$$\models^{\mathcal{I}}_{\Sigma} \neg(\phi \wedge \psi) \text{ and } \models^{\mathcal{I}}_{\Sigma} \neg(\neg\phi \wedge \neg\psi) \tag{2.3}$$

We denote the relation between two contradictory sentences by $R_C(\phi, \psi)$ or by a graph

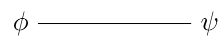


Figure 4. Geometrical representation of contradictory sentences.

Definition 2.3.2 (Aristotelian Relation of Contrariety)**.** Let $\mathcal{I} = \left(\mathbb{S}ig^{\mathcal{I}}, \boldsymbol{Sen}^{\mathcal{I}}, \boldsymbol{Mod}^{\mathcal{I}}, \models^{\mathcal{I}}\right)$ be an arbitrary institution and $\Sigma \in \mathbb{S}ig^{\mathcal{I}}$ and ϕ, ψ be propositions in $\boldsymbol{Sen}(\Sigma)$. Then the propositions ϕ, ψ are called contrary, if they cannot both be true.

$$\models^{\mathcal{I}}_{\Sigma} \neg(\phi \wedge \psi) \text{ and } \not\models^{\mathcal{I}}_{\Sigma} \neg(\neg\phi \wedge \neg\psi) \tag{2.4}$$

We denote the relation between two contrary sentences by $R_c(\phi, \psi)$ or by a graph

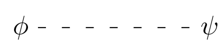


Figure 5. Geometrical representation of contrary sentences.

Definition 2.3.3 (Aristotelian Relation of Subcontrary)**.** Let $\mathcal{I} = \left(\mathbb{S}ig^{\mathcal{I}}, \boldsymbol{Sen}^{\mathcal{I}}, \boldsymbol{Mod}^{\mathcal{I}}, \models^{\mathcal{I}}\right)$ be an arbitrary institution and $\Sigma \in \mathbb{S}ig^{\mathcal{I}}$ and ϕ, ψ be propositions in $\boldsymbol{Sen}(\Sigma)$. Then the propositions ϕ, ψ are called subcontrary, if it is impossible for both to be false.

$$\not\models^{\mathcal{I}}_{\Sigma} \neg(\phi \wedge \psi) \text{ and } \models^{\mathcal{I}}_{\Sigma} \neg(\neg\phi \wedge \neg\psi) \tag{2.5}$$

We denote the relation between two subcontrary sentences by $R_s(\phi, \psi)$ or by a graph

Figure 6. Geometrical representation of subcontrary sentences.

Definition 2.3.4 (Aristotelian Relation of Subalternation)**.** Let $\mathcal{I} = \left(\mathbb{S}ig^{\mathcal{I}}, \boldsymbol{Sen}^{\mathcal{I}}, \boldsymbol{Mod}^{\mathcal{I}}, \models^{\mathcal{I}}\right)$ be an arbitrary institution and $\Sigma \in \mathbb{S}ig^{\mathcal{I}}$ and ϕ, ψ be propositions in $\boldsymbol{Sen}(\Sigma)$. Then the propositions ϕ, ψ are called subalternate, if the truth of the first (the "superaltern") implies the truth of the second ("the subaltern"), but not conversely.

$$\models^{\mathcal{I}}_{\Sigma} \phi \to \psi \text{ and } \not\models^{\mathcal{I}}_{\Sigma} \psi \to \phi \tag{2.6}$$

We denote the relation between two subalternate sentences by $R_S(\phi, \psi)$ or by a graph

$$\phi \longrightarrow \psi$$

Figure 7. Geometrical representation of subalternate sentences.

Definition 2.3.5 (Boethian Diagram)**.** Let $\mathcal{I} = \left(\mathbb{S}ig^{\mathcal{I}}, \boldsymbol{Sen}^{\mathcal{I}}, \boldsymbol{Mod}^{\mathcal{I}}, \models^{\mathcal{I}}\right)$ be an arbitrary institution and $\Sigma \in \mathbb{S}ig$. Then a *Boethian diagram* is an edge-labeled graph. The vertices of the graph are pairwise non-equivalent sentences $e_1, e_2, \ldots, e_n \in \boldsymbol{Sen}(\Sigma)$ and the edges of the graph are the Aristotelian relations (see Figure 8).

$$\phi \longrightarrow \psi$$

(a) Relation of subalternation

$$\phi \cdots\cdots\cdots\cdots \psi$$

(b) Relation of subcontrariety

$$\phi \text{ - - - - - - - } \psi$$

(c) Relation of contrariety

$$\phi \text{ ―――――― } \psi$$

(d) Relation of contradictoriness

Figure 8. The fundamental Aristotelian relations.

As for Definitions 2.3.6 and 2.3.7 below, we should note that Definition 2.3.6 is the common traditional square of opposition. However, the shape in the second Definition 2.3.7 is introduced, as we will see, in a natural way so that we can see how the square of opposition changes from one logical system to another logical system. For this reason, we call it the rhombus of opposition.

Definition 2.3.6 (Aristotelian Square)**.** Let $\mathcal{I} = \left(\mathbb{S}ig^{\mathcal{I}}, \boldsymbol{Sen}^{\mathcal{I}}, \boldsymbol{Mod}^{\mathcal{I}}, \models^{\mathcal{I}}\right)$, $\Sigma \in \mathbb{S}ig$ be an arbitrary institution and $p, q \in \boldsymbol{Sen}(\Sigma)$. Then an *Aristotelian Square* is a graph of the following form:

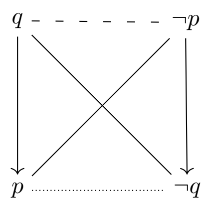


Figure 9. Aristotelian Square of Opposition

Definition 2.3.7 (Rhombus of Opposition)**.** Let $\mathcal{I} = \left(\mathbb{S}ig^{\mathcal{I}}, \boldsymbol{Sen}^{\mathcal{I}}, \boldsymbol{Mod}^{\mathcal{I}}, \models^{\mathcal{I}}\right)$, $\Sigma \in \mathbb{S}ig$ be an arbitrary institution and $p, q \in \boldsymbol{Sen}(\Sigma)$. Then a *rhombus of opposition* is

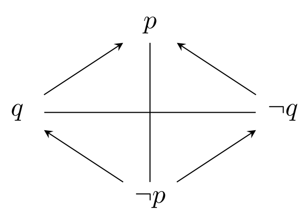


Figure 10. Aristotelian Rhombus of Opposition

2.3.2. The Example of PL – Square of opposition. In this subsection, we study the action of signature morphisms, i.e. how they affect the corresponding configurations.

The first example that we examine is that of propositional calculus. As shown in Figure 11 and Figure 12, after a signature morphism, the square structure is retained, if and only if the relationship remains unaltered; otherwise the square turns into a straight line segment. For the case of the straight line segment it is sufficient to imagine the possibility where $\sigma(p) = \sigma(q) = \chi$.

Fact 2.3.8. Let $\mathcal{I} = \left(\mathbb{S}ig^{\mathcal{I}}, \boldsymbol{Sen}^{\mathcal{I}}, \boldsymbol{Mod}^{\mathcal{I}}, \models^{\mathcal{I}}\right)$ be an institution with the traditional square, then for every $\sigma : \Sigma \to \Sigma'$ the square either remains invariant (Figure 11) or it turns into a line (Figure 12).

2.3.3. After the action of morphisms. In this section, we present several examples (shown in Figures 13, 14, 15 and 16) in which the square of opposition changes under the action of certain functors, i.e. we will illustrate how the square of opposition changes when we pass from one logical system to another.

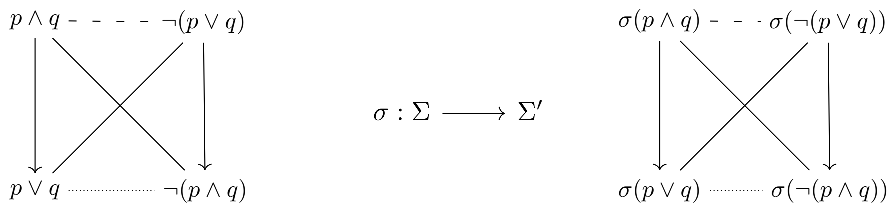


Figure 11. Square of opposition in institution with Boolean connectives

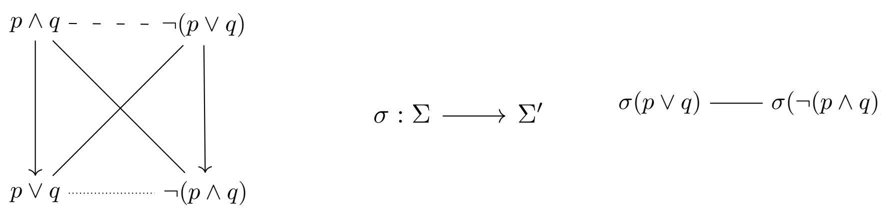


Figure 12. Line of opposition in institution with Boolean connectives

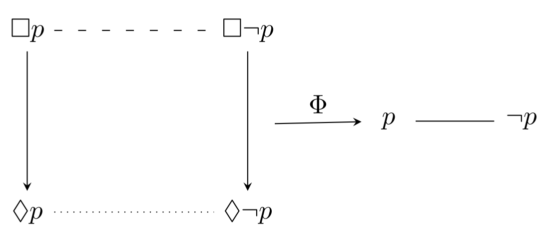


Figure 13. The modal system S5

The square of opposition of the modal logic S5 changes when the "forgetful" functor acts and assigns the shape to the primary logical system. In the case of the square of opposition, the configuration changes and becomes a straight-line segment, as the Sherwood-Czezowski Hexagon does. On the other hand, the Sesmat-Blanche and the Beziau hexagons become a rhombus. As we mentioned earlier, we can have a morphism $\Phi : \boldsymbol{FOL} \mapsto \boldsymbol{MFOL}$ which represents the projection of Modal Logic into First Order Logic.

2.4. Institution-theoretic treatment of the square of opposition

In general, in the square of opposition we have a relation between two sentences. We have defined the relations of sentences $R_i(\phi, \psi)$ where i belongs to $\{C, c, S, s\}$. In order to pass to dual relations $R_i^*(\phi^*, \psi^*)$, we appeal to the concept of the Galois

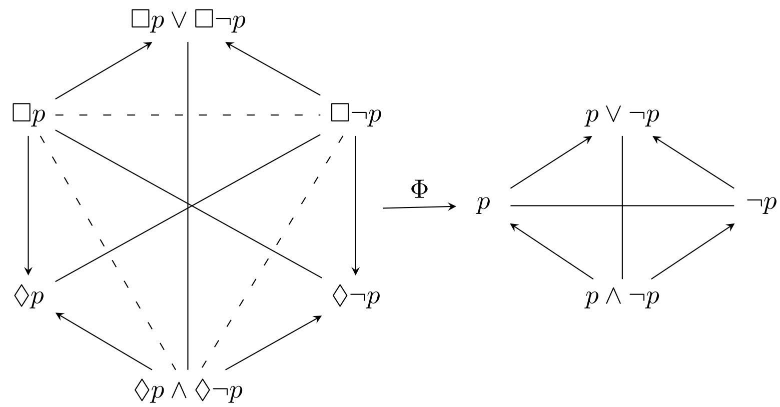


Figure 14. Sesmat-Blanche hexagon

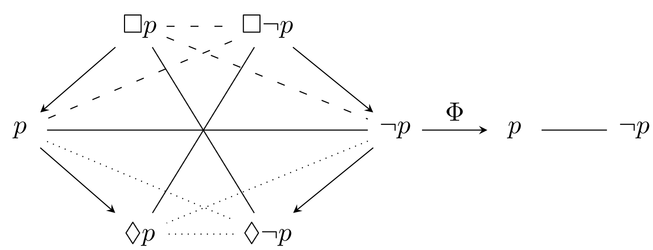


Figure 15. Sherwood-Czezowski hexagon

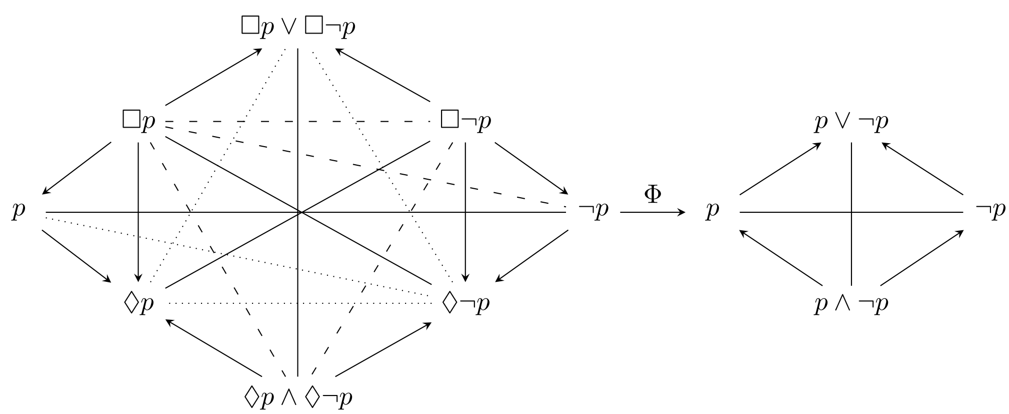


Figure 16. Beziau octagon for S5

connection, which is defined as follows:

$$* : R(\phi, \psi) \mapsto R^*(\phi^*, \psi^*) \tag{2.7}$$

Thus, the Galois connection forms the dual relation, as well as the dual square of opposition in a natural way [**42**, **17**].

2.4.1. Aristotelian Relations and the Galois connection. According to the previous section, we have four fundamental relations $R_i(\_,\_)$ where i belongs to $\{C, c, S, s\}$. In order to give an institution-independent form of these definitions using the Galois connection, we will first translate these relations in terms of the Galois connection.

1) $\models^{\mathcal{I}}_{\Sigma} \neg(\phi \wedge \psi)$. In terms of the Galois connection this means that:

$$\begin{aligned}
&\forall M \in \mathbf{Mod}(\Sigma)\left(M \models^{\mathcal{I}}_{\Sigma} \neg\phi \text{ or } M \models^{\mathcal{I}}_{\Sigma} \neg\psi\right) \Leftrightarrow \\
&\forall M \in \mathbf{Mod}(\Sigma)\left(M \in \overline{\phi^*} \text{ or } M \in \overline{\psi^*}\right) \Leftrightarrow \\
&\overline{\phi^* \cap \psi^*} = \overline{\phi^*} \cup \overline{\phi^*} = \mathbf{Mod}(\Sigma)
\end{aligned} \tag{2.8}$$

2) $\models^{\mathcal{I}}_{\Sigma} \neg(\neg\phi \wedge \neg\psi)$. In terms of the Galois connection this means that:

$$\begin{aligned}
&\forall M \in \mathbf{Mod}(\Sigma)\left(M \models^{\mathcal{I}}_{\Sigma} \phi \text{ or } M \models^{\mathcal{I}}_{\Sigma} \psi\right) \Leftrightarrow \\
&\forall M \in \mathbf{Mod}(\Sigma)\left(M \in \phi^* \text{ or } M \in \psi^*\right) \Leftrightarrow \\
&\phi^* \cup \psi^* = \mathbf{Mod}(\Sigma)
\end{aligned} \tag{2.9}$$

3) $\not\models^{\mathcal{I}}_{\Sigma} \neg(\phi \wedge \psi)$. In terms of the Galois connection this means that:

$$\begin{aligned}
&\exists M \in \mathbf{Mod}(\Sigma):\ M \models^{\mathcal{I}}_{\Sigma} \phi \wedge \psi \Leftrightarrow \\
&\exists M \in \mathbf{Mod}(\Sigma):\ M \models^{\mathcal{I}}_{\Sigma} \phi \;\&\; M \models^{\mathcal{I}}_{\Sigma} \psi \Leftrightarrow \\
&\exists M \in \mathbf{Mod}(\Sigma):\ M \in \phi^* \;\&\; M \in \psi^* \Leftrightarrow \\
&\exists M \in \mathbf{Mod}(\Sigma):\ M \in \phi^* \cap \psi^* \Leftrightarrow \\
&\overline{\phi^*} \cup \overline{\phi^*} \subset \mathbf{Mod}(\Sigma) \Leftrightarrow \\
&\overline{\phi^* \cap \psi^*} = \overline{\phi^*} \cup \overline{\psi^*} \neq \mathbf{Mod}(\Sigma)
\end{aligned} \tag{2.10}$$

4) $\not\models^{\mathcal{I}}_{\Sigma} \neg(\neg\phi \wedge \neg\psi)$. In terms of the Galois connection this means that:

$$\begin{aligned}
&\exists M \in \mathbf{Mod}(\Sigma):\ M \models^{\mathcal{I}}_{\Sigma} \neg\phi \wedge \neg\psi \Leftrightarrow \\
&\exists M \in \mathbf{Mod}(\Sigma):\ M \in \overline{\phi^*}\ \&\ M \in \overline{\psi^*} \Leftrightarrow \\
&\exists M \in \mathbf{Mod}(\Sigma):\ M \in \overline{\phi^*} \cap \overline{\psi^*} \Leftrightarrow \\
&\exists M \in \mathbf{Mod}(\Sigma):\ M \in \overline{\phi^* \cup \psi^*} \Leftrightarrow \\
&\phi^* \cup \psi^* \subset \mathbf{Mod}(\Sigma) \Leftrightarrow \\
&\phi^* \cup \psi^* \neq \mathbf{Mod}(\Sigma) \qquad (2.11)
\end{aligned}$$

We should note that in the initial definition we talked about relations between sentences. However, by introducing the concept of the Galois Connection, we will now talk now about relations between collections of models. Then, applying again the concept of Galois Connection, we pass to collections of sentences, i.e. essentially to relations of sentences again. Thus, in terms of relations we have the following scheme:

$$R(\phi,\psi) \xrightarrow{\ *\ } R^*(\phi^*,\psi^*) \xrightarrow{\ *\ } R^{**}(\phi^{**},\psi^{**}) \qquad (2.12)$$

This scheme is transferred in a natural way to the square's schemes. According to the following definitions we have:

Definition 2.4.1. Two sets of models ϕ^*, ψ^* are in dual contradictory relation $R^*_C(\phi^*,\psi^*)$ if $\overline{\phi^* \cap \psi^*} = \overline{\phi^*} \cup \overline{\psi^*} = \mathbf{Mod}(\Sigma)$ and $\phi^* \cup \psi^* = \mathbf{Mod}(\Sigma)$ which is equivalent to

$$\overline{\phi^*} = \psi^* \qquad (2.13)$$

Definition 2.4.2. Two sets of models ϕ^*, ψ^* are in a dual contrary relation $R^*_c(\phi^*,\psi^*)$ if

$$\overline{\phi^* \cap \psi^*} = \overline{\phi^*} \cup \overline{\psi^*} = \mathbf{Mod}(\Sigma) \text{ and } \overline{\phi^*} \cap \overline{\psi^*} \neq \emptyset \qquad (2.14)$$

Definition 2.4.3. Two sets of models ϕ^*, ψ^* are in dual subcontrary relation $R^*_s(\phi^*,\psi^*)$ if

$$\phi^* \cup \psi^* = \mathbf{Mod}(\Sigma) \text{ and } \phi^* \cap \psi^* \neq \emptyset \qquad (2.15)$$

Definition 2.4.4. Two sets of models ϕ^*, ψ^* are in dual subalternate relation $R^*_S(\phi^*,\psi^*)$ if

$$\phi^* \subset \psi^* \qquad (2.16)$$

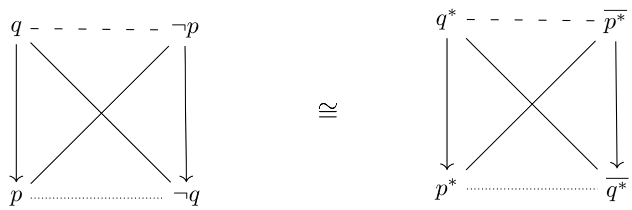


Figure 17. Transformation of square

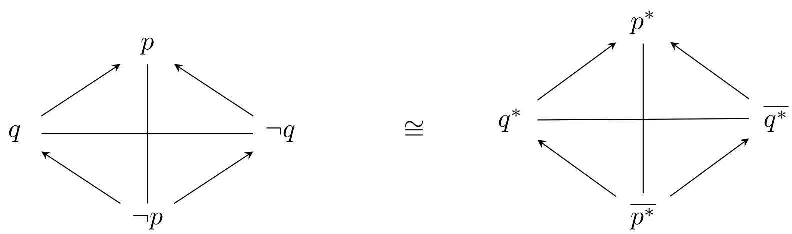


Figure 18. Transformation of rhombuses

2.4.2. The dual square of opposition. The scheme above is transferred in a natural way to the dual square's schemes.

Definition 2.4.5. Two sets of sentences ϕ^{**}, ψ^{**} are in a dual dual contradictory relation $R_C^{**}(\phi^{**}, \psi^{**})$, if $R_C^{*}(\phi^{***}, \psi^{***})$;
Two sets of sentences ϕ^{**}, ψ^{**} are in a dual dual contrary relation $R_c^{**}(\phi^{**}, \psi^{**})$, if $R_c^{*}(\phi^{***}, \psi^{***})$;
Two sets of sentences ϕ^{**}, ψ^{**} are in a dual dual subcontrary relation $R_s^{**}(\phi^{**}, \psi^{**})$, if $R_s^{*}(\phi^{***}, \psi^{***})$;
Two sets of sentences ϕ^{**}, ψ^{**} are in a dual dual subalternate relation $R_S^{**}(\phi^{**}, \psi^{**})$, if $R_S^{*}(\phi^{***}, \psi^{***})$

We know that $E^{*} = E^{***}$ and $\mathbb{M}^{*} = \mathbb{M}^{***}$. Therefore, we can obtain the following generalization for abstract set of models and sentences.

Definition 2.4.6. Two sets of models $\mathbb{D}, \mathbb{E}$ are in a dual contradictory relation $R_C^{*}(\mathbb{D}, \mathbb{E})$, if

$$\overline{\mathbb{D} \cap \mathbb{E}} = \overline{\mathbb{D}} \cup \overline{\mathbb{E}} = \mathbf{Mod}(\Sigma) \text{ and } \mathbb{D} \cup \mathbb{E} = \mathbf{Mod}(\Sigma) \tag{2.17}$$

Two sets of models $\mathbb{D}, \mathbb{E}$ are in a dual contrary relation $R_c^*(\mathbb{D}, \mathbb{E})$, if

$$\overline{\mathbb{D} \cap \mathbb{E}} = \overline{\mathbb{D}} \cup \overline{\mathbb{E}} = \mathbf{Mod}(\Sigma) \text{ and } \overline{\mathbb{D}} \cap \overline{\mathbb{E}} \neq \emptyset \tag{2.18}$$

Two sets of models $\mathbb{D}, \mathbb{E}$ are in a dual subcontrary relation $R_s^*(\mathbb{D}, \mathbb{E})$, if

$$\mathbb{D} \cup \mathbb{E} = \mathbf{Mod}(\Sigma) \text{ and } \mathbb{D} \cap \mathbb{E} \neq \emptyset \tag{2.19}$$

Two sets of models $\mathbb{D}, \mathbb{E}$ are in a dual subalternate relation $R_S^*(\mathbb{D}, \mathbb{E})$, if

$$\mathbb{D} \subset \mathbb{E} \tag{2.20}$$

Two sets of sentences D, E are in a dual dual contradictory relation $R_C^{**}(D, E)$, if their duals D^*, E^* are in a dual contradictory relation $R_C^*(D^*, E^*)$;
Two sets of sentences D, E are in a dual dual contrary relation $R_c^{**}(D, E)$, if their duals D^*, E^* are in a dual contrary relation $R_c^*(D^*, E^*)$;
Two sets of sentences D, E are in a dual dual subcontrary relation $R_s^{**}(D, E)$, if their duals D^*, E^* are in a dual subcontrary relation $R_s^*(D^*, E^*)$;
Two sets of sentences D, E are in a dual dual subalternate relation $R_S^{**}(D, E)$, if their duals D^*, E^* are in a dual subalternate relation $R_S^*(D^*, E^*)$;

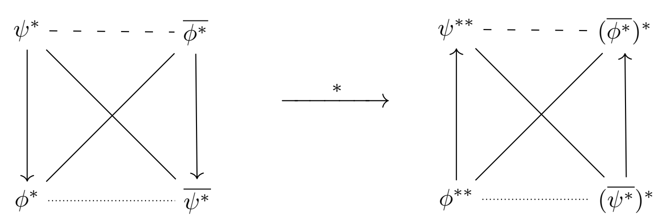


Figure 19. Dual square of opposition

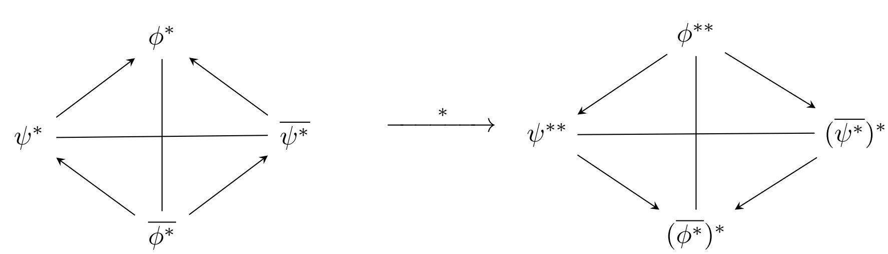


Figure 20. Dual rhombus of opposition

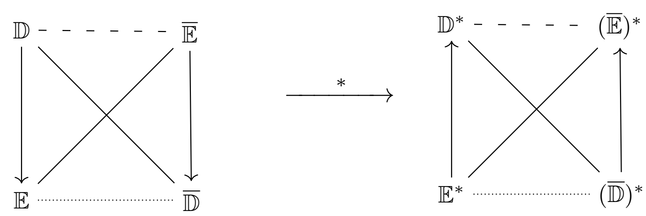


Figure 21. Generalized dual square of opposition

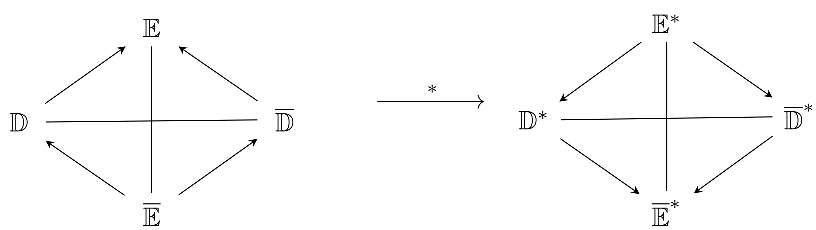


Figure 22. Generalized dual rhombus of opposition

2.5. Conclusions

In this Chapter we examined the transformations of the logical object conventionally called "square of opposition" undergoes under changes of semantics. For this reason, we appealed to concepts from category theory and the theory of institutions. By introducing the concept of the rhombus of opposition we examined the basic cases of configuration changes of the 'squares' of opposition inside a logical system and between different logical systems.

Furthermore, by introducing the concept of the Galois connection we showed the equilibrium that can be established between the sentences of the traditional square of opposition and the internal semantics of Boolean connectives, using them at a meta-level. Furthermore, the introduction of the concept of the dual square enabled us to examine not only squares for propositions but also squares for sets of sentences.

Since quite a few logical systems do not have internal connectives, it not useful to talk about proposition graphs, but about sets of models and sets of sentences. Therefore, we can now write basic relationships, such as contradiction.

This is the first time that different 'squares' of opposition were compared by using abstract model theory. We aim at integrating the different versions of the 'square' of opposition into this universal framework.

Chapter 3

Topological Semantics and Institutions

3.1. Introduction

ABSTRACT. In this chapter, we will introduce the notion of topological semantics in the framework of abstract model theory through institution-independent theory. This task will provide us with a method for expanding established models to topological models of any logic system present in Logic as well as computer science. Furthermore, we will examine the properties of semantic topological spaces according to the Kolmogorov classification [74, 71, 72]. The methodology we will present equips us with the appropriate tools in order to study semantic completeness through topological notions. Finally, we will prove a fundamental preservation result for abstract topo-modal satisfaction.

In "Logical topologies and Semantic Completeness" V. Goranko [68] set the foundation for a different kind of connection between logic and topology. He proposed a novel topological approach which allowed for studying the semantic completeness of a logical system with respect to one family of models taking into account that completeness is valid for a different family of models. More specifically, he proposed a topological approach to prove the semantic completeness of a logical system with respect to a class of "standard models" by utilizing the completeness result with respect to a larger class of "general models". He stated that:

Proposition 3.1.1 (Goranko)**.** *If the following two conditions hold for some topology $\mathcal{T}$ of general models, then the completeness of general models implies completeness with respect to standard models.*[1]

- *The class of standard models is dense to general models with respect to $\mathcal{T}$*
- *Validity is a continuous property with respect to $\mathcal{T}$.*

In First order logic (FOL) we can note all the models of the FOL as "general models" and the "standard models" to be only the finite models. Then we have the well known result of completeness with respect to general models and the failure of the completeness with respect to standard models. Goranko has pointed out that there is no general method for solving the problem described above, but usually some specific model-theoretic constructions which transform general into standard models while preserving satisfiability are applied. Furthermore, he has pointed out that standanrd topological methods and results have so far been under-utilized for solving purely logical problems. Our goal is to establish an appropriate framework for all of the above within an axiomatic setting. For this reason, we appeal to Institution-independent model theory [**24**].

Building on the expressive power provided by category theory we introduce the concept of topological semantics at the level of abstract model theory provided by the institution-independent framework. Our first step was to construct an abstract topological logic framework which will provide a method for systematic topological semantics extensions of logical systems from computer science and logic. The ultimate goal of this framework is to equip us with several appropriate model theoretical tools for proving semantic completeness on arbitrary Institutions via topological approach.

3.1.1. Topologic and Possible Worlds in institution-independent model theory. The starting point in standard logic is McKinsey and Tarski's topological interpretation of modal logic [**45, 44**] which introduces the connection between logic and topological space through studying the laws of the basic topological operators. Based on the modern truth-conditional format, the basic language $\mathcal{L}$ consists of a countable set P of proposition variables, the standard Boolean connectives, the modal operators $\Box$, $\Diamond$ and the topo-model i.e. a topological space $\langle X, \tau \rangle$ equipped

[1] Goranko V., Logical Topologies and Semantic Completeness

with a value function $\nu : P \to \mathcal{P}(X)$. This field saw rapid development and new tools like bisimulation and topo-games came to be added [3]. The perspective of modal logic has changed; it is crystal clear that we can deal with several geometrical structures like affine spaces, metric spaces, vector-based spaces etc. [4]

In 1938 Tarski [60] proved that S4 is complete with respect to topological spaces. In 1944 Mckinsey and Tarski [45] showed that S4 is the modal logic of real numbers. A number of interesting results can be found in several articles [21, 69] A different approach can be found in the works of Lawvere [22] and Goldblatt [28].

In Universal Logic, a different approach of logical systems is presented by Lewitzka in [[55], [54]] where he constructs a theory of logical representations (a logic map) to make use of the fact that every logical system can define a topology within its theory set.

Institution theory [24] is an important trend within so-called Universal Logic [7]. It is a categorical abstract model theory which formalizes the notion of a logical system, including syntax, semantics and the satisfaction relation between them. One of the many achievements of Institution theory has been to provide a conceptually elegant and unifying definition of the nature of logical systems [24]. It provides a complete form of abstract model theory, including signature morphisms, model reducts and mappings between logics noted as Institution-independent model theory.

From the Universal Logic view, Institution-independent model theory means the development of model theory in the very abstract setting of arbitrary institutions, which provides an efficient framework for doing model theory by translation [17] or borrowing via a mapping theory (homomorphisms) between institutions.

Possible worlds semantics or Kripke semantics is a classical development in the area on non-classical logics. In addition to their singnificance on logic, Kripke semantics have been applied to computer science and AI. In [16] and [18] the authors have developed the satisfaction of standard modalities possibility $\Diamond$ and necessity $\Box$ on top of an abstract satisfaction relation. This is the first work which has provided a method for systematic Kripke semantic extension at the categorical abstract model theoretic level provided by institutions.

3.1.2. Universal Topologic. Our aim is to construct topologies containing models of a logical system as their elements. The following definition, cited from [**55**, **54**, **16**], is what we need to begin our construction.

Theorem 3.1.2. *Let* Σ *be a signature of an Institution* $\mathcal{I} = \left(\mathbb{S}ig^{\mathcal{I}}, \boldsymbol{Sen}^{\mathcal{I}}, \boldsymbol{Mod}^{\mathcal{I}}, \models^{\mathcal{I}}\right)$*, then if we define for every set of* Σ*-sentences* E

$$E^* = \{M \in Mod(\Sigma) \mid M \models_\Sigma e \quad \forall e \in E\}$$

then the class $|Mod(\Sigma)|$ *of all* Σ*-models admits a natural topology, where the open sets are*

$$\tau_\Sigma = \left\{ \bigcup_{i \in I} E_i^* \mid \{E_i\}_{i \in I} \quad \textit{family of finite sets of } \Sigma\textit{-sentences} \quad \right\}$$

Making use of the above definition and of the possibilities afforded through the abstract level of institution-independent model theory, we may start to explore model topologies according to the Kolmogorov classification [**74**].

Definition 3.1.3. Let Σ be a signature of an institution $\mathcal{I} = \left(\mathbb{S}ig^{\mathcal{I}}, \boldsymbol{Sen}^{\mathcal{I}}, \boldsymbol{Mod}^{\mathcal{I}}, \models^{\mathcal{I}}\right)$ and M_1 and M_2 two models in $|\mathbf{Mod}(\Sigma)|$ such that for every $\phi \in \boldsymbol{Sen}(\Sigma)$

$$M_1 \models \phi \;\Leftrightarrow\; M_2 \models \phi$$

then $\mathcal{T}h(M_1) = \mathcal{T}h(M_2)$. Models M_1 and M_2 are equivalent, denoted as $M_1 \sim M_2$, if $\mathcal{T}h(M_1) = \mathcal{T}h(M_2)$.

Definition 3.1.4. Let Σ be a signature of an institution $\mathcal{I} = \left(\mathbb{S}ig^{\mathcal{I}}, \boldsymbol{Sen}^{\mathcal{I}}, \boldsymbol{Mod}^{\mathcal{I}}, \models^{\mathcal{I}}\right)$ and $\mathbf{Mod}(\Sigma)$ the class of its models; we define the class invariant under the relation $\sim$.

$$|\mathbf{Mod}(\Sigma)|/\sim$$

The next corollary comes naturally.

Proposition 3.1.5. *Let* Σ *be a signature of an institution* $\mathcal{I} = \left(\mathbb{S}ig^{\mathcal{I}}, \boldsymbol{Sen}^{\mathcal{I}}, \boldsymbol{Mod}^{\mathcal{I}}, \models^{\mathcal{I}}\right)$ *and* F *be the function which maps each point to its equivalence class*

$$F : |\mathbf{Mod}(\Sigma)| \to |\mathbf{Mod}(\Sigma)|/\sim$$

Then F *is continuous and defines the* ***Identification Semantic Topology (ISM)****.*

Theorem 3.1.6. *Let* Σ *be a signature of an institution* $\mathcal{I} = \left(\mathbb{S}ig^{\mathcal{I}}, \boldsymbol{Sen}^{\mathcal{I}}, \boldsymbol{Mod}^{\mathcal{I}}, \models^{\mathcal{I}}\right)$ *with negation and conjunction. Then the* ***ISM*** *topology is* T_2 *topology.*

Proof. We want to prove that for every $M_1 \neq M_2 \in |\mathbf{Mod}(\Sigma)|/\sim$ two disjoint open sets V_1, V_2 exist such that $M_1 \in V_1$ and $M_2 \in V_2$. Indeed, $M_1 \neq M_2$ implies that $Th(M_1) \neq Th(M_2)$ which means that there exists $\phi \in \boldsymbol{Sen}(\Sigma)$ such that

$$M_1 \models_\Sigma \phi \quad \text{and} \quad M_2 \models_\Sigma \neg\phi$$

That means

$$M_1 \in \{\phi\}^* = V_1 \quad \text{and} \quad M_2 \in \{\neg\phi\}* = V_2$$

with $V_1 \cap V_2 = \emptyset$

□

Theorem 3.1.7. *Let Σ be a signature of an institution $\mathcal{I} = \left(\mathbb{S}ig^{\mathcal{I}}, \boldsymbol{Sen}^{\mathcal{I}}, \boldsymbol{Mod}^{\mathcal{I}}, \models^{\mathcal{I}}\right)$ with negation and conjunction. Then the **ISM** topology is a regular topology.*

Proof. To understand the methodology, we will present two different situations.
First, let $x \in X$ be a point and $F \in X$ a closed set, such that $x \notin F$.
Let $X = |Mod(\Sigma)/\sim$ then if E^* and if $F = X \setminus E^*$ is closed and

$$M \in F \Leftrightarrow \exists \phi \in E : M \not\models_\Sigma \phi \Leftrightarrow \exists \phi \in E : M \models \neg\phi$$

Therefore, $M \in \{\neg\phi\}^*$. Therefore, an open set exists, namely $\{\neg\phi\}^*$ such that $M \in \{\neg\phi\}^*$.
Now, for all $M \in F$ $\exists \phi_M \in E : M \not\models_\Sigma \phi_M$; therefore, we set

$$V = \bigcup_{M \in F} \{\neg\phi_M\}*$$

and $F \subseteq V$.
Now $x \notin F$ implies that

$$x \notin X \setminus E^* \Leftrightarrow x \in E^* \Leftrightarrow x \models_\Sigma \phi \ \forall \phi \in E$$

which implies

$$x \in \bigcup_{M \in F} \{\phi_M\}^* = U$$

and $V \cap U = \emptyset$.
If $F = X \setminus \bigcup_{i \in I} E_i^* = \bigcap_{i \in I} X \setminus E_i^*$ then

$$M \in F \Leftrightarrow M \in \bigcap_{i \in I} X \setminus E_i^* \Leftrightarrow \forall i \in I \ M \in X \setminus E_i^* \Leftrightarrow$$

$$\forall i \in I \ \exists \phi_i^M \in E_i : M \models_\Sigma \neg\phi_i^M$$

we can set

$$V = \bigcup_{M \in F} \{\neg\phi_i^M\}^*$$

and

$$x \notin F \Leftrightarrow x \notin X \setminus \bigcup_{i \in I} E_i^* \Leftrightarrow x \notin \bigcap_{i \in I} X \setminus E_i^* \Leftrightarrow$$
$$\exists j : x \notin X \setminus E_j^* \Leftrightarrow \exists j : x \models_\Sigma \phi \forall \phi \in E_j$$

Therefore, there exists a $\phi \in E_j$ such that

$$x \in \{\phi\}^* = U$$

and $V \cap U = \emptyset$ □

Theorem 3.1.8. *Let Σ be a signature of an institution $\mathcal{I} = \left(\mathbb{S}ig^{\mathcal{I}}, \boldsymbol{Sen}^{\mathcal{I}}, \boldsymbol{Mod}^{\mathcal{I}}, \models^{\mathcal{I}}\right)$ with negation and conjunction. Then **ISM** topology is a normal topology.*

Proof. Let F_1 and F_2 be two closed sets; the goal is to find two disjoint open sets V_1 and V_2 such that

$$F_1 \subseteq V_1 \text{ and } F_2 \subseteq V_2 \text{ and } V_1 \cap V_2 = \emptyset$$

If $F_1 = X \setminus \bigcup_{i \in I} E_i^* = \bigcap_{i \in I} X \setminus E_i^*$ then

$$M \in F_1 \Leftrightarrow M \in \bigcap_{i \in I} X \setminus E_i^* \Leftrightarrow \forall i \in I \; M \in X \setminus E_i^* \Leftrightarrow$$
$$\forall i \in I \; \exists \phi_i^M \in E_i : M \models_\Sigma \neg\phi_i^M$$

we can set

$$V_1 = \bigcup_{M \in F_1} \{\neg\phi_i^M\}^*$$

Accordingly, if $N \in F_2$ then $N \notin F_1$ therefore there exists $\phi_i^N \in \bigcup_{i \in I} E_i^*$ such that $N \models_\Sigma$. We define

$$V_2 = \bigcup_{N \in F_2} \{\phi_i^N\}^*$$

Then $V_1 \cap V_2 = \emptyset$ □

Summing up all of the above, we reach the following theorem, which constitutes the starting point of our work on topological semantics.

Theorem 3.1.9 (Entailment Topology)**.** *Let* $(\mathbb{S}ig, \boldsymbol{Sen}, Pf)$ *be an Institution with proofs (proof system), then there is a topology (**Entailment topology**) such that the entailment logic of the proof system is sound and complete with respect to its entailment topological semantic i.e.*

$$\phi \vdash \psi \iff \psi^* \subseteq \phi^*$$

Proof. We define the topology (W, τ_w) where open sets are:

$$E^* = \{(\psi, E') \in \mathbf{Mod} \mid \psi[E] \subseteq E'\}$$

One basis of the topology is

$$\rho^* = \{(\psi, E') \in \mathbf{Mod} \mid \psi(\rho) \in E'\}$$

The proof of soundness and completeness arises in a natural way from Theorem 1.3.23 and Proposition 1.3.22 . □

For the preceding theorem to be meaningful, a theory on how to study the notion of completeness by utilizing topological tools needs to be constructed. We will set the basis of that theory in the following section.

3.2. Topological Semantics

Internal Topological Models. In this section we begin our attempt to construct a universal topological theory of logical systems. In order to achieve this, our first objective is to construct a (modal) topological institutional $\mathcal{I}_{\mathcal{T}} = \left(\mathbb{S}ig^{\mathcal{I}_{\mathcal{T}}}, \boldsymbol{Sen}^{\mathcal{I}_{\mathcal{T}}}, \mathbb{T}\text{-}\mathbf{Mod}^{\mathcal{I}_{\mathcal{T}}}, \models\right)$ from an old one $\mathcal{I} = \left(\mathbb{S}ig^{\mathcal{I}}, \boldsymbol{Sen}^{\mathcal{I}}, \boldsymbol{Mod}^{\mathcal{I}}, \models^{\mathcal{I}}\right)$ where $\mathbb{S}ig^{\mathcal{I}} = \mathbb{S}ig^{\mathcal{I}_{\mathcal{T}}}$. Our construction consists of several components:

(1) An extension of institution $\mathcal{I}$. The signatures remain the same, but new sentences are built from the sentences of $\mathcal{I}$ by approximation of sentences building Boolean connectives, quantifiers and modalities;

(2) Topological models built from the models of institution $\mathcal{I}$;

(3) The definition of a new modal satisfaction relation between the topological models and the new sentences;

The concept of a topological model can be defined internally to any "base" institution $\mathcal{I} = \left(\mathbb{S}ig^{\mathcal{I}}, \boldsymbol{Sen}^{\mathcal{I}}, \boldsymbol{Mod}^{\mathcal{I}}, \models^{\mathcal{I}}\right)$ providing the base models of the topological models, the sharing parameter being handled by a forgetful institution morphism $\left(\Phi^{\Delta}, \alpha^{\Delta}, \beta^{\Delta}\right)$ to a domain institution Δ providing the shared domains. Our methodology consists of three steps: The first step is to define the *Internal Topological Models*; the second step is to define a new functor $\mathbb{T}$-**Mod** which maps the "old" signatures $\mathbb{S}ig$ to a new category of models, the *topological-models*; finally, the third step is to define a functor $\boldsymbol{Sen}^{\mathcal{I}_{\mathcal{T}}}$ which will extend the "old" sentences to the new one and all of them with respect to the "old" $\mathcal{I} = \left(\mathbb{S}ig^{\mathcal{I}}, \boldsymbol{Sen}^{\mathcal{I}}, \boldsymbol{Mod}^{\mathcal{I}}, \models^{\mathcal{I}}\right)$.

We will start with the definition of *Internal Topological models*, an extension of the models of the initial-base Institution inspired by [**18**].

Definition 3.2.1 (Internal Topological Models)**.** Let $\left(\Phi^{\Delta}, \alpha^{\Delta}, \beta^{\Delta}\right) : \left(\mathbb{S}ig^{\mathcal{I}}, \boldsymbol{Sen}^{\mathcal{I}}, \mathbf{Mod}^{\mathcal{I}}\right) \to \Delta$ be an institution morphism from the base Institution to the Domain Institution, then for any Σ signature in $\mathbb{S}ig$, a Topological Σ-model $(\mathcal{T}, \tau)_{\Sigma}$ consists of

- a family of Σ-models $\mathcal{T} : I_{\mathcal{T}} \to |\mathbf{Mod}(\Sigma)|$ such that $\beta^{\Delta}_{\Sigma}(M_i) = \beta^{\Delta}_{\Sigma}(M_j)$ for every $i, j \in I_{\mathcal{T}}$
- τ a topology on $\mathcal{T}$

where $I_{\mathcal{T}}$ is an index set.

Remark 3.2.2. Henceforth we will use the following $\mathcal{T} = \bigcup_{i \in I_{\mathcal{T}}} M_i$ with $M_i \in |\mathbf{Mod}(\Sigma)|$;

Remark 3.2.3. Condition $\beta^{\Delta}_{\Sigma}(M_i) = \beta^{\Delta}_{\Sigma}(M_j)$ for every $i, j \in I_{\mathcal{T}}$ in definition 3.2.1 stems from the need to express that the models share something common at the level of abstract institution. For example, in the case of **MFOL** it is necessary for the models M_k to have the same underlying set; this is because we want quantification to work properly.

Remark 3.2.4. Let Σ be a signature, then the collection of the topological Σ-models form a category, namely the $\mathbb{T}$-$\mathbf{Mod}(\Sigma)$ category.

Morphisms of topological models. A topological Σ-model $(\mathcal{T}_1, \tau_1)$ is a topology, therefore it is natural to define Σ-model homomorphisms between two topological models in such a way as to preserve the mathematical structure of topological models. A homomorphism of Σ-topological models $(h, g) : (\mathcal{T}_1, \tau_1) \mapsto (\mathcal{T}_2, \tau_2)$ consists of :

- A function $h : I_{\mathcal{T}_1} \mapsto I_{\mathcal{T}_2}$ between the index sets such that the function $\gamma : \mathcal{T}_1 \mapsto \mathcal{T}_2$ defined by $\gamma(M_i^1) = \gamma(\mathcal{T}_1(i)) = \mathcal{T}_2(h(i)) = M_{h(i)}^2$ is a continuous function;
- An $I_{\mathcal{T}_1}$ index family of Σ-model homomorphisms $g = \{g_i : M_i^1 \to M_{h(i)}^2\}$. Family g can be regarded as a natural transformation $g : \mathcal{T}_1 \Rightarrow h; \mathcal{T}_2$ between functions (see Figure 1) – the functions can be regarded as functors $I_{\mathcal{T}_1} \mapsto |\mathbf{Mod}(\Sigma)|$ – such that $\beta_\Sigma^\Delta(g_i) = \beta_\Sigma^\Delta(g_j)$ for $i, j \in I_{\mathcal{T}_1}$;

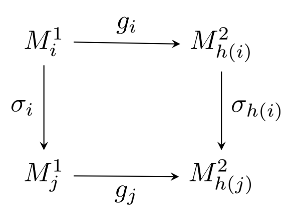


Figure 1

Remark 3.2.5. The topological Σ-models and their morphisms form a category labeled as $\mathbb{T}$-$\mathbf{Mod}(\Sigma)$.

This was the first step in the construction of the $\mathbb{T}$-**Mod** functor, as we just proved that the $\mathbb{T} - \mathbf{Mod}(\Sigma)$ forms a category. Now we must define the functor with respect to the arrows of $\mathbb{S}ig^{op}$.

Definition 3.2.6 ($\mathbb{T}$-**Mod** funtor)**.** Functor $\mathbb{T}$-**Mod** maps the $\phi : \Sigma \to \Sigma'$ to $\mathbb{T}$-$\mathbf{Mod}(\phi) : \mathbb{T}$-$\mathbf{Mod}(\Sigma') \to \mathbb{T}$-$\mathbf{Mod}(\Sigma)$.

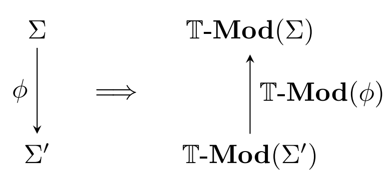


Figure 2

in such a way that :

- the functor maps each $(h', g') : (\mathcal{T}'_1, \tau'_1) \to (\mathcal{T}'_2, \tau'_2)$ to $\mathbb{T}\text{-}\mathbf{Mod}(\phi)(h, g) : \mathbb{T}\text{-}\mathbf{Mod}(\phi)\,(\mathcal{T}'_1, \tau'_1) \to \mathbb{T}\text{-}\mathbf{Mod}(\phi)\,(\mathcal{T}'_2, \tau'_2)$ which is defined as: $\mathbb{T}\text{-}\mathbf{Mod}(\phi)\,(\mathcal{T}'_i, \tau'_i) = (\mathcal{T}'_i ; \mathbf{Mod}(\phi), \mathbf{Mod}(\phi)\,[\tau'_i]) = (\mathcal{T}_i, \tau_i)$ where

$$
\begin{aligned}
\mathcal{T}_i &= \bigcup_{M \in \mathcal{T}_i} \{M\} = \\
&= \mathcal{T}'_i ; \mathbf{Mod}(\phi) = \\
&= \bigcup_{M' \in \mathcal{T}'_i} \{\mathbf{Mod}(\phi)(M')\} = \\
&= \bigcup_{k \in I_{\mathcal{T}'_i}} \left\{\mathbf{Mod}(\phi)(I_{\mathcal{T}'_i}(k))\right\} = \\
&= \bigcup_{k \in I_{\mathcal{T}'_i}} \{M_k\}
\end{aligned}
\tag{3.1}
$$

From equation 3.1 we can deduce that for every $M^i_k \in I_{\mathcal{T}_i}$ $M^i_k = \mathbf{Mod}(\phi)(I_{\mathcal{T}_i}(k))$ holds. This implies that we can define $I_{\mathcal{T}_i} = I_{\mathcal{T}'_i}$ for $i \in \{1, 2\}$ and $\mathbf{Mod}(\phi)\,[\tau'_i]$ as the minimum topology where map $\mathbf{Mod}(\phi)$ is open and continuous.

- $\mathbb{T}\text{-}\mathbf{Mod}(\phi)(h', g') = (h, g) = (h', \mathbf{Mod}(\phi)g')$ such that the square in Figure 3 is commutive

$$
\begin{array}{ccc}
\mathbf{Mod}(\phi)(M^1_i) & \xrightarrow{\mathbf{Mod}(\phi)g'_i} & \mathbf{Mod}(\phi)(M^2_{h(i)}) \\
\Big\downarrow {\scriptstyle \mathbf{Mod}(\phi)(\sigma_i)} & & \Big\downarrow {\scriptstyle \mathbf{Mod}(\phi)(\sigma_{h(i)})} \\
\mathbf{Mod}(\phi)(M^1_j) & \xrightarrow[\mathbf{Mod}(\phi)g'_j]{} & \mathbf{Mod}(\phi)(M^2_{h(j)})
\end{array}
$$

Figure 3

Fact 3.2.7. Function $\gamma : \mathcal{T}_1 \mapsto \mathcal{T}_2$ is a continuous function.

Proof. Let $V \subseteq \mathcal{T}_2$ be an open subset. Then

$$\begin{aligned} V &= \bigcup_{k \in \mathcal{T}_2^{-1}[V]} \left\{ M_k^2 \right\} = \\ &= \bigcup_{k \in J} \left\{ \mathbf{Mod}(\phi)(N_k^2) \right\} \end{aligned} \tag{3.2}$$

such that $\mathcal{T}_2^{-1}[V] = J$ and $N_k^2 \in \mathcal{T}_2'$. For the inverse image we have:

$$\begin{aligned} \gamma^{-1}[V] &= \left\{ M \in \mathcal{T}_1 \ : \ \gamma(M) = M_k^2 \text{ for some } k \in J \right\} = \\ &= \gamma^{-1} \left[\bigcup_{k \in J} M_k^2 \right] = \\ &= \bigcup_{k \in J} \gamma^{-1}[M_k] = \bigcup_{k \in J} \bigcup_{\lambda \in h^{-1}[k]} M_\lambda = \bigcup_{\lambda \in h^{-1}[J]} \{Y_\lambda\} \end{aligned} \tag{3.3}$$

Figure 4 helps us understand how to prove that $\gamma^{-1}[V]$ is an open set.

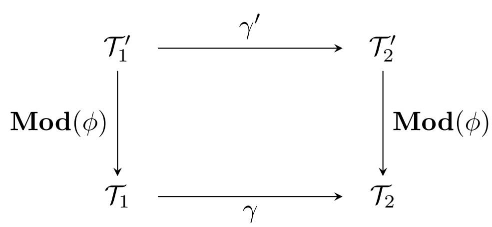


Figure 4

By definition of $\mathbf{Mod}(\phi)[\tau_i']$ set $\mathbf{Mod}^{-1}(\phi)[V] = \{N_k \in \mathcal{T}_2' \ : k \in J\}$ is an open set and

$$\gamma^{-1} \left[\mathbf{Mod}^{-1}(\phi)[V] \right] = \left\{ X \in \mathcal{T}_1' \ : \ \gamma(X) = N_k^2 \text{ for some } k \in J \right\} \tag{3.4}$$

is also an open set. The final step is to observe that $\mathbf{Mod}(\phi)$ is an open map, so the set

$$\begin{aligned} \mathbf{Mod}(\phi) \left[\bigcup_{k \in h^{-1}[J]} \{X_k\} \right] &= \{ Y \in \mathcal{T}_1 \ : \ Y_k \in \mathcal{T}_1 \ : \ Y_k = \mathbf{Mod}(\phi) X_k \} = \\ &= \bigcup_{k \in h^{-1}[J]} \{Y_k\} \end{aligned} \tag{3.5}$$

is also an open set. Hence, γ is a continuous function. □

The syntax of the topological institution is defined as follows:

Definition 3.2.8. Let $\left(\Phi^{\Delta}, \alpha^{\Delta}, \beta^{\Delta}\right) : \left(\mathbb{S}ig^{\mathcal{I}}, \boldsymbol{Sen}^{\mathcal{I}}, \mathbf{Mod}^{\mathcal{I}}\right) \to \Delta$ be an institution morphism from the base Institution to the Domain Institution, then we can extend the $\boldsymbol{Sen}^{\mathcal{I}}$ to a topological sentence functor $\boldsymbol{Sen}^{\mathcal{I}_{\mathcal{T}}} : \mathbb{S}ig \to \mathbb{S}et$ such that each $\boldsymbol{Sen}^{\mathcal{I}_{\mathcal{T}}}$ sentence is syntactically accessible from the base institution's sentences by:

- Boolean connectives;
- Topological Modalities $\Box, \Diamond$;
- $\mathcal{D}$-quantifiers, for a class $\mathcal{D}$ of signature morphisms stable under pushouts and such that any pushout between any morphism from $\mathcal{D}$ and any other signature morphism is an amalgamation square in the base Institution and gets mapped by Φ^{Δ} to an amalgamation square in the domain Institution.

Furthermore, we extend the new set of sentences in such a way so that the following holds: $\boldsymbol{Sen}(\phi)\,(\Box\rho) = \Box\boldsymbol{Sen}(\phi)(\rho)$;

Definition 3.2.8 could be described as follows:

(1) $\boldsymbol{Sen}^{\mathcal{I}}(\Sigma) \subseteq \boldsymbol{Sen}^{\mathcal{I}_{\mathcal{T}}}(\Sigma)$;

(2) $\phi \bullet \psi \in \boldsymbol{Sen}^{\mathcal{I}_{\mathcal{T}}}(\Sigma)$ for all $\phi, \psi \in \boldsymbol{Sen}^{\mathcal{I}_{\mathcal{T}}}(\Sigma)$ and for all $\bullet \in \{\Rightarrow, \wedge, \vee\}$;

(3) $\neg\phi \in \boldsymbol{Sen}^{\mathcal{I}_{\mathcal{T}}}(\Sigma)$ for all $\phi \in \boldsymbol{Sen}^{\mathcal{I}_{\mathcal{T}}}(\Sigma)$;

(4) $\bigstar\phi \in \boldsymbol{Sen}^{\mathcal{I}_{\mathcal{T}}}(\Sigma)$ for all $\bigstar \in \{\Box, \Diamond\}$ and for all $\phi \in \boldsymbol{Sen}^{\mathcal{I}_{\mathcal{T}}}(\Sigma)$;

(5) $(\forall\chi)\phi, (\exists\chi)\phi \in \boldsymbol{Sen}^{\mathcal{I}_{\mathcal{T}}}(\Sigma)$ for all $\phi \in \boldsymbol{Sen}^{\mathcal{I}_{\mathcal{T}}}(\Sigma)$ and $\chi : \Sigma \mapsto \Sigma' \in \mathcal{D}$;

To clarify, number (1) tells us that propositions in the base institution are also, in a natural way, propositions in the new modal institution. Numbers (2) and (3) describe how propositions in the (modal) topological institution are constructed, using Boolean connectors. Number (4) introduces modalities as topologic operators for the construction of sentences. Finally, number (5) demonstrates how to construct sentences with quantifiers. It is important to note here that quantifiers are utilized in an institution-theoretic manner. As noted in [14], not every signature morphism can serve as a quantifier. Therefore, those morphisms which can perform that function constitute quantification space $\mathcal{D}$.

Definition 3.2.9 (Topological Satisfaction)**.** Let $\mathcal{I}_{\mathcal{T}}$ be a topological institution and Σ a signature, then for every topological Σ-model $(\mathcal{T}, \tau)$ and each Σ sentence ϕ we define locally the satisfaction of $\boldsymbol{\phi}$ at the point $x \in \mathcal{T}$, denoted as $(\mathcal{T}, \tau) \models^{x}_{\Sigma} \phi$ as:

(1) $(\mathcal{T}, \tau) \models_{\Sigma}^{x} \phi$ if and only if $M_x \models_{\Sigma}^{\mathcal{I}} \phi$ whenever $\phi \in \boldsymbol{Sen}^{\mathcal{I}}(\Sigma)$;

(2) $(\mathcal{T}, \tau) \models_{\Sigma}^{x} \phi \wedge \psi$ if and only if $(\mathcal{T}, \tau) \models_{\Sigma}^{x} \phi$ and $(\mathcal{T}, \tau) \models_{\Sigma}^{x} \psi$ and similar for the others Boolean connectors in $\{\vee, \Rightarrow\}$;

(3) $(\mathcal{T}, \tau) \models_{\Sigma}^{x} \Box\phi$ if and only if $\exists \mathcal{O} \in \tau\, (x \in \mathcal{O}\ \&\ \forall y \in \mathcal{O}\ \ ((\mathcal{T}, \tau) \models_{\Sigma}^{y} \phi))$;

(4) $(\mathcal{T}, \tau) \models_{\Sigma}^{x} \Diamond\phi$ if and only if $\forall \mathcal{O} \in \tau\, (x \in \mathcal{O} \to \exists y \in \mathcal{O}\ y\, ((\mathcal{T}, \tau) \models^{y} \phi))$;

(5) $(\mathcal{T}, \tau) \models^{x} \forall\chi\phi$ if and only if $(\mathcal{T}', \tau') \models^{x} \phi$ for all χ expansion $(\mathcal{T}', \tau')$ of $(\mathcal{T}, \tau)$ such that $\mathbf{Mod}(\chi)\,(\mathcal{T}', \tau') = (\mathcal{T}, \tau)$;

(6) $(\mathcal{T}, \tau) \models^{x} \forall\chi\phi$ if and only if $(\mathcal{T}', \tau') \models^{x} \phi$ for all χ expansion $(\mathcal{T}', \tau')$ of $(\mathcal{T}, \tau)$ such that $\mathbf{Mod}(\chi)\,(\mathcal{T}', \tau') = (\mathcal{T}, \tau)$;

We can now define universal satisfaction as

$$(\mathcal{T}, \tau) \models^{\mathcal{I}_{\mathcal{T}}} \phi \text{ if and only if } (\mathcal{T}, \tau) \models^{x} \phi \text{ for all } x \in \mathcal{T}$$

Remark 3.2.10. It is important to note that $(\mathcal{T}, \tau) \models \neg\phi$ is not equivalent to $(\mathcal{T}, \tau) \not\models \phi$ because the latter means that there is an x such that $(\mathcal{T}, \tau) \not\models^{x} \phi$ and not that we have $(\mathcal{T}, \tau) \not\models^{x} \phi$ for all x which is equivalent to $(\mathcal{T}, \tau) \models \neg\phi$.

Before we prove our main theorem, we must prove the next fundamental theorem.

Theorem 3.2.11 (Topological Model Amalgamation)**.** *Let an institution morphism* $\left(\Phi^{\Delta}, \alpha^{\Delta}, \beta^{\Delta}\right) : (\mathbb{S}ig, \boldsymbol{Sen}, \mathbf{Mod}) \to \Delta$ *any commuting square of signature morphisms in* $\mathbb{S}ig$ *of signature morphisms in* $\mathbb{S}ig$ *such that:*

$$\begin{array}{ccc} \Sigma & \xrightarrow{\phi_1} & \Sigma_1 \\ {\scriptstyle \phi_2}\downarrow & & \downarrow{\scriptstyle \theta_1} \\ \Sigma_2 & \xrightarrow[\theta_2]{} & \Sigma' \end{array}$$

(1) *it is a model amalgamation square in the base institution;*

(2) Φ^{Δ} *maps it to a model amalgamation square in the domain institution,*

then it is a model amalgamation square with respect to the topological model functor $\mathbb{T}$-$\mathbf{Mod}$.

Proof. Let $(\mathcal{T}_1, \tau_1)$ be a Σ_1-topological model and $(\mathcal{T}_2, \tau_2)$ a Σ_2-topological model such that

$$\mathbb{T}\text{-}\mathbf{Mod}(\phi_1)\,(\mathcal{T}_1, \tau_1) = \mathbb{T}\text{-}\mathbf{Mod}(\phi_2)\,(\mathcal{T}_2, \tau_2) = (\mathcal{T}, \tau) \in \mathbb{T}\text{-}\mathbf{Mod}(\Sigma) \tag{3.6}$$

with $I_{\mathcal{T}'}$ index set. Then

$$(\mathcal{T}_1; \mathbf{Mod}(\phi_1), \mathbf{Mod}(\phi_1)[\tau_1]) = (\mathcal{T}_2; \mathbf{Mod}(\phi_2), \mathbf{Mod}(\phi_2)[\tau_2]) \Leftrightarrow$$
$$\mathcal{T}_1; \mathbf{Mod}(\phi_1) = \mathcal{T}_2; \mathbf{Mod}(\phi_2) \;\;\&\;\; \mathbf{Mod}(\phi_1)[\tau_1] = \mathbf{Mod}(\phi_2)[\tau_2] \Leftrightarrow$$
$$\bigcup_{M \in \mathcal{T}_1} \{\mathbf{Mod}(\phi)(M)\} = \bigcup_{M' \in \mathcal{T}_2} \{\mathbf{Mod}(\phi)(M')\} \;\;\&\;\; \mathbf{Mod}(\phi_1)[\tau_1] = \mathbf{Mod}(\phi_2)[\tau_2]$$

Given that the model amalgamation square holds in the base institution, we can define the new topological model $(\mathcal{T}', \tau') \in \mathbb{T}\text{-}\mathbf{Mod}(\Sigma)$ naturally. Figure 5 helps us understand the way of proof which will be followed.

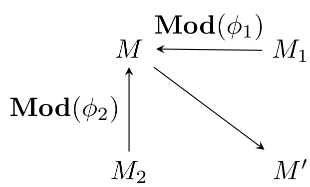


Figure 5

Let $(\mathcal{T}_1, \tau_1) \in \mathbb{T}\text{-}\mathbf{Mod}(\Sigma_1)$ and $(\mathcal{T}_2, \tau_2) \in \mathbb{T}\text{-}\mathbf{Mod}(\Sigma_2)$ be models such that equation 3.6 holds. Then for each pair $(M_1 \in \mathcal{T}_1, M_2 \in \mathcal{T}_2)$ with $\mathbf{Mod}(\phi_1)(M_1) = \mathbf{Mod}(\phi_2)(M_2)$ there exists a $M' \in \mathbf{Mod}(\Sigma')$ such that

$$\mathbf{Mod}(\theta_2)(M') = M_2 \text{ and } \mathbf{Mod}(\theta_1)(M') = M_1 \tag{3.7}$$

Taking into account that the model amalgamation square functions as a choice function, we define:

- $\mathcal{T}' = \bigcup_{M \in \mathcal{T}} \{M'_M \in \mathbf{Mod}(\Sigma') \;:\; M'_M = M_1 \otimes_{\phi_1, \phi_2} M_2\}$
- τ' the smaller topology where $\mathbf{Mod}(\theta_1)$ and $\mathbf{Mod}(\theta_2)$ are open and continuous maps.

Furthermore this extension is unique up to homeomorphism. Having demonstrated the construction, we need to prove that the Sharing Condition holds; namely we

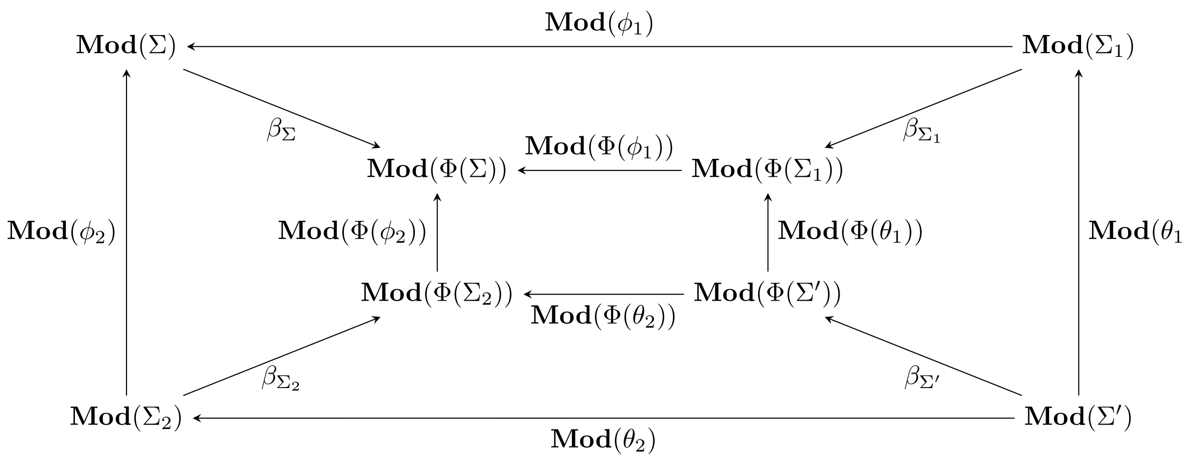


Figure 6

want to prove that for each $M'_i, M'_j \in \mathcal{T}'$ we get $\beta^{\Delta}_{\Sigma'}(M'_i) = \beta^{\Delta}_{\Sigma'}(M'_j)$.

Let $M'_i, M'_j \in \mathcal{T}'$ i.e. $M'_i, M'_j \in \mathbf{Mod}(\Sigma')$, working in the big square :
$M_i, M_j \in \mathbf{Mod}(\Sigma)$ such that
(3.8)
$M_i = \mathbf{Mod}(\phi_1)(M_i^1) = \mathbf{Mod}(\phi_2)(M_i^2)$ and $M_j = \mathbf{Mod}(\phi_1)(M_j^1) = \mathbf{Mod}(\phi_2)(M_j^2)$

Working on the base institution and due to the amalgamation property there exists exactly one $M'_i \in \mathbf{Mod}(\Sigma')$ such that :

(3.9) $$\mathbf{Mod}(\theta_1)(M'_i) = M_i^1 \text{ and } \mathbf{Mod}(\theta_2)(M'_i) = M_i^2$$

Also $M_i, M_j \in \mathbf{Mod}(\Sigma)$ such that :
(3.10)
$M_i = \mathbf{Mod}(\phi_1)(M_i^1) = \mathbf{Mod}(\phi_2)(M_i^2)$ and $M_j = \mathbf{Mod}(\phi_1)(M_j^1) = \mathbf{Mod}(\phi_2)(M_j^2)$

Working on the base institution and due to amalgamation property there exists exactly one $M'_j \in \mathbf{Mod}(\Sigma')$ such that :

(3.11) $$\mathbf{Mod}(\theta_1)(M'_j) = M_j^1 \text{ and } \mathbf{Mod}(\theta_2)(M'_j) = M_j^2$$

Now using the properties of the natural transformation β:

(1) $M_i^1, M_j^1 \in \mathbf{Mod}(\Sigma_1)$ such that $\beta^{\Delta}_{\Sigma_1}(M_i^1) = \beta^{\Delta}_{\Sigma_1}(M_j^1) \in \mathbf{Mod}\Phi(\Sigma_1)$
(2) $M_i^2, M_j^2 \in \mathbf{Mod}(\Sigma_2)$ such that $\beta^{\Delta}_{\Sigma_2}(M_i^2) = \beta^{\Delta}_{\Sigma_2}(M_j^2) \in \mathbf{Mod}\Phi(\Sigma_2)$

(3) $M_i, M_j \in \Sigma$ such that $\beta^{\Delta}_{\Sigma}(M_i) = \beta^{\Delta}_{\Sigma}(M_j) \in \mathbf{Mod}\Phi(\Sigma)$

We have

$$\mathbf{Mod}(\Phi(\phi_1))\beta^{\delta}_{\Sigma_1}(M_i^1) = \beta_{\Sigma}(\mathbf{Mod}(\phi_1)(M_i^1)) = \beta_{\Sigma}(M_i) \tag{3.12}$$

and

$$\mathbf{Mod}(\Phi(\phi_2))\beta^{\delta}_{\Sigma_1}(M_i^1) = \beta_{\Sigma}(\mathbf{Mod}(\phi_1)(M_i^1)) = \beta_{\Sigma}(M_i) \tag{3.13}$$

From Φ-amalgmation there is exactly one $C \in \Phi(\Sigma')$ such that

$$\mathbf{Mod}(\theta_2)C = \beta_{\Sigma}(M_i^2) \text{ and } \mathbf{Mod}(\theta_2)C = \beta_{\Sigma}(M_i^1) \tag{3.14}$$

However, $\mathbf{Mod}(\theta_1)\beta_{\Sigma'}(M_i') = \beta_{\Sigma_1}(\mathbf{Mod}\theta_1(M_i')) = \beta_{\Sigma_1}(M_i^1)$ so $C = \beta_{\Sigma'}(M_i')$, also $\mathbf{Mod}(\theta_2)\beta_{\Sigma'}(M_j') = \beta_{\Sigma_2}(\mathbf{Mod}\theta_2(M_i')) = \beta_{\Sigma}(M_i^1)$ and also $C = \beta_{\Sigma'}(M_j')$ and $\beta_{\Sigma'}(M_j') = \beta_{\Sigma'}(M_i')$ □

Theorem 3.2.12. *Given an institution morphism* $\left(\Phi^{\Delta}, \alpha^{\Delta}, \beta^{\Delta}\right) : (\mathbb{S}ig, \boldsymbol{Sen}, \mathbf{Mod}) \to \Delta$ *for any topological functor constructed by the previous description the* $\mathcal{I}_{\mathcal{T}} = \left(\mathbb{S}ig^{\mathcal{I}_{\mathcal{T}}}, \boldsymbol{Sen}^{\mathcal{I}_{\mathcal{T}}}, \mathbb{T} - \mathbf{Mod}^{\mathcal{I}_{\mathcal{T}}}, \models\right)$ *forms an institution.*

Proof. We should be able to prove the satisfaction condition for every signature morphism $\phi : \Sigma \to \Sigma'$

$$(\mathcal{T}', \tau') \models^{\mathcal{I}_{\mathcal{T}}}_{\Sigma'} \boldsymbol{Sen}^{\mathcal{I}_{\mathcal{T}}}(\phi)(\rho) \text{ iff } \mathbb{T} - \mathbf{Mod}^{\mathcal{I}_{\mathcal{T}}}(\phi)\,(\mathcal{T}', \tau') \models^{\mathcal{I}_{\mathcal{T}}}_{\Sigma} \rho \tag{3.15}$$

Note that when $\rho \in \boldsymbol{Sen}^{\mathcal{I}}(\Sigma)$, the relation 3.15 follows from the satisfaction condition of the base institution. The induction step can be checked easily for the Boolean connectives, and for the modalities.

The proof of the base case: Let $\rho \in \boldsymbol{Sen}^{\mathcal{I}}(\Sigma)$, then

$$\begin{aligned} \mathbb{T} - \mathbf{Mod}\,(\mathcal{T}', \tau') \models^{\mathcal{I}_{\mathcal{T}}}_{\Sigma} \rho \Leftrightarrow \\ \forall M \in \mathcal{T}'; \mathbf{Mod}(\phi) \models^{\mathcal{I}}_{\Sigma} \rho \Leftrightarrow \\ \forall M = \mathbf{Mod}(\phi)(Y) \text{ for some } Y \in \mathcal{T}' \models^{\mathcal{I}}_{\Sigma} \rho \Leftrightarrow \\ \forall M' \in \mathcal{T}' \models^{\mathcal{I}}_{\Sigma'} \boldsymbol{Sen}^{\mathcal{I}}(\phi)(\rho) \Leftrightarrow \\ (\mathcal{T}', \tau') \models^{\mathcal{I}_{\mathcal{T}}}_{\Sigma'} \boldsymbol{Sen}^{\mathcal{I}_{\mathcal{T}}}(\phi)(\rho) \end{aligned} \tag{3.16}$$

The next inductive step involves Boolean connectives. Let $\rho_1, \rho_2 \in \boldsymbol{Sen}(\Sigma)$ for which the inductive hypothesis holds, then

$$\begin{aligned} \mathbb{T} - \mathbf{Mod}\,(\mathcal{T}', \tau') \models^{\mathcal{I}_\mathcal{T}}_{\Sigma} \rho_1 \wedge \rho_2 \Leftrightarrow \\ \mathbb{T} - \mathbf{Mod}\,(\mathcal{T}', \tau') \models^{\mathcal{I}_\mathcal{T}}_{\Sigma} \rho_1 \text{ and } \mathbb{T} - \mathbf{Mod}\,(\mathcal{T}', \tau') \models^{\mathcal{I}_\mathcal{T}}_{\Sigma} \rho_2 \Leftrightarrow \\ (\mathcal{T}', \tau') \models^{\mathcal{I}_\mathcal{T}}_{\Sigma'} \boldsymbol{Sen}^{\mathcal{I}_\mathcal{T}}(\phi)(\rho_1) \text{ and } (\mathcal{T}', \tau') \models^{\mathcal{I}_\mathcal{T}}_{\Sigma'} \boldsymbol{Sen}^{\mathcal{I}_\mathcal{T}}(\phi)(\rho_2) \end{aligned} \tag{3.17}$$

The next inductive step involves modalities. Let $\rho \in \boldsymbol{Sen}^{\mathcal{I}_\mathcal{T}}(\Sigma)$ for which the inductive hypothesis holds, then

(3.18)

$$\begin{aligned} \mathbb{T} - \mathbf{Mod}(\mathcal{T}', \tau') \models^{\mathcal{I}_\mathcal{T}}_{\Sigma} \Box\rho \Leftrightarrow \\ \forall M \in \mathcal{T}'; \mathbf{Mod}(\phi) \models^{\mathcal{I}_\mathcal{T}}_{\Sigma} \Box\rho \Leftrightarrow \\ \forall M \text{ there exists } O_M \in \mathbf{Mod}(\phi)[\tau'] \text{ such that: } M \in O_M \ \&\ \forall N \in O_M \ N \models^{\mathcal{I}_\mathcal{T}}_{\Sigma} \rho \end{aligned}$$

If we set $O_{M'} = O_{\mathbf{Mod}^{-1}[O_M]}$, with $M = \mathbf{Mod}(\phi)(M')$. Then $O_{M'}$ is open and if $M' \in O_1$, then

(3.19)

$$\begin{aligned} \forall M' \text{ there exists } O_{M'} \in \tau' \text{ such that: } M' \in O_{M'} \ \&\ \forall Y' \in O_{M'} \ Y' \models^{\mathcal{I}_\mathcal{T}}_{\Sigma'} \boldsymbol{Sen}^{\mathcal{I}_\mathcal{T}}(\phi)\rho \\ \forall M \in \mathcal{T}' \models^{\mathcal{I}_\mathcal{T}}_{\Sigma} \Box \boldsymbol{Sen}^{\mathcal{I}_\mathcal{T}}(\phi)(\rho) \Leftrightarrow \\ \forall M \in \mathcal{T}' \models^{\mathcal{I}_\mathcal{T}}_{\Sigma} \boldsymbol{Sen}^{\mathcal{I}_\mathcal{T}}(\phi)(\Box\rho) \Leftrightarrow \\ (\mathcal{T}', \tau') \models^{\mathcal{I}_\mathcal{T}}_{\Sigma} \boldsymbol{Sen}^{\mathcal{I}_\mathcal{T}}(\phi)(\Box\rho) \end{aligned}$$

It is important to mention again the informal definition of universal quantification. We write $M \models^{\mathcal{I}}_{\Sigma} \forall\chi\rho_1$ when we get $M_1 \models^{\mathcal{I}}_{\Sigma_1} \rho_1$ **for all** χ-expansions M_1 of M (see page 94 [**17**]).

Let $\mathbb{T} - \mathbf{Mod}(\mathcal{T}', \tau') \models^{\mathcal{I}_\mathcal{T}}_{\Sigma} \rho = \forall\chi\rho_1$. If $\phi : \Sigma \to \Sigma'$ is a signature morphism which belongs to $\mathcal{D}$ and $\rho \in \boldsymbol{Sen}^{\mathcal{I}_\mathcal{T}}(\Sigma)$ is an universal χ-quantification sentence of a Σ_1-sentence ρ_1 and if the induction hypothesis holds for ρ_1, then for every $M \in \mathcal{T}'; \mathbf{Mod}(\phi)$ there exist $M' \in \mathbf{Mod}(\Sigma')$ with $M = \mathbf{Mod}(\phi)(M')$ such that:

$$M \models^{\mathcal{I}}_{\Sigma} \rho = \forall\chi\rho_1 \tag{3.20}$$

Let $\chi : \Sigma \to \Sigma_1$ be a morphism $\in \mathcal{D}$ such that $M = \mathbf{Mod}(\chi)(M_1)$, then

$$M_1 \models^{\mathcal{I}}_{\Sigma_1} \rho_1 \tag{3.21}$$

From Theorem 3.2.11 and the induction hypothesis (Figure 7) we can imply that for Σ_1' signature such that $\chi' : \Sigma' \to \Sigma_1'$, $M_1' \in \mathbf{Mod}(\Sigma_1')$ with $\mathbf{Mod}(\phi_1)M_1' = M_1$ and $\mathbf{Mod}(\chi')(M_1') = M'$ the following holds:

$$M_1' \models^{\mathcal{I}}_{\Sigma_1'} \boldsymbol{Sen}^{\mathcal{I}_{\mathcal{T}}}(\phi_1)\rho_1 \tag{3.22}$$

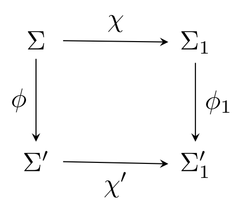


Figure 7. Pushout square

Equation 3.22 implies

$$M' \models^{\mathcal{I}}_{\Sigma'} \forall \chi' \boldsymbol{Sen}^{\mathcal{I}_{\mathcal{T}}}(\phi_1)\rho_1 \tag{3.23}$$

Therefore we have the purpose

$$(\mathcal{T}', \tau') \models^{\mathcal{I}_{\mathcal{T}}}_{\Sigma'} \forall \chi' \boldsymbol{Sen}^{\mathcal{I}_{\mathcal{T}}}(\phi_1)\rho_1 \tag{3.24}$$

$\square$

3.3. Categorical Topological Untraproducts

In the previous section we examined how to expand a classic logical system to one with topological semantics. In this section we will examine how to introduce the notion of ultraproducts in this new modal extension. Definitions 1.2.11 and 1.2.10 serve as the basis for the definitions which follow.

Definition 3.3.1. For a signature Σ in an institution, a Σ-sentence ρ is :

- preserved by $\mathbb{F}$-filtered factors if $\prod_{\mathcal{F}} A_i \models_\Sigma \rho$ implies $\{i \in I \mid A_i \models_\Sigma \rho\} \in \mathcal{F}$;
- preserved by $\mathbb{F}$-filtered products if $\{i \in I \mid A_i \models_\Sigma \rho\} \in \mathcal{F}$ implies $\prod_{\mathcal{F}} A_i \models_\Sigma \rho$;

for each filter $\mathcal{F} \in \mathbb{F}$ over a set I and for each family $\{A_i\}_{i \in I}$ of Σ-models.

Remark 3.3.2. Note that $\{i \in I \mid A_i \models_\Sigma \rho\} \in \mathcal{F}$ is equivalent with $\exists J \in \mathcal{F},\ \& \forall i \in J\ A_i \models_\Sigma \rho$.

The following definition comes as a complement to definition 3.3.1.

Definition 3.3.3. Let $\mathbb{F}$ be a class of filters. For a signature Σ, a sentence ρ is:

(1) topo-modally preserved by $\mathbb{F}$-filtered factors, when for each $i \in I_{\mathcal{T}_\mathcal{F}}$, $\prod_{\mathcal{F}} (\mathcal{T}_j, \tau_j) = (\mathcal{T}_\mathcal{F}, \tau_\mathcal{F}) \models^i_\Sigma \rho$ there exists $k \in \mu_J^{-1}(i)$ and $J \in \mathcal{F}$ such that $(\mathcal{T}_j, \tau_j) \models^{k_j}_\Sigma \rho$ for all $j \in J$;

(2) topo-modally preserved by $\mathbb{F}$-filtered products, when for each $i \in I_{\mathcal{T}_\mathcal{F}}$ there exists $k \in \mu_J^{-1}(i)$ and $J \in \mathcal{F}$ such that $(\mathcal{T}_j, \tau_j) \models^{k_j}_\Sigma \rho$ for all $j \in J$ it implies $\prod_{\mathcal{F}} (\mathcal{T}_j, \tau_j) = (\mathcal{T}_\mathcal{F}, \tau_\mathcal{F}) \models^i_\Sigma \rho$;

for each filter $\mathcal{F} \in \mathbb{F}$ over a set I and for every family $\{(\mathcal{T}_j, \tau_j)\}_{j \in I}$ of toplogical models

3.3.1. Filtered products of topological models. The aim of this section is to develop an extension of the institution-independent method of ultraproducts of [18] to topological semantics and to topo-modal satisfaction. The first step of our proof is to show that categorical filtered products can be lifted from the categories of the base models to the categories of topological models. Following that, we develop the first ultraproduct fundamental theorem for the topo-modal satisfaction.
Let us assume that:

- a class $\mathbb{F}$ of filters;
- an institution morphism from a base institution to a domain institution

$$\left(\Phi^\Delta, \alpha^\Delta, \beta^\Delta\right) : \mathcal{I} = \left(\mathbb{S}ig^\mathcal{I}, \boldsymbol{Sen}^\mathcal{I}, \boldsymbol{Mod}^\mathcal{I}, \models^\mathcal{I}\right) \to \Delta \tag{3.25}$$

such that:

(FP): for each signature Σ the category of Σ-models $\mathbf{Mod}(\Sigma)$ has products and $\mathbb{F}$-filtered products which are preserved by the β^Δ_Σ;

(LI): for each signature Σ, β^Δ_Σ lifts isomorphisms, i.e. if $\beta^\Delta_\Sigma(M)$ is isomorphic to N there is an M' such that $\beta^\Delta_\Sigma(M') = N$;

Proposition 3.3.4. *For all signatures Σ, the category of topological models* $\mathbb{T}$-$\mathbf{Mod}(\Sigma)$ *has filtered products.*

Proof. Let $\mathcal{F}$ be any filter over a set I and let $\{(T_j, \tau_j) \mid j \in I\}$ be an I-index family of topological models. For each $J \in \mathcal{F}$ we denote the topological product

$$\prod_{j \in J}(\mathcal{T}_j, \tau_j) = \left(\prod_{j \in J} \mathcal{T}_j,\ \tau_J\right) = (\mathcal{T}_J, \tau_J) \tag{3.26}$$

where τ_J is the standard product topology. It is important to mention that the elements of the $\prod_{j \in J} \mathcal{T}_j$ have the form $\prod_{j \in J} M_j^{k_j}$ where $M_j^{k_j} \in \mathcal{T}_j$ models in $\mathbf{Mod}(\Sigma)$. Let $\{\langle \mathcal{T}_i, \tau_i \rangle\}_{i \in I}$ be a family of topological models, such that $\mathcal{T}_i : I_{\mathcal{T}_i} \mapsto \mathbf{Mod}(\Sigma)$. We know that the $\mathbf{Mod}(\Sigma)$ has products, which means that for any $\{M_i\}_{i \in I}$ family of Σ-models product $\prod_{i \in I} M_i$ is also a Σ-model. So we can define

$$\mathcal{T}_I = \prod_{i \in I} \mathcal{T}_i : I_{\prod_{i \in I} \mathcal{T}_i} = \prod_{i \in I} I_{\mathcal{T}_i} \mapsto \mathbf{Mod}(\Sigma) \tag{3.27}$$

such that for every $k = \prod_{i \in I} k_i \in \prod_{i \in I} I_{\mathcal{T}_i}$ we have

$$\prod_{i \in I} \mathcal{T}_i(k) = \prod_{i \in I} M_i^{k_i} \tag{3.28}$$

such that for each $i \in I$ $M_i^{k_i} \in \mathcal{T}_i$. Furthermore, for all $k, k' \in \prod_{i \in I} I_{\mathcal{T}_i}$ we get $\beta_\Sigma^\Delta \left(\prod_{i \in I} M_i^{k_i}\right) = \beta_\Sigma^\Delta \left(\prod_{i \in I} M_i^{k'_i}\right)$. In order to construct $\prod_{\mathcal{F}} \mathcal{T}_i$ we will work as follows: as a first step we will work on category of **Set** so as to construct the appropriate index set $I_{\mathcal{T}_\mathcal{J}}$. For every $J \in \mathcal{F}$ and for all $j \in J$, we have an index set $I_{\mathcal{T}_j}$ to $\mathcal{T}_j$. The category of sets has the products; this implies that $\prod_{j \in J} I_{\mathcal{T}_j}$ belongs to the category of sets. Taking the colimit as a new index set, $\prod_{\mathcal{F}} I_{\mathcal{T}_i} = \mathcal{I}_\mathcal{F}$, with $p_{J',J}$ being the canonical projections as shown in Figure 8. we can now move onward to

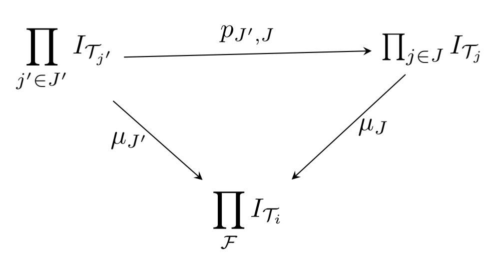


Figure 8

the next step.

Setting canonical projections $p_{k',k} : \prod_{i \in J'} M_i \mapsto \prod_{i \in J} M_i$ for every $J \subseteq J' \in \mathcal{F}$, in a natural way for every $k \in \mu_{J'}^{-1}(i)$ and $k' \in \mu_{J'}^{-1}(i)$ with $p_{J',J}(k') = k$ we have the new colimit (see Figure 9). As for the set defined through the previous

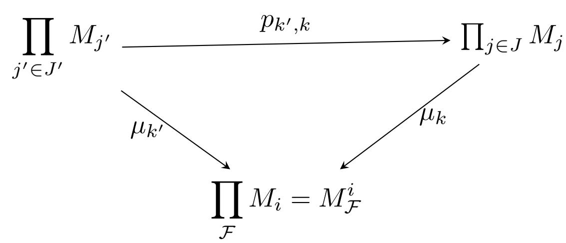


Figure 9

construct (Equation 3.29) we attribute to it the reduct topology as per its standard definition.

$$\mathcal{T}_{\mathcal{F}} = \prod_{\mathcal{F}} \mathcal{T}_i : I_{\mathcal{F}} \mapsto \mathbf{Mod}(\Sigma) \tag{3.29}$$

With the previous construct we achieved attributing a colimit $\mathbb{F}$-filtered product in $\mathbf{Mod}(\Sigma)$, to every $i \in I_{\mathcal{T}_{\mathcal{J}}}$, that colimit being a $\mathbb{F}$-filtered product in $\mathbf{Mod}(\Sigma)$. Furthermore it holds that for every $i \in I_{\mathcal{F}}$ and for all $(k_j)_{j \in J}$, the $\prod_{\mathcal{F}} M_i = M_{\mathcal{F}}^i$ is the filtered product modulo $\mathbb{F}$ of the family $\{M_j^{k_j} \mid j \in I\}$. $\square$

Theorem 3.3.5 (First Topological Modal Fundamental Theorem)**.** (1) *Each sentence of the base institution which is preserved by $\mathbb{F}$-filtered products in the base institution is also topo-modally preserved by $\mathbb{F}$-filtered products of topological models;*

(2) *Each sentence of the base institution which is preserved by $\mathbb{F}$-filtered factors in the base institution is also topo-modally preserved by $\mathbb{F}$-filters factors of topological models;*

Proof. Let $\mathcal{F} \in \mathbb{F}$ be a filter over a set I of a family $\{(\mathcal{T}_j, \tau_j\}_{j \in I}$.

(1) If $i \in I_{\mathcal{T}_{\mathcal{F}}}$ such that there exists $k \in \mu_J^{-1}(i)$ and $J \in \mathcal{F}$, then from hypothesis we have $\prod_{\mathcal{F}} M_i \models_{\Sigma}$ and from Proposition 3.3.4 we have the purpose.

(2) If $i \in I_{\mathcal{T}_\mathcal{F}}$ with $(\mathcal{T}_\mathcal{F}, \tau_\mathcal{F}) \models_\Sigma \rho$, then from Proposition 3.3.4 we have $\prod_\mathcal{F} M_i \models_\Sigma \rho$, which implies that $\{i \in I \mid M_i^{k_i} \models_\Sigma \rho\} \in \mathcal{F}$. If we set $J = \{i \in I \mid M_i^{k_i} \models_\Sigma \rho\}$ and $k' = p_{I,J}(k)$, $\mu_J = \mu_I \circ p_{J,I}$ we have the purpose.

□

3.4. Conclusions

In general we can say that a coalgebra consists a set (or a category) X , labelled as state space, and a function $\xi : X \mapsto \mathcal{T}(X)$ where the elements of $\mathcal{T}(X)$ called transitions from X. In the language of category theory:

Definition 3.4.1. Let $\mathcal{L}$ be a category and $\mathcal{T} : \mathcal{L} \mapsto \mathcal{L}$ an endofunctor, then a $\mathcal{T}$-coalgebra is a pair (X, ξ), where Z is an object in $\mathcal{L}$ and ξ is an arrow such that $\xi : X \mapsto \mathcal{T}(X)$.

Definition 3.4.2. Let $\mathcal{T} : \mathcal{L} \mapsto \mathcal{L}$ be an endofunctor, then a morphism between two coalgebras (X, ξ) and (Y, γ) is a morphism $f : X \mapsto Y$ such that the following diagram commutes:

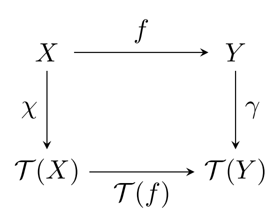


Figure 10

Example 3.4.3 (Kripke models)**.** This example is coming from the area of Standard Modal Logic. A Kripke model for a set of atomic formulas is a triple $\mathcal{M} = (W, \{R^m\}_{m \in \mathrm{MOD}}, V)$ where W is a nonempty set whose elements called as points or states or possible worlds. Each R^m is a binary relation on W and V is the valuation function, which assigns a subset of the domain to each basic propositional symbol of the language. According Definition 3.4.1, we can define Kripke models [26] through a coalgebraic formalization:

- $\mathrm{next}(w) = \{w' \in W \mid wRw'\}$ is the set of states that are possibly next of w;
- $\mathrm{prop}(w) = \{p \in P \mid w \in V(p)\}$ is the formulas which are true in w;

- the Kripke model is $(W, \xi$, where $\xi =< \mathrm{next}, \mathrm{prop} >: W \mapsto \mathcal{P}(W) \times \mathcal{P}(P)$, and
- the standard modalities can be defined as $[\Box\phi]_\xi = \{w \in W \mid \xi(w) \subseteq [\phi]_\xi\}$

In this chapter we defined the topological semantic extension of an arbitrary logical system and we examined the extension of classic results from ultraproduct theory. In the following, our aim will be to examine the new topolocial semantics standards so that the theorem extensions hold for Łoś sentences. The next for our work is to develop an extension of the institution-independent method of ultraproducts to coalgebraic semantics and to colgebraic-modal satisfaction.

Chapter 4

Generic Structures

Abstract. Analyzing diagrams which form generative classes we describe definable sets and their links in generic structures as well as cardinality bounds for these definable sets, finite or infinite. Introducing basic characteristics for definable sets in generic structures we compare them with each others and with cardinalities of these sets. We introduce calculi for (type-)definable sets allowing for comparing their cardinalities [**29**, **73**]. In terms of these calculi we prove a Trichotomy Theorem. Using these calculi we characterize the possibility to construct a generic structure of a given generative class. The notion of the definable set is one of the basic notions in Model Theory. Studying definable sets one can observe what properties can be described by formulas. Definable sets play an important role in describing the structural properties of elementary theories both in general and for valuable classes [**70**, **41**, **15**]. Cardinalities of definable sets for superstructures and Fraïssé structures were examined in [**77**, **33**]. In [**33**], examples for semantic generative classes which the forbid existence of Fraïssé limits, for uncountable cases, are proposed. In this chapter, considering syntactic approach to generic constructions and their limits [**65**, **62**, **63**, **67**, **66**, **58**] we study and compare basic characteristics for definable sets in generic structures.

4.1. Preliminaries

This chapter is organized as follows. In Section 1 we present preliminary notions and necessary results. In Section 2 we introduce fragments of definable sets in generic structures, characterize (co)finite definable sets, and describe bounds for finite definable sets and their covers. Basic characteristics and their bounds for infinite definable sets are described in Section 3. In Section 4 we introduce calculi for definable and type-definable sets allowing to compare and control their cardinalities. In terms of these calculi, we prove a Trichotomy Theorem. Using these calculi, in

Section 5, we introduce the notion of meeting of cardinality contradiction and, for given generative class, characterize the existence of a generic structure. In section 6 [34] we define and study lattices in generative classes associated with generic structures. It is shown that these lattices can be non-distributive and, moreover, sufficiently arbitrary enough. The height and width of the lattices is described. A model-theoretic criterion for linear ordering is proved and these linear orders are described. Boolean algebras generated by the considered lattices are described.

Considering syntactic approach to generic constructions and their limits [65, 62, 63, 67, 66, 58, 75] we study lattices in generative classes associated with generic structures. We show that these lattices can be non-distributive. Standard notions on the lattice theory can be found in [8, 25, 10].

We consider collections of sentences and formulas in first order logic in a language Σ. Thus, as usual, $\vdash$ means proof from no hypotheses deducing $\vdash \varphi$ for a formula φ of language Σ, which may contain function symbols and constants. If deducing φ, hypotheses can be used in a set Φ of formulas and we write $\Phi \vdash \varphi$. Usually Σ will be fixed in context and not mentioned explicitly. Henceforth we write $X, Y, Z, \ldots$ for finite sets of variables and denote finite sets of elements by $A, B, C, \ldots$, as well as finite sets in structures or structures with finite universes themselves.

In the diagrams, $A, B, C, \ldots$ denote finite sets of constant symbols disjoint from the constant symbols in Σ. We denote the vocabulary with the constants from A adjoined by $\Sigma(A)$. $\Phi(A), \Psi(B), \mathrm{X}(C)$ stand for Σ-*diagrams* (of sets A, B, C), that is, *consistent sets of* $\Sigma(A)$-, $\Sigma(B)$-, $\Sigma(C)$-*sentences*, respectively. These sets A, B, C are called *universes* of correspondent diagrams.

In the following we assume that for any considered diagram $\Phi(A)$, if a_1, a_2 are distinct elements in A, then $\neg(a_1 \approx a_2) \in \Phi(A)$. This means that if c is a constant symbol in Σ, then there is at most one element $a \in A$ such that $(a \approx c) \in \Phi(A)$.

If $\Phi(A)$ is a diagram and B is a set, we denote the set $\{\varphi(\bar{a}) \in \Phi(A) \mid \bar{a} \in B\}$ by $\Phi(A)|_B$. Similarly, for a language Σ, we denote the restriction of $\Phi(A)$ to the set of formulas in the language Σ by $\Phi(A)|_\Sigma$.

Definition 4.1.1. [65, 62, 63, 67, 66, 58] We denote by $[\Phi(A)]^A_B$ the diagram $\Phi(B)$ obtained by replacing a subset $A' \subseteq A$ by a set $B' \subseteq B$ of constants disjoint from Σ and with $|A'| = |B'|$, where $A \setminus A' = B \setminus B'$. Similarly we call the consistent set of formulas denoted by $[\Phi(A)]^A_X$ type $\Phi(X)$ if it is the result of a bijective substitution into $\Phi(A)$ of variables of X for the constants in A. In this case, we say that $\Phi(B)$ is a *copy* of $\Phi(A)$ and a *representative* of $\Phi(X)$. We also denote the diagram $\Phi(A)$ by $[\Phi(X)]^X_A$.

Remark 4.1.2. If the vocabulary contains functional symbols, then diagrams $\Phi(A)$ containing equalities and inequalities of terms can generate both finite and infinite structures. The same effect is observed for purely predicate vocabularies if it is written in $\Phi(A)$, then the model for $\Phi(A)$ should be infinite. For instance, diagrams containing axioms for finitely axiomatizable theories have this property.
By the definition, for any diagram $\Phi(A)$, each constant symbol in Σ appears in some formula of $\Phi(A)$. Thus, $\Phi(A)$ can be considered as $\Phi(A \cup K)$, where K is the set of constant symbols in Σ.

We now give conditions on a partial ordering of a collection of diagrams sufficient for it to determine a structure. We modify some of the conditions for structures by d to signify they are conditions on diagrams and not structures.

Definition 4.1.3. [**65**, **62**, **63**, **67**, **66**, **58**] Let Σ be a vocabulary. We say that $(\mathbf{D}_0; \leqslant)$ (or $\mathbf{D}_0$) is *generic*, or *generative*, if $\mathbf{D}_0$ is a class of Σ-diagrams of finite sets so that $\mathbf{D}_0$ is partially ordered by a binary relation $\leqslant$ such that $\leqslant$ is preserved by bijective substitutions, i. e., if $\Phi(A) \leqslant \Psi(B)$ and $A' \subseteq B'$ such that $[\Phi(A)]^A_{A'} = \Phi(A')$ and $[\Psi(B)]^B_{B'} = \Psi(B')$ are defined, then $[\Phi(A)]^A_{A'}$, $[\Psi(B)]^B_{B'}$ are in $\mathbf{D}_0$ and $[\Phi(A)]^A_{A'} \leqslant [\Psi(B)]^B_{B'}$.[1] Furthermore:

(1) if $\Phi(A) \in \mathbf{D}_0$, then for any quantifier free formula $\varphi(\bar{x})$ and any tuple $\bar{a} \in A$ either $\varphi(\bar{a}) \in \Phi(A)$ or $\neg\varphi(\bar{a}) \in \Phi(A)$;

(2) if $\Phi \leqslant \Psi$ then $\Phi \subseteq \Psi$;[2]

(3) if $\Phi \leqslant \mathrm{X}$ $\Psi \in \mathbf{D}_0$, and $\Phi \subseteq \Psi \subseteq \mathrm{X}$, then $\Phi \leqslant \Psi$;

(4) a diagram $\Phi_0(\emptyset)$ is the least element of the system $(\mathbf{D}_0; \leqslant)$;

(5) (the *d-amalgamation property*) for any diagrams $\Phi(A)$, $\Psi(B)$, $\mathrm{X}(C) \in \mathbf{D}_0$, if there exist injections f_0: $A \to B$ and g_0: $A \to C$ with $[\Phi(A)]^A_{f_0(A)} \leqslant \Psi(B)$

[1] Note that $\mathbf{D}_0$ is closed under bijective substitutions since $\leqslant$ is preserved by bijective substitutions and $\leqslant$ is reflexive.

[2] Note that $\Phi(A) \leqslant \Psi(B)$ implies $A \subseteq B$, since if $a \in A$ then $(a \approx a) \in \Phi(A)$, so $\Phi(A) \leqslant \Psi(B)$ implies $\Phi(A) \subseteq \Psi(B)$ and we have $(a \approx a) \in \Psi(B)$, whence $a \in B$.

and $[\Phi(A)]^A_{g_0(A)} \leqslant \mathrm{X}(C)$, then there are a diagram $\Theta(D) \in \mathbf{D}_0$ and injections $f_1\colon B \to D$ and $g_1\colon C \to D$ for which $[\Psi(B)]^B_{f_1(B)} \leqslant \Theta(D)$, $[\mathrm{X}(C)]^C_{g_1(C)} \leqslant \Theta(D)$ and $f_0 \circ f_1 = g_0 \circ g_1$; diagram $\Theta(D)$ is called the *amalgam* of $\Psi(B)$ and $\mathrm{X}(C)$ over the diagram $\Phi(A)$ and witnessed by the four maps (f_0, g_0, f_1, g_1) (see Figure 1);

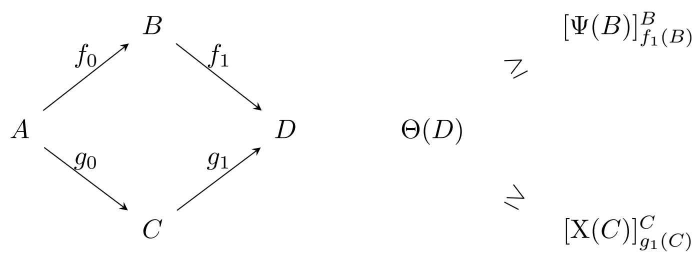


Figure 1

(6) (the *local realizability property*) if $\Phi(A) \in \mathbf{D}_0$ and $\Phi(A) \vdash \exists x\, \varphi(x)$, then there is a diagram $\Psi(B) \in \mathbf{D}_0$, $\Phi(A) \leqslant \Psi(B)$ and an element $b \in B$ for which $\Psi(B) \vdash \varphi(b)$;

(7) (the *d-uniqueness property*) for any diagrams $\Phi(A), \Psi(B) \in \mathbf{D}_0$ if $A \subseteq B$ and the set $\Phi(A) \cup \Psi(B)$ is consistent, then $\Phi(A) = \{\varphi(\bar{b}) \in \Psi(B) \mid \bar{b} \in A\}$.

(8) A diagram Φ is called a *strong subdiagram* of a diagram Ψ if $\Phi \leqslant \Psi$.

Example 4.1.4. Consider a vocabulary $\Sigma = \{<, f\}$ where $<$ is a linear order and f is a unary function. For $A = \{a\}$ and assuming that $\forall x(f(x) > x)$ and $a = a$ belongs to a diagram $\Phi(A)$, it follows that the diagram consists of the formulas $f^{(n+1)}(a) > f^{(n)}(a)$ for every $n \in \omega$. Thus the finite set A generates an infinite structure.

Example 4.1.5. Consider a vocabulary $\Sigma = \{<, f\}$ where $<$ is a linear order and f is a function. For $B = \{a, b\}$ and assuming that $\forall x \forall y(x < f(x, y) < y)$ and $a = a, b = b$ belong to a diagram $\Phi(B)$, it follows that the diagram consists of the formulas $x < f^{(n)}(x, y) < y$ for every $n \in \omega$. Thus the finite set B generates an infinite structure where the order is dense.

Definition 4.1.6. A diagram $\Phi(A)$ is said to be (*strongly*) *embeddable* in a diagram $\Psi(B)$ if there is an injection $f\colon A \to B$ such that $[\Phi(A)]^A_{f(A)} \subseteq \Psi(B)$ $([\Phi(A)]^A_{f(A)} \leqslant$

$\Psi(B)$). The injection f, in this instance, is called a (*strong*) *embedding* of diagram $\Phi(A)$ in diagram $\Psi(B)$ and is denoted by $f\colon \Phi(A) \to \Psi(B)$. A diagram $\Phi(A)$ is said to be (*strongly*) *embeddable* in a structure $\mathcal{M}$ if $\Phi(A)$ is (strongly) embeddable in a diagram $\Psi(B)$, where $\mathcal{M} \models \Psi(B)$. The corresponding embedding $f\colon \Phi(A) \to \Psi(B)$, in this case, is called a (*strong*) *embedding* of diagram $\Phi(A)$ in structure $\mathcal{M}$ and is denoted by $f\colon \Phi(A) \to \mathcal{M}$.
Let $\mathbf{D}_0$ be a class of diagrams, $\mathbf{P}_0$ be a class of structures of some language and $\mathcal{M}$ be a structure in $\mathbf{P}_0$. The class $\mathbf{D}_0$ is *cofinal* in the structure $\mathcal{M}$ if for each finite set $A \subseteq M$ there is a finite set B, $A \subseteq B \subseteq M$ and a diagram $\Phi(B) \in \mathbf{D}_0$ such that $\mathcal{M} \models \Phi(B)$. The class $\mathbf{D}_0$ is *cofinal* in $\mathbf{P}_0$ if $\mathbf{D}_0$ is cofinal in every structure of $\mathbf{P}_0$. We denote the class of all structures $\mathcal{M}$ by $\mathbf{K}(\mathbf{D}_0)$, under the condition that $\mathbf{D}_0$ is cofinal in $\mathcal{M}$, and a subclass of $\mathbf{K}(\mathbf{D}_0)$ by $\mathbf{P}$ such that each diagram $\Phi \in \mathbf{D}_0$ is true in some structure in $\mathbf{P}$.

Now we extend the relation $\leqslant$ from the generative class $(\mathbf{D}_0; \leqslant)$ to a class of subsets of structures in the class $\mathbf{K}(\mathbf{D}_0)$.

Definition 4.1.7. Let $\mathcal{M}$ be a structure in $\mathbf{K}(\mathbf{D}_0)$ and A and B be finite sets in $\mathcal{M}$ with $A \subseteq B$. We call A a *strong subset* of the set B (in the structure $\mathcal{M}$) and write $A \leqslant B$, if there exist diagrams $\Phi(A), \Psi(B) \in \mathbf{D}_0$, for which $\Phi(A) \leqslant \Psi(B)$ and $\mathcal{M} \models \Psi(B)$.
A finite set A is called a *strong subset* of a set $M_0 \subseteq M$ (in the structure $\mathcal{M}$), where $A \subseteq M_0$, if $A \leqslant B$ for any finite set B such that $A \subseteq B \subseteq M_0$ and $\Phi(A) \subseteq \Psi(B)$ for some diagrams $\Phi(A), \Psi(B) \in \mathbf{D}_0$ with $\mathcal{M} \models \Psi(B)$. If A is a strong subset of M_0 then, as above, we write $A \leqslant M_0$. If $A \leqslant M$ in $\mathcal{M}$ then we refer to A as a *self-sufficient set* (in $\mathcal{M}$).

Notice that, by the d-uniqueness property, diagrams $\Phi(A)$ and $\Psi(B)$ specified in the definition of strong subsets are defined uniquely. A diagram $\Phi(A) \in \mathbf{D}_0$, corresponding to a self-sufficient set A in $\mathcal{M}$, is said to be a *self-sufficient diagram* (in $\mathcal{M}$).

Definition 4.1.8. [**65, 62, 63, 67, 66, 58**] A class $(\mathbf{D}_0; \leqslant)$ possesses the *joint embedding property* (JEP) if for any diagrams $\Phi(A), \Psi(B) \in \mathbf{D}_0$ there is a diagram $\mathrm{X}(C) \in \mathbf{D}_0$ such that $\Phi(A)$ and $\Psi(B)$ are strongly embeddable in $\mathrm{X}(C)$.

Clearly, every generative class has JEP since JEP means the d-amalgamation property over the empty set.

Definition 4.1.9. [**65**, **62**, **63**, **67**, **66**, **58**] A structure $\mathcal{M} \in \mathbf{P}$ has *finite closures* with respect to the class $(\mathbf{D}_0; \leqslant)$, or is *finitely generated over* Σ, if any finite set $A \subseteq M$ is contained in some finite self-sufficient set in $\mathcal{M}$, i. e., there is a finite set B with $A \subseteq B \subseteq M$ and $\Psi(B) \in \mathbf{D}_0$ such that $\mathcal{M} \models \Psi(B)$ and $\Psi(B) \leqslant \mathrm{X}(C)$ for any $\mathrm{X}(C) \in \mathbf{D}_0$ with $\mathcal{M} \models \mathrm{X}(C)$ and $\Psi(B) \subseteq \mathrm{X}(C)$. A class $\mathbf{P}$ has *finite closures* with respect to the class $(\mathbf{D}_0; \leqslant)$, or it is *finitely generated over* Σ, if each structure in $\mathbf{P}$ has finite closures (with respect to $(\mathbf{D}_0; \leqslant)$).

Clearly, an at most countable structure $\mathcal{M}$ has finite closures with respect to $(\mathbf{D}_0; \leqslant)$ if and only if $M = \bigcup\limits_{i \in \omega} A_i$ for some self-sufficient sets A_i with $A_i \leqslant A_{i+1}$, $i \in \omega$.

Note that the finite closure property is defined modulo Σ and does not correlate with the cardinalities of algebraic closures. For instance, if Σ contains infinitely many constant symbols then $\mathrm{acl}(A)$ is always infinite whereas a finite set A can or cannot be extended to a self-sufficient set.

Additionally, for the finite closures of sets A we consider finite self-sufficient extensions B in a given structure $\mathcal{M}$ with respect to $(\mathbf{D}_0; \leqslant)$ only and B can be both a universe of a substructure of $\mathcal{M}$ or not. Moreover, it is permitted that corresponding diagrams $\Psi(B)$ can have only finite, finite and infinite, or only infinite models.

Thus, for instance, a finitely axiomatizable theory without finite models and with a generative class $(\mathbf{D}_0; \subseteq)$, containing diagrams for all finite sets and with axioms in diagrams, has identical finite closures whereas each diagram in $\mathbf{D}_0$ has only infinite models.

Definition 4.1.10. [**65**, **62**, **63**, **67**, **66**, **58**] A structure $\mathcal{M} \in \mathbf{K}(\mathbf{D}_0)$ is $(\mathbf{D}_0; \leqslant)$-*generic*, or a *generic limit for the class* $(\mathbf{D}_0; \leqslant)$ and denoted by $\mathrm{glim}(\mathbf{D}_0; \leqslant)$, if it satisfies the following conditions:

(a): $\mathcal{M}$ has finite closures with respect to $\mathbf{D}_0$;

(b): if $A \subseteq M$ is a finite set, $\Phi(A), \Psi(B) \in \mathbf{D}_0$, $\mathcal{M} \models \Phi(A)$ and $\Phi(A) \leqslant \Psi(B)$, then there exists a set $B' \leqslant M$ such that $A \subseteq B'$ and $\mathcal{M} \models \Psi(B')$;

Theorem 4.1.11. [65, 62, 63, 58] *For any generative class* $(\mathbf{D}_0; \leqslant)$ *with at most countably many diagrams the copies of which form* $\mathbf{D}_0$*, there exists a* $(\mathbf{D}_0; \leqslant)$*-generic structure.*

Theorem 4.1.12. [67, 58] *Every* ω*-homogeneous structure* $\mathcal{M}$ *is* $(\mathbf{D}_0; \leqslant)$*-generic for some generative class* $(\mathbf{D}_0; \leqslant)$*.*

Thus any first-order theory has a generic model and therefore can be represented by it.

Definition 4.1.13. [65, 62, 63, 67, 66, 58] A generative class $(\mathbf{D}_0; \leqslant)$ is *self-sufficient* if the following *axiom of self-sufficiency*, or *coherence* axiom, holds:

If $\Phi, \Psi, \mathrm{X} \in \mathbf{D}_0$, $\Phi \leqslant \Psi$, and $\mathrm{X} \subseteq \Psi$, then $\Phi \cap \mathrm{X} \leqslant \mathrm{X}$.

Theorem 4.1.14. [65, 62, 63, 58] *Let* $(\mathbf{D}_0; \leqslant)$ *be a self-sufficient class,* $\mathcal{M}$ *be at most countable* $(\mathbf{D}_0; \leqslant)$*-generic structure and* $\mathbf{K}$ *be the class of all models of* $T = \mathrm{Th}(\mathcal{M})$ *which has finite closures. Then the generic structure* $\mathcal{M}$ *is homogeneous.*

Thus, since any ω-homogeneous structure can be considered as generic with respect to a generic class with complete diagrams, a countable structure $\mathcal{M}$ is homogeneous if and only if it is generic for an appropriate self-sufficient generative class $(\mathbf{D}_0; \leqslant)$.

4.2. Definable sets, their fragments and finite cardinalities

Let $(\mathbf{D}_0; \leqslant)$ be a generative class in a language Σ, and $\mathcal{M}$ be a $(\mathbf{D}_0; \leqslant)$-generic structure. Take a Σ-formula $\phi(\bar{x})$ and a tuple $\bar{a} \in M$. Considering inductive steps of the construction of $\phi(\bar{x})$ we observe the relation $\mathcal{M} \models \phi(\bar{a})$:

(1) if $\phi(\bar{x})$ is a quantifier-free formula, then $\mathcal{M} \models \phi(\bar{a})$ if and only if $\Phi(A) \vdash \phi(\bar{a})$ in the propositional calculus for some/any $\Phi(A) \in \mathbf{D}_0$ satisfying $\mathcal{M} \models \Phi(A)$ with $\bar{a} \in A$;

(2) if $\phi(\bar{x})$ has a form $\exists y \psi(\bar{x}, y)$, then $\mathcal{M} \models \phi(\bar{a})$ if and only if $\mathcal{M} \models \psi(\bar{a}, b)$ for some/any $\Psi(B) \in \mathbf{D}_0$ satisfying $\mathcal{M} \models \Psi(B)$ with $\bar{a}, b \in B$;

(3) if $\phi(\bar{x})$ is a Boolean combination of formulas $\psi_i(\bar{x})$ with defined satisfaction $\mathcal{M} \models \psi_i^{\delta_i}(\bar{a})$, $\delta_i \in \{0, 1\}$, $i \in \{1, \ldots, n\}$, then $\mathcal{M} \models \phi(\bar{a})$ if and only if

the Boolean function $f(x_1, \dots, x_n)$ has the value $f(\delta_1, \dots, \delta_n) = 1$ for the combination $\phi(\bar{x})$ and variables x_i corresponding to $\psi_i(\bar{x})$;

(4) as usual, $\mathcal{M} \models \forall y \psi(\bar{x}, y)$ if and only if $\mathcal{M} \models \neg\exists y \neg\psi(\bar{x}, y)$.

If the generic structure $\mathcal{M}$ is finite, then there is a diagram $\Phi(M) \in \mathbf{D}_0$. Thus, $\Phi(M)$ completely defines the relation $\mathcal{M} \models \phi(\bar{a})$. Therefore in the following mainly we consider infinite $\mathcal{M}$.

Now we fix a definable set $X = X_\phi = \{\bar{a} \mid \mathcal{M} \models \phi(\bar{a})\}$ in the generic structure $\mathcal{M}$. For any $\Phi(A) \in \mathbf{D}_0$ with $\mathcal{M} \models \Phi(A)$ we denote by $X_{\Phi(A)}$ the finite $\Phi(A)$–*fragment* $X \cap A^{l(\bar{a})}$ of X. We will omit $\Phi(A)$– if that diagram is fixed or clear from the context.

Combining X from its finite fragments, we have:

$$X = \bigcup_{\Phi(A)} X_{\Phi(A)}. \tag{4.1}$$

Having Equality (4.1) we will also say that the diagrams $\Phi(A)$ *cover* the set X. Clearly, Equality (4.1) stays true taking only minimal diagrams $\Phi(A)$ such that all coordinates of each tuple $\bar{a} \in X$ are contained in some A. Moreover, if $(\mathbf{D}_0; \leqslant)$ is self-sufficient it suffices to take least diagrams $\Phi(A)$ with that property. In these cases, Equality (4.1) holds since described $\Phi(A)$ cover the set X. The latter case means that there are atoms covering X.

Recall that a partially ordered set $\langle U; \leq \rangle$ is said to be *downward* (*upward*) *directed* if for each $x, y \in U$ there exists $z \in U$ such that $z \leq x$ and $z \leq y$ (respectively, $x \leq z$ and $y \leq z$).

Remark 4.2.1. Following the definition of generic structure and the amalgamation property we have the *finite upward direction for X relative to finite fragments*: for any finite fragments $X_1, X_2 \subseteq X$ there is a finite fragment $Y \subseteq X$ such that $X_1 \cup X_2 \subseteq Y$.

Studying definable sets X in $\mathcal{M}$ it is natural to start with (co)finite sets.

Proposition 4.2.2. *The following conditions are equivalent for the definable set X:*

(1) *X is finite;*

(2) *$X = X_{\Phi(A)}$ for some $\Phi(A) \in \mathbf{D}_0$;*

(3) *there is* $\Phi(A) \in \mathbf{D}_0$ *such that* $X_{\Phi(A)} = X_{\Psi(B)}$ *for any* $\Psi(B)$ *with* $\mathcal{M} \models \Psi(B)$ *and* $\Phi(A) \leqslant \Psi(B)$.

Proof. (1) $\Rightarrow$ (2). If X is finite then we have the following cases:

(1) $X \subseteq X_{\Phi(A)}$ for some $\Phi(A) \in \mathbf{D}_0$; then $X = X_{\Phi(A)}$ since $X \supseteq X_{\Phi(A)}$;

(2) $X = \bigcup_{i=1}^{n} X_{\Phi_i(A_i)}$ for some $n > 1$ and $\Phi_i(A_i) \in \mathbf{D}_0$; then we consider induction on n:

- for $n = 2$, if $X = X_{\Phi(A_1)} \cup X_{\Phi(A_2)}$ then by the amalgamation property there exists a diagram $\Psi_2(C_2) \in D_0$ such that $\Phi(A_1)$ and $\Phi(A_2)$ are identically strongly embeddable in $\Psi_2(C_2)$ and by the definition of generic structure $\mathcal{M}$, $\mathcal{M} \models \Psi_2(C_2)$ with $X = X_{\Psi_2(C_2)}$;
- if $\Psi_{n-1}(C_{n-1})$ is already defined covering $\Phi_1(A_1), \ldots, \Phi_{n-1}(A_{n-1})$ we repeat the process above for $\Psi_{n-1}(C_{n-1})$ and $\Phi_n(A_n)$ taking their amalgam $\Psi_n(C_n)$ such that $\Psi_{n-1}(C_{n-1})$ and $\Phi_n(A_n)$ are identically strongly embeddable in $\Psi_n(C_n)$, and $\mathcal{M} \models \Psi_n(C_n)$. Hence we have $X = X_{\Psi_n(C_n)}$ for some $\Psi_n(C_n) \in \mathbf{D}_0$.

(2) $\Rightarrow$ (1). If $X = X_{\Phi(A)}$ for some $\Phi(A) \in D_0$ then X is finite by the definition of $X_{\Phi(A)}$.

(2) $\Leftrightarrow$ (3) is true again applying Equality (4.1) and Remark 4.2.1, since $\Phi(A) \leqslant \Psi(B)$ implies $X_{\Phi(A)} \subseteq X_{\Psi(B)}$. □

Remark 4.2.3. Proposition 4.2.2 means that a definable sets is finite if and only if it can be covered by a unique diagram.

The following propositions are implied by Remark 4.2.3 and the standard finite combinatorics.

Proposition 4.2.4. *If* $X \subseteq M^n$ *is finite with* $X = X_{\Phi(A)}$, *where* $|A| = k$, *then* $|X| \leq k^n$.

Definition 4.2.5. Let $X \subseteq M^n$ be finite, $X = X_{\Phi(A)}$ with $|A| = k$, U be a covering set of diagrams $\Psi \in \mathbf{D}_0$ for X, where $\mathcal{M} \models \Psi$, $\Psi \leqslant \Phi(A)$, $|U| = m$. Then U is called a *m-cover*, or simply an *cover*, of X with respect to A. If $V \subseteq U$ is again a cover of X then V is called a *$|V|$-subcover*, or simply a *subcover*, of U.

Proposition 4.2.6. *Any cover* U *of* X *contains an* s*-subcover* V *with* $s \leq k$.

Proposition 4.2.7. *Any finite X has a 1-cover $\{\Psi\}$. Moreover, if $(\mathbf{D}_0; \leqslant)$ is self-sufficient then the 1-cover $\{\Psi\}$ can be chosen with the least Ψ.*

Proposition 4.2.2 immediately implies

Corollary 4.2.8. *For the definable set X the following conditions are equivalent:*

(1) *X is cofinite;*

(2) *$M \setminus X = (M \setminus X)_{\Phi(A)}$ for some $\Phi(A) \in \mathbf{D}_0$;*

(3) *$M \setminus X \subseteq Y_{\Phi(A)}$ for some $\Phi(A) \in \mathbf{D}_0$ and a definable set Y;*

(4) *there is $\Phi(A) \in \mathbf{D}_0$ such that $(M \setminus X)_{\Phi(A)} = (M \setminus X)_{\Psi(B)}$ for any $\Psi(B)$ with $\mathcal{M} \models \Psi(B)$ and $\Phi(A) \leqslant \Psi(B)$.*

Remark 4.2.9. Since each diagram $\Phi(A) \in \mathbf{D}_0$ implies a quantifier-free diagram over A, if $X = X_{\{\phi\}}$ for a quantifier-free formula ϕ, in view of Proposition 4.2.2 we can characterize quantifier-free formulas for (co)finite definable sets in terms of $(\mathbf{D}_0; \leqslant)$, counting the number of tuples satisfying ϕ $(\neg\phi)$ and not taking the generic structure $\mathcal{M}$.

Thus, if $\mathrm{Th}(\mathcal{M})$ has quantifier elimination, then the set of algebraic types is defined in terms of $(\mathbf{D}_0; \leqslant)$, too.

Definition 4.2.10. [**70, 15, 5, 51**]. A structure $\mathcal{N}$ is *definably minimal* if any subset of N, definable by a formula $\phi(x, \bar{a})$, $\bar{a} \in N$, is either finite or cofinite. A structure $\mathcal{N}$ is *strongly minimal* if each $\mathcal{N}' \succ \mathcal{N}$ is definably minimal.

Remark 4.2.11. Consider the set $\mathbf{F}$ (respectively $\mathbf{CF}$ of Σ-formulas having (co)finitely many solutions in $\mathcal{M}$). We set $\mathbf{FCF} = \mathbf{F} \cup \mathbf{CF}$. Clearly, $\mathbf{F}$ and $\mathbf{CF}$ are closed under positive Boolean combinations preserving arity and under projections. Thus, $\mathbf{FCF}$ is closed under Boolean combinations preserving arity and under projections. Moreover, it holds for Σ_A-formulas, where $A \subseteq M$, forming the set $\mathbf{FCF}_A$. In such a case we write $\mathbf{FCF}_\varnothing$ for $\mathbf{FCF}$. Thus, restrictions of the Morleyzation of $\mathcal{M}$ to some unions of $\mathbf{FCF}_A$ form definably minimal fragments of that Morleyzation. Moreover, if a sublanguage Σ' of Σ forms the set of formulas in the union of $\mathbf{FCF}_A$ with one free variable then the Σ'-restriction of $\mathcal{M}$ is definably minimal. Preserving the definable minimality under elementary extensions of $\mathcal{M}$ we get strongly minimal restrictions. Characterizing (co)finite definable sets in Proposition 4.2.2 and Corollary 4.2.8 we deduce a characterization for the definable minimality of generic structures. Considering appropriate extensions of $(\mathbf{D}_0; \leqslant)$ we can also characterize the strong minimality.

Clearly, fragments (in particular, the greatest fragment) of a finite definable set allow, to count its cardinality.

4.3. Bounds for infinite definable sets in generic structures

We will now define characteristics allowing to find lower and upper bounds and the cardinality for an *infinite* definable set X.
For a diagram $\Phi(A) \in \mathbf{D}_0$ the set of copies $\Phi(B)$ of $\Phi(A)$ we denote by $c_{\Phi(A)}(X)$ such that $X_{\Phi(B)} \neq \varnothing$, and the set of copies $\Phi(B)$ of $\Phi(A)$ by $c^m_{\Phi(A)}(X)$ such that $\Phi(B)$ is minimal among diagrams $\Psi \in \mathbf{D}_0$ with $\mathcal{M} \models \Psi$ and $X_\Psi \neq \varnothing$. Since X is infinite and each of its Φ-fragment is finite, in view of Equality (4.1) we have:

$$|X| \leq \sum_{\Phi(A)} |c^m_{\Phi(A)}(X)| \leq \sum_{\Phi(A)} |c_{\Phi(A)}(X)|. \tag{4.2}$$

Remark 4.3.1. The first inequality in (4.2) can be strict if $\mathbf{D}_0$ is not closed under intersections. Indeed, in such a case a singleton $\{a\}$ can be a subset of unboundedly many sets A in minimal diagrams $\Phi(A) \in \mathbf{D}_0$. Thus, having, say, countable X we can form uncountably many minimal diagrams for each element $a \in X$ producing $\sum_{\Phi(A)} |c^m_{\Phi(A)}(X)| > \omega$. At the same time, if $\mathbf{D}_0$ is self-sufficient then each finite subset X_0 of X defines a set A with least diagram $\Phi(A) \in \mathbf{D}_0$ containing all coordinates of tuples in X_0 and, collecting all these least diagrams, we get

$$|X| = \sum_{\Phi(A)} |c^m_{\Phi(A)}(X)|.$$

The arguments above show that diagrams in $\mathbf{D}_0$, containing these $\Phi(A)$, can produce the strict inequality simultaneously

$$\sum_{\Phi(A)} |c^m_{\Phi(A)}(X)| < \sum_{\Phi(A)} |c_{\Phi(A)}(X)|.$$

Remark 4.3.2. The inequalities in (4.2) become equalities if $|X| = |M|$, since diagrams in the union $\bigcup\limits_{\Phi(A)} c_{\Phi(A)}(X)$ form a covering of M by finite sets.

In general, for the equality

$$|X| = \sum_{\Phi(A)} |c_{\Phi(A)}(X)| \tag{4.3}$$

it suffices to take a minimal covering, for X, subset of the set of diagrams $\Phi(A)$. In particular, Equality (4.3) holds if chosen diagrams $\Phi(A)$ are disjoint with respect to A and cover X.

Proposition 4.3.3. *A covering set U of diagrams $\Phi(A)$ for X (with $\mathcal{M} \models \Phi(A)$) is minimal if and only if for each $\Phi(A) \in U$ there is a coordinate a_i for a tuple $\bar{a}$ in X such that a_i belongs to A and does not belong to universes B of other diagrams $\Psi(B) \in U$.*

Proof. Let U be minimal and some $\Phi(A) \in U$ not have coordinates a_i tuples $\bar{a}$ in X such that $a_i \in A$ and does not belong to universes B of other diagrams $\Psi(B) \in U$. Then $U \setminus \{\Phi(A)\}$ is covering too, contradicting the minimality of U.

Conversely, if for each $\Phi(A) \in U$ there is a coordinate a_i for a tuple $\bar{a}$ in X such that a_i belongs to A and does not belong to universes B of other diagrams $\Psi(B) \in U$, then $U \setminus \{\Phi(A)\}$ is not covering. Therefore, U is minimal. □

Remark 4.3.4. If the covering set U consists of diagrams for singletons, then U is minimal. More generally, any covering set U, with universes A for $\Phi(A) \in U$ of bounded cardinalities, contains a minimal covering subset U_0.

In contrast to this, taking a covering set U of diagrams $\Phi_n(A_n)$ with $A_n \subset A_{n+1}$, $n \in \omega$, one can not find a minimal covering subset U_0.

Remark 4.3.5. In any case, taking an infinite set $X \subseteq M^n$, whether it is definable or not, and indexing elements of X we can choose step-by step a covering set U of diagrams $\Phi(A) \in \mathbf{D}_0$ such that $|X| = |U|$.

We denote by $d(\mathbf{D}_0, X)$ (respectively, $d^m(\mathbf{D}_0, X)$) the cardinality of a minimal set W of diagrams $\Phi(A) \in \mathbf{D}_0$ such that each diagram in $c_{\Phi(A)}(X)$ ($c^m_{\Phi(A)}(X)$) has a copy in W. We set $d(\mathbf{D}_0) = d(\mathbf{D}_0, M)$ and $d^m(\mathbf{D}_0) = d^m(\mathbf{D}_0, M)$. By the definition of generic structure, $d(\mathbf{D}_0)$ is the cardinality of minimal set W' of diagrams in $\mathbf{D}_0$, with non-empty universes, such that each diagram in $\mathbf{D}_0$ has a copy in W'. Respectively, $d^m(\mathbf{D}_0)$ is the cardinality of minimal set W' of diagrams in $\mathbf{D}_0$ such that each minimal diagram $\Phi(A) \in \mathbf{D}_0$ with nonempty A has a copy in W'.

Remark 4.3.6. Clearly, $d^m(\mathbf{D}_0, X) \leq d^m(\mathbf{D}_0)$ and $d^m(\mathbf{D}_0, X) \leq d(\mathbf{D}_0, X) \leq d(\mathbf{D}_0)$. Moreover, $d^m(\mathbf{D}_0, X)$ and even $d^m(\mathbf{D}_0)$ can be finite: if, for instance, the generic structure $\mathcal{M}$ realizes only finitely many n-types, for some $n \geq 1$, and all n-element sets $\{a_1, \ldots, a_n\}$, $a_i \in M$, are represented by diagrams in $\mathbf{D}_0$. If $\mathcal{M}$ realizes unique 1-type and some (equivalently, all) singletons $\{a\}$, $a \in M$, are represented by diagrams in $\mathbf{D}_0$ then $d^m(\mathbf{D}_0) = 1$. Therefore, in general, the only estimation for $d^m(\mathbf{D}_0)$ is $d^m(\mathbf{D}_0) \geq 1$.

At the same time, $d(\mathbf{D}_0, X)$ are always infinite (for nonempty X and infinite $\mathcal{M}$).

The value $d(\mathbf{D}_0)$ is bounded by the cardinality of the language Σ: $d(\mathbf{D}_0) \leq 2^{\max\{|\Sigma|,\omega\}}$. Besides, $d^m(\mathbf{D}_0, X)$ and $d(\mathbf{D}_0, X)$ are monotone with respect to the second coordinate and, by Remark 4.3.6, $d(\mathbf{D}_0, X)$ are always infinite (although $d^m(\mathbf{D}_0, X)$ can be finite).

Proposition 4.3.7. *For any definable set $Y \supseteq X$ in the generic structure $\mathcal{M}$,*

$$\omega \leq d(\mathbf{D}_0, X) \leq d(\mathbf{D}_0, Y) \leq d(\mathbf{D}_0) \leq 2^{\max\{|\Sigma|,\omega\}}.$$

Remark 4.3.8. In contrast with Proposition 4.3.7, in general, we cannot assert a fixed inequality comparing $|X|$ and $d(\mathbf{D}_0, X)$, and even $|X|$ and $d^m(\mathbf{D}_0, X)$. Indeed, X can be covered by copies of a fixed minimal diagram $\Phi(A)$. In such an instance, $d^m(\mathbf{D}_0, X) = 1$ implying $d^m(\mathbf{D}_0, X) < |X|$. If $d(\mathbf{D}_0, X) = \omega$ we can get both $d(\mathbf{D}_0, X) = |X|$, if X is countable, and $d(\mathbf{D}_0, X) < |X|$, if X is uncountable.

Now, having $d^m(\mathbf{D}_0, X) = d^m(\mathbf{D}_0, X) = d(\mathbf{D}_0) = 2^{\max\{|\Sigma|,\omega\}}$ we face three possibilities: $d(\mathbf{D}_0, X) > |X|$ taking a countable definable set X with $2^{\max\{|\Sigma|,\omega\}}$ diagrams which are not copies of one another and the universes of which have nonempty intersections with X; $d(\mathbf{D}_0, X) = |X|$ or $d(\mathbf{D}_0, X) < |X|$ for X of appropriate cardinalities. We have similar possibilities for $d^m(\mathbf{D}_0, X)$. But if the generative class $(\mathbf{D}_0; \leqslant)$ is self-sufficient, then minimal diagrams $\Phi(A)$ should be least producing the inequality $d^m(\mathbf{D}_0, X) \leq |X|$, which is strict if each least diagram, for X, is a copy of a diagram in a finite list.

We denote by $\mathrm{sc}(X)$ the value $\sup\{|c_{\Phi(A)}(X)| \mid \Phi(A) \in \mathbf{D}_0\}$ producing the supremum of numbers of copies $\Phi(B)$ for diagrams $\Phi(A)$ in $\mathbf{D}_0$ with nonempty fragments of X, and by $\mathrm{sc}^m(X)$ the value $\sup\{|c^m_{\Phi(A)}(X)| \mid \Phi(A) \in \mathbf{D}_0\}$ producing the supremum of numbers of minimal copies $\Phi(B)$ for diagrams $\Phi(A)$ in $\mathbf{D}_0$ with nonempty fragments of X. We set $\mathrm{sc}(\mathcal{M}) = \mathrm{sc}(M)$ and $\mathrm{sc}^m(\mathcal{M}) = \mathrm{sc}^m(M)$.

Remark 4.3.9. Clearly, $\mathrm{sc}^m(X) \leq \mathrm{sc}(X)$, $1 \leq \mathrm{sc}(X)$, and as above for $d(\mathbf{D}_0, X)$, three possibilities $\mathrm{sc}(X) < |X|$, $\mathrm{sc}(X) = |X|$, $\mathrm{sc}(X) > |X|$ can be realized. Replacing sc by sc^m we again have three realizable relations.

Equation (4.2) implies

$$|X| \leq d^m(\mathbf{D}_0, X) \cdot \mathrm{sc}^m(X) \leq d(\mathbf{D}_0, X) \cdot \mathrm{sc}(X). \tag{4.4}$$

Inequalities (4.4) can be strict and they can be equalities. In particular, we have

$$|M| = d^m(\mathbf{D}_0, M) \cdot \mathrm{sc}^m(M) = d(\mathbf{D}_0, M) \cdot \mathrm{sc}(M) = d(\mathbf{D}_0) \cdot \mathrm{sc}(\mathcal{M}). \tag{4.5}$$

Since $d(\mathbf{D}_0, X) \cdot \mathrm{sc}(X)$ is monotone with respect to X and both for $d(\mathbf{D}_0, X)$ and $\mathrm{sc}(X)$, then using Proposition 4.3.7 and equations (4.4), (4.5) we have the following:

Proposition 4.3.10. *For any definable set $Y \supseteq X$ in the generic structure $\mathcal{M}$,*

$$|X| \leq d(\mathbf{D}_0, X) \cdot \mathrm{sc}(X) \leq d(\mathbf{D}_0, Y) \cdot \mathrm{sc}(Y) \leq |M| = d(\mathbf{D}_0) \cdot \mathrm{sc}(\mathcal{M}).$$

Clearly, the estimate in Proposition 4.3.10 is precise since the cardinality of X can vary from 0 to $|M|$. Note also that, in general case, $\mathrm{sc}(\mathcal{M})$ can vary from 1 to $|M|$.

The following examples illustrate Proposition 4.3.10.

Example 4.3.11. Take a generic structure $\mathcal{M}$ consisting of $\lambda \geq \omega$ constants forming the language. We have that any unary definable set X is either finite of cofinite, $d(\mathbf{D}_0) = \lambda$, $\mathrm{sc}(\mathcal{M}) = 1$, and $|M| = \lambda = d(\mathbf{D}_0) = d(\mathbf{D}_0) \cdot \mathrm{sc}(\mathcal{M})$.

Example 4.3.12. Take a structure $\mathcal{M}'$ in the empty language and with $\lambda \geq \omega$ elements. We again have either finite of cofinite unary definable sets, $d(\mathbf{D}_0) = \omega$, $\mathrm{sc}(\mathcal{M}) = \lambda$, and thus $|M| = \lambda = \omega \cdot \lambda = d(\mathbf{D}_0) \cdot \mathrm{sc}(\mathcal{M})$.

Example 4.3.13. Extending Example 4.3.12 let $\mathcal{M}''$ be a structure in the language with μ disjoint unary predicates P_i, $|P_i| = \lambda_i$, $i \in \mu$, $\sum\limits_{i\in\mu} \lambda_i = \lambda$. Diagrams $\Phi(A)$, in a generative class $(\mathbf{D}_0; \leqslant)$ for $\mathcal{M}''$, describe finite cardinalities λ_i, $|P_j| \geq \omega$, and colorings $i \in \mu \cup \{\infty\}$ [**65, 61**] of elements $a \in A$ corresponding to $\models P_i(a)$ or $\models p_\infty(a)$, where $p_\infty(x) = \{\neg P_i(x) \mid i \in \mu\}$. If there are finite λ_i then the choices of elements $a \in A$ in correspondent predicates P_i can be counted by standard formulas of finite combinatorics. Having only infinite λ_i we get $\max\{\mu, \omega\}$ possibilities forming $d(\mathbf{D}_0)$ and $\mathrm{sc}(\mathcal{M}) = \lambda$ producing $|M| = \lambda = \max\{\mu, \omega\} \cdot \lambda = d(\mathbf{D}_0) \cdot \mathrm{sc}(\mathcal{M})$.

In general, concerning unary predicates, $d(\mathbf{D}_0)$ can vary from ω to $\max\{2^\mu, \omega\}$ and $\mathrm{sc}(\mathcal{M})$ has a cardinality between 1 and λ.

Example 4.3.14. (cf. [**33**, Proof of Theorem 2.1]) Let $\mathcal{N}$ be a structure in the language with $\mu \geq \omega$ equivalence relations E_i such that $E_0 = N^2$, each E_i-class is divided into k_i E_{i+1}-classes, $k_i \in \omega \setminus \{0, 1\}$, $i \in \mu$, and every intersection of a $\subseteq$-chain of E_i-classes X_i, $i \in \mu$, has κ elements for some fixed $\kappa > 0$.

Clearly, $|\mathcal{N}| = 2^\mu \cdot \kappa$. In particular, if $\kappa \leq \omega$ then $|\mathcal{N}| = 2^\mu$. Since $\mathrm{Th}(\mathcal{N})$ has μ complete types, $d(\mathbf{D}_0) = \mu$, too. Therefore $\mathrm{sc}(\mathcal{N}) = 2^\mu \cdot \kappa$, and if $\kappa \leq \omega$ then $\mathrm{sc}(\mathcal{N}) = 2^\mu$.

4.4. Links between definable sets

In this section we consider links between definable sets implying cardinality estimations for these definable sets and forcing for these cardinalities. Thus, in families of definable sets, we consider relations which allow us to compare cardinalities of these definable sets.

Definition 4.4.1. If X and Y are definable sets in a structure $\mathcal{M}$, $X = \phi(\mathcal{M}, \bar{a})$, $Y = \psi(\mathcal{M}, \bar{b})$, $\bar{a}, \bar{b} \in M$, $|X| = \lambda$, $|Y| = \mu$, then we write $X \Rightarrow_{\mu,\mathcal{M}} Y$, $X\,{}_{\lambda,\mathcal{M}}\!\Leftarrow Y$, and $X\,{}_{\lambda,\mathcal{M}}\!\Leftrightarrow_{\mu,\mathcal{M}} Y$. If $X \Rightarrow_{\mu,\mathcal{N}} Y'$ (respectively, $X\,{}_{\lambda,\mathcal{N}}\!\Leftarrow Y$; $X\,{}_{\lambda,\mathcal{N}}\!\Leftrightarrow_{\mu,\mathcal{N}} Y$) for any $\mathcal{N}$ such that $\bar{a}, \bar{b} \in N$, $\mathcal{M} \prec \mathcal{N}$ or $\mathcal{M} \succ \mathcal{N}$, $X = \phi(\mathcal{N}, \bar{a})$, $Y' = \psi(\mathcal{N}, \bar{b})$ ($X' = \phi(\mathcal{N}, \bar{a})$, $Y = \psi(\mathcal{N}, \bar{b})$; $X' = \phi(\mathcal{N}, \bar{a})$, $Y' = \psi(\mathcal{N}, \bar{b})$, and $|X'| = \lambda$ or $|Y'| = \mu$), then we say that X *forces* the cardinality μ for Y (Y *forces* the cardinality λ for X; X and Y mutually force cardinalities λ and μ), written $X \Rightarrow_\mu Y$ ($X\,{}_\lambda\!\Leftarrow Y$; $X\,{}_\lambda\!\Leftrightarrow_\mu Y$). Here X' (respectively, Y') is called a *copy* of X (Y).

Replacing λ by $\leq\lambda$ or $\geq\lambda$ or $<\lambda$ or $>\lambda$, and/or μ by $\leq\mu$ or $\geq\mu$ or $<\mu$ or $>\mu$, we get a series of related notions and notations, for instance, $X_{\leq\lambda,\mathcal{N}}\!\Leftrightarrow_{>\mu,\mathcal{N}} Y$.

Having $X \Rightarrow_{\mu,\mathcal{M}} Y$, $X \Rightarrow_{\leq\mu,\mathcal{M}} Y$, or $X \Rightarrow_{\geq\mu,\mathcal{M}} Y$ for any X we write $\Rightarrow_{\mu,\mathcal{M}} Y$, $\Rightarrow_{\leq\mu,\mathcal{M}} Y$, or $\Rightarrow_{\geq\mu,\mathcal{M}} Y$ respectively.

Example 4.4.2. Taking a structure $\mathcal{M}$ with infinite disjoint unary predicates P_0 and P_1 of cardinalities λ and μ respectively and without any links, we have $X_{\lambda,\mathcal{M}}\!\Leftrightarrow_{\mu,\mathcal{M}} Y$ for $X = P_0(\mathcal{M})$ and $Y = P_1(\mathcal{M})$, whereas $X_{\lambda}\!\not\Leftrightarrow_{\mu} Y$, even $X \not\Rightarrow_{\mu} Y$ and $X_{\lambda}\!\not\Leftarrow Y$. If $\lambda \geq \mu$, we can extend the language for $\mathcal{M}$ by a function $f\colon P_0 \to P_1$ which guarantee $X \Rightarrow_{\mu} Y$.

The example confirms that the relation $X \Rightarrow_{\mu} Y$ is not preserved under language restrictions.

The following properties for definable sets are obvious.

1.: If Y is finite, then $X \Rightarrow_n Y$ for some unique $n \in \omega$ and for any/some X. Conversely, if Y is infinite, then $X \not\Rightarrow_n Y$ for any $n \in \omega$ and for any/some X. Thus, we have $\Rightarrow_{<\omega} Y$ for finite Y and $\not\Rightarrow_{<\omega} Y$ for infinite one.

2.: If Y is infinite, then $X \Rightarrow_{\geq\omega} Y$ for any/some X. Conversely, if Y is finite, then $X \not\Rightarrow_{\geq\omega} Y$ for any/some X. Thus we have $\Rightarrow_{\geq\omega} Y$ for infinite Y and $\not\Rightarrow_{\geq\omega} Y$ for finite ones.

3.: (Monotony) If $X \Rightarrow_{\leq\lambda} Y$, $\lambda \leq \mu$ and $Y \subseteq Z$, then $X \Rightarrow_{\leq\mu} Z$.

4.: (Transitivity) If X, Y, Z are definable sets in a structure $\mathcal{M}$, $X \Rightarrow_{\lambda} Y'$ and $Y' \Rightarrow_{\mu} Z'$ for any copies Y' and Z' of Y and Z, respectively, then $X \Rightarrow_{\mu} Z$. The same is true replacing λ by $\leq\lambda$ or $\geq\lambda$, and μ by $\leq\mu$ or $\geq\mu$.

5.: If X_i are disjoint subsets of Y and $\Rightarrow_{\geq\lambda_i} X_i$, $i \in \kappa$, then $\Rightarrow_{\geq\sum\limits_{i\in\kappa}\lambda_i} Y$. In particular, if Y is implied by λ disjoint nonempty definable sets then $\Rightarrow_{\geq\lambda} Y$.

6.: If X and Y have a definable function $f\colon X \to Y$ and $|Y| = \lambda$ then $X_{\geq\lambda}\!\Leftarrow Y$. In particular, if X and Y have a definable bijection $f\colon X \leftrightarrow Y$, then for any λ $X_{\lambda}\!\Leftrightarrow_{\lambda} Y$.

Property 6 can be generalized taking, for instance, an infinite Y and a definable relation $R \subset X \times Y$ such that each $a \in X$ has *uniformly* finitely many R-images, i.e., the sets $R(a, \mathcal{M})$ have bounded finite cardinalities. In such an instance we have $X_{\geq|Y|}\!\Leftarrow Y$. Similarly, having a definable *almost bijection* $R \subset X \times Y$ with

uniformly finitely many R-images and R-preimages, then, for infinite X and Y, we get $|X| = |Y|$ and, moreover, $X\,_{|X|}\!\Leftrightarrow_{|X|} Y$.
The following example shows that the condition $X\,_\lambda\!\Leftrightarrow_\lambda Y$ does not imply that there exists a definable almost bijection.

Example 4.4.3. Take a structure $\mathcal{M}$ in a language $\{P_0, P_1, f\}$ with disjoint unary predicates P_0 and P_1 and a language with unary acyclic operation f in universe $P_0 \cup P_1$ such that each element $a \in P_i$ has infinite $f^{-1}(a) \subset P_{1-i}$, $i \in \{0, 1\}$. By the definition of f, we have $|P_0| = |P_1|$ for any $\mathcal{N} \equiv \mathcal{M}$. Thus, $X\,_\lambda\!\Leftrightarrow_\lambda Y$, where $X = P_0(\mathcal{M})$ and $Y = P_1(\mathcal{M})$.

Note that we have a similar effect, with $X\,_\lambda\!\Leftrightarrow_\lambda Y$, replacing f by a relation $R \subset (X \times Y) \cup (Y \times X)$ with infinite $R^{-1}(a)$ and uniformly bounded finite $R(a)$, $a \in X \cup Y$.

For definable sets $X = \phi(\mathcal{M}, \bar{a})$ and $Y = \psi(\mathcal{M}, \bar{b})$ we denote the set of solutions by $X \vee Y$, in $\mathcal{M}$, of a formula $\phi(\bar{x}, \bar{a}) \vee \psi(\bar{y}, \bar{b})$, by $X \wedge Y$ — of $\phi(\bar{x}, \bar{a}) \wedge \psi(\bar{y}, \bar{b})$, by $\neg X$ — of $\neg\phi(\bar{x}, \bar{a})$, by $\forall x X$ — of $\forall x \phi(\bar{x}, \bar{a})$, by $\exists x X$ — of $\exists x \phi(\bar{x}, \bar{a})$.

7.: If $\Rightarrow_{<\omega} X$ and $\Rightarrow_{<\omega} Y$, then $\Rightarrow_{<\omega} X \wedge Y$, $\Rightarrow_{<\omega} X \wedge \neg Y$, $\Rightarrow_{<\omega} X \vee Y$.

8.: If $X \Rightarrow_{\leq\lambda} Y$ and $X \Rightarrow_{\leq\lambda} Z$ for some infinite λ, then $X \Rightarrow_{\leq\lambda} Y \vee Z$. If $X \Rightarrow_\lambda Y$ and $X \Rightarrow_{<\lambda} Z$ for some infinite λ, then $X \Rightarrow_\lambda Y \vee Z$.

9.: If $X \Rightarrow_\lambda Y$ for some infinite λ and $\Rightarrow_{<\omega} Z$, then $X \Rightarrow_\lambda Y \vee Z$ and $X \Rightarrow_\lambda Y \wedge \neg Z$.

10.: For every variable x, if $X \Rightarrow_{\leq\lambda} Y$, then $X \Rightarrow_{\leq\lambda} \forall x Y$ and $X \Rightarrow_{\leq\lambda} \exists x Y$, and, by Monotony, if $X \Rightarrow_{\leq\lambda} \exists x Y$ then $X \Rightarrow_{\leq\lambda} \forall x Y$

The properties above allow us to define a calculus $\mathcal{D}$ with formulas for definable sets, and calculi $\mathcal{D}_\mathcal{M}$ for definable sets in structures $\mathcal{M}$. If $|\mathcal{M}| = \lambda_0 \geq \omega$, then the calculus $\mathcal{D}$ for the family of definable sets X in $\mathcal{M}$ is restricted to the cardinalities $\lambda \leq \lambda_0$, producing the calculus $\mathcal{D}_{\lambda_0}$ together with the calculus $\mathcal{D}_\mathcal{M}$. In particular, if $\lambda_0 = \omega$, then we have the calculus $\mathcal{D}_\omega$ saying that definable sets are either finite or infinite, without comparing infinite cardinalities.

Definition 4.4.4. If for any set X of elements in a model $\mathcal{M}$ of a theory T, then the union of sets of solutions of formulas $\varphi(x, \bar{a})$, $\bar{a} \in X$, such that $\mathcal{M} \models \exists^{=n} x\, \varphi(x, \bar{a})$

for some $n \in \omega$ is said to be an *algebraic closure* of X. The algebraic closure of X (in $\mathcal{M}$) is denoted by $\mathrm{acl}(X)$.

Definition 4.4.5. The sets X and Y are *cardinality independent* if for any λ and μ, $X \not\Rightarrow_\mu Y$ and $X\,{}_\lambda\!\not\Leftarrow Y$. Otherwise, if, for instance, $X \Rightarrow_\mu Y$, we say that Y is *cardinality dependent* with respect to X. Having $X'{}_\lambda\!\Leftrightarrow_\mu Y'$ for some λ, μ with X' and Y' as in Definition 4.4.1, we say that X and Y are *cardinality dependent*, and if $\lambda = \mu$ we write $X \sim Y$.

Clearly, all finite definable sets are cardinality dependent. At the same time, the following theorem, describing cardinality links between infinite definable sets X and Y, shows that if $X\,{}_\lambda\!\Leftrightarrow_\mu Y$ then $\lambda = \mu$.

Theorem 4.4.6. (Trichotomy Theorem) *For any infinite definable sets X and Y in a structure $\mathcal{M}$, either X and Y are cardinality independent, or exactly one of X and Y is cardinality dependent with respect to another, or $X \sim Y$.*

Proof. Let $X = \phi(\mathcal{M}, \bar{a})$, $Y = \psi(\mathcal{M}, \bar{b})$, $\bar{a}, \bar{b} \in M$, $|X| = \lambda$, $|Y| = \mu$. Without loss of generality we assume that $\bar{a} \in X$ and $\bar{b} \in Y$. Considering $\mathrm{acl}(X)$ and $\mathrm{acl}(Y)$ we have the following possibilities:

- **i):** $Y' \backslash \mathrm{acl}(X) \neq \emptyset$ and $X' \backslash \mathrm{acl}(Y) \neq \emptyset$ for some $Y' = \psi(\mathcal{N}_1, \bar{b})$ and $X' = \phi(\mathcal{N}_2, \bar{a})$, where $\mathcal{M} \prec \mathcal{N}_1$, $\mathcal{M} \prec \mathcal{N}_2$, $X = \phi(\mathcal{M}, \bar{a}) = \phi(\mathcal{N}_1, \bar{a})$, $Y = \psi(\mathcal{M}, \bar{b}) = \psi(\mathcal{N}_2, \bar{b})$;
- **ii):** either $Y' \subseteq \mathrm{acl}(X)$ or $X' \subseteq \mathrm{acl}(Y)$ but not both for all X' and Y' as above;
- **iii):** $Y' \subseteq \mathrm{acl}(X')$ and $X' \subseteq \mathrm{acl}(Y')$ for all $X' = \phi(\mathcal{N}, \bar{a})$ and $Y' = \psi(\mathcal{N}, \bar{b})$, where $\mathcal{M} \prec \mathcal{N}$ or $\mathcal{N} \prec \mathcal{M}$, $\bar{a}, \bar{b} \in N$.

In case i), X and Y are cardinality independent since, using compactness, the structures $\mathcal{N}_1$ and $\mathcal{N}_2$ can be chosen with unbounded cardinalities for $Y' \setminus \mathrm{acl}(X)$ and $X' \setminus \mathrm{acl}(Y)$, preserving X and Y, respectively.

In case ii) we immediately get exactly one of the conditions $X \Rightarrow_\mu Y$ and $X\,{}_\lambda\!\Leftarrow Y$ implying that exactly one of X and Y is cardinality dependent with respect to another.

Case iii) means that cardinalities of X' and Y' correlate. These cardinalities are unbounded but $|Y'| \leq |\mathrm{acl}(X')| \leq \max\{|X'|, \Sigma(\mathcal{M}), \omega\}$ and $|X'| \leq |\mathrm{acl}(Y')| \leq \max\{|Y'|, \Sigma(\mathcal{M}), \omega\}$. Choosing a cardinality $\lambda > \max\{\Sigma(\mathcal{M}), \omega\}$ for X' and Y'

we observe that the conditions $Y' \subseteq \mathrm{acl}(X')$ and $X' \subseteq \mathrm{acl}(Y')$ imply $X'_{\lambda}\Leftrightarrow_{\lambda} Y'$. Hence, $X \sim Y$. □

Remark 4.4.7. Arguments for Theorem 4.4.6 show that $\sim$ is an equivalence relation for every structure $\mathcal{M}$. By the transitivity and the reflexivity for $\Rightarrow$. we have a preorder on the set of definable sets in $\mathcal{M}$ forming an order for $\sim$-quotients.

Definition 4.4.8. [**64**] (cf. [**65, 49, 57, 6**]). Let T be a complete theory, $\mathcal{M} \models T$. We consider *closed* (under the natural topology) nonempty sets $\mathbf{p}(x) \subseteq S^1(\varnothing)$, i. e., sets $\mathbf{p}(x)$ such that $\mathbf{p}(x) = \bigcap\limits_{i\in I}[\phi_{\mathbf{p},i}(x)]$, where $[\phi_{\mathbf{p},i}(x)] \rightleftharpoons \{p(x) \in S^1(\varnothing) \mid \phi_{\mathbf{p},i}(x) \in p(x)\}$ for some formulas $\phi_{\mathbf{p},i}(x)$ of T.
For closed sets $\mathbf{p}(x), \mathbf{q}(y) \subseteq S(\varnothing)$ of types, realized in $\mathcal{M}$, we take all $(\mathbf{p},\mathbf{q})$-*preserving*, $(\mathbf{p},\mathbf{q})$-*semi-isolating*, $(\mathbf{p} \to \mathbf{q})$-, or $(\mathbf{q} \leftarrow \mathbf{p})$-*formulas* $\phi(x,y)$ of T, i. e., formulas for which if $a \in M$ realizes a type in $\mathbf{p}(x)$, then every solution of $\phi(a,y)$ realizes a type in $\mathbf{q}(y)$. Now, for each such a formula $\phi(x,y)$, we define a binary relation $R_{\mathbf{p},\phi,\mathbf{q}} \rightleftharpoons \{(a,b) \mid \mathcal{M} \models \phi(a,b) \wedge \mathbf{p}(a)\}$, where $\models \mathbf{p}(a)$ means that a realizes some type in $\mathbf{p}$. If $(a,b) \in R_{\mathbf{p},\phi,\mathbf{q}}$, (a,b) is called a $(\mathbf{p},\phi,\mathbf{q})$-*arc*.
If, in addition, $\phi(x,y)$ is a $(\mathbf{p} \leftrightarrow \mathbf{q})$-*formula*, i. e., it is both a $(\mathbf{p} \to \mathbf{q})$- and a $(\mathbf{q} \to \mathbf{p})$-formula, then the set $[a,b] \rightleftharpoons \{(a,b),(b,a)\}$ is said to be a $(\mathbf{p},\phi,\mathbf{q})$-*edge*.

The definition above can be obviously transformed for sets $\mathbf{p}(\bar{x}), \mathbf{q}(\bar{y}) \subseteq S(A)$, where types in $\mathbf{p}(\bar{x})$, respectively in $\mathbf{q}(\bar{y})$, have a same arity for free variables.

Remark 4.4.9. The arguments above for definable sets stay valid for type-definable sets [**50**], i.e., for sets $X = \mathbf{p}(\mathcal{M}) = \{\bar{a} \in M \mid \mathcal{M} \models \mathbf{p}(\bar{a})\}$. Here, for links between type-definable sets X and $Y = \mathbf{q}(\mathcal{M})$, we use $(\mathbf{p},\mathbf{q})$-preserving formulas and $(\mathbf{q},\mathbf{p})$-preserving formulas.

In particular, if a $(\mathbf{p},\mathbf{q})$-preserving formula ϕ defines a bijection between arbitrary copies X' and Y' of X and Y respectively, then ϕ defines $(\mathbf{p},\phi,\mathbf{q})$-edges confirming the relation $X\,_{\lambda}\Leftrightarrow_{\lambda} Y$. At the same time, $(\mathbf{p},\phi,\mathbf{q})$-arcs, as above, can confirm the relation $X \Rightarrow_{\leq\lambda} Y$.

4.5. Meetings of cardinality contradictions and criteria of existence of generic structure

In this section we prove the criteria of existence of generic structure in terms of forcing for cardinalities.

Definition 4.5.1. We say that a generative class $(\mathbf{D}_0;\leqslant)$ *forces* cardinality λ (respectively, $\leq\lambda$, $\geq\lambda$, $<\lambda$, $>\lambda$) for a (type-)definable set X, written $(\mathbf{D}_0;\leqslant)\Rightarrow_\lambda X$ ($(\mathbf{D}_0;\leqslant)\Rightarrow_{\leq\lambda} X$, $(\mathbf{D}_0;\leqslant)\Rightarrow_{\geq\lambda} X$, $(\mathbf{D}_0;\leqslant)\Rightarrow_{<\lambda} X$, $(\mathbf{D}_0;\leqslant)\Rightarrow_{>\lambda} X$) if the union of $\Phi(A)$-fragments for X, where $\Phi(A)\in\mathbf{D}_0$, has cardinality λ (a cardinality $\leq\lambda$, $\geq\lambda$, $<\lambda$, $>\lambda$).

For a generative class $(\mathbf{D}_0;\leqslant)$, we say that a (type-)definable set X *meets a contradiction for its cardinality* if $(\mathbf{D}_0;\leqslant)\Rightarrow_{\leq\lambda} X$ and $(\mathbf{D}_0;\leqslant)\Rightarrow_{>\lambda} X$ for some cardinality λ.

Example 4.5.2. (cf. [**33**, Proof of Theorem 2.1]) Consider a structure $\mathcal{N}$ in Example 4.3.14 with $\mu\geq\omega$ sequential equivalence relations E_i, whose chains of E_i-classes, $i\in\mu$, have unique elements in intersections and are forming a unary predicate P_0. Now we extend P_0 and the language $\{P_0\}\cup\{E_i\mid i\in\mu\}$ by:

1): a disjoint unary predicate P_1 which is divided by $\lambda>2^\mu$ disjoint infinite unary predicates Q_i;

2): a function $f\colon P_0\to P_1$ such that $f^{-1}(a)$ is infinite for every $a\in P_1$.

The resulted hypothetical structure $\mathcal{N}'$ has the universe $P_0\cup P_1$. We denote the generative class consisting of all diagrams being copies of quantifier free diagrams for finite subsets of $\mathcal{N}'$ by $(\mathbf{D}_0;\subseteq)$. As shown in Example 4.3.14, $|P_0|=2^\mu$. Therefore $(\mathbf{D}_0;\subseteq)\Rightarrow_{2^\mu} P_0$.
At the same, time by Property 5 for λ definable sets $X_i=Q_i$ and $Y=P_1$ we have $\Rightarrow_{>2^\mu} P_1$, and by Property 6 the definable function $f\colon P_0\to P_1$ confirms that $P_0\ {}_{>2^\mu}\!\Leftarrow P_1$.
Having $(\mathbf{D}_0;\subseteq)\Rightarrow_{2^\mu} P_0$ and $(\mathbf{D}_0;\subseteq)\Rightarrow_{>2^\mu} P_0$ we observe that X meets a contradiction for its cardinality. Hence the $(\mathbf{D}_0;\subseteq)$-generic structure $\mathcal{N}'$ does not exist.

The following example modifies Example 4.5.3 producing a meeting of cardinality contradiction for type-definable sets.

Example 4.5.3. We take Example 4.5.3 and replace the structure of P_1 with $\mu'>2^\mu$ sequential equivalence relations E'_j, whose chains of E'_j-classes, $j\in\mu$, have infinitely many elements in intersections. Now, we observe that formulas $P_0(x)$ and $P_1(x)$ isolate complete 1-types. Introducing a language function f, as in Example 4.5.3, we again meet the cardinality contradiction for $X=P_0$ which is forced with $Y=P_1$.

This example can be easily transformed replacing definable sets P_0 and P_1 by correspondent type-definable sets with non-isolated $p_0(x), p_1(x) \in S(\emptyset)$. To this purpose, it suffices to introduce two sequences of predicates $P_{0,n}, P_{1,n}$, $n \in \omega$, satisfying the following conditions:

i): $P_{k,0} = P_k$, $k \in \{0,1\}$;

ii): $P_{k,n} \supset P_{k,n+1}$, $k \in \{0,1\}$, where $P_{0,n} \setminus P_{0,n+1}$ consists of infinitely many E_0-classes and $P_{1,n} \setminus P_{1,n+1}$ consists of infinitely many E'_0-classes;

iii): if $a \in P_{1,n} \setminus P_{1,n+1}$ then $f^{-1}(a) \in P_{0,n} \setminus P_{0,n+1}$;

iv): $\bigcap\limits_{n\in\omega} P_{kn}$ has infinitely many E_0-classes.

We denote the (unique) complete nonisolated 1-type which is isolated by the set $\{P_{k,n}(x) \mid n \in \omega\}$, $k \in \{0,1\}$ by $p_k(x)$.

The formula $f(x) \approx y$ defines links between a type-definable set X of realizations of $p_0(x)$ and a type-definable set Y of realizations of $p_1(y)$. As in Example 4.5.3 we have $|X| = 2^{\mu}$, $|Y| = 2^{\mu'}$, $|X| < |Y|$ by choice of μ', but the links with respect to $f(x) \approx y$ imply $|X| \geq |Y|$. Thus, X meets the cardinality contradiction.

Note that since formula-definable sets consist of type-definable sets, lower cardinality bounds for type-definable sets imply similar bounds for formula-definable ones.

Now we will show that meetings of cardinality contradictions for (type)-definable sets are the only reason why generic structures, for given generative class, can not exist.

Theorem 4.5.4. *For any generative class* $(\mathbf{D}_0;\leqslant)$ *the following conditions are equivalent:*

(1) *there exists a* $(\mathbf{D}_0;\leqslant)$*-generic structure;*

(2) *there are no type-definable sets* X *constructed with respect to* $(\mathbf{D}_0;\leqslant)$ *such that these* X *meet contradictions for their cardinality;*

(3) *there are no definable sets* X *constructed with respect to* $(\mathbf{D}_0;\leqslant)$ *such that these* X *meet contradictions for their cardinality.*

Proof. (1) $\Rightarrow$ (2) is obvious since having a structure $\mathcal{M}$ we can not meet cardinality contradictions for (type-)definable sets in $\mathcal{M}$. (2) $\Rightarrow$ (3) is also obvious because each definable set is type-definable.

(3) $\Rightarrow$ (1). If $\mathbf{D}_0$ consists of copies of a countable set of diagrams, then a $(\mathbf{D}_0;\leqslant)$-generic structure exists by Theorem 4.1.11.

Now, we assume that there are only $\nu > \omega$ diagrams the copies of which form the generative class $(\mathbf{D}_0;\leqslant)$ and start to construct a required generic structure $\mathcal{M}$. We enumerate the set $\mathcal{X}$ of all pairs $(\Phi(X),\Psi(X,Y))$, $X\cap Y=\varnothing$, such that $\Phi(A)\leqslant\Psi(B)$ for some sets A and B, $A\subseteq B$, and in this case we assume that any such pair is enumerated ν times. Since the set of all considered types $\Phi(X)$ has the cardinality ν, then the set of enumerated pairs has also the cardinality ν: $\mathcal{X}=\{(\Phi_i(X_i),\Psi_i(X_i,Y_i))\mid i\in\nu\}$, $(\Phi_0(X_0),\Psi_0(X_0,Y_0))=(\Phi_0(\emptyset),\Phi_0(\emptyset))$. Using the enumeration of $\mathcal{X}$, we construct, by transfinite induction, some consistent set $S=\bigcup\limits_{i\in\nu}S_i$ of propositions in a language $\Sigma\cup\hat{K}$, where $\hat{K}$ is a set with ν constant symbols that do not occur in Σ and appear in the diagrams in $\mathbf{D}_0$. In this case, for each step i, the set S_i will equal some *closed* set $U_i(V_i)\in\cup\mathbf{D}_0$ containing all previous sets $U_j(V_j)$. "Closed" here means that each finite subset of $U_i(V_i)$ is contained in a diagram belonging to $\mathbf{D}_0$.

In the initial step $i=0$, we set $S_0\rightleftharpoons\Phi_0(\varnothing)$, where $\Phi_0(\varnothing)$ is the diagram that exists by Axiom (iv).

Assume that for a step $i>0$, the sets $S_j=U_j(V_j)$, $j<i$, are already constructed and $V_j\subset\hat{K}$.

If i is a limit ordinal we set $S_i=U_i(V_i)=\bigcup\limits_{j<i}U_j(V_j)$ and observe that S_i is consistent and closed.

If $i=j+1$, we consider the pair $(\Phi_i(X_i),\Psi_i(X_i,Y_i))$. If $U_j(V_j)$ does not contain sets $\Phi_i(A)$ such that $\Phi_i(A)\in U_j(V_j)$, we set $S_i\rightleftharpoons S_j$. If such sets $\Phi_i(A)$ exist, we enumerate all of them: $\Phi_k(A_k)$, $k\in\mu$. Now, using the d-amalgamation property, we find step-by-step a consistent set $\mathcal{D}_i$ of diagrams $\Theta\in\mathbf{D}_0$ satisfying the following conditions:

- all new constant symbols in Θ belong to $\hat{K}$;
- if $\Theta\subseteq U_j(V_j)$, then $\Theta\in\mathcal{D}_i$;

• each diagram $\Theta_1, \Theta_2 \in \mathcal{D}_i$ is identically strongly embeddable in some their amalgam Θ;

• for any $\Phi_i(A) \in U_j(V_j)$ and for some B with $\Phi_i(A) \leq \Psi_i(A, B)$, $\Psi_i(A, B)$ is strongly embeddable in some Θ, identically over $\Phi_i(A)$;

• $\mathcal{D}_i$ is generated by the items above.

The process for $\mathcal{D}_i$ is realizable since definable sets composed by diagrams Θ do not meet contradictions for their cardinality.

Now we set $S_i = \cup \mathcal{D}_i$.

Since for any step of construction the set S_i is consistent, the set $S = \bigcup\limits_{i \in \nu} S_i$ is consistent as well. Note also that, since the diagrams Θ contain quantifier free subdiagrams and these Θ, by amalgams, form a upward directed set, set K' of all constant symbols in language Σ united with the set of all constant symbols in $\hat{K}$, that appear in S, has a quantifier free diagram which is a subdiagram of S.

For K', we define an equivalence relation $\sim$ such that $c_1 \sim c_2 \Leftrightarrow (c_1 \approx c_2) \in S$. Since for any diagram $\Phi(A) \in \mathbf{D}_0$ and distinct elements $a_1, a_2 \in A$ we have $\neg(a_1 \approx a_2) \in \Phi(A)$, then each $\sim$-class contains exactly one element in K'.

Now, we define structure $\mathcal{M}$ with a universal set of $\sim$-classes $\tilde{c}$ and with predicate and functional symbols interpreted by the following rules:

• if $P \in \Sigma$ is a n-ary predicate symbol, then

$$\mathcal{M} \models P(\tilde{c}_1, \ldots, \tilde{c}_n) \Leftrightarrow P(c_1, \ldots, c_n) \in S;$$

• if $f \in \Sigma$ is a n-ary functional symbol, then

$$\mathcal{M} \models (f(\tilde{c}_1, \ldots, \tilde{c}_n) \approx \tilde{c}) \Leftrightarrow (f(c_1, \ldots, c_n) \approx c) \in S.$$

A routine check shows the correctness of the definition for satisfaction. It should be mentioned that, by the local realizability property, for any term $t(x_1, \ldots, x_l)$ of the language Σ and for any constant symbols $c_1, \ldots, c_l$ in K', there exists a constant symbol c in K' such that $(t(c_1, \ldots, c_l) \approx c) \in S$.

We are going to prove that for any $\Theta(C) \in \mathrm{D}_0$ with $\Theta(C) \subseteq S$, structure $\mathcal{M}$ satisfies all formulas deducible from $[\Theta(C)]^{C}_{\tilde{C}}$, where $\tilde{C} = \{\tilde{c} \mid c \in C\}$. Let $\varphi \rightleftharpoons \varphi(\tilde{c}_1, \ldots, \tilde{c}_n)$ be an arbitrary formula that is a consequence of $[\Theta(C)]^{C}_{\tilde{C}}$. Consider a formula $\psi \rightleftharpoons \psi(\tilde{c}_1, \ldots, \tilde{c}_n)$ that is equivalent to φ and is in the prenex normal form. We shall show that $\mathcal{M} \models \psi$, using the induction on the number of quantifiers in ψ. If ψ is quantifier free, then $\mathcal{M} \models \psi$, because language symbols in $\Sigma \cup K'$ are interpreted as described in $S \supseteq \Theta(C)$. Now consider two possible induction steps.

If ψ equals $\forall x\, \chi(x)$, then $\mathcal{M} \models \chi(a)$ for any $a \in M$, since $\Theta(C) \vdash \forall x\, \chi(x)$ implies $S \vdash \forall x\, \chi(x)$ and, hence, we get $S \vdash \chi(c)$ for any constant symbol $c \in K'$.

If ψ equals $\exists x\, \chi(x)$, then $\mathcal{M} \models \chi(b)$ for some $b \in M$, because, by the local realizability property and by the construction of S, $\Theta(C) \vdash \exists x\, \chi(x)$ implies $S \vdash \chi(c)$ for some constant symbol $c \in K'$.

Now we shall show that $\mathcal{M}$ has finite closures. Since any finite set of elements in $\mathcal{M}$ is contained in some set $\tilde{C}$ with $\mathcal{M} \models [\Theta(C)]_{\tilde{C}}^{C}$, it suffices to see that the set $\tilde{C}$ is self-sufficient in $\mathcal{M}$. This is true, however since by Axioms (iii), (vii), and by transitivity of the relation $\leqslant$, diagram $[\Theta(C)]_{\tilde{C}}^{C}$ is a strong subdiagram of any diagram $\Delta(\tilde{D}) \in \mathbf{D}_0$, where $\tilde{C} \subseteq \tilde{D}$ and $\mathcal{M} \models \Delta(\tilde{D})$.

Therefore, by the construction of S, any diagram $\Phi_i(A) \in S$, where $\Phi_i(A) \leqslant \Psi_i(B)$, is extended to some self-sufficient diagram $\Psi_i(B')$, we get $\tilde{A} \subseteq \tilde{B}'$, $\tilde{B}' \leqslant M$, and $\mathcal{M} \models \Psi_i(\tilde{B}')$.

Thus, all properties of $(\mathbf{D}_0; \leqslant)$-genericity for structure $\mathcal{M}$ are satisfied and $\mathcal{M} = \mathrm{glim}(\mathbf{D}_0; \leqslant)$. □

4.6. Lattices associated with generic structures

Note that if $(\mathbf{D}_0; \leqslant)$ be a self-sufficient class, then for $\Phi, \Psi \in \mathbf{D}_0$ with consistent $\Phi \cup \Psi$, we have $\Phi \cap \Psi \in \mathbf{D}_0$ and there is the least amalgam $\Theta \in \mathbf{D}_0$ containing $\Phi \cup \Psi$. We denote $\Phi \cap \Psi$ by $\Phi \wedge \Psi$ and Θ by $\Phi \vee \Psi$. Hence, for a $(\mathbf{D}_0; \leqslant)$-generic structure $\mathcal{M}$, the set $L(\mathcal{M}, \mathbf{D}_0, \leqslant)$ of all diagrams $\Phi \in \mathbf{D}_0$ with $\mathcal{M} \models \Phi$ forms a lattice $\mathcal{L} = \langle L(\mathcal{M}, \mathbf{D}_0, \leqslant); \wedge, \vee \rangle$. Structure $\mathcal{L}$ is called the *lattice associated with the generic structure* $\mathcal{M}$.

Note that lattice $\mathcal{L}$ can be non-distributive admitting both lattice M_3 and lattice P_5 (see Figures 2(a) and 2(b), respectively).

Indeed, having always $(\Phi \wedge \Psi) \vee (\Phi \wedge \mathrm{X}) \leqslant \Phi \wedge (\Psi \vee \mathrm{X})$ and $\Phi \vee (\Psi \wedge \mathrm{X}) \leqslant (\Phi \vee \Psi) \wedge (\Phi \vee \mathrm{X})$, we can consider a 3-element structure $\mathcal{M}$ with constants a, b, c and diagrams $\Phi = \Phi(\{a\})$, $\Psi = \Psi(\{b\})$, $\mathrm{X} = \mathrm{X}(\{c\})$, $\Theta = \Theta(\{a, b, c\})$ such that $\Phi \vee \Psi = \Phi \vee \mathrm{X} = \Psi \vee \mathrm{X} = \Theta$. We have $(\Phi \wedge \Psi) \vee (\Phi \wedge \mathrm{X}) = \Phi_0 \vee \Phi_0 = \Phi_0$ whereas

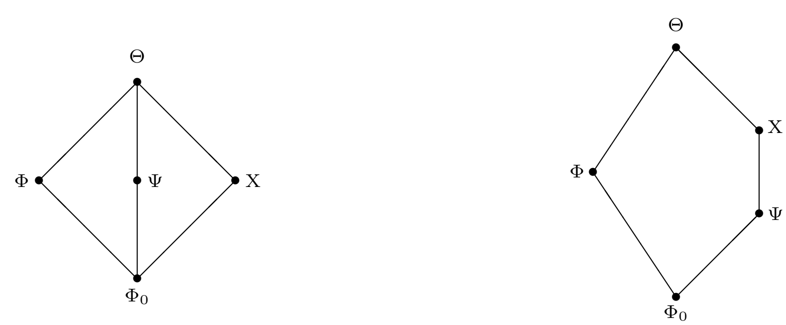
(a) (b)

Figure 2

$\Phi \wedge (\Psi \vee \mathrm{X}) = \Phi \wedge \Theta = \Phi$, and similarly $\Phi \vee (\Psi \wedge \mathrm{X}) = \Phi \vee \Phi_0 = \Phi$ whereas $(\Phi \vee \Psi) \wedge (\Phi \vee \mathrm{X}) = \Theta \wedge \Theta = \Theta$, realizing lattice M_3 (see Figure 3(a)).

Considering a 4-element structure $\mathcal{M}$ with constants a, b, c, d and diagrams $\Phi = \Phi(\{a\})$, $\Psi = \Psi(\{b\})$, $\mathrm{X} = \mathrm{X}(\{b, c\})$, $\Theta = \Theta(\{a, b, c, d\})$ such that a and b are separated by a formula $\varphi(x)$ with $\varphi(a) \in \Phi$, $\neg\varphi(b) \in \Psi$, $\Phi \vee \Psi = \Phi \vee \mathrm{X} = \Theta$ we realize lattice P_5 (see Figure 3(b)).

Thus we have the following:



(a) (b)

Figure 3

Proposition 4.6.1. For any self-sufficient class $(\mathbf{D}_0; \leqslant)$ and a $(\mathbf{D}_0; \leqslant)$-generic structure $\mathcal{M}$, structure $\langle L(\mathcal{M}, \mathbf{D}_0, \leqslant); \wedge, \vee\rangle$ is a lattice which can be a non-distributive and a non modular lattice also.

Note that in the way above arbitrary finite lattices and their superatomic limits can be constructed.

Studying lattices $\mathcal{L}$ we recall that the *height* (respectively, *width*) of $\mathcal{L}$ is the supremum of cardinalities of (anti)chains in $\mathcal{L}$. The height of $\mathcal{L}$ is denoted by $h(\mathcal{L})$ and the width is denoted by $w(\mathcal{L})$.

Since each element $\Phi(A)$ in $\mathcal{L}$ corresponds to a finite set $A \subseteq M$, the height of $\mathcal{L}$ is at most countable and $h(\mathcal{L}) < \omega$ if and only if $\mathcal{M}$ is finite. Finite $h(\mathcal{L})$ can vary from 2 to $|M| + 1$ (counting the least diagram $\Phi_0(\varnothing)$ and the greatest diagram $\Psi(M)$). Indeed, considering a structure $\mathcal{N}$ consisting of n distinct constants $c_1, \ldots, c_n$ we can take, for any positive $k \leq n$ the following chain of diagrams:

$$\Phi_0(\varnothing) \subset \Phi_1(\{a_1, \ldots, a_{n-k+1}\}) \subset \Phi_2(\{a_1, \ldots, a_{n-k+2}\}) \subset \ldots \subset \Phi_k(\{a_1, \ldots, a_n\}),$$

containing the formulas $(a_i \approx c_i)$. Lattice $\mathcal{L}$ for structure $\mathcal{N}$ with the described set of diagrams is linearly ordered with $h(\mathcal{L}) = k + 1$.

At the same time, for finite $\mathcal{M}$, $w(\mathcal{L})$ can vary from 1 to $|M|$ taking, for instance, diagrams $\Phi_1(\{a_1, \ldots, a_{n-k+1}\}), \Phi'_2(\{a_{n-k+1}\}), \ldots, \Phi'_k(\{a_n\})$ with $(a_i \approx c_i)$ for the n-element example $\langle M; c_1, \ldots, c_n\rangle$ as above. Continuing the process of adding new constants, we obtain $1 \leq w(\mathcal{L}) \leq \omega$ for countable $\mathcal{M}$.

Since $h(\mathcal{L}) \leq \omega$, we have $w(\mathcal{L}) = |M|$, if $\mathcal{M}$ is uncountable.

Summarizing the arguments we have the following:

Theorem 4.6.2. *For any self-sufficient class* $(\mathbf{D}_0; \leqslant)$ *and a* $(\mathbf{D}_0; \leqslant)$*-generic structure* $\mathcal{M}$*, the lattice* $\mathcal{L} = \langle L(\mathcal{M}, \mathbf{D}_0, \leqslant); \wedge, \vee\rangle$ *has the following characteristics:* (1) $1 < h(\mathcal{L}) \leq |M|+1$ *if* $\mathcal{M}$ *is finite and* $h(\mathcal{L}) = \omega$ *if* $\mathcal{M}$ *is infinite;* (2) $1 \leq w(\mathcal{L}) \leq |M|$ *if* $\mathcal{M}$ *is at most countable and* $h(\mathcal{L}) = |M|$ *if* $\mathcal{M}$ *is uncountable. All values in the described intervals can be realized in appropriate generic structures.*

By 4.6.2 having a linearly ordered $\mathcal{L}$, we obtain $|\mathcal{L}| \leq \omega$ and diagrams $\Phi(A)$ if $\mathcal{L}$ corresponds to finite sets A forming at most countable well-ordered set. Therefore,

$\mathcal{L}$ is well-ordered as well, and if $\mathcal{L}$ is infinite, i. e., $\mathcal{M}$ is countable, then $\mathcal{L}$ is ordered by the type ω. Thus we have the following:

Corollary 4.6.3. If the lattice $\mathcal{L} = \langle L(\mathcal{M}, \mathbf{D}_0, \leqslant); \wedge, \vee\rangle$ is linearly ordered then $\mathcal{L}$ is at most countable and well-ordered, being finite for finite $\mathcal{M}$ and having the type ω for countable $\mathcal{M}$.

Definition 4.6.4. A structure $\mathcal{N}$ is *almost rigid* if for any $\bar{a} \in N$ the type $\mathrm{tp}(\bar{a})$ has finitely many realizations in $\mathcal{N}$.

Theorem 4.6.5. *For a generic structure $\mathcal{M}$ with a class $\mathbf{K}$ of all models of $T =$ $\mathrm{Th}(\mathcal{M})$ which has finite closures, the following conditions are equivalent:*

- **(1):** *$\mathcal{M}$ has a linearly ordered lattice $\mathcal{L}$;*
- **(2):** *$\mathcal{M}$ has a linearly ordered lattice $\mathcal{L}$ modulo finitely many incomparable elements;*
- **(3):** *$\mathcal{M}$ is almost rigid and at most countable.*

Proof. (1) $\Leftrightarrow$ (2) holds, since finitely many incomparable elements can be replaced by their amalgam.

(1) $\Rightarrow$ (3). If $\mathcal{M}$ has a linearly ordered lattice $\mathcal{L}$ then $\mathcal{M}$ can not be uncountable in view of Corollary 2.3. So by Theorem 1.6, $\mathcal{M}$ is homogeneous. Now, if $\mathcal{M}$ is not almost algebraic, then for any generic class $(\mathbf{D}_0; \leqslant)$ of diagrams the copies of which are satisfied in $\mathcal{M}$ there is a diagram $\Phi(A) \in \mathbf{D}_0$ with $\mathcal{M} \models \Phi(A)$, such that $\mathrm{tp}(A)$ has infinitely many realizations in $\mathcal{M}$. Therefore, there is a copy $\Phi(A')$ of $\Phi(A)$ such that A' is incomparable with A. Thus, $\langle L(\mathcal{M}, \mathbf{D}_0, \leqslant)\rangle$ is not linearly ordered.

(3) $\Rightarrow$ (1). Clearly, finite structures have two-element lattices $\mathcal{L}$ which are linearly ordered. Now for the countable almost rigid structure $\mathcal{M}$ we construct a required linearly ordered lattice $\mathcal{L}$ and a corresponding self-sufficient generic class starting with the diagram $\Phi_0(\varnothing) = \mathrm{Th}(\mathcal{M})$. Let linearly ordered diagrams $\Phi_0(\varnothing) \subset$ $\Phi_1(A_1) \subset \ldots \subset \Phi_n(A_n)$ be already constructed with $\mathcal{M} \models \Phi_n(A_n)$ and without copies satisfied in $\mathcal{M}$. We choose an element $b_n \in M \setminus A_n$ and consider the (finite) set B_n of all elements $b \in M$ such that $\mathrm{tp}(b/A_n) = \mathrm{tp}(b_n/A)$. Now we set $A_{n+1} =$ $A_n \cup B_n$ $\Phi_{n+1}(A_{n+1}) = [\mathrm{tp}_{X_{n+1}}(A_n \cup B_n)]^{X_{n+1}}_{A_n \cup B_n}$. Clearly, $\mathcal{M} \models \Phi_{n+1}(A_{n+1})$, $\Phi_n(A_n) \subset \Phi_{n+1}(A_{n+1})$ and $\Phi_{n+1}(A_{n+1})$ does not have copies satisfied in $\mathcal{M}$.

Extending the set $\{\Phi_n(A_n) \mid n \in \omega\}$ by all possible copies we form a generative class $(\mathbf{D}_0; \leqslant)$ with the linearly ordered lattice $\langle L(\mathcal{M}, \mathbf{D}_0, \leqslant); \wedge, \vee\rangle$. $\square$

Since ω-saturated (infinite) structures are not almost rigid we obtain the following:

Corollary 4.6.6. *There are no infinite ω-saturated structures $\mathcal{M}$ with linearly ordered lattices $\mathcal{L}$.*

By the definition the lattice $\mathcal{L}$ has relative complements if and only if for each $\Phi(A) \leqslant \mathrm{X}(B)$ there is a diagram $\Psi(C) \leqslant \mathrm{X}(B)$ (being the relative complement of $\Phi(A)$) such that $C \subseteq B \setminus A$ and $\mathrm{X}(B) = \Phi(A) \vee \Psi(C)$. Since unions $\Phi(A) \vee \Psi(C)$ are always diagrams over finite sets, the distributive lattice $\mathcal{L}$ with relative complements is a Boolean algebra if and only if $\mathcal{L}$ is finite, i. e., $\mathcal{M}$ is finite.

Summarizing the arguments we have the following:

Proposition 4.6.7. (1) *The lattice $\mathcal{L}$ has relative complements if and only if for each $\Phi(A) \leqslant \mathrm{X}(B)$ there is a diagram $\Psi(C) \leqslant \mathrm{X}(B)$ such that $C \subseteq B \setminus A$ and $\mathrm{X}(B) = \Phi(A) \vee \Psi(C)$.*
(2) *The distributive lattice $\mathcal{L}$ with relative complements is a Boolean algebra if and only if $\mathcal{M}$ is finite.*

Clearly, any finite structure $\mathcal{M}$ has only finite lattices $\mathcal{L}$ such that being Boolean algebras their cardinalities $|\mathcal{L}|$ can vary from 2 to $2^{|M|}$.

Considering the dynamics of the lattices $\mathcal{L}$ and applying Proposition 1.7 we note the following.

Proposition 4.6.8. *Let $\mathcal{M}$ and $\mathcal{N}$ be any ω-homogeneous structures with $\mathcal{N} \prec \mathcal{M}$; $(\mathbf{D}_0; \leqslant)$ and $(\mathbf{D}'_0; \leqslant')$ be self-sufficient classes such that $\mathcal{M}$ is $(\mathbf{D}_0; \leqslant)$-generic, $\mathcal{N}$ is $(\mathbf{D}'_0; \leqslant')$-generic, and $(\mathbf{D}'_0; \leqslant')$ is a restriction of $(\mathbf{D}_0; \leqslant)$ coordinated with $\mathcal{N} \prec \mathcal{M}$; $\mathcal{L} = \langle L(\mathcal{M}, \mathbf{D}_0, \leqslant); \wedge, \vee\rangle$, $\mathcal{L}' = \langle L(\mathcal{N}, \mathbf{D}'_0, \leqslant'); \wedge, \vee\rangle$. Then $\mathcal{L}'$ is isomorphic to a quotient of $\mathcal{L}$. If, moreover, the restriction $(\mathbf{D}'_0; \leqslant')$ of $(\mathbf{D}_0; \leqslant)$ is conservative then $\mathcal{L}'$ is a sublattice of $\mathcal{L}$.*

4.7. Conclusions

Ravi Rajani is developing the theory of generic structures at a general, unified level but it is not strictly categorical. Furthermore, Kubis in [40] is developing a

category-theoretic approach to homogeneous structures and in [39] introducing the Katetov functors which provide a universal way to construct Fraïssé limits.
Our goal is to unify all these different approaches under the Institution-Independent model theory. The institution theory gives us a tool to expand the main results of the generic constructions and generic limits in abstract logical systems.

Regarding the theory of generic limits and generic constructions for standard first order logic we will work in the institutional first order logic framework, based on $\mathcal{D}$-first order fragment [59]. The first goal is to merge the institutional definition of diagrams with the standard theory of diagrams in a universal way. Furthermore, we will attempt to give abstract categorical conditions on partial ordering of a collection of Σ-diagrams with respect to Definition 4.1.3. To this end, we introduce the Σ_d-amalgamation property, the semantic existential *local realizability property* and the Σ_d-uniqueness property expanding Definition 4.1.3.

To the extent that cardinalities of generic structures as well as their languages can unbounded it is natural to generalize generic constructions for the restrictions $\mathcal{I}_h = \left(\mathbf{Sig}^{\mathcal{I}_h}, \mathbf{Sen}^{\mathcal{I}_h}, \mathbf{Mod}^{\mathcal{I}_h}, \models^{\mathcal{I}_h}\right)$ of institutions $\mathcal{I}$ to the class of homogeneous structures. These restrictions $\mathcal{I}_h$ are called *h-institutions*.

Taking a h-institution $\mathcal{I}_h$ and applying Theorem 4.1.12 we can replace theories and their homogeneous models with appropriate generative classes obtaining a *g-institution* $\mathcal{I}_g$. Since by Theorems 4.1.11 and 4.1.12 generative classes allow to reconstruct countable generic structures, up to isomorphism, and their theories, we have:

Theorem 4.7.1. *Any h-institution $\mathcal{I}_h$ can be transformed to a g-institution $\mathcal{I}_g$ such that countable models in $\mathbf{Mod}^{\mathcal{I}_h}$ can be reconstructed by generative classes in $\mathcal{I}_g$, up to isomorphism, and sentences in $\mathbf{Sen}^{\mathcal{I}}$ satisfied by these models as well.*

An advantage of g-institutions is that these categories are syntactic, do not contain semantic objects allowing to describe semantic links in syntactic way.

Again applying Theorems 4.1.11 and 4.1.12 we immediately get the following theorem clarifying links between institutions and generic structures via first order fragments of these institutions.

Theorem 4.7.2. *Let* $\mathcal{I} = \left(\mathbf{Sig}^{\mathcal{I}}, \mathbf{Sen}^{\mathcal{I}}, \mathbf{Mod}^{\mathcal{I}}, \models^{\mathcal{I}}\right)$ *be an Institution with* $\mathcal{D} \subseteq$ **Sig** *of signature morphisms then every countable* $\mathcal{D}$-***First Order Fragment*** *has generic models and therefore can be represented by them.*

Chapter 5

Applications of Logic in A.I.

ABSTRACT. In this chaper we present an early stage of a decision-making system tasked with selecting suitable statistical models by employing mathematical interpretations. This work deals with the new field of Explainable AI (XAI) and it combines concepts and principles from the fields of statistics, decision theory, artificial intelligence, symbolic artificial intelligence and modelling of human behavior. The primary goal of the proposed approach is to address errors that occur resulting from the misuse of statistical methods. In practice, such errors often occur either owing to the use of inappropriate statistical methods or the wrong interpretations of results. The proposed approach relies on the LPwNF (Logic Programming without Negation as Failure) framework of non-monotonic reasoning as provided by Gorgias [**36**, **35**]. The proposed system enables automatic selection of the appropriate statistical method, based on the characteristics of the problem and the sample. The expected impact could be twofold: it can enhance the use of statistical systems like R and, combined with a Java-based interface of Gorgias, make non-monotonic reasoning easy to use in the proposed context.

5.1. Introduction

In the age of Big Data, understanding and interpreting scientific results from the fields of Artificial Intelligence, Artificial Learning and Statistics is becoming all the more pressing. What is at stake is not only popularizing but, above all, adopting AI techniques from scientific fields such as medical science, where AI has proven to be both of great use and a necessity. Interpretation depends on a multitude of factors. It depends not only one selecting a suitable model with explainable decisions, but also on whether the whole modelling process is accurate and consistent. In other words, are accurate data being used? Is the applied model mathematically suitable? Is the result valid?

There is a pressing need for explanatory AI in healthcare. The motivation for a theory explaining AI models in healthcare is clear; end users as well as the critical significance of accurate prediction mandate transparency – for the sake of both user involvement and patient safety. However, a simple interpretation of the predictions of an algorithm process is insufficient. How interpretations are communicated to end users, how they become incorporated in users' work flows and, finally, how they are interpreted or misinterpreted by end users is a matter deserving of thorough investigation. Two types of interpretation/interpretability can be found in XAI, labeled as post-hoc and ante-hoc (Fellmeth and Horwitz, 2009). Post-hoc interpretability refers to "after the fact", i.e. that which happens after the fact in question. In the case of XAI, it explains what the model predicts based on what is readily explainable. Ante-hoc interpetability refers to "before the fact", i.e. that which happens before the fact in question. In this paper, when referring to interpreting a conclusion, we refer both to a set of justifications for a specific result and to a rigorous description of the decision making process.

However, being able to provide a clear explanation is predicated upon an understanding of the behavior of users, who do not necessarily grasp all the details of the modelling process followed. Therein lies the innovation of this research paper. However, despite its significance, describing a system that approximates human behavior remains a problematic endeavor. Mathematical theories being a part of symbolic logic mandate a rigorous and mathematically consistent mode of description.

For example, if someone mentioned that there is a plumber from Athens, it would come naturally to most people to conclude that there are people from Athens who are not plumbers. In the framework of a logical system, a syllogism of that type is not obvious at all. Take another example – which might be considered self referential: if you have been reading this thesis beginning from page 1 and have reached this line, you would readily assume that the whole thesis is written in English. This type of syllogism, characterized by a high degree of scientific uncertainty, is very difficult to frame within a classic logical system.

Propositional calculus and first order logic, the two most common classic logical systems, fail to frame such syllogisms owing in part to monotonicity. Monotonicity is a property mandated by a number of logical systems – such as the ones mentioned

– and it basically consists in the following: If we have already proven a conclusion based on a set of beliefs, even if we expand the set of beliefs in question, our conclusion would not change. However, monotonicity, which is a property of logical systems, is not a property of typical systems which better describe human behavior, human modes of thinking and how humans handle information.

A further simple example of the above would be the following: If you ask someone "Can birds fly?" what are the chances that, instead of answering "yes", they would answer "it depends on the bird"? Probably not too high. This is due to the fact that the proposition "birds can fly" is equivalent to the proposition "typical birds can fly". If you were told that Tweety is a bird, you would conclude that he can fly, in the absence of evidence to the contrary. In the framework of a typical logical system, any expansion of our beliefs should not alter this conclusion about Tweety; however, if you found out that the real life Tweety is actually a penguin, your belief on whether he can fly would change, without this leading to contradiction. This is precisely why non-monotonic logical systems, which can handle this type of situation, are necessary and this is the reason we resort to Gorgias [2].

How are all these interconnected? It has been documented in the literature [56, 14, 27] that proper methodology in statistical analysis has not been always followed and various errors occur on all levels of the research process. Errors appear in the initial stages, such as during literature review when basic research questions are specified. Errors related to p-values, statistical tests, usage of statistical symbols as well as failure to summarize data and demonstrate the findings mathematically, [20, 27] all occur frequently during the important statistical/mathematical stages. Such errors highlight the need for developing a system that supports all aspects of the research process. The system ought to support researchers through the entire data analysis process.

Naturally, a heated debate has begun on the role that artificial intelligence (AI) can play in human society. A large number of scientists think that AI may not replace human thought but extend it, but that presupposes that we have mathematical theories which can explain human behaviour. A field where mathematicians try to construct theories about human behaviour is mathematical logic, the theories on human reasoning.

In the field of mathematical logic, many logical systems have been proposed for the representation of human reasoning, human behaviour etc. An important goal is to deal with contradiction. In standard monotonic logics, proof is a sequence of propositions in which if a proposal occurs simultaneously with its negation, then the system is inconsistent; this is not, however, the case with human behaviour. Our goal is not to address why this happens but how we can use non-monotonic logics to make a system more efficient.

Our case study is on the field of statistics and its application to analysis of medical data [**35**]. A large number of errors in relation to data analysis, can be found in medical research papers [**56, 32, 30, 20, 1, 14**]. Despite the fact that information is now accessible and doctors are generally considered to be highly educated researchers, errors in the use of statistical methods occur frequently. Our aim is, therefore, to build a new decision-making system – an assistant for the researcher who wants to use statistical data processing tools.

In this chapter we propose an information system that uses argumentation logic, specifically the LPwNF framework provided by Gorgias [**13, 19, 2**], to augment and enhance the use of statistical methods. The information system supports automatic selection of the proper method, based on the problem at hand the features of the statistical sample and the applicability of the statistical method according to previously defined parameters. The proposed commited approach aims to limit errors related to statistics applicability by users who often lack the knowledge and/or relevant skills. Some of the errors include conceptual misrepresentations, inappropriate use of statistical analytics software, or faulty interpretation of results [**56, 32, 30, 20, 1, 14, 27**].

The main advantage of the proposed system is based on the proof procedure employed by Gorgias; more specifically, it can give an explanation of the answer in relation to the logical rules employed. In this way, the proposed system provides correct answers and through repeated use the system trains the user regarding to proper use of statistical methods.

Furthermore, within the proposed system, it is still possible to express the logical rules that define its behavior in alignment with the relevant mathematical theorems of statistical analysis, which leads towards the verification of employed rules. This is possible because one can trace the logical rules that drive the program and interpret the result through Gorgias' functionality. Within this framework, a fully developed system could be used to provide verifiable answers in the same way a mathematician would. The system could be used in many different fields and provide valuable support to a large userbase. To demonstrate its potential, we are going to examine common errors in applied statistics and statistical software and explain how the LPwNF framework provided by Gorgias can limit the occurrence of such errors. A special purpose interface has been developed in Java to integrate with Gorgias, based on SWI-Prolog and system R so that the full benefits of argumentation logic in statistical analysis may be outlined.

5.2. System design

In logical systems, such as propositional logic and first order logic, if a theory contains a contradiction, then it is inconsistent. Furthermore, in standard logical systems, the proof system is monotonic, i.e. if, in the linear path of a proof, types $\phi, \neg\phi$ occurs then the theory is inconsistent. We can overcome this by using non-monotonic logic, which handles the simultaneous truth of a proposition and its negation in a completely different way.

We use a system like Logic Programming without Negation as Failure (LPwNF) [19]. In the context of logical programming without negation as failure (LPwNF), logical programmes are non-monotonic theories wherein each program is treated as a collection of propositions from which we must choose an appropriate subset, called an "extension".

Example 5.2.1. [19] Consider the following programme - set of rules :

$$\texttt{bird}(x) \rightarrow \texttt{fly}(x) \tag{5.1}$$

$$\texttt{penguin}(x) \rightarrow \neg\texttt{fly}(x) \tag{5.2}$$

$$\texttt{penguin}(x) \rightarrow \texttt{bird}(x) \tag{5.3}$$

$$\texttt{bird}(Tweety) \tag{5.4}$$

From this set of rules we can conclude $\texttt{fly}(Tweety)$ because we can extract it from the first rule and there is no way to extract $\neg\texttt{fly}(Tweety)$. If we add $\texttt{penguin}(Tweety)$ as a statement then we can derive $\texttt{fly}(Tweety)$ and $\neg\texttt{fly}(Tweety)$ from the rules of the program. This example may seem basic, but it is fundamentally important to our work. In essence, we described mathematically what everyone understands: although birds usually can fly, penguins, which are also birds, cannot fly. Such reasoning is not invalid, because we are used to exceptions to the rules, but when we go about modelling it within a rigid monotonic logical system, we will face a problem.

We can assume that the non-expert can understand that the previous system of rules-propositions allows for contradiction, and it thus constitutes an inconsistent set of propositions, in the framework of standard logical systems. This could lead to confusion if one does not take into account that the logical system is non-monotonic. In this way, we allow a set of rules to be consistent and at the same time to prove a proposition and its negation. In standard logical systems, we define only the consistent set of formulas, but in our underlying logic there are new definitions, such as "acceptable", "weak conclusion" and "strong conclusion" [**19, 13, 48**].

5.3. Proofs within non standard logics

Here are the basic definitions of the logical programming framework without negation as a failure:

Definition 5.3.1 (Programme)**.** We define a programme $(K, <)$ to be a set of rules K and a priority relationship $<$ which is defined over the set K.

Definition 5.3.2 (Attack)**.** Let $(K, <)$ be a programme and $T, T' \subseteq K$. T' attacks T if and only if $L, T_1 \subseteq T'$ and $T_2 \subseteq T$ exist such that:

(1) $T_1 \vdash_{min} L$ and $T_2 \vdash_{min} \neg L$

(2) $(\exists r' \in T_1, r \in T_2 \ \text{ such that } r' < r) \Rightarrow (\exists r' \in T_1, r \in T_2 \ \text{ such that } \ r < r')$

where $T \vdash_{min} L$ means that T is a minimal set such that $T \vdash L$.

However, in the priority relationship we have defined, the second rule is stronger than the first rule, so the conclusion derived from it, ¬fly($Tweety$), prevails.

Definition 5.3.3 (Consistent set of rules)**.** A set of rules T is consistent if for every ground literal k such that $T \vdash k$ we can not have $T \vdash \neg k$.

Example 5.3.4. For example, $\{a, \neg b \rightarrow a\}$ is consistent but set $\{a, \neg b \rightarrow a, b\}$ is not.

Definition 5.3.5 (Acceptable)**.** Let $(K, <)$ be a programme and T a close subset of K, then set T is acceptable if and only if

(1) T is a consistent set of rules, and

(2) for each $T' \subseteq K$, if T' attacks T then T attacks T'.

Definition 5.3.6 (Weak Conclusion)**.** Let $(K, <)$ be a programme and k be a ground literal, then k is a weak conclusion of the programme, if k applies to a maximum acceptable set (maximum admissible set) of K.

Definition 5.3.7 (Strong Conclusion)**.** Let $(K, <)$ and k be a stable literal, then k is a strong conclusion of the programme if k is valid to any maximal admissible set of K.

Given the basic definitions, we can now describe a proof procedure within the Framework of Logical Programming without negation as a failure. The proof procedure in this context consists of two types of derivation, type A derivation and type B derivation, when constructing an acceptable subset of rules.

Every type A derivation produces a part of the theory we need. Type A derivation produces one part of theory, which is sufficient to produce an original X goal, while at the same time the other derivation provides us with rules in order to fight back or to defend the theory being produced. These counterattacks are manufactured from the type B derivation. In the context of a Type B derivation, once an attack has been detected, a new type A derivation is created to defend against this attack.

Within a type B derivation, a rule r_i is given as initial input with a head l_i, which has has been used in a type A derivation. The goal of the type B derivation is to block all possible proofs of $\neg l$, the contrary of the conclusion of r_i. More specifically, the proof procedure takes into account these proofs of $\neg l_i$ whose rule r_j with $\neg l_i$ as the head is higher than rule r_i that has imported the type B derivation.

A type B derivation acts on a set of goals, which are parts of the initial goal $\leftarrow l_i$. If for such a rule r_k there is a type A derivation of the negation of the literal in the head of r_k, then this way of proving $\neg l_i$ can be counterattacked. Therefore, the rules used in the previous procedure are added to the set that has been constructed. At some step (the final step) of a type B derivation the empty clause is derived, which means that the initial goal X fails.

Example of a proof. Let there be a programme with rules:

$r_1 : fly(X) \leftarrow bird(X)$

$r_2 : \neg fly(X) \leftarrow penguin(X)$

$r_3 : penguin(X) \leftarrow walkslikepeng(X)$

$r_4 : \neg penguin(X) \leftarrow \neg flatfeet(X)$

$r_5 : bird(X) \leftarrow penguin(X)$

$r_6 : bird(T)$

$r_7 : walkslikepeng(T)$

$r_8 : \neg flatfeet(T)$

If we look at the previous set of rules in a standard logic system, then a proof for $fly(T)$ would either not exist or the system would be considered inconsistent, since we would have proof of $fly(T)$ from r_1 and r_6 and proof of $\neg fly(T)$ from r_7, r_3 and r_2. Here, however, is where non-monotonic logic is steps in so as eliminate contradictions. First of all, we have the priorities among the rules $r_2 > r_1$ and $r_4 > r_3$.

Now, if we try to prove $fly(T)$, then based on the procedure we have described first, a Type A derivation will begin which will try to prove $fly(T)$. The r_1 rule is used to prove $fly(T)$, but there is rule r_2 which attacks rule r_1 as the head of r_2; $\neg fly(X)$ is the negation of the head of r_1 and rule r_2 takes priority over r_1.

Thus begins a B-type derivation which tries to prove $\neg fly(T)$. Rules r_2 and r_3 are used to prove the negation $\neg fly(T)$. As rule r_4 exists, it begins a type A derivation to defend against this attack. This type A derivation successfully demonstrates $\neg penguin(T)$, defending against the attack on the original Type A derivation and eventually proving the original goal $fly(T)$.

5.4. System implementation

5.4.1. Gorgias. Gorgias [2] is a general framework of argumentation theory. It can form the basis of a dynamic policy framework, incomplete information notwithstanding. Gorgias' syntax is based on Prolog. The predicates of Gorgias are divided into three categories:

- abducibles
- defeasible

- background (non-defeasible)

The literals are represented by Prolog terms: A negation literal is called `neg(L)`. Furthermore, the language for the representation of the theories is defined by the rules below:

```
rule(Signature, Head, Body).
```

where

- `Head` is a literal
- `Body` is a list of literals
- `Signature` is a compound term consisting of the name of the rule along with selected variables from `Head` and the `Body` of the rule.

The special predicate `prefer/2` is used to locally codify the relative priority of the theory's rules. For example, the following means that the rule with signature `Sig1` has higher priority than the rule with signature `Sig2`, if the conditions in `Body` apply:

```
rule(Signature, prefer(Sig1,Sig2), Body).
```

Abducible literals are declared using the special predicate `abducible/2`; for example:

```
abducible(Literal, Preconditions).
```

Finally, the statement `conflict (Sig1, Sig2)` indicates that the rules with signatures `Sig1` and `Sig2` collide. In many cases `conflict (Sig1, Sig2)` is true if and only if the `Head` of the `Sig1` and `Sig2` are opposite literals.

Remark 5.4.1. An SWI-Prolog installation is needed in order to use Gorgias. Using the syntax we mentioned, we can describe our own worlds adding the following two lines to the top of the file:

```
:  - compile (′../ lib / gorgias.pl′).
:  - compile (′../ ext / lpwnf.pl′).
```

The first line loads the system while the second one is a collection of rules which define a hierarchical relationship between the arguments used by the "attack" relationship to encode the related power of arguments.

5.4.2. Representation of Knowledge and Belief. To express the rules, conflicts and preferences between them, we will normally use Prolog terms using the predicates of the Gorgias system as we have previously defined.

Example 5.4.2. To say that something can fly when it is a bird and that something cannot fly when it is a penguin we write:

```
rule (r1 (X), fly (X), [bird (X)]).
rule (r2 (X), neg (fly (X)), [penguin (X)]).
```

In this example, it is clear that these two rules are in conflict when something is both a penguin and a bird:

```
rule (f1, bird (tweety), []).
rule (f2, penguin (tweety), []).
```

To solve the conflict, we use the special hypothesis `prefer/2`. Therefore, in our example we have:

```
rule (pr1 (X), prefer (r2 (x), r1 (x)), []).
```

We saw that Gorgias has the above syntax structure `(signature, head, body)`. The head is a literal, the body is a list of literals and the signature is a sign:

```
rule (signature, preferred (Sig1, Sig2), Body)
```

But how do we get answers in Gorgias? Our answers are essentially proofs of arguments, as we saw in the previous section. The question we ask is:

```
prove(Goals, Delta)
```

Where `Goals` are a list of affirmatives or no literals and `Delta` is a proof of that question. So, in our previous example:

```
prove([neg (fly (tweety))], Delta).
```

Gorgias responds with the accepted argument which supports the question we asked:

```
Delta = [f2, r2 (tweety)]
```

This is worth mentioning because there is a preference rule named `pr1`, if we put the question

```
prove([fly (tweety)], Delta)
```

Gorgias will not be able to find a solution because the `r2` rule attacks the `r1` rule and (as it is stronger) there is no defence against it. If we subtract the preference rule `pr1` (adding % at the beginning of the line), then the system can find a solution (i.e. an acceptable argument) for both questions.

5.5. Review and maintenance

5.5.1. Chi square test implementation. For calculating the statistics we will use language R. Our information system will choose the appropriate statistical test depending on the characteristics of the sample, and, whether the conditions are satisfied, it will then use Gorgias to interpret these results and then to test hypotheses. This architecture is chosen because:

- R is the basic scientific data analysis tool and provides many ready-made statistical functions.
- The underlying logic of the interpretation is concentrated in a Gorgias rule file and it fully corresponds to what is reported in statistics literature. If in the future a part of the rules needs to be corrected or extended, the software code will not need to be modified in its entirety.
- The user, who is often not a expert, will be trained in statistical methods and will use the tools of statistical methods correctly.

In order to show that our system works, we will conduct a statistical test to check if there is any dependence between two nominal variables: A and B. One way we can do this by applying a χ^2–independent test. To apply this test, we typically represent the data from one sample size n in the form of a 2×2 **contingency table** (Table 1). This matrix is a contingency Table where we have two subpopulations (the categories of the variable B) and two cases namely “successes” and “failures” (the categories of the variable A).

The user usually considers that the conditions of the Central Limit Theorem apply, which in our case translates into the condition: all the expected frequencies being ≥ 5.

The programming language R supplies us with the appropriate functions, such as the `chisq.test ()` and `fisher.test ()`, which implements Fisher’s exact test

	Success	**Failure**	**Total**
First subpopulation	n_{11}	n_{12}	n_1
Second subpopulation	n_{21}	n_{22}	n_2
Total	n_1	n_2	n

Table 1. Contingency Table

when the conditions of the Central Limit Theorem do not apply. Afterwards, the user uses standard statistical analysis software to check the appropriate conditions to see whether two features of the population are independent.

In the following section, we will be using AL to build our application, which will explain to the user how to select the appropriate statistical test as well as which hypothesis we accept. In this case, it is not necessary for the user to know all the "details" of mathematical statistics. The programme will work as a "mathematician" and will suggest answers. This can help to avoid errors which may occur when non-experts use statistical tools, for example in medicine.

Over time, such an application can easily expand without its operation being corrupted. All that would be added to the programme would be new rules for the interpretation of statistical results and questions for the user to answer about the type of data and the type of problem.

5.5.2. Implementing a Java Interface with R. In order to enable Java to communicate with R, we will use rJava to supply R with the Java interface library. This allows us to have the results as a Java variable. We will implement–using the JRI–the `Statistics.java` class that contains the relevant functions.

The next step is to apply the χ^2 independence test or Fisher's exact test using the `chisq ()` and `fisher ()` functions. The results of these tests are stored in a different variable for each contingency table and, therefore, we can retrieve them with several functions.

5.5.3. Policy for Statistics in Gorgias. In the previous section, we described the `Statistics` class responsible for handling R and the collection of statistical results. Based on what we said in the introduction, these results will be interpreted in our system using the functions of interface `Logic`. We will utilize the underlying logic of Gorgias, which allows us to express our beliefs, as a mathematician

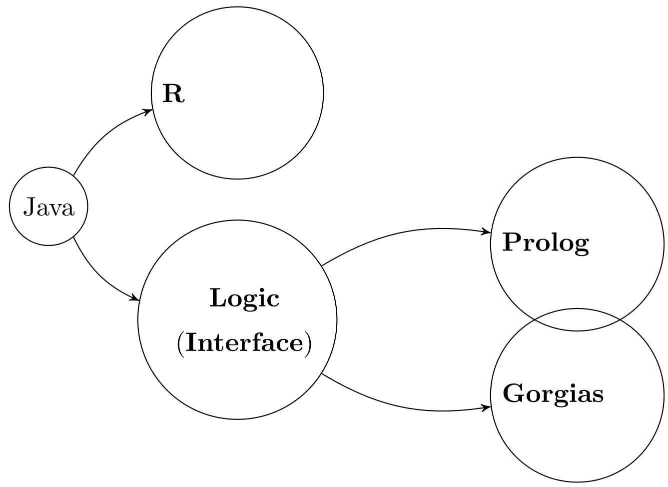


Figure 1. The basic structure of our system

would. Furthermore, through the production of the acceptable argument made by Gorgias,we can inform the user of what rules and data have led to this answer.

We have thus managed to use Gorgias as a knowledge database of the characteristics of a population. Maybe, in the future, this knowledge database can be extended to include additional queries. Furthermore, the policy defined by the Gorgias rules avoids calling functions that will not be needed. For example, Fisher's exact test is running only when the χ^2 test is invalid. Finally, thanks to the ability to set preferences between rules and higher order preferences (as described in the Chapter 5), the extension of the rules and the introduction of new rules is easy without need to change the old ones, and we can use tools such as Gorgias-B and SoDA development methodology (Software Development for Argumentation).

5.5.4. Example Usage. To demonstate how the application functions as well as the format the results are presented in, we will try three different context tables for the variables A and B from samples of four different populations. The crosstabs of our case studies are [**23**]:

The final application response and hypothesis testing with the help of Gorgias is:

```
Cannot reject null hypothes7s, data1 are independent
Why? [[h0_is_not_rejected(data1)]]
```

	Defective products	Non-defective products
First Production Process	12	288
Second Production Process	20	380

Table 2. First crosstab of our case study

	Defective products	Non-defective products
Third Production Process	12	288
Fourth Production Process	100	380

Table 3. Second crosstab of our case study

	Non-Smoker	Smoker	Heavy Smoker
Man	28	8	22
Woman	26	2	14

Table 4. Third crosstab of our case study

	Defective products	Non-defective products
Fifth Production Process	1	10
Sixth Production Process	14	15

Table 5. Fourth crosstab of our case study

```
Null hypothesis rejected, data2 are codependent.
Why? [[f1, f6, f5, chisq_is_valid(data2), f5, chisq_rejects_h0(data2)]]

Cannot reject null hypothesis, data3 are independent
Why? [[h0_is_not_rejected(data3)]]

Null hypothesis rejected, data4 are codependent.
Why? [[f1, f17, f16, fisher_rejects_h0(data4)]]

Gorgias state
-------------
consult(stats.pl).
```

```
rule(f1,significance(0.050000000000000044),[]).
rule(f2,chisq(data1),[]).
rule(f3,chisq_pvalue(data1,0.6570195719690067),[]).
rule(f4,chisq_minexpected(data1,13.714285714285714),[]).
rule(f5,chisq(data2),[]).
rule(f6,chisq_pvalue(data2,1.386470999319574E-10),[]).
rule(f7,chisq_minexpected(data2,43.07692307692308),[]).
rule(f9,chisq_pvalue(data3,0.22674842690343286),[]).
rule(f10,chisq_minexpected(data3,4.2),[]).
rule(f11,fisher(data3),[]).
rule(f12,fisher_pvalue(data3,0.24022012054762612),[]).
rule(f14,chisq_pvalue(data4,0.05485393990013243),[]).
rule(f15,chisq_minexpected(data4,4.125),[]).
rule(f16,fisher(data4),[]).
rule(f17,fisher_pvalue(data4,0.030221989999279542),[]).
```

By interpreting these results, we see that, for Table 2, a χ^2 test was valid but insufficient evidence was found and the null hypothesis H_0 was rejected. In Table 4, the result of the χ^2 test provided sufficient data to reject the null hypothesis and this is reflected in the accepted argument. As for Table 5, Fisher's exact test had to be applied because the expected frequencies that resulted from `chisq.test ()` made χ^2 invalid.

5.6. Conclusions

We have suggested that the framework of logic programming without negation as failure (LPwNF) provided by Gorgias can be applied to the use of existing statistical packages like R and provide ease of use and correctness. More specifically, a common source of errors in the use of statistics in various fields, like medicine and business, stems from the misuse of statistical methods and misinterpretation of the results, since the researchers are often not experts in the field of statistical analysis. The system we propose acts as an intelligent agent that solves this problem by automatically selecting the appropriate method and interpreting the result like a mathematician while also explaining the answer to the user with references to the relevant rules and conditions. Furthermore, by keeping the logic of Gorgias rule file, and using of custom ordering relation of rules and default reasoning (abduction), there are development benefits, such as increased encapsulation, easier extensibility without disrupting current functionality and verification of correctness due to the natural mapping of rules to the relevant mathematical theorems.

The information system we presented is in its early state. Additional work is needed to integrate other basic parametric and non-parametric statistical tests and to integrate the process of selecting the correct statistical test. The latter requires the creation of a specialized knowledge base, but the creation and integration of such a knowledge base will be a simple matter, if the fundamental structure of our cognitive bases has been described successfully. Finally, the methodology we described can bring the benefits of non-monotonic logic, to different fields, such as legal reasoning and business cases, and combine existing software packages which use different technologies, with Gorgias and thus augment their use.

Bibliography

1. Hanif A and Ajmal T, *Statistical errors in medical journals*, Ann King Edward Med Univ **17** (2011), 178–182.
2. Kakas A.C. and Mancarella P., *On the semantics of abstract argumentation*, Journal of Logic and Computation **23** (2013), no. 5, 991–1015.
3. Marco Aiello and Johan Van Benthem, *Logical patterns in space*, University of Amsterdam, 1999, pp. 5–25.
4. Marco Aiello, Ian Pratt-Hartmann, and Johan van Benthem J., *Handbook of spatial logics*, Springer, Berlin, 2007.
5. J. T. Baldwin and A. H. Lachlan, *On strongly minimal sets*, Journal of Symbolic Logic **36** (1971), no. 1, 79–96.
6. Baizhanov Bektur, Sudoplatov Sergey, and Verbovskiy Viktor, *Conditions for non-symmetric relations of semi-isolation*, Siberian Electronic Mathematical Reports **9** (2012), 161–184.
7. Jean-Yves Beziau, *Universal logic: An anthology*, Springer, New York, 2012.
8. G. Birkhoff., *Lattice theory*, American Mathematical Soc, 1964.
9. R. Blanché, *Sur l'opposition des concepts*, Theoria **9** (1953).
10. S. Burris and H. P. Sankappanavar, *A course in universal algebra*, Springer-Verlag, New York, 2011.
11. Rod Burstall and Joseph Goguen, *The semantics of clear, a specification language*, Abstract Software Specifications. Lecture Notes in Computer Science **86** (1980), 292–332.

12. Gianfranco (Eds.) Béziau, Jean-Yves; Basti, *The square of opposition. a cornerstone of thought.*, Springer, New York, 2017.

13. Kakas A. C., Mancarella P., and Dung Phan Minh, *The acceptability semantics for logic programs*, Proceedings of the Eleventh International Conference on Logic Programming (1994), 504–519.

14. Glenn T. Clark and Roseann Mulligan, *Fifteen common mistakes encountered in clinical research*, Journal of Prosthodontic Research **55** (2011), no. 1, 1 – 6.

15. Marker D., *Model theory: An introduction*, Springer, Berlin, 2002.

16. Răzvan Diaconescu, *Institution-independent ultraproducts.*, Fundam. Inform. **55** (2003), 321–348.

17. Răzvan Diaconescu, *Institution-independent model theory*, Springer, New York, 2008.

18. Răzvan Diaconescu and Petros Stefaneas, *Ultraproducts and possible worlds semantics in institutions*, Theoretical Computer Science **379** (2007), no. 1, 210–230.

19. Y. Dimopoulos and A. Kakas, *Logic programming without negation as failure*, Proceedings of the Eleventh International Conference on Logic Programming (1995), 369–384.

20. Ilker Ercan, Berna Yazıcı, Yaning Yang, Guven Ozkaya, Sengul Cangur, Bulent Ediz, and Ismet Kan, *Misusage of statistics in medical research*, European Journal of General Medicine **4** (2007), no. 3. 4. 10.29333/ejgm/82507.

21. L. Esakia, *Diagonal constructions, löb's formula and cantor's scattered space*, Logical and Semantical Studies (1981), 128–143.

22. Lawvere F., *Quantifiers and sheaves*, Actes du Congre International des Mathematiciens (1970), 329–334.

23. D. Fouskakis, *Data analysis using r*, Tsotras, Athens, 2013.

24. Joseph Goguen and Rod Burstall, *Institutions: Abstract model theory for specification and programming*, Journal of the Association for Computing Machinery **39** (1992), no. 1, 95–146.

25. G. Grätzer, *General lattice theory*, Birkhauser Basel, 2003.

26. Ichiro Hasuo, *Modal logics for coalgebras*, AIST Programming Science Group Technical Report, 2003.

27. Tyson H. Holmes, *Ten categories of statistical errors: a guide for research in endocrinology and metabolism*, American Journal of Physiology-Endocrinology and Metabolism **286** (2004), no. 4, E495–E501, PMID: 15010353.

28. Goldblatt Robert I., *Grothendieck topology as geometric modality*, Mathematical Logic Quarterly **27** (1981), no. 31-35, 495–529.

29. S. V. Sudoplatov I. Kiouvrekis, P. Stefaneas, *Definable sets in generic structures and their cardinalities*, Matematicheskie trudy **20** (2017), no. 2, 52–79.

30. Young J., *Statistical errors in medical research – a chronic disease?*, Swiss Med Wkly **137** (2007), 41–43.

31. N Jacobson, *Basic algebra i*, W. H. Freeman, New York, 1985.

32. Murphy J.R, *Statistical errors in immunologic research*, Journal of Allergy and Clinical Immunology **114** (2004), no. 6, 1259–1263.

33. Kudaĭbergenov K., *On fraïssé's theorem for uncountable classes of finitely generated structures*, Siberian Advances in Mathematics **28** (2018), 60–64.

34. Y. Kiouvrekis, S. V. Sudoplatov, and P. Stefaneas, *Lattices in generative classes*, Sib. Èlektron. Mat. Izv **16** (2019), 1752–1761.

35. Yiannis Kiouvrekis, Petros Stefaneas, and Aggeliki Kokkinaki, *An argumentation based statistical support tool*, 4th International Workshop Applied Methods of Statistical Analysis Nonparametric Methods in Cybernetics and System Analysis, AMSA 2017 (Boris Lemeshko, Narayanaswamy Balakrishnan, and Mikhail Nikulin, eds.), Applied Methods of Statistical Analysis, Novosibirsk State Technical University, January 2017, pp. 227–234 (English).

36. Yiannis Kiouvrekis, Petros Stefaneas, Angelika Kokkinaki, and Nikos Asimakis, *Limiting the impact of statistics as a proverbial source of falsehood*, Information Systems (Cham) (Marinos Themistocleous and Paulo Rupino da Cunha, eds.), Springer International Publishing, 2019, pp. 433–442.

37. Yiannis Kiouvrekis, Petros Stefaneas, and Ioannis Vandoulakis, *On the transformations of the square of opposition*, Jean-Yves Beziau, Arthur Buchsbaum, Ioannis Vandoulakis (eds) Sixth World Congress on the Square of Opposition Crete, November 1–5, Handbook of Abstracts (2018), 50–51.

38. ———, *On the transformations of the square of opposition from the point of view of institution model theory*, Studies in Universal Logic,, 2020, Accepted.

39. Wiesław Kubiś and Dragan Mašulović, *Katětov functors*, Applied Categorical Structures **25** (2017), no. 4, 569–602.

40. Wiesław Kubiś, *Fraïssé sequences: category-theoretic approach to universal homogeneous structures*, Annals of Pure and Applied Logic **165** (2014), no. 11, 1755 – 1811.

41. Ershov Yu. L., *Definability and computability*, Consultants Bureau, New York, 1996.

42. S. Mac Lane., *Categories for the working mathematician*, Springer, New York, 1978.

43. F. William Lawvere, *Diagonal arguments and cartesian closed categories*, Lecture Notes in Mathematics **92** (1969), 134–145.

44. J. C. C. McKinsey, *A solution of the decision problem for the lewis systems s2 and s4, with an application to topology*, The Journal of Symbolic Logic **6** (1941), no. 4, 117–134.

45. J. C. C. McKinsey and Alfred Tarski, *The algebra of topology*, Annals of Mathematics **45** (1944), no. 1, 141–191.

46. Bert Mendelson, *Introduction to topology*, Dover, New York, 1999.

47. Alessio Moretti, *Non-linear modal graphs: the simplest bifurcation inside n-opposition theory*, UNILOG (2007), 87.

48. Spanoudakis Nikolaos, Elena Constantinou, Koumi Adamos, and Kakas Antonis, *Modeling data access legislation with gorgias*, (2017).

49. Anand Pillay, *Countable models of stable theories*, Proceedings of the American Mathematical Society **89** (1983), no. 4, 666–672.

50. ———, *Stable theories, pseudoplanes and the number of countable models*, Annals of Pure and Applied Logic **43** (1989), no. 2, 147 – 160.

51. Baisalov E. R., Meirembekov K. A., and Nurtazin A. T., *Definably minimal models*, Model Theory in Kazakhstan. Collection of papers in the memory of A. D. Taimanov / ed. M. M. Erimbetov (2006), 14–28.

52. Mac Lane S., *One universe as a foundation for category theory*, Reports of the Midwest Category Semina **106** (1969), 192–200.

53. Hans Smessaert and Lorenz Demey, *Logical geometries and information in the square of oppositions*, Journal of Logic, Language and Information **23** (2014), 527–565.

54. Lewitzka Steffen, *Abstract logics, logic maps, and logic homomorphisms*, Logica Universalis **1** (2007), no. 2, 243–276.

55. ———, *A topological approach to universal logic: Model-theoretical abstract logics*, Logica Universalis (2007), 35–61.

56. Alexander M. Strasak, Qamruz Zaman, Karl P. Pfeiffer, Georg Göbel, and S. V. SudoplatovHanno Ulmer, *Statistical errors in medical research–a review of common pitfalls*, Swiss Med Wkly **137** (2007), no. 3-4, 44–49.

57. S. V. Sudoplatov, *Hypergraphs of prime models and distributions of countable models of small theories*, Journal of Mathematical Sciences **169** (2010), no. 5, 680–695.

58. Sergey V. Sudoplatov, Yiannis Kiouvrekis, and Petros Stefaneas, *Generic constructions and generic limits*, Algebraic Modeling of Topological and Computational Structures and Applications (2017), 375–398.

59. ———, *Generic constructions and generic limits*, Algebraic Modeling of Topological and Computational Structures and Applications (Cham) (Sofia Lambropoulou, Doros

Theodorou, Petros Stefaneas, and Louis H. Kauffman, eds.), Springer International Publishing, 2017, pp. 375–398.

60. Alfred Tarski, *Der aussagenkalkul und die topologie*, Fund. Math. **31** (1938), 103–134.
61. Sudoplatov S. V., *Inessential combinations and colorings of models*, Siberian Mathematical Journal **44** (2003), no. 5, 883–890.
62. Sudoplatov S. V., *Syntactic approach to constructions of generic models*, Algebra and Logic **46** (2007), no. 2, 134–146.
63. ______, *The lachlan problem*, NSTU, Novosibirsk, 2009.
64. ______, *Algebras of distributions of formulas with respect to generalized semi-isolation*, Algebra and Model Theory 9. Collection of papers. Novosibirsk: NSTU (2013), 67–100.
65. ______, *Classification of countable models of complete theories*, Novosibirsk: NSTU (2014).
66. ______, *enerative classes generated by sets of diagrams*, Algebra and Model Theory 10. Collection of papers. Edited by A.G. Pinus, K.N. Ponomaryov, S.V. Sudoplatov, E.I. Timoshenko (2015), 163–174.
67. ______, *Generative and pre-generative classes*, Proceedings of the 10^{th} Panhellenic Logic Symposium (2015), 30–34.
68. Goranko Valentin, *Logical topologies and semantic completeness*, Lecture Notes in Logic, p. 68–79, Cambridge University Press, 2004.
69. Shehtman Valentin, *Modal logics of domains on the real plane*, Studia Logica **42** (1983), 63–80.
70. Hodges W., *Model theory*, Cambridge: Cambridge University Press, 1993.
71. Kiouvrekis Y., *Agent-behavior system: An intorduction to a topological approach*, Algebra and Model Theory 11 (Novosibirsk) (A.G.Pinus and E. N. Poroshenko, eds.), NSTU, 2017, pp. 81–85.
72. ______, *Remarks on abstract logical topologies: an institutional approach.*, BOOK OF ABSTRACTS, 16th International Congress on Logic, Methodology and Philosophy of Science and Technology (Prague, 5–10 August 2019), 2019, p. 258.
73. ______, *Dualities, modalities and institutions: A coalgebraic perspective*, Symmetry: Art and Science **1-4** (2020), Accepted, Special issue editor: Evgenios Avgerinos. Editors: Ioannis Vandoulakis and Dénes Nagy.
74. Kiouvrekis Y. and Stefaneas P., *Topological semantics in institutions with proofs*, Algebra and Model Theory 10 (Novosibirsk) (A.G.Pinus, K.N.Ponomaryov, S.V.Sudoplatov, and E.I.Timoshenko, eds.), NSTU, 2015, pp. 92–100.

75. Kiouvrekis Y., Stefaneas P., and Sudoplatov S. V., *Definable sets in generic structures and their cardinalities*, Siberian Advances in Mathematics **28** (2018), no. 1, 39–52.

76. Noson S. Yanofsky, *A universal approach to self-referential paradoxes, incompleteness and fixed points*, The Bulletin of Symbolic Logic **9** (2003), no. 3, 362–386.

77. Kudaibergenov K. Zh., *Cardinalities of definable sets in superstructures over models*, Siberian Advances in Mathematics **20** (2010), no. 1, 58–67.

Appendix A

Implementation

A.1. Logic Interface

```
package logic;
import java.util.List; public interface Logic {
public abstract void load(String file);
public abstract void claim(String condition);
public abstract void claim(String condition, String
   label);
public abstract boolean test(String condition);
public abstract List<String> query(String variable,
   String condition);
public abstract List<List<String>> query(List<String>
   variables, String condition); public abstract
   List<List<String>> why(); // explain the last
   test()/query()
public abstract List<String> listPredicates(); public
   abstract void disclaimAll();
public abstract void disclaimLast();
public abstract void disclaim(String condition); public
   abstract String negate(String condition);
```

```
}
```

A.2. Prolog Class

```
package logic;
import java.util.ArrayList; import java.util.LinkedList;
   import java.util.List; import java.util.Map;
import org.jpl7.Query; import org.jpl7.Term;
public class Prolog implements Logic {
private Query q;
private List<List<String>> explanation = new
   ArrayList<>(); private final List<String> claims =
   new LinkedList<>(); private final List<String>
   loadedFiles = new ArrayList<>();
@Override
public void load(String file) {
String string = "consult('" + file + "')."; q = new
   Query(string);
q.hasSolution();
loadedFiles.add(file);
}
@Override
public void claim(String condition) {
if (claims.contains(condition)) { // avoid duplicate
   claims disclaim(condition);
}
String string = "assert(" + condition + ")."; q = new
   Query(string);
q.hasSolution();
claims.add(condition);
}
@Override
public void claim(String condition, String label) {
```

```
claim(condition); }
@Override

public void disclaimLast() {
String string = "retract(" + claims.get(claims.size() -
    1) + ")."; q = new Query(string);
q.hasSolution();
claims.remove(claims.size() - 1);
}
@Override
public boolean test(String condition) { //throws
    $\hookleftarrow$ org.jpl7.PrologException {
String string = condition + "."; boolean bool;
q = new Query(string);
try {
bool = q.hasMoreSolutions();
q.close(); // we don't care for more solutions }
    catch(org.jpl7.PrologException e) {
bool = false; }
explanation = new ArrayList<>(); explanation.add(new
    ArrayList<>()); return bool;
}
@Override
public List<String> query(String variable, String
    condition) {
List<String> listOfVariables = new ArrayList<>();
    listOfVariables.add(variable);
List<List<String>> listOfLists = query(listOfVariables,
    condition); List<String> reply = new ArrayList<>();
for (List list : listOfLists) { reply.add((String)
    list.get(0));
}
```

```
return reply; }
@Override
public List<List<String>> query(List<String> variables,
   String  condition) {
Map<String, Term> terms;
List<List<String>> reply = new ArrayList(); explanation
   = new ArrayList<>(); // reset explanation q = new
   Query(condition + ".");

while (q.hasNext()) {
terms = q.next();
List<String> solution = new ArrayList<>(); for (String
   variable : variables) {
solution.add(terms.get(variable).toString()); }
reply.add(solution);
explanation.add(new ArrayList<>()); }
return reply; }
@Override
public List<List<String>> why() {
return explanation; }
@Override
public List<String> listPredicates() {
List<String> predicates = new ArrayList<>(); for (String
   claim : claims) {
predicates.add(claim + "."); }
return predicates; }
@Override
public void disclaimAll() {
for (int i = claims.size() - 1; i >= 0; i--) {
q = new Query("retract(" + claims.get(i) + ").");
   q.hasSolution();
claims.remove(i);
```

```
} }
@Override
public String toString() {
String reply = "";
for (String file : loadedFiles) {
reply += "consult(" + file + ").\n"; }
for (String predicate : listPredicates()) { reply +=
    predicate + "\n";

}
}
return reply; }
@Override
public void disclaim(String condition) {
if (claims.indexOf(condition) != -1) {
q = new Query("retract(" + condition + ").");
    q.hasSolution();
claims.remove(condition);
} }
@Override
public String negate(String condition) {
return "not(" + condition + ")"; // not(foo).
    deprecated, use \+ foo. }
```

A.3. Gorgias Class

```
package logic;
import java.util.ArrayList; import
    java.util.LinkedHashMap; import java.util.List;
import java.util.Map;
import org.jpl7.Query;
import org.jpl7.Term;
public class Gorgias implements Logic { private Query q;
```

```
private final String factPrefix = "f"; private int
    factCounter = 0;
private List<List<String>> explanation = new
    ArrayList<>();
private final Map<String, String> claimsToRules = new
    LinkedHashMap<>(); private final List<String>
    loadedFiles = new ArrayList<>();
@Override
public void load(String file) {
q = new Query("consult('" + file + "').");
    q.hasSolution();
loadedFiles.add(file);
//q.close(); // is this necessary?
}
@Override
public void claim(String condition) {
claim(condition, ""); }
@Override
public void claim(String condition, String label) {
if (claimsToRules.containsKey(condition)) { // avoid
    duplicate claims disclaim(condition);
}
if (label.equals("")) {
label = factPrefix + ++factCounter; }

q = new Query(wrapAssert(condition, label));
q.hasSolution();
//q.close(); // is this necessary?
claimsToRules.put(condition, "rule(" + label + "," +
    condition +
",[])"); }
@Override
```

```
public void disclaimLast() {
String lastClaim = (String)
    claimsToRules.keySet().toArray()[claimsToRules.size()
    - 1];
String lastRule = claimsToRules.get(lastClaim); q = new
    Query("retract(" + lastRule + ").");
    q.hasSolution(); claimsToRules.remove(lastClaim);
}
private String wrapAssert(String condition, String
    label) { String string;
string = "assert(rule("; string += label;
string += "," + condition; string += ",[]))).";
return string; }
@Override
public boolean test(String condition) {
Map<String, Term> termmap;
boolean bool;
explanation = new ArrayList<>(); // reset explanation q
    = new Query(wrapProve(condition,"Delta"));
bool = q.hasMoreSolutions();
if (bool) {
termmap = q.next();
List<String> deltaList = new ArrayList<>();
for (Term term : termmap.get("Delta").toTermArray()) {
deltaList.add(term.toString()); }
explanation.add(deltaList); }
q.close(); // we don't care for more solutions return
    bool;

}
private String wrapProve(String condition, String delta)
    { String string;
```

```
string = "prove([";
string += condition;
string += "]," + delta + ")."; return string;
}
@Override
public List<String> query(String variable, String
   condition) {
List<String> listOfVariables = new ArrayList<>();
   listOfVariables.add(variable);
List<List<String>> listOfLists = query(listOfVariables,
   condition); List<String> reply = new ArrayList<>();
for (List list : listOfLists) { reply.add((String)
   list.get(0));
}
return reply; }
@Override
public List<List<String>> query(List<String> variables,
   String <- condition) {
Map<String, Term> terms;
List<List<String>> reply = new ArrayList<>();
String delta = getDelta(variables);
explanation = new ArrayList<>(); // reset explanation q
   = new Query(wrapProve(condition, delta));
while (q.hasNext()) {
terms = q.next();
List<String> solution = new ArrayList<>(); for (String
   variable : variables) {
solution.add(terms.get(variable).toString()); }
reply.add(solution);
List<String> solutionDelta = new ArrayList<>(); for
   (Term term : terms.get(delta).toTermArray()) {
solutionDelta.add(term.toString()); }
```

```
explanation.add(solutionDelta); }


return reply; }
private String getDelta(List<String> variables) { String
   delta = "Delta";
if (variables.indexOf(delta) != -1) {
for (String variable : variables) { delta += variable;
} }
return delta; }
@Override
public List<List<String>> why() {
return explanation; }
@Override
public List<String> listPredicates() {
List<String> rules = new ArrayList<>();
for (String claim : claimsToRules.keySet()) {
rules.add(claimsToRules.get(claim) + "."); }
return rules; }
@Override
public void disclaimAll() {
String[] claims = claimsToRules.keySet().toArray( new
   String[claimsToRules.keySet().size()]
);
for (int i = claimsToRules.size() - 1; i >= 0; i--) {
q = new Query("retract(" + claims[i] + ").");
   q.hasSolution(); claimsToRules.remove(claims[i]);
}
factCounter = 0; }
@Override
public String toString() {
String reply = "";
```

```
}
for (String file : loadedFiles) {
reply += "consult(" + file + ").\n";
}
for (String predicate : listPredicates()) {
reply += predicate + "\n"; }
return reply; }
@Override
public void disclaim(String claim) {
if (claimsToRules.containsKey(claim)) {
q = new Query("retract(" + claimsToRules.get(claim) +
   ")."); q.hasSolution();
claimsToRules.remove(claim);
} }
@Override
public String negate(String condition) {
return "neg(" + condition + ")"; }
```

A.4. Examples using Logic, Prolog, Gorgias classes

```
import java.util.ArrayList; import java.util.List;
   import logic.*;
public class logic {
public static void testLogic(Logic l) {
System.out.println("\n> Claim alice mother, bob father
   of charlie, <- david...");
l.claim("mother(alice,charlie)");
   l.claim("father(bob,charlie)");
   l.claim("mother(alice,david)");
   l.claim("father(bob,david)");
```

```
System.out.println("\n> Is alice mother, bob father of
    charlie?");
    System.out.println(l.test("mother(alice,charlie)"));
    System.out.println(l.test("parent(bob,charlie)"));
System.out.println("\n> Is bar foo of baz?"); try {
System.out.println(l.test("foo(bar,baz)")); } catch
    (org.jpl7.PrologException e) {
System.out.println("a prolog error caught");
//e.printStackTrace();
}
System.out.println("\n> Parents of charlie?");
    List<String> solution;
solution = l.query("X", "parent(X,charlie)"); for
    (String string : solution) {
System.out.println(string); }
System.out.println("\n> Why?"); List<List<String>>
    explanations;
explanations = l.why();
for (int i = 0; i < explanations.size(); i++) {
System.out.print(solution.get(i) + " because ");
for (String argument : (List<String>)
    explanations.get(i) ) {
System.out.print(argument + ", ");

}
System.out.println(); }
System.out.println("\n> Parent/child pairs?");
    List<List<String>> parents_children; List<String>
    variables = new ArrayList<>(); variables.add("X");
variables.add("Y");
parents_children = l.query(variables, "parent(X,Y)");
    for (List parent_child : parents_children) {
```

```
System.out.println(parent_child); }
System.out.println("\n> Why?"); for (List explanation :
    l.why()) {
System.out.println(explanation); }
}
public static void banana(Logic l) { boolean b;
System.out.println("\n> About bananas");
bananaQuestions(l);
System.out.println("\n> Claim banana is too ripe and ask
    again"); l.claim("tooripe(b)", "banana_b_is_ripe");
bananaQuestions(l);
}
public static void bananaQuestions(Logic l) { boolean b;
b = l.test("color(b,yellow)"); System.out.println("-
    Banana yellow? " + b); System.out.println("--
    because " + l.why()); b = l.test("color(b,black)");
    System.out.println("- Banana black? " + b);
    System.out.println("-- because " + l.why()); b =
    l.test("taste(b,sweet)"); System.out.println("-
    Banana sweet? " + b); System.out.println("-- because
    " + l.why()); b = l.test("taste(b,mushy)");
    System.out.println("- Banana mushy? " + b);


System.out.println("-- because " + l.why()); }
public static void multiple() { // doesn't work, there
    is always one <- instance of prolog
Logic a, b;
String time_a, time_b;
a = new Gorgias();
b = new Gorgias();
a.claim("time(day)"); b.claim("time(night)");
```

```
time_a = a.query("X", "time(X)").get(0); time_b =
   b.query("X", "time(X)").get(0);
   System.out.println("a says " + time_a);
   System.out.println("b says " + time_b);
}
public static void listAndClean(Logic l) {
   System.out.println("\nPredicates before <-
clean:\n-----------------------");
for (String predicate : l.listPredicates()) {
System.out.println(predicate); }
l.disclaimAll();
System.out.println("\npredicates after <-
   clean:\n-----------------------");
for (String predicate : l.listPredicates()) {
   System.out.println(predicate);
}
System.out.println("\nClaim
   something:\n----------------");
   l.claim("foo(bar,baz)");
System.out.println(l);
System.out.println("Test it:\n-------");
   System.out.println(l.test("foo(bar,baz)"));
System.out.println("\nDisclaim last:\n--------------");
   l.disclaimLast();
System.out.println(l);


System.out.println("Test it:\n-------");
System.out.println(l.test("foo(bar,baz)")); }
```

```
public static void numbers(Logic l) {
   System.out.println("\nnumbers:\n--------");
   System.out.println("claim a is 4.0 and b is 6.0");
   l.claim("valueof(a,4.0)", "a_is_4");
   l.claim("valueof(b,6.0)", "b_is_6");
   System.out.println(
"is a larger than 5? " + l.test("largerthan(a,5.0)") +
   ", because " + <- l.why() + "\n" +
"is b larger than 5? " + l.test("largerthan(b,5.0)") +
   ", because " + <- l.why()
); }
public static void queryWithNoSolution(Logic l) {
   System.out.println("\nChildren of
   Charlie\n-------------------");
   System.out.println(l.query("X",
   "parent(charlie,X)")); System.out.println("Result is
   empty? " + l.query("X", <-
"parent(charlie,X)").isEmpty()); }
public static void testNegation(Logic l) {
   System.out.println("\nTest
   Negation\n-------------"); boolean b;
String cond;
cond = "increasing(1,2,3)";
b = l.test(cond);
System.out.println(cond + "?\t" + b + ", because " +
   l.why());
b = l.test(l.negate(cond));
System.out.println("not " + cond + "?\t" + b + ",
   because " + l.why());
cond = "decreasing(1,2,3)";
b = l.test(cond);
```

```
System.out.println(cond + "?\t" + b + ", because " +
    l.why());
b = l.test(l.negate(cond));
System.out.println("not " + cond + "?\t" + b + ",
    because " + l.why());
cond = "increasing(3,2,1)"; b = l.test(cond);


System.out.println(cond + "?\t" + b + ", because " +
    l.why());
b = l.test(l.negate(cond));
System.out.println("not " + cond + "?\t" + b + ",
    because " + l.why());
cond = "decreasing(3,2,1)";
b = l.test(cond);
System.out.println(cond + "?\t" + b + ", because " +
    l.why());
b = l.test(l.negate(cond));
System.out.println("not " + cond + "?\t" + b + ",
    because " + l.why());
System.out.println("\nHow negation looks: " +
    l.negate("foo(bar)")); }
public static void main(String[] args) {
System.out.println("\nProlog\n-------"); Logic prolog =
    new Prolog(); prolog.load("prolog.pl");
    testLogic(prolog);
System.out.println("\nGorgias\n-------"); Logic gorgias
    = new Gorgias(); gorgias.load("gorgias.pl");
    testLogic(gorgias);
gorgias.load("banana.pl"); banana(gorgias);
listAndClean(prolog); listAndClean(gorgias);
prolog.load("numbers.pl");
```

```
gorgias.load("numbers.pl"); // we don't have to load
   both because <- there is always one prolog running
numbers(prolog); numbers(gorgias);
queryWithNoSolution(prolog);
   queryWithNoSolution(gorgias);
prolog.load("increasing.pl");
   gorgias.load("increasing.pl"); testNegation(prolog);

} }
testNegation(gorgias);
System.out.println("\nTypes of
   objects\n----------------");
   System.out.println("prolog is " +
   prolog.getClass()); System.out.println("gorgias is "
   + gorgias.getClass());
```

A.5. Connection Java - R, Statistics.java

```
import org.rosuda.JRI.REXP; import
   org.rosuda.JRI.Rengine;
public class Statistics {
private final Rengine re;
public Statistics() {
String[] args = {"--vanilla"};
re = new Rengine(args, false, null);
}
public void load(String rfile) {
String string = "source('" + rfile + "')";
   re.eval(string);
}
public void closeR() { re.end();
}
public void printData(String variable) { REXP x;
```

```
System.out.println(x = re.eval(variable)); }
/* ----------- */ /* CHISQ STUFF */ /* ----------- */
public void chisq(String variable) {
String string = variable + "_chisq = chisq.test(" +
   variable + ")"; re.eval(string);
}
public double getChisqStat(String variable) { String
   string = variable + "_chisq$statistic"; return
   re.eval(string).asDouble();
}
public int getChisqDf(String variable) {
} }
String string = variable + "_chisq$parameter";
return re.eval(string).asInt(); }
public double getChisqPValue(String variable) { String
   string = variable + "_chisq$p.value"; return
   re.eval(string).asDouble();
}
public double getMinExpected(String variable) {
String string = "min(" + variable + "_chisq$expected)";
   return re.eval(string).asDouble();
}
/* ------------ */ /* FISHER STUFF */ /* ------------ */
public void fisher(String variable) {
String string = variable + "_fisher = fisher.test(" +
   variable + ")"; re.eval(string);
}
public double getFisherPValue(String variable) { String
   string = variable + "_fisher$p.value"; return
   re.eval(string).asDouble();
```

A.6. Connection R - Gorgias, stats.java

```
import logic.*; public class stats {
public static void chisq(Logic l, Statistics s, String
    data) { s.chisq(data);
double pvalue = s.getChisqPValue(data); double
    minExpected = s.getMinExpected(data);
l.claim("chisq(" + data + ")"); // because many
    data/objects can be <- loaded
l.claim("chisq_pvalue(" + data + "," + pvalue + ")");
l.claim("chisq_minexpected(" + data + "," + minExpected
    + ")"); }
public static void fisher(Logic l, Statistics s, String
    data) { s.fisher(data);
double pvalue = s.getFisherPValue(data);
l.claim("fisher(" + data + ")"); // because many
    data/objects can be <- loaded
l.claim("fisher_pvalue(" + data + "," + pvalue + ")"); }
public static void main(String[] args) {
//String data = "data3";
String[] allData = {"data1", "data2", "data3", "data4"};
    double confidence = 0.95;
double significance = 1 - confidence;
Statistics s = new Statistics();
Logic l = new Gorgias();
l.load("stats.pl");
l.claim("significance(" + significance + ")");
for (String data: allData) {
s.load(data + ".txt"); chisq(l, s, data);
} }
if (l.test("chisq_valid(" + data + ")")) { } else {
l.disclaim("chisq(" + data + ")"); // not needed anymore
    but <- nice to have - remind that chisq is not valid
    for this data
```

```
fisher(l, s, data); }
boolean rejecth0 = l.test("rejecth0(" + data + ")"); if
    (rejecth0) {
System.out.println("Null hypotheses rejected, " + data +
    " are <- codependent.");
} else {
System.out.println("Cannot reject null hypotheses, " +
    data + <-
" are independent");
l.test(l.negate("rejecth0(" + data + ")")); // just to
    fill <-
the explanation
}
System.out.println("Why? " + l.why());
    System.out.println();
}
System.out.println("Gorgias state");
    System.out.println("-------------");
    System.out.println(l);
// close R
s.closeR();
```

A.7. Hypothesis Test in Gorgias stats.pl

```
% stats.pl - chisq & fisher
% load gorgias
:- compile('lib/gorgias.pl'). :- compile('ext/lpwnf.pl').
rule(chisq_is_valid(X), chisq_valid(X),
[chisq(X)]).
rule(chisq_is_not_valid(X), neg(chisq_valid(X)),
[chisq(X), chisq_minexpected(X,MinExpected), MinExpected
    < 5]).
```

```
rule(prefer_chisq_is_not_valid(X),
   prefer(chisq_is_not_valid(X),
chisq_is_valid(X)),
[]).
rule(h0_is_not_rejected(X), neg(rejecth0(X)),
[]).
rule(chisq_rejects_h0(X), rejecth0(X),
[chisq(X),
chisq_valid(X), chisq_pvalue(X,Pvalue),
   significance(Significance), Pvalue < Significance]).
rule(prefer_chisq_rejects_h0(X),
   prefer(chisq_rejects_h0(X),
h0_is_not_rejected(X)),
[]).
rule(fisher_rejects_h0(X), rejecth0(X),
[fisher(X),
fisher_pvalue(X,Pvalue), significance(Significance),
   Pvalue < Significance]).
rule(prefer_fisher_rejects_h0(X),
   prefer(fisher_rejects_h0(X),
[]).
h0_is_not_rejected(X)),[]).
```

A.8. List of publications

The following is a list of all publications originating from this thesis, including articles under review, articles accepted in periodicals or collective works, abstracts presented at conferences following review and presentations made throughout the composition of the thesis.

A.8.1. List of papers under review which are submitted in Journals.

JUR1: Kiouvrekis Yiannis, Topological Semantics in Institutions (submitted)

A.8.2. Papers in Journals.

J1: Y. Kiouvrekis, P. Stefaneas, S. V. Sudoplatov, "Lattices in generative classes", Sib. Elektron. Mat. Izv., 16 (2019), 1752–1761 https://doi.org/10.33048/semi.2019.16.123 [**34**]

J2: Kiouvrekis,Y.,Stefaneas,P.& SudoplatovS., Definable Sets in Generic Structures and their Cardinalities S.V. Sib. Adv. Math. (2018) 28: 39. https://doi.org/10.3103/S1055134418010030

J3: Kiouvrekis, Y., Stefaneas, P. & Sudoplatov S., Definable sets in generic structures and their cardinalities, Matematicheskie trudy. 2017. Vol. 20, N 2. P. 52-79. DOI: https://doi.org/10.17377/mattrudy.2017.20.203

J4: Kiouvrekis Y. Dualities, Modalities and Institutions: A Coalgebraic Perspective (Accepted to Symmetry: Art and Science, (International Journal of the International Society for the Interdisciplinary Study of Symmetry). Editors: Ioannis Vandoulakis and Denes Nagy.

A.8.3. Papers in chapters of books.

CIB1: Yiannis Kiouvrekis, Petros Stefaneas and Ioannis Vandoulakis , On The Transformations of the Square of Opposition from the point of view of Institution Model Theory, (ACCEPTED at Logica Universalis, Springer)

CIB2: Sudoplatov, S. V., Kiouvrekis, Y., & Stefaneas, P. (2017). Generic constructions and generic limits. Algebraic Modeling of Topological and Computational Structures and Applications. AlModTopCom 2015. Springer Proceedings in Mathematics & Statistics, (Vol. 219, pp. 375-398). Springer, Cham. https://doi.org/10.1007/978-3-319-68103-0_18

A.8.4. Papers in Conferences.

CWR1: Kiouvrekis, Y., Stefaneas, P., Kokkinaki, A., & Asimakis, N. (2019). Limiting the impact of statistics as a proverbial source of falsehood. In M. Themistocleous, M. Themistocleous, & P. R. da Cunha (Eds.), Information Systems - 15th European, Mediterranean, and Middle Eastern Conference, EMCIS 2018, Proceedings (pp. 433-442). (Lecture Notes in Business Information Processing; Vol. 341). Springer Verlag. https://doi.org/10.1007/978-3-030-11395-7_33

CWR2: Kiouvrekis Yiannis, Kokkinaki Aggeliki, Stefaneas Petros, An argumentation based statistical support tool, AMSA 2017, Krasnoyarsk, Russia, 18-22 September, 2017: Proceedings of the International Work- shop. - Novosibirsk: NSTU publisher, 2017. - 227-233. ISSN 2313-870X

CWR3: Kiouvrekis Y., Stefaneas P. Topological semantics in institutions with proofs, Algebra and Model Theory 10. Collection of papers /Eds.: A.G.Pinus, K.N.Ponomaryov, S.V.Sudoplatov, and E.I.Timoshenko. - Novosibirsk : NSTU, 2015. - P. 92-100.

CWR4: Kiouvrekis Y. Agent-behavior system: An introduction to a topological approach, Algebra and Model Theory 11. Collection of papers / Collection of papers edited by A. G. Pinus, E. N.Poroshenko, - Novosibirsk : NSTU, 2017 - P. 81–85

A.8.5. Abstracs in conferences.

AWR1: Kiouvrekis Y. Dualities, Modalities and Institutions, ISSC 2016 - International Conference, The Logics of Image: Visualization, Iconicity, Imagination and Human Creativity Santorini, 25-30 July 2016, THE BOOK OF ABSTRACTS, p.91-92

AWR2: Kiouvrekis, Y., Sudoplatov, S. & Stefaneas P., On definable sets in generic structures, Handbook of the 6th World Congress and School on Universal Logic June 16–26, 2018 Vichy, France. Vichy: Vichy University, 2018. , P. 233-234.

AWR3: Kiouvrekis, Y., Sudoplatov, S. & Stefaneas P., Calculi for definable sets, Handbook of the 6th World Congress and School on Universal Logic June 16–26, 2018 Vichy, France. Vichy: Vichy University, 2018. P. 240-241.

AWR4: Kiouvrekis, Y., Sudoplatov, S. & Stefaneas P., On lattices in generative classes, Handbook of the 6th World Congress and School on Universal Logic June 16–26, 2018 Vichy, France. Vichy: Vichy University, 2018. P. 243-244

AWR5: Kiouvrekis, Y. Remarks on abstract logical topologies: an institutional approach., Book of abstracts 16th international congress on logic, methodology and philosophy of science and technology, bridging across academic cultures, Prague, 5–10 August 2019 -P. 258

A.8.6. Presentations.

P1: Remarks on Abstract Logical Topologies: An Institutional Approach, 16th International congress on logic, methodology and philosophy of science and technology Prague, Aug 2019

P2: Rules: Logic and Applications Workshop, Athens December 19-20, 2018, Formal Methods and Blockchain Models, for a universal blockchain logic.

P3: On The Transformations of the Square of Opposition from the point of view of Institution Model Theory, Square of Opposition - Crete - November 1-5, 2018

P4: On definable sets in generic structures, with S. Suduplatov 24/7/2018, Model theory workshop, 6th World Congress and School on Universal Logic, June 16–26, 2018 Vichy, France -On lattices in generative classes with S. Sudoplatov; 24/7/2018, Model theory workshop, 6th World Congress and School on Universal Logic, June 16–26, 2018 Vichy, France

P5: Calculi for definable sets with S. Sudoplatov 24/7/2018, Model theory workshop, 6th World Congress and School on Universal Logic, June 16–26, 2018 Vichy, France

P6: Topological Semantics and fundamental theorems via Topology – 25/7/2018, Model theory workshop, 6th World Congress and School on Universal Logic, June 16–26, 2018 Vichy, France,

P7: Invited speaker in 12th International Summer School-Conference "Problems Allied to Universal Algebra and Model Theory" Erlagol-2017, June 23-29, 2017 at Novosibirsk, Russia

P8: Dualities, Modalities and Institutions, Yiannis Kiouvrekis ISSC 2016 - International Conference, The Logics of Image: Visualization, Iconicity, Imagination and Human Creativity Santorini, 25-30 July 2016

P9: 3rd International Conference on Cryptography, Cyber Security and Information Warfare, 2016 TITLE: Towards Abstract Logics for Secure Communication International Congress on Mathematics, MICOM 2015, Research towards an Institution Independent Model Theory

P10: 22nd InternationalWorkshop on Algebraic Development Techniques (WADT 2014), Fixed Point Logics as Institutions, Yiannis Kiouvrekis,Petros Stefaneas

P11: Logic and Utopia 2014, A Universal Approach to teaching Logic, Yiannis Kiouvrekis

Index