Spring 2016

Constructing a Categorical Framework of Metamathematical Comparison Between Deductive Systems of Logic

Alex Gabriel Goodlad
Bard College, ag6697@bard.edu

Constructing a Categorical Framework of Metamathematical Comparison Between Deductive Systems of Logic

A Senior Project submitted to
The Division of Science, Mathematics, and Computing
of
Bard College

by
Alex Goodlad

Annandale-on-Hudson, New York
May, 2016

Abstract

The topic of this paper in a broad phrase is "proof theory". It tries to theorize the general notion of "proving" something using rigorous definitions, inspired by previous less general theories. The purpose for being this general is to eventually establish a rigorous framework that can bridge the gap when interrelating different logical systems, particularly ones that have not been as well defined rigorously, such as sequent calculus. Even as far as semantics go on more formally defined logic such as classic propositional logic, concepts like "completeness" and "soundness" between the "semantic" and the "deductive system" is too arbitrarily defined on the specific system that is applied to for it to carry as an adequate definition. What we shall do then is come up with an adequate definition for a characterization of *every* logic that one has worked with, and show what can be done with it for a few basic logical systems that include classic propositional logic, intuitionistic propositional logic and intuitionistic sequent calculus. To make this definition work with eloquence, we go the category theory route of constructing a category with objects that correspond to collections of logical formulae and arrows that correspond to deductions from one such collection to another.

Contents

Abstract **1**

Dedication **4**

Acknowledgments **5**

1 Introduction **6**

2 Preliminaries **10**
- 2.1 Definition by Recursion and Structural Induction 10
- 2.2 Natural Deduction Informally Presented 13
- 2.3 Preliminary Category Theory 15
- 2.4 The Free Category 22
- 2.5 The Path Category and the Cartesian Path Category 24

3 Language and Inference **30**
- 3.1 Languages 31
- 3.2 Formula Vectors 34
- 3.3 Inferences and Proofs 36
- 3.4 Natural Deduction and Hypotheses 40
- 3.5 Functoral Semantics 42
- 3.6 Valuation Logic 47

4 Classical Propositional Logic **50**
- 4.1 The Classical Language and Deductive System 51
- 4.2 Soundness and Completeness with Ordinary Classical Propositional Logic . 56
- 4.3 Classical Semantics 59
- 4.4 Classical Completeness and Consequences 62

5 Sequent Calculus **66**
5.1 The Metalanguage and Sequents 67
5.2 Intuitionistic Sequent Logic Definition 68
5.3 The Cut Rule 76
5.4 Properties of Intuitionistic Sequent Logic 81

6 Conclusion and Future Work **88**
6.1 Fixing Mistakes and Tightening Rigor 88
6.2 Exploring More on "Deduction Class Equivalence" 89
6.3 Generalizing for Infinite Formula-Vectors 89
6.4 Generalizing Consistency and Satisfiability 89
6.5 Exploring Intuitionistic Logic 89
6.6 Exploring Linear Logic 91
6.7 Exploring More Languages 94
6.8 Exploring Predicate Logic 94

Bibliography **95**

Dedication

To my grandmother Mathilde and father Peter

Acknowledgments

There are too many people to be thankful for to list. But I just want to thank my senior project advisor Bob McGrail, who has been an excellent logician mentor who I think has guided me well in my aspirations to be a logician. Next I want to thank Ethan Bloch who I extremely look up to as a mathematician and professor. As not only the writer of my first ever abstract math textbook *Proofs and Fundamentals* but also my instructor for that class, he has really inspired me to pursue my undergraduate studies–and ultimately my graduate studies–in mathematics. Finally, I want to give thanks to my project board Japeth Wood and James Belk, and also the senior project supervisor John Cullinan.

Of course I am thankful for my family, who really supported me every step of the way in all these years of attending Bard College and seeking my degree. I want to give one final thanks to the three math students who I feel really had an impact on me as a math student at Bard. The first is Jeffrey Pereira, whose positive enthusiasm when it came the subject of mathematics I still remember like it was yesterday, even though he graduated three years ago. The next person is Andy Huynh, who I would call my mentor, and I would say is the main reason I have gotten far as I have with studying mathematics. Finally, I want to give the best of regards to Erik Lovece, who really is one of my best friends at Bard and is yet another math alumnus who set a good example for me at Bard. As of writing this, he is part of the national guard and his services for this country should be greatly acknowledged.

1 Introduction

So the main question at hand is what exactly does this paper seek out to *do*. While a very necessary question, it is not a very *easy* one to anwer since proof theory in general is somewhat of an obscure discipline with motivations that, although existant, do not come to mind immediately. In this case, what I am doing has not really been mathematically constructed, so much as given a mathematical definition. I shall nevertheless do what I can to explain what it is that is being done and why it is being done this way

I shall start with the easy part first, which is *why*. Proof theory–to the extent that there is first order logic, classical logic, intuitionistic logic, sequent calculus, and linear logic–has not been rigorously defined in a way that is adequate for discussing *all* these logics together in unison. While one can talk about these structures in an intuitive way that captures the right idea when analyzing the logic as its own structure, there is a lot that gets lost from the lack of formality when it comes to comparing two deductive systems. An example of comparing two deductive systems that is very popular in logic, since the discipline came to be, is showing that we can derive aspects (or all) of one system from another.

For example, we have a notion of "soundness" and "completeness" between the deductive system of classical logic and the semantic system of classical logic, but that notion does not really extend generally once we get to things like sequent calculus. What also does not extend generally very well is what an "inference rule" exactly is or a "logical axiom". We have a horizontal bar style of deduction/inference, but comparing two different logical systems again becomes problematic because the definitions have not been unified between these sorts of things.

What this paper proposes to fix this is to define things like language, inference rules, deductions, soundness and completion, and finally logic, very generally using category theory, which gives us the tools to do this very effectively. After defining all these things, it is shown how they apply to very familiar systems of logic. Various propositional varieties of logic, which include classical propositional, intuitionistic propositional, and intuitionistic sequent calculus, get a chapter's worth of attention each, but various widely known predicate logics get attention, too, in my variety of examples. The exact organization of how I do this is as follows:

Chapter 2 covers the preliminary information that one needs to know which includes a theoretical treatment of structural induction, an informal presentation of horizontal bar style deductions and sequent calculus, category theory, and a little bit of graph theory. It is expected that the reader has a fairly comprehensive background in mathmatical logic and category theory in particular, but this chapter does provide the necessary background information in these subjects should the reader need it. Moreover, Chapter 2 is very crucial in deriving a very important, yet infrequently oberved concept in category theory known as the "free category". While a category under the name of "free category" is discussed in MacLane, this turns out to be only the *special* case of a general sort of category that happens to be majorly useful in this paper.

Chapter 3 lays down the definitions, much of which have been shown to exist by virtue of the free category that was derived in Chapter 2. In the first section, we define languages. Afterwards, the construction of a category called the“deductive category”–intended to standardize all logical deductive systems–shall take place. First the objects will be constructed in one section, and the arrows shortly afterwards. A crucial section lays down the general concept of “soundness” (and analogously “completeness”) between two logics, which will allow us to assess the ability for one deductive system to “imply” another. Other sections make a few general applications to the deductive category to more general verions of useful logical constructs such as valuation mappings and hypothesis construction.

Chapter 4 applies this categorical treatment of logic to the most simple of the conventional logic: Classical propositional logic. Here, the language and inference rules are defined in the most minimal, yet eloquent, way possible. From this, the familiar properties of propositional logic are derived and it is then shown to be sound and complete with the very familiar truth table treatment of classical propositional logic.

Chapter 5 covers the final application of this categorical treatment to definitely the most complicated deductive logic that this paper derives in rigorous detail. And that is intuitionistic sequent calculus. As a language that uses a construct as exotic as sequents with a very bizarre method of doing inferences with propositions, it is necessary to define a whole new language that is different from any that one is familar with. This new language in general is called the “metalanguage” since the idea of categorical logic here is to make inferences on objects in a language and sequent calculus makes these inferences on a class of objects that correspond to the metalanguage of propositions. It then only makes sense to mathematically construct a language that corresponds to a “metalanguage” of a base language and observe derivability with respect to that language.

Once this metalanguage has been created, the attention of Chapter 5 will then focus to the construction of intuitionistic sequent calculus. Again the minimal amount of inference rules and symbols needed to create this logic will be used, meaning a lot of Chapter 6 will be setting up a theoretical framework that will effectively derive these other rules. The main rule at hand is called the "cut rule", or "cut elimination", which turns out to take a proof so long and technical that the process (particularly the proof itself) gets its own section. Once all the desired rules of intuitionistic sequent calculus are established, the next sections derive some properties of major note, which include its untimate relationship with propositional logic as it is conventionally done.

Finally, Chapter 6 will conclude this paper, and lay the road map as to where to go with this theory next. Most vitally this will include deriving linear logic, as this turns out to be a major motivation for creating this categorical system. An informal discussion of linear logic is then in order. But other future topics will be discussed such as the possibility of generalizing this logic to infinite vectors (functions) of propositions, defining a general notion of "satisfiability" and "consistency", and finally a little bit on how one can apply this system to more complicated predicate systems.

2 Preliminaries

This chapter explores some preliminary mathematical concepts that are both obscure and required to understand in order to proceed with the theoretical conquest of this paper. Section 2.1 deals with the bread and butter to much of our constructions and proofs of their qualities: Definition by Recursion and Structural Induction. Section 2.2 presents an informal presentation of Natural Deduction, which will be a style of proof that this text will adapt. Section 2.3 is a crash course on Category Theory, which is mostly there to establish the terminology and notation that shall be used throughout. Section 2.4 is devoted to deriving the "free category"–a very useful tool in category theory that will allow us to establish a robust definition for logical deductions. Finally, Section 2.5 is provides preliminary information on graph theory and then uses graph theory and the free category to derive some very necessary tools.

2.1 Definition by Recursion and Structural Induction

Induction is useful. If there is anywhere that one will see this fact, it will be in this paper. Knowledge shall be assumed of Peanno's Axioms and the many variants of mathematical induction (such as strong induction). But induction is used to the point where it can definitely get obscure, hence clear the obscurity up with this section. Note that it will be assumed that the reader has been exposed to the Peanno Axioms, but for anyone in need of reference can find a treatment, here. [1]

First, acknowledgement shall be made about Definition by Recursion. It appears so much in mathematics that one often forgets that it happens, until one runs into a strange implementation of Definition by Recursion and thinks to oneself, “But you can’t do that!” It’s very intuitive why, but theoretically it is not straightforward why it is that given a function $k\colon E \longrightarrow E$ one can conjure up a unique function $f\colon \mathbb{N} \longrightarrow E$ such that $fs = kf$, where $s\colon \mathbb{N} \longrightarrow \mathbb{N}$ is the successor function. Here is the theorem.

Theorem 2.1.1. *(Definition By Recursion) Let E be a set. Given a function $k\colon E \longrightarrow E$ and $e \in E$, there exists a unique function $f\colon \mathbb{N} \longrightarrow E$ such that*

$$f(n) = \begin{cases} e, & \textit{if } n = 1 \\ k(f(n-1)), & \textit{if } n > 1. \end{cases}$$

The proof to this theorem can be found in **Theorem 2.5.5** of Bloch’s Real Analysis text. [1]

To understand structural induction, what should be first cleared up is what exactly *is* “structural induction”? It’s when there is a mathematical structure that is “inductive” in the sense that there is a “basic” or “atomic” component and then a “recursive” component, where one form new objects by taking objects in the element (these objects playing the role of “recursion”) and then forming new objects. How this differs from original induction is that now the structure itself does not correspond in any way to the natural numbers, or any peanno object for that matter. Oftentimes the structure doesn’t even share the same cardinality. For example, a vector space, which has a basis, is an example of a structurally inductive set since every element can be formed by operations that bring about recursion that stems from a basis.

What does, however, make the structure inductive is the recursion. The structure starts with "basic" component which one can call the "order one" instance of our structure, since one cannot generate the occurence of a given "basic" object from other objects (in the same way that in Peanno's Axioms one cannot get the number 1 from any input of our successor function). Then the structure has "recursive component" where an object is labeled "order n" based on how many acts of recursion were done from the base case to get the new object.

Rigorously, a set which is structurally inductive can be thought as a set E along with a set $\mathcal{C}$ of functions $f\colon E^{n_f} \longrightarrow E$ such that there is a base case set A such that for every $G \subseteq E$, if $A \subseteq G$ and G is closed under the function class $\mathcal{C}$, then $G = E$. If one thinks about it, this is really the generalized Peanno Axioms because the Peanno Axioms can be viewed this way two with $\{1\}$ as the base case set and then the function class as $\{s\}$, where s is again a successor function

This theorem establishes how easily constructible this sort of set is. Given any set and any set of function classes on that set, we can form a subset that is structurally inductive. This sort of thing is exactly what we shall need for our theoretical constructions starting in the next chapter.

Theorem 2.1.2. *Let W be a set. Let $\mathcal{C}$ be a collection of functions $f\colon W^{m_f} \longrightarrow W$, for some $m_f \geqslant 1$. Let A be a set disjoint from $f(W^{m_f})$, for each $f \in \mathcal{C}$ and the corresponding $m_f \geqslant 1$. Then there exists some $E \subseteq W$ such that $A \subseteq G \subseteq E$ and $f(e_1, \dots, e_{m_f}) \in G$, for any $f \in \mathcal{C}$ and $e_1, \dots, e_{m_f} \in G$, imply $G = E$.*

Proof. Define G_n recursively for $n \geqslant 1$ as follows

$$E_n = \begin{cases} A, & \text{if } n = 1, \\ \bigcup_{f \in \mathcal{C}} f(E_{n-1}^{m_f}), & \text{otherwise.} \end{cases}$$

We shall prove that $E = \bigcup_{n \geqslant 1} E_n$ is a set with the desired properties. Note first that $A \subseteq E$. Let G be a set such that $A \subseteq G \subseteq E$ and $f(e_1, \dots, e_{m_f}) \in G$, for any $f \in \mathcal{C}$ and

$e_1, \ldots, e_{m_f} \in G$. We need to prove $E \subseteq G$. Let $p \in E$. Then $p \in E_n$, for some $n \geqslant 1$. We shall prove that $p \in G$ by induction on n. For $n = 1$, we have $p \in A \subseteq G$. For our inductive step, if $p' \in G$, for all $p' \in E_n$, we find $p \in f(E_n^{m_f})$, for some $f \in \mathcal{C}$; so it follows by inductive hypothesis that $p = f(e_1, \ldots, e_{m_f})$, for $e_1, \ldots, e_{m_f} \in G$, hence $p \in G$ by hypothesis. $\square$

2.2 Natural Deduction Informally Presented

In logic, a mathematical process called "natural deduction", or more generally "horizontal bar style proofs", is a nice thing. It's nice because oftentimes deriving new formulas in a logical scheme is a very technical, step-heavy process that could use an eloquent shorthand method of laying out, as opposed to simply *writing* all the steps by paragraphs and words. However, eloquence comes at the price of obscurity. These horizontal bar conventions are rather notation heavy; therefore, they are unclear, and morover do not have much of a rigorous definition behind them. Chapter 3 will try to resolve the latter problem and provide a rigorous definition, but rigorous definitions do not always provide provide clarity. This section shall try to provide clarity and alleviate such confusion.

Natural deduction is a system that is typically used in propositional/predicate logic with the usual $\wedge, \vee, \neg, \rightarrow, \leftrightarrow, \bot, \top$ connectives and $\forall, \exists$ quantifiers. What the horizontal bar style does is try to incorporate various inference rules that one has in a given system in the style of an input-output formula. On the top is one or more input formulas, and on the bottom is an output formula; in other words, a single bar plays the role of taking one formula and "infering" a new formula. To the right side of these horizontal bars is the type of inference rule that it is, which is essentially the label for an overall formula. What will be helpful at this time is some examples, so here are a few familiar ones

$$\frac{\phi}{\phi \vee \varphi} \vee I1, \quad \frac{\varphi}{\phi \vee \varphi} \vee I2, \quad \frac{\phi \quad \phi \rightarrow \varphi}{\varphi} \rightarrow E,$$
$$\frac{\phi \quad \varphi}{\phi \wedge \varphi} \wedge I, \quad \frac{\phi \wedge \varphi}{\phi} \wedge E1, \quad \frac{\phi \wedge \varphi}{\varphi} \wedge E2,$$

As one can see, these inference rules provide an algorithm to be able to make a multistep process that would make up a lot of tedious writing into a proof in a few lines. For example, proving $\phi \wedge \varphi$ derives $(\theta \vee \phi) \vee \varphi$, one writes

$$\dfrac{\dfrac{\dfrac{\phi \wedge \varphi}{\phi}\,\wedge E1}{\theta \vee \phi}\,\vee I2}{(\theta \vee \phi) \vee \varphi}\,\vee I1.$$

So far this is pretty accessible for the most part (assuming somewhat of a background in symbolic logic), but there is one more roadblock to understand that is not very clear at first, and the lack of a great "rigorous" definition does not help, and this the use of hypotheses. A very motivating way of understanding the $\rightarrow$ connective is really concieving it as implication in the colloquial sense of "$p \rightarrow q$ means assuming p, we deduce q." So to provide an inference rule for $\rightarrow$, this is where hypotheses notation comes in with making a hypothesis p and deducing things from such hypothesis to show we can indeed deduce q, which allows us to infer $p \rightarrow q$. Any hypothesis formula in natural deduction one places a box $[\phi]$ around it to indicate that it is a hypothesis. From this, inference rules can be formed using hypotheses and conclusions, with vertical dots indicating steps of inference in between the hypotheses and the conclusion. Here are two such inference rules in propositional logic

$$\dfrac{\phi \vee \varphi \quad \begin{matrix}[\phi]\\ \vdots\\ \theta\end{matrix} \quad \begin{matrix}[\varphi]\\ \vdots\\ \theta\end{matrix}}{\theta}\,\vee E, \quad \dfrac{\begin{matrix}[\phi]\\ \vdots\\ \varphi\end{matrix}}{\phi \rightarrow \varphi} \rightarrow I.$$

For example, to prove that one go from no formula at all to derive $\phi \rightarrow \phi \vee \varphi$, the proof is presented as

$$\dfrac{\dfrac{[\phi]}{\phi \vee \varphi}\,\vee 1}{\phi \rightarrow (\phi \vee \varphi)} \rightarrow I.$$

One confusing thing, however, to deal with is when there is more than one hypothesis happening at once. What is the protocol, there? Well now natural deduction starts to become a story of "charging" and "discharging" hypotheses. In other words, one can "charge" a hypothesis statement anytime in the deduction and inference rules such as $\vee E$ and $\to I$ play the roll of "discharging" hypotheses after they have been made. A legitimate deduction has all its hypotheses that were made at any point of the deduction discharged. To eliminate ambiguity, the boxes and discharging inference rule uses have upper script numbers that correspond to which inference rule cancels what. For example, in the deduction from no formula to derive $((\phi \wedge \varphi) \to \theta) \to (\phi \to (\varphi \to \theta))$, one writes

$$\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{[\phi]^2 \quad [\varphi]^1}{\phi \wedge \varphi}\wedge I \quad [(\phi \wedge \varphi) \to \theta]^3}{\theta} \to E}{\varphi \to \theta} \to I^1}{\phi \to (\varphi \to \theta)} \to I^2}{((\phi \wedge \varphi) \to \theta) \to (\phi \to (\varphi \to \theta))} \to I^3.$$

For further inquiries, it is recommended to explore Van Dalen's introductory logic text. [3]

2.3 Preliminary Category Theory

This section shalll introduce some Category theory. For any outside source inquirey, Mac Lane's text is strongly recommended. [7] Here is the basic definition of a category.

Definition 2.3.1. A **Category** $\mathcal{C}$ is class of objects $Ob(\mathcal{C})$ and a class of arrows $Ar(\mathcal{C})$ with the following properties.

1. Composition. For every arrow $A \overset{f}{\mapsto} B$ and $B \overset{g}{\mapsto} A$, there exists some arrow $A \overset{gf}{\mapsto} C$.

2. Associativity. For every arrow f, g, h, it holds that $(fg)h = f(gh)$.

3. Identity. For every object A, there exists some arrow $A \overset{1_A}{\mapsto} A$ such that $f1_A = f$ and $1_A g = g$, for every arrow $A \overset{f}{\mapsto} B$ and $C \overset{g}{\mapsto} A$.

For an arrow $A \overset{f}{\mapsto} B$, a **right inverse** is an arrow $B \overset{f'}{\mapsto} A$ such that $ff' = 1_B$. For an arrow $A \overset{f}{\mapsto} B$, a **left inverse** is an arrow $B \overset{f'}{\mapsto} A$ such that $f'f = 1_A$. For an arrow $A \overset{f}{\mapsto} B$, an **inverse** is an arrow $B \overset{f'}{\mapsto} A$ that is both a right inverse and a left inverse. △

Remark 2.3.2. With regards to notation, it is standard in category theory to denote a category $\mathcal{C}$ as both a set of objects and a set of arrows. In other words, when one says $A \in Ob(\mathcal{C})$, it is pretty standard to write A is in $\mathcal{C}$, and same for $f \in Ar(\mathcal{C})$. So if no ambiguity arises, be alert to see $\mathcal{C}$ used interchangably with instances where the object class is being referenced or those where the arrow class is being referenced. ♢

Here are a few good examples of commonly-used categories.

Example 2.3.3. The first (and most intuitive) category is the category **Set** of all sets. The object class is all sets and the arrow class is all functions between those sets.

There are similar "set with structure" categories. One is **Grp** with the object class of all groups and arrow class of all homomorphisms between those groups. Another is **Top** with the object class of all topological spaces and arrow class of all continuous functions between those spaces. Yet another analogous example, which for the purposes of this paper will be very useful, is the category **Grph** with the object class of all graphs and arrow class the class of all homomorphisms between those graphs. Note that the last section of this chapter will expand on particularly the graph category. ♢

Example 2.3.4. A couple of other categories set don't involve functional arrows include the following:

One is the category generated by a group $(G, *)$. The object class is actually the singular set itself and the arrows are the elements in the group, which can be thought of as having the same domain and codomain. Notice that this is a category since group elements $g, f \in G$ are by definition closed under its operation $*$, hence $g * f$ forms a compsition arrow. It is furthermore easy to verify that e suffices for the identity arrow.

Another category is one formed by a poset $(P, \leqslant)$. We can think of the relation $a \leqslant b$ as an arrow between the elements $a, b \in P$. The property of transitivity gives us composition since given $a \leqslant b$ and $b \leqslant c$ we can always form $a \leqslant c$. Since arrows in the poset between domains are unique by construction, we find the relexive property $a \leqslant a$ gives us an identity arrow for each element. $\Diamond$

Example 2.3.5. Now some categories formulated from other categories shall be defined. Given a category $\mathcal{C}$, the dual category $\mathcal{C}^{op}$ (sometimes called the "opposite category") has the object class the same as $\mathcal{C}$ and the arrow class is formulated as follows: For every arrow $A \overset{f}{\mapsto} B$ in $\mathcal{C}$, there exists an arrow $B \overset{f^{op}}{\mapsto} A$ in $\mathcal{C}^{op}$. It is easy to verifty that $\mathcal{C}^{op}$ is a category.

Given two categories $\mathcal{C}_1$ and $\mathcal{C}_2$, a product category $\mathcal{C}_1 \times \mathcal{C}_2$ has the object class $Ob(\mathcal{C}_1) \times Ob(\mathcal{C}_2)$ and the arrow class $Ar(\mathcal{C}_1) \times Ar(\mathcal{C}_2)$. Composition forms from the operation $(g_1, g_2)(f_1, f_2) := (g_1 f_1, g_2 f_2)$ and the identity is $(1_{A_1}, 1_{A_2})$ for any object (A_1, A_2). This can be easily generalized to a family of categories $\{\mathcal{C}_\alpha\}_{\alpha \in J}$ for an arbitrary index J to form the product category $\prod_{\alpha \in J} \mathcal{C}_\alpha$ with the object class $\prod_{\alpha \in J} Ob(\mathcal{C}_\alpha)$ ad the arrow class $\prod_{\alpha \in J} Ar(\mathcal{C}_\alpha)$.

Given two categories $\mathcal{C}_1$ and $\mathcal{C}_2$, one can also form a coproduct category $\mathcal{C}_1 + \mathcal{C}_2$. The object class is the union $(Ob(\mathcal{C}_1) \times \{1\}) \cup (Ob(\mathcal{C}_2) \times \{2\})$ and the arrow class is the union $(Ar(\mathcal{C}_1) \times \{1\}) \cup (Ar(\mathcal{C}_2) \times \{2\})$. Here, it holds that the categorical structure of the arrows are preserved, and there are no arrows that map any object in $Ob(\mathcal{C}_1) \times \{1\}$ to any object in $Ob(\mathcal{C}_2) \times \{2\}$, since the object and arrow classes are disjoint by obvious design.

Like product categories, this can be generalized to a family of categories $\{\mathcal{C}_\alpha\}_{\alpha \in J}$ for an arbitrary index J to form the product category $\coprod_{\alpha \in J} \mathcal{C}_\alpha$ with the object class $\bigcup_{\alpha \in J}(Ob(\mathcal{C}_\alpha) \times \{\alpha\})$ and the arrow class $\bigcup_{\alpha \in J}(Ar(\mathcal{C}_\alpha) \times \{\alpha\})$. $\Diamond$

Remark 2.3.6. One should take special notice to how the co-product category was formed. Not only will this category prove useful in proving a major result very soon, but the way it was constructed brings note to a little known, yet useful process called the "disjoint union". This is a procedure that is typically done when one wants to either include the same element more than once or to include something "different" but for some reason is mathematically convenient to denote it under the same name. $\Diamond$

Next this section defines the thing that in some ways is more important than the category itself when it comes to how it gets utilized. This of course is the functor, which essentially maps one category to another. The thing to understand is that it is essentially a two piece function that maps objects to objects and then arrows to arrows, in a way that preserves the structure of how the arrows are arranged. In other words, if there is an arrow $A \overset{f}{\mapsto} B$, the functor F makes sure that some arrow $F(A) \overset{F(f)}{\mapsto} F(B)$ exists, and hence our domain and codomain get preserved in the mapping of F. The definition is as follows.

Definition 2.3.7. A **Functor** $F\colon \mathcal{C} \longrightarrow \mathcal{D}$ is a mapping from the category $\mathcal{C}$ to the category $\mathcal{D}$. More specifically a functor is a bi-product of two functions $F_{Ob}\colon Ob(\mathcal{C}) \longrightarrow Ob(\mathcal{D})$ and $F_{Ar}\colon Ar(\mathcal{C}) \longrightarrow Ar(\mathcal{D})$ such that the following conditions are satisfied:

1. Arrow domain and codomain structure is preserved under F. Hence, every arrow $A \overset{f}{\mapsto} B$ in $\mathcal{C}$ gets mapped to the arrow $F_{Ob}(A) \overset{F_{Ar}(f)}{\mapsto} F_{Ob}(B)$ in $\mathcal{D}$.
2. $F_{Ar}(1_A) = 1_{F_{Ob}(A)}$, for every object A in $\mathcal{C}$
3. $F_{Ar}(gf) = F_{Ar}(g) \circ F_{Ar}(f)$, for every arrow $A \overset{f}{\mapsto} B$ and $B \overset{g}{\mapsto} C$ in $\mathcal{C}$.

A category $\mathcal{C}$ is called **isomorphic** with $\mathcal{D}$ if there exists a functor $F\colon \mathcal{C} \longrightarrow \mathcal{D}$ such that there exists an inverse functor F^{-1}. $\triangle$

Remark 2.3.8. A few additional things to note about functors. First is that although it is "clear notation" to differentiate the object component of the functor from the arrow component, the functor is *really* one function that maps arrows to arrows. This is because the composition properties *imply* that the objects, which are the domain and codomain of each arrow, are consistently mapped to a single object that is the domain and codomain of the single mapped arrow.

Functors, as functions, also have the property that a given functor $F\colon \mathcal{C} \longrightarrow \mathcal{D}$ is injective if and only if F has a left inverse and F is surjective if and only if F has a right inverse (as a function typically does). Moreover, it is easy to verify that the image of a functor $F(\mathcal{D})$ is a category.

Finally, one should note the general limitations of isomorphism functors as a way of denoting "categorical equivilance". This is because the the identity functor is only one functor object that is equivilent up to so-called "natural isomorphism", which arises in the "category of functors", and captures more generally what it means to think of a category as "essentially the same". $\Diamond$

In category theory, there is a distinction one needs to make between "small" and "large" categories. Now that functors have been defined it is a good place to now make this distinction.

Definition 2.3.9. Basically, a **small category** is a category whose object class can be expressed as a set in ZFC set theory. A **large category**, on the other hand, is a category that is *not* small. In other words, the object class *cannot* be expressed as a set in ZFC set theory (such as the "universal set"). Furthermore, any functor whose domain and codomain is a small category will be called a **small functor**, and any functor whose domain *or* codomain is large will analogously be called a **large functor**. $\triangle$

Remark 2.3.10. Unfortunately, for the rigor-bent reader, this is not a rigorous definition since the involved axiomatic set theory brings much more technicality than it does insight to the theory this paper deals with. The curious reader looking for a more rigorous treatment should consult Bloch's introductory proofs text for the ZFC axioms, [2] MacLane for much more standard treatment of large categories. [7] The reader may also appreciate for a starting explanation of the NBG axioms used to rigorously define the "proper classes" for which make up the object classes of large categories. [9] ◊

Example 2.3.11. It is now a good time to define a the category **Cat** of all small categories. The object class is the class of small categories and the arrow class is the class of all small functors between those small classes. ◊

Next in the list of vital definitions is the notion of products and coproducts. A product as an object with projections to a family of objects is no new one. The co-product is a family of objects that all commonly project to the coproduct. Interestingly, a co-product in $\mathcal{C}$ is the product in $\mathcal{C}^{op.}$, and vice versa. Provided is first a definition of the specific case of product and coproduct where the family has specifically two elements (as this is the easier one to digest on first glance). Then the more general case is defined, which for this paper's purposes has its much needed utility.

Definition 2.3.12. Given two objects A and B in a category $\mathcal{C}$, a **cartesian bi-product** $A \times B$ of A and B in a category $\mathcal{C}$ is an object with the following properties.
1. There exists two projection arrows $A \times B \overset{\pi_A}{\mapsto} A$ and $A \times B \overset{\pi_B}{\mapsto} B$.
2. Given a collection of arrows $Y \overset{f_A}{\mapsto} A$ and $Y \overset{f_B}{\mapsto} B$, there exists a unique arrow $Y \overset{f}{\mapsto} A \times B$ such that $f\pi_A = f_A$ and $f\pi_B = f_B$.

A **cartesian co-bi-product** $A + B$ of A and B in $\mathcal{C}$ is an object with the following two properties.
1. There exists two projection arrows $A \overset{\mu_A}{\mapsto} A + B$ and $B \overset{\mu_A}{\mapsto} A + B$.
2. Given a collection of arrows $A \overset{f_A}{\mapsto} Y$ and $B \overset{f_B}{\mapsto} Y$, there exists a unique arrow $A + B \overset{f}{\mapsto} Y$ such that $\mu_A f = f_A$ and $\mu_B f = f_B$. △

Definition 2.3.13. Given a family of objects X_i in a category $\mathcal{C}$ under an index set I, a **cartesian product** $\prod_{i \in I} X_i$ is an object with the following properties.

1. For every $k \in I$, there exists a projection arrow $\prod_{i \in I} X_i \overset{\pi_k}{\mapsto} X_k$.

2. Given a collection of arrows $Y \overset{f_k}{\mapsto} X_k$, for each $k \in I$, there exists a unique arrow $Y \overset{f}{\mapsto} \prod_{i \in I} X_k$ such that $f\pi_k = f_k$, for each $k \in I$.

A **cartesian co-product** $\coprod_{i \in I} X_i$ in $\mathcal{C}$ is an object with the following two properties.

1. For every $k \in I$, there exists a projection arrow $X_k \overset{\mu_i}{\mapsto} \coprod_{i \in I} X_i$.

2. Given a collection of arrows $X_k \overset{f_k}{\mapsto} Y$, there exists a unique arrow $\coprod_{i \in I} X_i \overset{f}{\mapsto} Y$ such that $\mu_k f = f_k$, for each $k \in I$. △

Now is the point to discuss what is very central to this paper's application of category theory, and that is so-called "diagrams".

Definition 2.3.14. A **diagram** on a category $\mathcal{C}$ is a functor $F\colon J \longrightarrow \mathcal{C}$, where J is some category (that one might call the "index category").

A **small diagram** is a diagram $F\colon J \longrightarrow \mathcal{C}$ whose index category (but not necessarily the codomain category $\mathcal{C}$) J is small. Any diagram whose index category is not small shall be called a **large diagram**. △

Note that it does not matter what exactly the given "index category" *is*–in a similar way to an arbitrary index set–so much as it is does its job of taking a collection of objects, and describing its categorical structure of objects and arrows (in the category). In a similar way that an index set is used to index a family of elements in a set, an index category is used to index the objects in the category as well as the arrows between them.

Now here is one more definition before moving on to two more sections of applications. First is a cone, which is an intuitive geometric description of what is going on because a cone has a single point with a lot of points (in many cases uncountably many!) that connect to that point, which is more or less what is going on here. A limit on a diagram we can describe as essentially the "smallest cone" that exists–the measurement of size being the existance of a unique arrow from single point in one cone to a single point in another cone, such that the structure of the other cone commutes with that arrow.

Definition 2.3.15. In a category $\mathcal{C}$, a **cone** (N, ψ) in a diagram $F\colon J \longrightarrow \mathcal{C}$ is an object N with a family of arrows $N \overset{\psi_X}{\mapsto} X$ such that for any arrow $X \overset{f}{\mapsto} Y$ in J, we have $F(f)\psi_X = \psi_Y$.

A cone (L, φ) in the diagram F is a limit if given any cone (N, ψ), there exists a unique arrow $N \overset{u}{\mapsto} L$ such that for any arrow $X \overset{f}{\mapsto} Y$ in J, we find the arrows $L \overset{\varphi_X}{\mapsto} X$ and $L \overset{\varphi_Y}{\mapsto} Y$ and arrows $N \overset{\psi_X}{\mapsto} X$ and $N \overset{\psi_Y}{\mapsto} Y$ that exist by definition of a cone are such that $u\varphi_X = \psi_X$ and $u\varphi_Y = \psi_Y$. $\triangle$

Remark 2.3.16. It is pretty easy to verify that a limit (L, φ) in a given diagram is unique up to object isomorphism. This fact will be very useful in this paper. $\Diamond$

2.4 The Free Category

This section derives a concept that is very valuable, and that is the notion of a "free category". Now the "free category" has definitely been mentioned before in standard category theory texts. However, they are not entirely the holy grail "free category" that this paper seek, and neither are they really a "less general version", although in some respects it can be thought that way. Although that is not to say that this *other* "free category" is not useful as it definitely is, and is co-integrated heavily with the new free category

What the free category *is* intuitively is a category that one can generate "for free" from a diagram that essentially constructs a category with desirable properties. For example, one may want to make into products, or even co-products, while still preserving its original arrows, *without* adding arrows that are unneeded, or worse doesn't have desirable properties. Can it be done? With the free category, we can. However, its existence is not so straightforward. And that is why it is proven here.

First some additional vocabulary must be presented, which is the notion of categorical completeness.

Definition 2.4.1. A category $\mathcal{C}$ is small-complete if every small diagram $F\colon J \longrightarrow \mathcal{C}$ has a limit. △

While this definition seems rather arbitrary, it is more powerful than it appears. The power shall be demonstrated this after showing that **Cat** is a category that meets the "small complete" criterion. It does so with major ease, actually, since after all it *is* the category of ALL (small) categories, with just about every category possible, including categories that make up the limit of *any* diagram. Ok, he proof is not that simple, but it's surprisingly not that complicated. It requires a bit of an understanding of how to commute arrows, particularly with co-products. The theorem shall now be presented.

Theorem 2.4.2. **Cat** *is small-complete.*

Proof. Let $F\colon J \longrightarrow$ **Cat** be a small diagram. Let $\mathcal{V}$ be the set of cones on F. Let $L = \coprod_{(N,\psi)\in\mathcal{V}} N$ be the coproduct category of categories N indexed by the cone (N,ψ) on F. Let $u_N\colon N \longrightarrow L$ be the co-projection. For each X in J, let $\psi_{NX}\colon N \longrightarrow F(X)$ be the component functors in each cone (N,ψ) on F. It follows that there exists a unique mapping ϕ_X such that $u_N\phi_X = \psi_{NX}$, for every cone (N,ψ).

We shall first prove that (L,ϕ) is a cone on F, and in doing so prove that (L,ϕ) is a limit. Given a cone (N,ψ), we find u_N as the co-projection arrow is unique. Furthermore, we find that given the arrow $F(X) \overset{F(f)}{\mapsto} F(Y)$ in **Cat**, we have ψ_X and ψ_Y such that

$$\psi_X F(f) = \psi_Y$$

$$u_N \phi_X F(f) = u_N \phi_Y$$

$$\phi_X F(f) = \phi_Y .$$

□

Now the free category in its *most general* sense can be defined.

Definition 2.4.3. Given a small diagram $F\colon J \longrightarrow$ **Cat**, the the limit category $\mathcal{C}'$ of F, which exists by **Theorem 2.4.2**, we shall call the **Free Category** generated by the small diagram F. △

2.5 The Path Category and the Cartesian Path Category

This paper is one that definitely incorporates a lot of category theory. But in doing so, there is a healthy dose of graph theory as well. Graph theory meets category is essentially a vital application of the free category. For more graph theory, one should consult Churchill, but some brief graph theory definitions are provided here. [8]

Definition 2.5.1. A graph $G = \langle V(G), E(G)\rangle$ is a set $G(V)$ called the **vertices** with a multiset $G(E) \subseteq V(G) \times V(G) \times I$, for some index set I, called the **edges**. The index set I will often be called $E(G)$ index class.

Given a graph G, a **path** $p = (p_E, p_V)$ of order $n \geqslant 0$ in a graph G is a finite sequence of edges $p_E = (e_1, \ldots, e_n)$ and a function $p_V\colon \{0, \ldots, n\} \longrightarrow V(G)$ such that $e_k = (p(k - 1), p(k), i_k)$, for each $1 \leqslant k \leqslant n$, for some index i_k in the $E(G)$ index class.

A **graph homomorphsim** $h\colon G \longrightarrow \mathcal{C}$ between a graph G and a category $\mathcal{C}$ is a mapping between them in the following way: h consists of two functions $h_V\colon V(G) \longrightarrow Ob(\mathcal{C})$ and

$h_E\colon E(G) \longrightarrow Ar(\mathcal{C})$ such that the following condition is satisfied: $h_E((x, y, i)) = f$, for some arrow $h_V(x) \overset{f}{\mapsto} h_V(y)$.

A **graph homomorphsim** $h\colon G_1 \longrightarrow G_2$ between a graph G_1 and another graph G_2 is a mapping between them in the following way: h consists of two functions $h_V\colon V(G_1) \longrightarrow V(G_2)$ and $h_E\colon E(G_1) \longrightarrow E(G_2)$ such that the following condition is satisfied: $h_E((x, y, i_1)) = (h_V(x), h_V(y), i_2)$, for some index i_2 in the $E(G_2)$ index class.

Any graph homomorphism h (of both graph and categorical varieties) is said to be a **graph isomorphism** if h_V and h_E is bijective. △

In this definition, one should also note that there is yet another instance of the multiset, this time with the definition of edges. This is a practical necessity, since more than one edge may connect a given pair of vertices, in the same way that there may be more than one arrow in a category.

Speaking of which, graphs can be thought of in a way as its own category, with paths (as defined earlier) playing the roll as the arrow, with the domain and codomain being the start and endpoints of the path respectively. This works because this gives the graph structure identity and composition. For identity, one has the trivial path of (1_x), with an empty sequence of edges and a function $1_x\colon \{0\} \longrightarrow G(V)$ defined by $1_x(0) = x$. For composition, given two paths $((e_1, \ldots, e_{n_1}), p_{1V})$ and $((e'_1, \ldots, e'_{n2}), p_{2V})$ such that $p_1(n_1) = p_2(0)$, one can form the path $((e_1, \ldots, e_{n_1}, e'_1, \ldots, e'_{n2}), p_{1V}p_{2V})$, where $p_{1V}p_{2V}\colon \{1, \ldots, n_1 + n_2\}$ defined by $p_{1V}p_{2V}(n) = p_1(n)$ for $1 \leqslant n \leqslant n_1$ and $p_{1V}p_{2V}(n) = p_2(n)$ for $n_1 \leqslant n \leqslant n_2$.

Definition 2.5.2. The **path category** G' of a graph G has the object class of vertices $G(V)$ and arrows the class of all paths $G(P)$. △

Defining the category seems easy. But deriving a way to generate functors between two path categories? Not so much. It would be *really* nice if one could have a functor $\mathbf{C}\colon \mathbf{Grph} \longrightarrow \mathbf{Cat}$ that can generate a functor that goes from a graph homomorphism between two graphs and a functor in the path categories. Turns out there *is* such a functor, but in order to derive this effectively, one needs the free category. It turns out the path category can be constructed in another way, which is going to be the diagram provided in the following definition.

Definition 2.5.3. The **graph diagram** $F\colon J \longrightarrow \mathbf{Cat}$ of a graph G shall be defined as follows. $F(J)$ is the category of all categories $F(X)$ for X in J such that there exists a graph homomorphism $Ob(F(X))$ is surjectively mapped by a graph homomorphism $h\colon G \longrightarrow F(X)$ such that h_V surjectively maps $G(V)$ to $Ob(F(X))$. The arrows $F(f)$ for f in J are functors $F(f)\colon F(X) \longrightarrow F(Y)$ such that given two graph homomorphisms $h_Y\colon G \longrightarrow F(X)$ and $h_Y\colon G \longrightarrow F(Y)$ such that $(h_Y)_V$ and $(h_Y)_V$ are surjective and $F(f) \circ h_X = h_Y$. (in other words $F(f)_{Ob.} \circ (h_X)_V = (h_Y)_V$ and $F(f)_{Ar.} \circ (h_X)_E = (h_Y)_E$)

The **free category** G^* generated by a graph G is the free category generated by the graph diagram F. △

Lemma 2.5.4. *Let $h\colon G \longrightarrow \mathcal{C}$ be a graph homomorphism between a graph G and a category $\mathcal{C}$. Let G' be the path category. Then there exists a functor $H\colon G' \longrightarrow \mathcal{C}$ such that $H_{Ob.}(x) = h_V(x)$ and $H_{Ar.}(((e), p_V)) = h_E(e)$.*

Proof. Let $H\colon Ar(G') \longrightarrow Ar(\mathcal{C})$ be defined by

$$H(f) = \begin{cases} 1_{h_V(a)}, & \text{if } f = 1_a \text{ for some vertex } a \text{ in } G, \\ h_E(e), & \text{if } f \text{ is order 1, where } f = ((e), p_V), \\ H(f')H(s), & \text{if } f = f's, \text{ where } f' \text{ is a path and } s \text{ is an order 1 path.} \end{cases}$$

We shall first show H is a well-defined function by induction on the order of arrows in G'. Note that H is defined on all order zero and one arrows of G'. Given that H is defined on order n arrows, and f is an order $n+1$ arrow, we find $f = f's$ for some order n arrow f' and some order 1 arrow s, which completes our inductive hypothesis.

Next we show that H satisfies the essental conditions for a functor as an arrow mapping. The condition that for all vertices a in G, there exists some object A in $\mathcal{C}$ such that $H(1_a) = 1_A$, is is immediate. Given an arrow $f = f_1 f_2$, the condition that $H(f) = H(f_1)H(f_2)$ follows by very straightforward induction on the order of f_2.
By construction, we find H is such that $H_{Ob.}(x) = h_V(x)$ and $H_{Ar.}(((e), p_V)) = h_E(e)$. □

Lemma 2.5.5. *Let G be a graph and G' be the path category of G, and $F \colon J \longrightarrow$ **Cat** be the graph diagram. G' is an object of $F(J)$.*

Proof. Define $h_V \colon G(V) \longrightarrow G'$ by $h_V = id_{G(V)}$ and $h_E \colon G(E) \longrightarrow G'$ by $h_E(e) = ((e), p_V)$. This forms a graph homomorphism $h \colon G \longrightarrow G'$. □

Now for the important theorem.

Theorem 2.5.6. *Let G be a graph. The free category G^* generated by G is isomorpihic to the path category G' of G.*

Proof. Let $F \colon J \longrightarrow$ **Cat** be the graph diagram. For every object X in J, choose a graph homomorphism $h_X \colon G \longrightarrow F(X)$ with the property that $(h_X)_V$ is surjective. By **Lemma 2.5.4**, we can choose a functor $H \colon G' \longrightarrow \mathcal{C}$ such that $H_{Ar.}(((e), p_V)) = h_E(e)$.
We shall first prove that (G', H) is a cone. Given an arrow $F(X) \overset{F(f)}{\mapsto} F(Y)$ in the graph diagram, we shall prove that $F(f) \circ H_X = H_Y$ by induction on the order of the path p. If p is order zero, then $p = 1_a$, for some vertex a, and

$$H_Y(1_a) = 1_{h_Y(a)} = 1_{(F(f) \circ h_X)(a)} = (F(f) \circ H_X)(1_a).$$

If $p = ((e), p_V)$, and is order 1, we find that $H_Y(p) = h_Y(e) = (F(f) \circ h_X)(e) = (F(f) \circ H_X)(p)$. If p is order $n+1$ and our inductive hypothesis holds for order n, then $p = p's$ for some order n path p' and order 1 path s, hence

$$H_Y(p) = H_Y(p')H_Y(s) = (F(f) \circ H_X)(p')(F(f) \circ H_X)(s) = (F(f) \circ H_X)(p).$$

We shall now prove that G' and G^* are isomorphic. We find by **Lemma 2.5.5** that G' is an object in $F(J)$. Note there is some cone (G^*, ϕ) that forms a limit in F. Then there exists some unique $u\colon G' \longrightarrow G^*$ such that $\phi_{G'} \circ \mu = H_{G'}$. It is easy to verify that $H_{G'} = id_{G'}$. It follows that $\phi_{G'}$ is a left inverse of μ, hence by uniqueness of μ we conclude that μ is a categorical isomorphism. □

While the path category and the free functor has very proven utility, it turns out that for the purposes of this paper, one needs to construct something fairly more sophisticated and unfortunately less intuitive. The motivation, however, is simple. It would sometimes be very nice to construct a category out of a graph that not only *is* a category but is a category that has various desired qualities, namely cartesian products, without taking away any originally desired arrows. Turns out that a bit more creativity with small diagrams provides exactly this results.

Definition 2.5.7. The **cartesian graph diagram** $F\colon J \longrightarrow \mathbf{Cat}$ of a graph G shall be defined as follows. $F(J)$ is the category of all cartesian categories $F(X)$ for X in J such that $Ob(F(X))$ is surjectively mapped by a graph homomorphism $h\colon G(V) \longrightarrow Ob(F(X))$. The arrows $F(f)$ for f in J are cartesian-preserving functors $F(f)\colon F(X) \longrightarrow F(Y)$ such that there exsts surjective graph homomorphisms $h_X\colon G(V) \longrightarrow Ob(F(X))$ and $h_Y\colon G(V) \longrightarrow Ob(F(Y))$ such that $F(f) \circ h_X = h_Y$.
The **free cartesian category** G^* generated by a graph G is the free category generated by the cartesian graph diagram F. △

This next theorem, like the earlier one, shows what the free cartesian category *is.* Essentially, it is the path category except with more arrows–to be precise, exactly the arrows needed to assure that it is a cartesian category, no more, and no less. The free cartesian category, as a result, manifests itself in the following way. At this current stage of research, although this is an accepted fact, a proof is currently in the working stage (and not quite finished at this time) so this will simply be an accepted construct.

Theorem 2.5.8. *Let G be a graph. The free cartesian category G^* generated by G has the following characteristics.*

1. G^* is in the cartesian graph diagram.

2. There exists an embedding functor $\kappa\colon \mathbf{C}(G) \longrightarrow G^*$.

3. f is an arrow in G^* is an arrow if and only if f is in $\kappa(\mathbf{C}(G))$ or $f = f'\Pi_{i\in I} f_i$ such that I is a small index, and f' and $\{f_i\}_{i\in I}$ are in G^*.

Lemma 2.5.9. *Let G be a graph and $\mathcal{C}$ be a cartesian category. Let $H\colon \mathbf{C}(G) \longrightarrow \mathcal{C}$ be a functor and G^* be the free cartesian category generated by G. Then there exists a cartesian functor $CH\colon G^* \longrightarrow \mathcal{C}$.*

Also this corollary will be very important throughout this text.

Corollary 2.5.10. *Let G_1 and G_2 be graphs. Let G_1^* be the free categories generated by G_1. Let G_1^{**} and G_2^{**} be the free cartesian categories generated by G_1 and G_2. The following are equivalent.*

1. There exists a graph homomorphism $h\colon G_1 \longrightarrow G_2^{**}$.

2. There exists a functor $H\colon G_1^* \longrightarrow G_2^{**}$.

3. There exists a cartesian functor $CH\colon G_1^{**} \longrightarrow G_2^{**}$.

3
Language and Inference

This Chapter is the center of what this project seeks to do, which is to not only come up with an adequate definition of “logic”, but also to find a way to appropriately compare two logics. In logic, it seems like there is not a lot of great ways to compare logic besides the normal paradigm of classical logic and model theory.

The first section deals with languages and a general analogue that fits most of the languages that have appeared in logic.

The section section deals with formula vectors, which will suffice to be the objects of a categorical treatment that will be implemented.

Construction of these formula vector objects set stage for the third section, which very importantly defines what a logic is and provides a precise definiton of the arrows in this categorical definition provided. Vital to this is a rigorous definition for inference maps which overlay the horizontal bar inference scheme.

The fourth section quickly touches on how versatile the inference definition can be by showing that it can portray hypotheses in the way they ought to be.

The fifth section makes a sort of semantics that will prove very helpful in this logic as they are "functoral semantics". This will allow us to compare logics and give us a richer sense of "soundness" and "completeness".

Finally, for more general notions of semantics, a general notion of "truth tables" will be created as a class of "valuation mappings" that one can think of as "possible worlds" as sometimes one thinks of in model theory.

3.1 Languages

Whether one knows it or not, when one reads, writes, and speaks math, one is working with a language that is not English! Or rather, math is a language that deep to its core has a very constructive nature, whereas with a literary language such as English one typically takes the constructive rules one has (such as grammar, sentence, even idiomatic structure) as they are, not really concerned with analyzing the nature of the language's construction itself.

While most mathemeticians do not really go as in depth with the language's construction as the typical logician, most concern themselves with the basic structure in what they are studying. For example, with a group, topology, poset, etc., they concern themselves just as much with the inherent structure of what the object is as we are with its consequences.

This first definition tries to stay true to this strong tradition in mathematics hereby described–which is often called "rigor". It also tries to generally define what logicians have defined for as long as Tarski, which is the notion of "language": Something mathemeticians use in every day math.

Definition 3.1.1. A **Language** $\mathcal{L}$ consists most basically of a class of formula called $Form$. This class is defined recursively as follows.

1. **Atomic Formula**: A class of atomic formula *Atom*.

2. **Propositional Connectives**: con_j, for each $j \in J$, where J is a finite index set. Each propositional connective is a mapping of a finite integer $n \geqslant 0$ of propositions to another proposition $con_j \colon Form^n \longrightarrow Form$ In other words, if $\phi_1, \ldots, \phi_n$ are all propositions then $con_j(\phi_1, \ldots, \phi_n)$ forms a new proposition that is linguistically distinct from any atomic prior proposition. △

As one might infer from this definition, math is hard, but at least the language behind it is simple. In other words, there are these "atomic" things that is taken at the most basic level of formula composition and then join formulae together with connectives to make new formula. And the kicker: ALL of our statements in the language can be broken down into that atomic formulae joined together by connectives. If only English behaved that way. Here are some examples that are prevalent in logic.

Example 3.1.2. The most basic example of a language is a propositional language with *Atom* being composed of "propositional variables" $p_1, p_2, \ldots, p_n, \ldots$ and "propositional constants" $\bot, \top$ with connectives $\vee, \wedge, \neg, \rightarrow$. This language shall be more rigorously in Chapter 4, and this paper will work with variants of this sort of language throughout. ◊

Example 3.1.3. One of the most applied sort of language of all logic is called the "first order predicate language". This language is more complex in the sense that its *Atom* class is constructed from a $Term$ class made up of the following.

1. Variables: A class of variables $Vars$ consisting of one symbol for each natural number $x_1, x_2, \ldots, x_n, \ldots$.

2. Constants: A class of constant symbols c_k indexed by some arbitrary K. What exactly (and how many of) these constant symbols there are varies from language to language.

3. Functions: For each $n \in \mathbb{N}$, a class of countably many n-ary function symbols $f^n \colon Term^n \longrightarrow Term$.

The class *Atom* is then constructed through the usual $\bot, \top$ constants. But our "propositional variables" are more elaborately n-ary relation symbols R^n (up to countably many for each $n \in \mathbb{N}$), such as $=$ (this one appears quite frequently), which take n terms and form our atomic formulas. This is to say, $R^n(t_1, \ldots, t_n) \in Atom$, for $(t_1, \ldots, t_n) \in Term^n$. We then have the usual connectives $\vee, \wedge, \neg, \rightarrow$, as our last example. But we also have quantifiers $\forall, \exists$, which are not exactly connectives but can be defined using connectives. We define $Qx \colon Form \longrightarrow Form$ as a connective, for every variable x and a given quantifier Q. So $\forall x \phi$ and $\exists x \phi$ would be a formulae, and $\forall x$ and $\exists x$ would be a connectives. $\Diamond$

Example 3.1.4. The next language is an expanded version of the "First Order Predicate Language" called the "Second Order Predicate Language." There is the class $Term1$, which are "first order terms" which contain only the variable and constant classes as described in the last example. But what really distinguishes this language is that there are a class of second order terms called $Term2$, which essentially contains "predicate symbols" that show up in n-ary form for each $n \in \mathbb{N}$. These n-ary predicate symbols are either variables $X_1^n, X_2^n, \ldots, X_m^n, \ldots$, for which are indexed by $\mathbb{N}$, or constants P_k^n indexed by an arbitrary K_n. Atomic formulas are not of the form $T^n(t_1, \ldots, t_n)$, where $T^n \in Term2$ and $(t_1, \ldots, t_n) \in Term1$. The connectives and quantifiers are still $\vee, \wedge, \neg, \rightarrow$ $, \bot, \top$ and $\forall, \exists$, except that the quantifiers are now able to quantify variables from $Term2$ in addition to those from $Term1$ (in other words $\forall X^n$ and $\exists X^n$ are connectives for each n-ary variable). $\Diamond$

Example 3.1.5. This one is not prevalent in logic...but maybe it should be. A general kind of language that we shall be dealing with (a term that I as the writer made up, but for good reason) is called the "Meta-Language" $\mathcal{M}$. The idea is that when there is a language and the metalinguistic usage becomes so vast with symbols and rules, such as with sequents, that it necessitates a language of its own. This language shall be defined more rigorously in Chapter 5, but the basic idea is that formulas $Form_{\mathcal{L}}$ in a base language $\mathcal{L}$ become our new class of terms, and our class of formulas $Form_{\mathcal{M}}$ become n-ary relation symbols that operate on $Form_{\mathcal{L}}$. $\Diamond$

Keep in mind that within the context of this paper, the main (and most inherently basic) language dealt with is that of the propositional variety. While predicate languages (ones that utilize quantifiers) are very important in mathematics (and not to mention computer science), introducing quantifiers in a reasonably rigorous way involve a level of technical sophistication that clouds what is really going on with the nature of deductive systems.

3.2 Formula Vectors

Typically in logic, one makes theorems about a relation between a given *set* of formulas and a *single* formula. That is to say, the relations look something like $\Gamma \vdash \phi$ or $\Gamma \models \phi$, where Γ is a set of formulas (of arbitrary cardinality) and ϕ is a formula. This is a *very unideal* relation to work with since the relation is one that relates between two different domains of objects. Wouldn't it be nicer if relations like $\vdash$ and $\models$ were binary relations *on* a given domain?...especially if such relations happened to have had a transitive property where $\Gamma_1 \vdash \Gamma_2$ and $\Gamma_2 \vdash \Gamma_3 \implies \Gamma_1 \vdash \Gamma_3$?

Turns out that can be done. This is where the category theory will come in. In a potential categorical scheme of a given language, which is the plan for this section, this definition gives the notion of objects in the category that will be worked with. However, there are two important disclaimers to address as to why this definition was chosen to be the way it is, which at first definitely appears odd.

The first disclaimer: "formula vectors" were decided instead of "sets of formulas". This is because vector objects are better-suited for some of the later definitions than sets ever will be; and while there is trade offs that results from this definition, they are nothing that does not have a straightforward fix. The second disclaimer follows from the first: Exclusively *finite* "formula vectors" were decided on instead of possibly *infinite* formula vectors (which could have been done by defining my formula vectors using functional mappings $f\colon J \longrightarrow Form$, for any index set J of cardinality $|J| \leqslant |Form|$). The decision was not including infinite vectors because they are a big complication that makes the system harder to work with, and for less theorecal insight than some of the other things that could be given atention.

Definition 3.2.1. Given a language $\mathcal{L}$, a $\mathcal{L}$-vector is an ordered pair of $\mathcal{L}$-formulas. The set of all $\mathcal{L}$-vecters we shall denote as $\mathbb{V}$. Note that in addition to all ordered pairs of order $n \geqslant 1$,the "empty" vector shall be included with no tuples and call it $\vec{0}$.

Additionally defined are some operations on $\mathbb{V}$. Suppose $\Gamma = \langle \theta_1, \ldots, \theta_n \rangle$ and $\Delta = \langle \phi_1, \ldots, \phi_m \rangle$.

First defined is $\langle -, - \rangle$ on $\mathbb{V}$ by $\langle \Gamma, \Delta \rangle = \langle \theta_1, \ldots, \theta_n, \phi_1, \ldots, \phi_m \rangle$. An anologous more general definition will be given for $\underbrace{\langle -, \ldots, - \rangle}_{\text{n times}}$.

Then defined the **projection mapping** π_i on $\mathbb{V}$ for each $i \geqslant 1$, provided that a given vector has an i-th tuple. This is to say that $\pi_i(\Gamma) = \theta_i$, for every $1 \leqslant i \leqslant n$, but is undefined for all $i > n$.

Next, defined is the **order** of a given $\mathcal{L}$-vector as the number of tuples it contains, and notate it $|\Gamma|$, that is to say $|\Gamma| = n$.

Finally defined is the binary relation $\preceq$ on $\mathbb{V}$, so then $\Gamma \preceq \Delta$ if and only if for every $1 \leqslant i \leqslant |\Gamma|$, there is some $1 \leqslant j \leqslant |\Delta|$ such that $\pi_i(\Gamma) = \pi_j(\Delta)$. △

3.3 Inferences and Proofs

Out of everything in this whole paper, this specific section out of all sections of all is perhaps the MOST IMPORTANT out of everything written. This is because the rest of this paper cannot do what it has done without first going back and placing sufficiently general definitions that can be rigorously worked with. Once that has been accomplished, the nature of deductions, the nature of metamath, becomes arithmatic–and that is just beautiful!

First, here is a definition of an "inference rule" that is pretty standard for the most part.

Definition 3.3.1. In a language $\mathcal{L}$, an **Inference Rule** is an ordered pair $\langle \Gamma, \phi \rangle$, where Γ is an $\mathcal{L}$-formula vector and ϕ is an $\mathcal{L}$-formula. △

Now here is something that is not really standard at all. When dealing with inference rules, many mathematical texts simply take it for granted that we can take a collection of these "inference rules" and make an algorithmic "horizontal bar style"' equation that is supposed to make sense. But in order to work meaningfully with this "horizontal bar style" thing, one should, like anything, define it rigorously, and this is done here as standard as a functional mapping.

Definition 3.3.2. An **Inference Map** I is a family of inference rules $\{\langle \Gamma, \phi_\Gamma \rangle\}_{\Gamma \in \Omega}$ indexed by some $\Omega \subseteq \mathbb{V}$. In other words, it corresponds to a function $I \colon \Omega \longrightarrow \mathbb{V}$. When defining inference maps, very standard horizontal bar notation shall be used. So $I(\Gamma) = \Delta$ shall be written as

$$\frac{\pi_1(\Gamma) \quad \pi_2(\Gamma) \quad \ldots \quad \pi_{|\Gamma|}(\Gamma)}{\Delta}\, I,$$

although it will be very convenient to not just write the projections spaced, but simply

$$\frac{\Gamma}{\Delta} I.$$

△

Next we define a logic, which is something of a new abstract concept, inspired by the "set with structure" motif. Except what a logic is, and what it really always has been, is a language with additional inference structure that allow one to make deductions. The structure will be a set of inference maps that will establish the framework for how the logi to work.

Definition 3.3.3. A **Logic** $\mathscr{L}$ is an ordered pair $\langle \mathcal{L}, \mathscr{T} \rangle$, where $\mathcal{L}$ is a language and $\mathscr{T}$ is a finite family of inference rules. △

Now, with the inference rules, it's crucial to generate a deductive system that behaves categorically. By defining inference maps, essentially half the battle has been won, which is to have our "one step proofs", by going from the inference input to the inference output. But now one must be able to 1. make proofs that combine our "one step proofs" and 2. find a way to keep our proof closed under products in the way that one would want. This is where "free cartesian categories"–which was focused in chapter two towards deriving– come in very handy. This is because free categories essentially say that the category that one "wants" to have exists, as one shall witness from the following definition.

Definition 3.3.4. A language $\mathcal{L}$, an **Logic** is an ordered pair $\langle \Gamma, \Delta \rangle$, where Γ, Δ are $\mathcal{L}$-formula vectors. The **Deductive Graph** $\mathbf{G}(\mathscr{L})$ generated by $\mathscr{L}$ is defined as follows:
Vertices: All formula vectors in $\mathcal{L}$.
Edges: An edge $\langle \Gamma, \Delta \rangle$ between two formula vectors Γ and Δ exist if and only if
1. There is a projection map π_i such that $\Gamma \overset{\pi_i}{\mapsto} \Delta$. In other words, if $\Gamma = \langle \phi_1, \ldots, \phi_n \rangle$, then $\Delta = \phi_i$, for some $1 \leqslant i \leqslant n$. These edges shall be called **projection edges**

2. There is an inference map $I \in \mathscr{T}$ such that

$$\frac{\Gamma}{\Delta} I.$$

These edges shall be called **inference edges**.

Let the **Deductive Category** $C(\mathscr{L})$ be the free *cartesian* category generated by $\mathbf{G}(\mathscr{L})$.

$\triangle$

Remark 3.3.5. If there exists a path between two formula vectors Γ and Δ in a given logic we shall metalinguistically write $\Gamma \gg \Delta$...or $\Gamma \vdash \Delta$, $\Gamma \models \Delta$, and other symbols corresponding to standard notation in the particular logic with which we are working. $\Diamond$

Now here is proof that this category has desired properties.

Lemma 3.3.6. *Let $\Gamma, \Delta_1, \Delta_2$ be formula vectors in a logic $\mathscr{L}$. Suppose $\Gamma \gg \Delta_1$ and $\Delta_1 \gg \Delta_2$. Then $\Gamma \gg \Delta_2$.*

Proof. By hypothesis there exists two arrows $\Gamma \overset{f}{\mapsto} \Delta_1$ and $\Delta_1 \overset{g}{\mapsto} \Delta_2$ in $C(\mathscr{L})$, so our conclusion follows from existance of the arrow $\Gamma \overset{gf}{\mapsto} \Delta_2$. $\square$

Lemma 3.3.7. *Let Γ, Δ be formula vectors in $\mathscr{L}$. If $\Gamma \gg \pi_i(\Delta)$ for each $1 \leqslant i \leqslant |\Delta|$, then $\Gamma \gg \Delta$.*

Proof. Follows from $C(\mathscr{L})$ being a cartesian category. $\square$

Proposition 3.3.8. *Let Γ_1, Γ_2 be formula vectors in $\mathscr{L}$. If $\Gamma_1 \preceq \Gamma_2$, then $\Gamma_2 \gg \Gamma_1$.*

Proof. Suppose $\Gamma_1 \preceq \Gamma_2$. Then by definition, we have $\Gamma_2 \gg \pi_i(\Gamma_1)$, for each $1 \leqslant i \leqslant |\Gamma_1|$. It follows from **Lemma 2.3.7** that $\Gamma_2 \gg \Gamma_1$. $\square$

Corollary 3.3.9. *Let $\Gamma_1, \Gamma_2, \Delta$ be formula vectors in $\mathscr{L}$. If $\Gamma_1 \gg \Delta$ and $\Gamma_1 \preceq \Gamma_2$, then $\Gamma_2 \gg \Delta$.*

It shall moreover be useful to define "logical equivalence" in the category $\mathbf{CC}(\mathscr{L})$, since categorically it is possible to run into scenarios where both $\Gamma \gg \Delta$ and $\Delta \gg \Gamma$. This holds a lot of theoretical significance since being able to deduce one thing from another and vice versa captures the essence of "logical equivalence"–similar to what has been captured before in a language with something like $\phi \leftrightarrow \varphi$. But this new convention can do something more general *metalinguistically.*

Definition 3.3.10. Let Γ and Δ be formula vectors in a given logic $\mathscr{L} = \langle \mathcal{L}, \mathscr{T} \rangle$. If $\Gamma \gg \Delta$ and $\Delta \gg \Gamma$, we shall call these two vectors **Logically Equivalent** and denote this quality with the notation $\Gamma \ll\gg \Delta$ (or alternatively $\Gamma \dashv\vdash \Delta$, $\Gamma \Dashv\vDash \Delta$, etc.).
Furthermore, the definition of derivability shall be extended in terms of inference maps. Sn inference map $I \colon \Omega \longrightarrow \mathbb{V}$ shall be said to be **Derivable** in a given logic $\mathscr{L}$, and write $\mathscr{L} \gg I$ if $\Gamma \gg I(\Gamma)$, for every $\Gamma \in \Omega$. $\triangle$

It is firstly important to understand $\ll\gg$ is an equivalence relation.

Lemma 3.3.11. *In a given logic $\mathscr{L}$, $\ll\gg$ is an equivalence relation.*

Proof. Reflexivity follows from **Lemma 3.3.8** since $\Gamma \preceq \Gamma$, for every formula vector Γ. Symmetry follows by definition. Transitivity follows directly from **Lemma 3.3.6**. $\square$

One should ask themselves what one would "want" to be logically equivalent. What should be logically eqivalent, first off, is when two vectors of the same order contain the same tuples but are not necessarily the same order. Actually...more generally want two formula vectors Γ, Δ such that $\Gamma \preceq \Delta$ and $\Delta \preceq \Gamma$ to be logically equivalent, which follows from two uses of **Proposition 2.3.7**. The following corollaries shall be presented to make the point clear, but with their proofs omitted.

Corollary 3.3.12. *Let Γ, Δ be formula vectors such that $\Gamma \preceq \Delta$ and $\Delta \preceq \Gamma$. Then $\Gamma \ll\gg \Delta$.*

Corollary 3.3.13. *Let $\langle \phi_1, \ldots, \phi_n \rangle$ be a formula vector in a logic $\mathscr{L}$ and let σ be a permutation on n. Then $\langle \phi_1, \ldots, \phi_n \rangle \lll\ggg \langle \phi_{\sigma(1)}, \ldots, \phi_{\sigma(n)} \rangle$.*

Definition 3.3.14. We shall define **Basic Arrows** in a given logic $\mathscr{L}$ as follows:

1. If Γ, Δ are formula vectors such that $\Delta \preceq \Gamma$, we shall call the unique arrow–that exists by virtue of projection edges existing from Γ to $\pi_i(\Delta)$, for each $1 \leqslant i \leqslant |\Delta|$–a **Projection Arrow**

2. If Γ, Δ are formula vectors such that $\Delta \npreceq \Gamma$ and there exists an inference edge (Γ, Δ), then the arrow $\Gamma \overset{P}{\mapsto} \Delta$ mapped by (Γ, Δ) is called an **Inference Arrow**

Any arrow that is not an basic arrow is called a **composite arrow**. We shall call any arrow between Γ and Δ generated by a family of arrows from Γ to $\pi_i(\Delta)$, for each $1 \leqslant i \leqslant |\Delta|$, a **product arrow**.

Finally, the **order** of an arrow will be defined recursively as follows: Given a single-product composite arrow, the order is defined as the number of inference arrows used to compose it. Given a product arrow, take the maximum order of the projection arrows used to compose it. △

Do note that the nature of the free category allows this definition to work properly.

3.4 Natural Deduction and Hypotheses

While my definition for inference rules is a very rigorous definition that embodies essentially what they *are*, up until this point there has not been a rigorous treatment of hypotheses...such as the sort of chargin and discharging of hypotheses that one would frequently see in classical/intuitionistic propositional/predicate logic, such as

$$\dfrac{\begin{array}{c}[\phi]\\ \vdots\\ \varphi\end{array}}{\phi \to \varphi} \to I$$

In this section, the much needed answer is provided.

The answer is doing what will be done a lot in this paper, which is to be clever with the domain of the inference map. The inference map will map all inference vectors such that a hypothesized vector incorporated with theoriginal vector reaches a given conclusion vector. The input of the inference map, is then defined into components that are the hypothesis component and conclusion component. This gets us something rigorous that essentially makes hypotheses work the way one would like.

Definition 3.4.1. A **hypothesis inference map** in a given language $\mathcal{L}$ is an inference map $H\colon \Omega_\Gamma \longrightarrow \mathbb{V}$, where $\Omega_\Gamma \subseteq \mathbb{V}$ such that for each $\Delta \in \Omega_\Gamma$, we have $\Delta := \langle \Phi, \Delta' \rangle$ such that $\langle \Gamma, \Phi \rangle \gg \Delta'$. Γ shall be called the **hypothesis componenet** of Δ and Δ' shall be called the **conclusion component**. Such an inference map is written

$$\dfrac{\begin{matrix}[\Gamma]\\ \vdots \\ \Delta'\end{matrix}}{H(\Delta)}\, H,$$

where Γ and Δ' are the hypothesis and conclusion components respectively. △

More generally, there might be more than one hypothesis and conclusion component. For instance,

$$\dfrac{\phi \vee \varphi \quad \begin{matrix}[\phi]\\ \vdots \\ \theta\end{matrix} \quad \begin{matrix}[\varphi]\\ \vdots \\ \theta\end{matrix}}{\theta}\, \vee E,$$

in the usual propositional logic. To deal with that, there inference map could instead compose of as many vectors of hypothesis and conclusion components as needed. This is done in the next definition

Definition 3.4.2. Extending the earlier definition, a hypothesis inference map $H\colon \Omega_\Gamma \longrightarrow \mathbb{V}$ in a given language $\mathcal{L}$ more generally is defined as follows. $\Gamma := \langle \Gamma_1, \ldots, \Gamma_n \rangle$ and $\Delta := \langle \Phi, \Delta'_1, \ldots, \Delta'_n \rangle$ such that for every $1 \leqslant i \leqslant n$, we have $\langle \Gamma_i, \Phi \rangle \gg \Delta'_i$. Each Γ_i and Δ'_i

are called hypothesis and conclusion components for each $1 \leqslant i \leqslant n$, and shall be written

$$\dfrac{\begin{array}{ccc}[\Gamma_1] & & [\Gamma_n] \\ \vdots & & \vdots \\ \Delta'_1 & \dots & \Delta'_n\end{array}}{H(\Delta)}\,H,$$

Where Γ_i and Δ'_i are hypothesis and conclusion components respectively for each $1 \leqslant i \leqslant n$. △

3.5 Functoral Semantics

This section is where the law of the land is laid down as far as "semantics" go. Typically in logic (as it is presented in a typical introductory course), one thinks of semantics as a disdinct notion from that of so-called "natural deduction" and "proof theory". This paper is of the opinion that this approach misses the point at what semantics intends to do, which is capture the meaning and interpretation of a given system. Really the way "semantics" as one has seen, i.e. truth tables and models, captures the way a given system works through entailment, is really in and of itself a kind of logical deduction.

What is meant by this? This means that the deductive category we've been using captures the essence of model theory AS WELL AS proof theory. There are objects that are generated by the language, and there are arrows that are brought about by "truth tables" which embody the most exquisite of inference rules in a way that shall soon be explored. That is all there really is to it.

In this perspective, so-called "soundness" and "completeness" shed a new, more *complete*, light. As a result of this categorical treatment, one can now talk about soundness and completeness as a relation between two different "logics", as opposed to one between "truth" and "provability". And yes, truth table/interpretation entailment in a given language IS a logic, now, different from so-called "propositional logic" or "propositional calculus".

Now what is important with soundness from one logics $\mathscr{L}_1$ to $\mathscr{L}_2$ is two things: 1. That every derivation arrow in $C(\mathscr{L}_1)$ can be expressed as a derivation arrow in $C(\mathscr{L}_2)$ via a functor F 2. The image of the functor F captures essentially *all* of the logical system of $C(\mathscr{L}_2)$. To clarify what is meant by 2, here is what one would not want for a definition of soundness: Simply having a more existance of functors between the two logics. The would make logics sound between each other that *really* shouldn't be, via a fixed functor that just sends a bunch of arrows to an identity arrow.

One way to fix this problem is to require that F is surjective, which definitely fixes up a lot. However, this seems like *too strong* of a criteria. What about two logics $\mathscr{L}_1$ and $\mathscr{L}_2$ such that the language of $\mathscr{L}_1$ is a sublanguage of $\mathscr{L}_2$. It is certainly the case, for example, (and something that will be derived in the next chapter) that all of classical propositional logic can be derived using the connectives $\rightarrow, \perp$. One states this because every statement in propositional logic is *logically equivalent* to some formula that is formed only using $\rightarrow, \perp$.

It seems like the remedy is for the criteria to be existence of a surjective functor F between an equivalence class category $C(\mathscr{L}_1)/ \ll\gg$ and $C(\mathscr{L}_2)/ \lhd \rhd$, where $\gg$ and $\rhd$ denotes derivability in $\mathscr{L}_1$ and $\mathscr{L}_2$ respectively, and for this paper that in the end will be the convention that shall be adapted. However, the main hurdle to this definition is that a "quotient category" $C(\mathscr{L}_1)/ \ll\gg$ in the desired sense has not been defined yet. What is desired is a functor Q that maps objects to their equivalence classes in the sense that the only thing that changes about the arrow structure is that the domain and codomain become the analogous equivalence class. In other words, each arrow is injectively mapped by Q even though the objects are surjectively mapped by Q to the quotient classes.

The question is if such an arrow exists. Turns out it does, but a few specific tools need to be defined before proving such a notion.

Definition 3.5.1. Let G be a graph. Let $\sim$ be an equivalence relation on V. The **quotient graph** $G/\sim$ of G with respect to $\sim$ is a graph such that $G/\sim(V)$ is the quotient set $G(V)/\sim$ and two equivalence classes $[u],[v]$ and an index i of the $G(E)$ index set forms an edge $([u],[v],i)$ in $G/\sim(E)$ if and only if (u,v,i) forms an edge in $G(E)$.

The forgetful functor $\mathbf{U}\colon \mathbf{Cat} \longrightarrow \mathbf{Grph}$ is the functor defined by $\mathbf{U}(\mathcal{C}) \overset{\mathbf{U}(H)}{\mapsto} \mathbf{U}(\mathcal{D})$ being a graph homomorphism between two graphs defined in the following way.

1. For any category $\mathcal{C}$, $\mathbf{U}_{Ob.}(\mathcal{C})$ is the graph with the vertices as $Ob(\mathcal{C})$ and edges (a,b,f) indexed by the arrows $a \overset{f}{\mapsto} b$ in $\mathcal{C}$.

2. For any functor $F\colon \mathcal{C} \longrightarrow \mathcal{D}$, $\mathbf{U}(F)\colon \mathbf{U}(\mathcal{C}) \longrightarrow \mathbf{U}(\mathcal{D})$ is the graph homomorism (easy to verify it as such) Such that $\mathbf{U}(F)_V(a) = F_{Ob.}(a)$ and $\mathbf{U}(F)_E((a,b,f)) = (F(a),F(b),F(f))$.

It is easy to verify that $\mathbf{U}$ is a functor. A treatment of the forgetful functor can be found in Mac Lane. [7] △

Now the existance of a desired functor Q can be recognized.

Lemma 3.5.2. *Given any logic $\mathscr{L}$, there exists a $Q\colon C(\mathscr{L}) \longrightarrow D$, where D is a category, such that.*

1. $Q(\Gamma) = [\Gamma]$, where $[\Gamma]$ is the Γ equivalence class of the relation $\ll\gg$.

2. Q is a cartesian functor.

3. Q maps arrows injectively.

Proof. Let $\mathscr{L}$ be a logic. Let $G = \mathbf{U}(C(\mathscr{L}))$, where $\mathbf{U}\colon \mathbf{Cat} \longrightarrow \mathbf{Grph}$ is the forgetful functor. Let $G/\ll\gg$ be the quotient graph generated by the relation $\ll\gg$. Let $q\colon G \longrightarrow G/\ll\gg$ be the quotient homomorphism from G to $G/\ll\gg$. By design we find q_E is injective. Let $D = \mathbf{C}(G)$, where $\mathbf{CC}\colon \mathbf{Grph} \longrightarrow \mathbf{Cat}$ is the free cartesian category functor. Define $h_E\colon G \longrightarrow \mathbf{C}(G)$ by $h_E(e) = f_e$, where f_e is the order 1 path corresponding to the edge

e. It is easy to verify that $h_E q_E = (hq)_E$ is injective, hence the functor $Q\colon C(\mathscr{L}) \longrightarrow D$ generated by hq exists that follows conditions 1.-3. as advertised. □

Now soundness can be defined as desirable.

Definition 3.5.3. Given any logic $\mathscr{L}$, the **quotient deductive category** is the image $Q(C(\mathscr{L}))$ of the functor Q as it is described in **Lemma 3.5.1**. This category shall be written $C(\mathscr{L})/\ll\gg$. Moreover, Q shall be called **quotient functor** of the category $C(\mathscr{L})$.
Let $\mathscr{L}_1$ and $\mathscr{L}_2$ be two logics. $\mathscr{L}_1$ is said to be **sound** on $\mathscr{L}_2$ if there exists a cartesian functor $F^*\colon C(\mathscr{L}_1)/\ll\gg\longrightarrow C(\mathscr{L}_2)/\ll\gg$ such that $F^*_{Ob.}$ is surjective. $\mathscr{L}_1$ is said to be **complete** on $\mathscr{L}_2$ if $\mathscr{L}_2$ is sound on $\mathscr{L}_1$.
Two logics $\mathscr{L}_1$ and $\mathscr{L}_2$ are said to be **deduction class equivilent** if there exists a cartesian functor $F^*\colon C(\mathscr{L}_1)/\ll\gg\longrightarrow C(\mathscr{L}_2)/\lhd\rhd$ such that $F^*_{Ob.}$ is bijective. △

The first thing to notice about this definition is that there is this notion of **deduction class equivalence**, which is really classifying two logics as "essentially the same". This notion of "essentially the same" is inspired by why one views propositional classical logic defined only in a language with the $\rightarrow, \perp$ connectives as the same logic as one defined with more connectives: And that is because there is a one-to-one correspondence between derivability inference maps. Also, one might notice that completeness has been defined here as well, which is really just soundness in the opposite direction. Once one sees some of the applications of this definition, this link will be very evident.

To get into some of the applications of this definition of soundness, it will help to define some special cases where the criteria is met. These next few lemmas will derive some of these conditions. The first is simply finding the functor F between two logic categories $C(\mathscr{L}_1)$ and $C(\mathscr{L}_2)$ that is surjectively maps the object class. It suffices to assume that a graph homomorphism exists since this paper has shown that constructing the two is equivalent by **Corollary 2.5.11**.

Lemma 3.5.4. *Let $\mathscr{L}_1$ and $\mathscr{L}_2$ be two logics on the same language. Let $\gg$ and $\rhd$ be the derivability relations for $\mathscr{L}_1$ and $\mathscr{L}_2$ respectively. If there exists a graph homomorphism $h\colon G(\mathscr{L}_1) \longrightarrow C(\mathscr{L}_2)$ such that h_V is surjective, then $\mathscr{L}_1$ is sound on $\mathscr{L}_2$.*

Proof. Define $H^*\colon C(\mathscr{L}_1)/\ll\gg \longrightarrow C(\mathscr{L}_2)/\lhd\rhd$ such that $H^*([f]) = [H(f)]$. It is easy to verify that H^* is a well-defined functor. Given $[\Delta]$ in $C(\mathscr{L}_2)/\lhd\rhd$, we find there is some Γ in $C(\mathscr{L}_1)$ such that $H(\Gamma) = \Delta$, hence $[\Delta] = [H(\Gamma)] = H^*([\Gamma])$. $\square$

Now an even more special case will be derived that will bring lots of utility these next few chapters.

Proposition 3.5.5. *Let $\mathscr{L}_1$ and $\mathscr{L}_2$ be two logics on the same language $\mathfrak{L}$. Let $\gg$ and $\rhd$ be the derivability relations for $\mathscr{L}_1$ and $\mathscr{L}_2$ respectively. If there exists a graph homomorphism $h\colon G(\mathscr{L}_1) \longrightarrow C(\mathscr{L}_2)$ such that $h_V(\Gamma) = \Gamma$, then $\mathscr{L}_1$ is sound on $\mathscr{L}_2$. Moreover, the following are eqivalent.*

1. There exists a graph homomorphism $h\colon G(\mathscr{L}_1) \longrightarrow C(\mathscr{L}_2)$ such that $h_V(\Gamma) = \Gamma$

2. $\Gamma \gg \Delta$ implies $\Gamma \rhd \Delta$.

3. The inference maps of $\mathscr{L}_1$ are all derivable by $\mathscr{L}_2$.

Proof. Soundness follows directly from **Lemma 3.5.4**. For the next part, we shall do 1. $\implies$ 2. $\implies$ 3. $\implies$ 1.

1. $\implies$ 2. There exists a graph homomorphism $h\colon G(\mathscr{L}_1) \longrightarrow C(\mathscr{L}_2)$ such that $h_V(\Gamma) = \Gamma$. Then there exists a functor $H\colon C(\mathscr{L}_1) \longrightarrow C(\mathscr{L}_2)$ such that $H(\Gamma) = \Gamma$. Our conclusion follows.

2. $\implies$ 3. Suppose $\Gamma \gg \Delta$ implies $\Gamma \rhd \Delta$. Given an inference map I in $\mathscr{L}_1$, we find that $\Gamma \rhd I(\Gamma)$, since $\Gamma \gg I(\Gamma)$.

3. $\implies$ 1. Suppose the inference maps of $\mathscr{L}_1$ are derivable by $\mathscr{L}_2$. For every inference edge (Γ, Δ, i) in $G(\mathscr{L}_1)$, we find $\Gamma \rhd \Delta$. Then for every inference edge $e = (\Gamma, \Delta, i)$, choose

some $\Gamma \overset{f_e}{\mapsto} \Delta$ in $C(\mathscr{L}_2)$. Define $h_E \colon E(G(\mathscr{L}_1)) \longrightarrow Ar(C(\mathscr{L}_2))$ by $h_E(e) = f_e$. This forms the graph homomorphism $h \colon G(\mathscr{L}_1) \longrightarrow C(\mathscr{L}_2)$ as advertised. □

3.6 Valuation Logic

Now, we bring our attention to the next concept of this section, and that is what I shall call "valuation logic". Essentially, it is a generalized notion of "truth tables" , or more rigorously, "model theory" of first-order/propositional languages.

Definition 3.6.1. Let $\mathcal{L}$ be a language. A **Valuation Class** on $\mathcal{L}$ (which we shall also call a $\mathcal{L}$-**Valuation Class**), is a class Val of "valuation mappings" $v \colon Form \longrightarrow \{0, 1\}$. Moreover, for any $\mathcal{L}$-valuation class Val, the **Valuation Inference Map** V generated by Val is the inference map such that

$$\frac{\Gamma}{\Delta}\, V,$$

if and only if $v(\pi_i(\Gamma)) = 1$, for all $v \in Val$ and $1 \leqslant i \leqslant |\Gamma|$, implies $v(\pi_j(\Delta)) = 1$, for all $1 \leqslant j \leqslant |\Delta|$.
Note that $\mathscr{L}_V = \langle \mathcal{L}, \{V\} \rangle$ forms a logic, which we shall call a **Valuation Logic**, or a V-**Logic** when being specific to the particular semantic inference map V. △

As seen from the definition, valuation is defined as a logic, but just because it's layed it out "proof theoretically" does not mean there is a lot of proving that is done on this system. Valuation logics still retain their model theoretic structure and hence their applications. For the "deduction" to determine whether $\Gamma \vDash_V \Delta$ is often to declare "Δ is always true in $\mathscr{L}_V$." In fact, the following proposition says one can find $\Gamma \vDash_V \Delta$ if and only if there exists an inference edge. This result corresponds valuation mappings as they are traditionally done, since $\vDash_V$ is a metalinguistic relation that holds if and only if the right side is true for all possible valuations that the left side is true. This principle holds in this more general definition.

Proposition 3.6.2. *$\Gamma \vDash_V \Delta$ if and only if there exists an inference edge (Γ, Δ, i) in $G(\mathscr{L}_V)$ generated by V.*

Proof. (Γ, Δ, i) in $G(\mathscr{L}_V)$ implying $\Gamma \vDash_V \Delta$ is trivial. Conversely suppose $\Gamma \vDash_V \Delta$. Then there exists some arrow $\Gamma \overset{f}{\mapsto} \Delta$. We shall prove this by structural induction by arrows in $C(\mathscr{L}_V)$ that the domain and codomain corresponds to an edge. In the base case that f is an order 1 path, if f is an inference edge, we are done. If f is a projection edge, then $\Delta \preceq \Gamma$, so $v(\pi_k(\Gamma)) = 1$, for each $1 \leqslant k \leqslant |\Gamma|$, implies $v(\pi_k(\Delta)) = 1$, for each $1 \leqslant k \leqslant |\Delta|$. In the first inductive step where $f = f's$ for an order n path $\Gamma \overset{f'}{\mapsto} \Gamma'$ and an order 1 path $\Gamma' \overset{s}{\mapsto} \Delta$. We find $v(\pi_k(\Gamma)) = 1$, for all valuations v and $1 \leqslant k \leqslant |\Gamma|$ implies $v(\pi_k(\Gamma')) = 1$, for all valuations v and $1 \leqslant k \leqslant |\Gamma'|$, which further implies $v(\pi_k(\Delta_j)) = 1$, for all $1 \leqslant k \leqslant |\Delta_j|$ and for all $1 \leqslant j \leqslant n$.
In the inductive step, suppose $f = f'(\Pi_{j=1}^{n} f_k)$ such that the inductive hypothesis applies to $\Gamma \overset{f'}{\mapsto} \Gamma'$ and $\Gamma \overset{f_j}{\mapsto} \Delta_j$ for each $1 \leqslant j \leqslant n$. By inductive hypothesis, we find $v(\pi_k(\Gamma)) = 1$, for all valuations v and $1 \leqslant k \leqslant |\Gamma|$ implies $v(\pi_k(\Gamma')) = 1$, for all valuations v and $1 \leqslant k \leqslant |\Gamma'|$, which further implies $v(\pi_k(\Delta_j)) = 1$, for all $1 \leqslant k \leqslant |\Delta_j|$ and for all $1 \leqslant j \leqslant n$. Since $\Delta = \langle \Delta_1, \ldots, \Delta_n \rangle$, we conclude $v(\pi_k(\Gamma)) = 1$, for all valuations v and $1 \leqslant k \leqslant |\Gamma|$ implies $v(\pi_k(\Delta)) = 1$, for all valuations v and $1 \leqslant k \leqslant |\Delta|$. □

Now, it's a good idea to look at some examples!

Example 3.6.3. Let us look at the complete semantic for propositional classic logic. This can be straightforwardly defined as the class of valuation maps V such that each $v \in V$ has the following two properties: 1. $v(\bot) = 0$. 2. $v(\phi \to \varphi) = max\{1 - v(\phi), v(\varphi)\}$. ◊

Example 3.6.4. Classic Predicate First Order Logic is a little more complicated, since generally we look at semantics in terms of so-called "interpretations" of "language-structures" (often notated $\mathfrak{A}$ for a given language structure and $L(\mathfrak{A})$ for the interpretation) in place of valuation mappings. But that doesn't mean we can be clever with our valuation mappings and not capture the semantics of predicate logic. Each interpretation of a given language structure $L(\mathfrak{A})$ valuation map can be regarded as a valuation map $v_{L(\mathfrak{A})}\colon Prop \longrightarrow \{0,1\}$ that works given a condition associated with the variable assignment of $L(\mathfrak{A})$. So if ϕ is atomic, then $v_{L(\mathfrak{A})}(\phi) = 1$ if and only if $\phi_{L(\mathfrak{A})} \in R^n_{\mathfrak{A}}$ for some n-ary relation (including "="). With connectives, we have the propositional rules stated in **Example 2.5.6** plus the quantifier rule: $v(\forall x\phi) = min\{v(\phi[x|a]) \mid a \in |\mathfrak{A}|\}$, where $|\mathfrak{A}|$ is the set of constant symbols corresponding to $\mathfrak{A}$. $\Diamond$

4
Classical Propositional Logic

This chapter shall explore how this paper's rigorous treatment of logic in the previous Chapter works on one of the more basic and popular form of logics: Classic propositional logic. Note that with classical propositional logic, there are a great many ways to define the language and axiomize the system. As any mathematician knows, there are the connectives $\wedge, \vee, \rightarrow, \leftrightarrow, \neg, \bot, \top$. As any logic-savy mathematician definitely knows, we can form classical propositional logic from a language consisting of two connectives/constants and a few inference maps. All the other connectives we can derive *in terms* of those two. For example, a language with propositional variables plus the connectives $\vee$ and $\neg$, we are able to talk about $\wedge$ as $\phi \wedge \varphi := \neg(\neg\phi \vee \neg\varphi)$, $\rightarrow$ as $\phi \rightarrow \varphi := \neg\phi \vee \varphi$, $\bot$ as $\bot := p_1 \wedge \neg p_1$, for some atomic proposition p_1, and so on.

In this paper, the minimal approach shall be used by defining the language in terms of the least amount of connectives, since it leads to much more efficient proofs with less redundant cases. But instead of $\vee, \neg$, it will be most convenient to use $\rightarrow, \bot$ for the simple reason that it is most eloquent in defining this system, especially bearing in mind that intuitionistic propositional logic is around the corner in the next chapter. The way propositional logic is derived, here, ought to be as analogous to intuitionistic as possible, not only to save tedious derivations but to make clear what exactly is changing between the two systems.

The sections of the chapter will break down as follows. In the first section, the *deductive* system shall be defined and derive some of its properties will be derived, both the usual ones but also some new ones that are unique this paper's categorical approach. The second section shall show that the minimal approach to the logic results in a "deduction class equivalent" logic to a similar sort of logic that defines all its connectives initially. The third section shall define the valuation logic (as it is usually done) that this paper claims classical propositional logic is sound and complete on. Finally, the fourth section shall quickly verify the result classical propositional logic is sound and complete on this valuation logic.

4.1 The Classical Language and Deductive System

Definition 4.1.1. The **Classical Propositional Language** denoted $\mathcal{L}_C$ is defined as follows

1. The class *Atom.* contains an element p_i, called a **Propositional Variable**, for each $i \geqslant 1$.

2. The connectives in $\mathscr{L}_C$ include the constant $\perp$ and the binary connective $\rightarrow$.

Furthermore, we shall further define the connectives $\neg, \vee, \wedge, \leftrightarrow, \top$ as shorthand notation for the following:

1. $\neg\phi := \phi \rightarrow\perp$

2. $\phi \vee \varphi := \neg\phi \rightarrow \varphi$,

3. $\phi \wedge \varphi := \neg(\neg\phi \vee \neg\varphi)$,

4. $\phi \leftrightarrow \varphi := (\phi \rightarrow \varphi) \wedge (\varphi \rightarrow \phi)$,

5. $\top := \neg \perp$. △

before defining exact logic on $\mathcal{L}_C$, this next definition will be provided as a reference to what some prominent inference maps of note in $\mathcal{L}_C$.

Definition 4.1.2. These inference maps on $\mathcal{L}_C$ will be named as follows:

1.

$$\frac{\begin{array}{c}[\phi]\\ \vdots\\ \varphi\end{array}}{\phi \to \varphi} \to I, \quad \frac{\phi \quad \phi \to \varphi}{\varphi} \to E,$$

2.

$$\frac{\begin{array}{c}[\phi]\\ \vdots\\ \bot\end{array}}{\neg\phi} \neg I, \quad \frac{\phi \quad \neg\phi}{\varphi} \neg E,$$

3.

$$\frac{\phi}{\phi \vee \varphi} \vee I1, \quad \frac{\varphi}{\phi \vee \varphi} \vee I2, \quad \frac{\phi \vee \varphi \quad \begin{array}{c}[\phi]\\ \vdots\\ \theta\end{array} \quad \begin{array}{c}[\varphi]\\ \vdots\\ \theta\end{array}}{\theta} \vee E,$$

4.

$$\frac{\phi \quad \varphi}{\phi \wedge \varphi} \wedge I, \quad \frac{\phi \wedge \varphi}{\phi} \wedge E1, \quad \frac{\phi \wedge \varphi}{\varphi} \wedge E2,$$

5.

$$\frac{\bot}{\phi} \bot, \quad \frac{}{\top} \top,$$

6.

$$\frac{\phi \to \varphi \quad \varphi \to \phi}{\phi \leftrightarrow \varphi} \leftrightarrow I, \quad \frac{\phi \leftrightarrow \varphi}{\phi \to \varphi} \leftrightarrow E1, \quad \frac{\phi \leftrightarrow \varphi}{\varphi \to \phi} \leftrightarrow E2,$$

7.

$$\frac{\neg\neg\phi}{\phi} RAA, \quad \frac{}{\phi \vee \neg\phi} LEM.$$

$\triangle$

Now we are not quite up to the point where we define classical propositional logic yet; the definition shall first be motivated–and the careful way this language was defined–to show that contrary to popular belief, one is not obligated to have the law of excluded middle LEM or for that matter reducto ad absurdum RAA as an inference maps for the logic, for the language is set up such that $\rightarrow I, \rightarrow E$ alone can derive LEM, and $\rightarrow I, \rightarrow E, \vee E, \bot$ can derive RAA. In fact, the proposition below tells even more: The logic $\langle \mathcal{L}_C, \{\rightarrow I, \rightarrow E, \vee E, \bot\}\rangle$ is sound and complete on $\langle \mathcal{L}_C, \{\rightarrow I, \rightarrow E, RAA\}\rangle$.

Proposition 4.1.3. *1. For the logic $\mathscr{L} = \langle \mathcal{L}_C, \{\rightarrow I, \rightarrow E\}\rangle$, we find that LEM is derivable.*
2. The logic $\mathscr{L}_C = \langle \mathcal{L}_C, \{\rightarrow I, \rightarrow E, \vee E, \bot\}\rangle$ is sound and complete with the logic $\mathscr{L}' = \langle \mathcal{L}_C, \{\rightarrow I, \rightarrow E, RAA\}\rangle$.

Proof. 1. Note that $\phi \vee \neg\phi := \neg\phi \rightarrow \neg\phi$. We find

$$\frac{[\neg\phi]^1}{\phi \vee \neg\phi}\,\vee I^1.$$

2. It shall suffice to prove that RAA is derivable by $\mathscr{L}_C$ and $\bot$ and $\vee E$ is derivable by $\mathscr{L}'$. The derivations go as follows:
RAA in $\mathscr{L}_C$.

$$\dfrac{\dfrac{}{\phi \vee \neg\phi}\,LEM \qquad [\phi]^1 \qquad \dfrac{\dfrac{[\neg\phi]^1 \quad \neg\neg\phi}{\bot}\,\vee E}{\phi}\,\bot}{\phi}\,\vee E^1.$$

$\vee E$ in $\mathscr{L}'$.

$$\cfrac{\cfrac{\cfrac{\begin{matrix}\cfrac{\cfrac{\cfrac{\begin{matrix}[\phi]^1\\ \vdots\\ \theta\end{matrix} \qquad [\neg\theta]^2}{\perp}\to E}{\neg\phi}\to I^1 \qquad \phi\vee\varphi}{\varphi}\to E \\ \vdots \\ \theta\end{matrix} \qquad\qquad [\neg\theta]^2}{\perp}\to E}{\neg\neg\theta}\to I^2}{\theta}\,RAA.$$

$\perp$ in $\mathscr{L}'$.

$$\cfrac{\cfrac{\perp}{\neg\neg\phi}\to I^1}{\phi}\,RAA.$$

□

Since the proof of this proposition shows that all inference maps of $\mathscr{L}$ are derivable in $\mathscr{L}_C$ and vice versa, **Proposition 3.5.5** deduces the following corollary.

Corollary 4.1.4. *Let $\mathscr{L}$ and $\mathscr{L}_C$ (as they are defined in **Proposition 4.1.3**) are deduction class equivalent.*

It has been seen before that the logic $\langle\mathcal{L}_C, \{\to I, \to E, RAA\}\rangle$ is sufficient for classical propositional logic; however, $\langle\mathcal{L}_C, \{\to I, \to E, \vee E, \perp\}\rangle$ is just as sufficient a choice...apart from the fact that the latter has more inference maps. But this one time, efficiency and conciseness shall be sacrificed for a method of proof that will be more analogous to intuitionistic propositional logic down the road.

Definition 4.1.5. The logic $\mathscr{L}_C = \langle\mathcal{L}_C, \{\to I, \to E, \vee E, \perp\}\rangle$ shall be called **Classical Propositional Logic**. △

Now for the big proposition showing that the other inference maps of note can be derived.

Proposition 4.1.6. *The inference rules $\neg I$, $\neg E$, $\vee I1$, $\vee I2$, $\wedge I$, $\wedge E1$, $\wedge E2$, $\top$, $\leftrightarrow I$, $\leftrightarrow E1$, $\leftrightarrow E2$ are all derivable in $\mathscr{L}_C$.*

Proof. Note throughout the proof that $\neg\phi := \phi \to\bot$.

$\neg I$. Since $\neg\phi := \phi \to\bot$, this is just a special case of $\to I$

$\neg E$.

$$\dfrac{\dfrac{\phi \quad \neg\phi}{\bot}\to E,}{\varphi}\bot\,.$$

$\vee I1$. Note that $\phi \vee \varphi := (\phi \to\bot) \to \varphi$. We find

$$\dfrac{\dfrac{\dfrac{\phi \quad [\neg\phi]^1}{\bot}\to E}{\varphi}\bot}{\phi \vee \varphi}\to I^1.$$

$\vee I2$.

$$\dfrac{\varphi}{\phi \vee \varphi}\to I^1.$$

$\wedge I$. Remember that $\phi \wedge \varphi := \neg(\neg\phi \vee \neg\varphi)$. By $\vee E$ proved in part 2., we find

$$\dfrac{\dfrac{[\neg\phi \vee \neg\varphi]^2 \quad \dfrac{\phi \quad [\neg\phi]^1}{\bot}\to E \quad \dfrac{\varphi \quad [\neg\varphi]^1}{\bot}\to E}{\bot}\vee E^1}{\phi \wedge \varphi}\to I^2.$$

$\wedge E1$. By $\vee I1$ proved in part 1., we find

$$\dfrac{\dfrac{\dfrac{\dfrac{[\neg\phi]^1}{\neg\phi \vee \neg\varphi}\vee I1 \quad \phi \wedge \varphi}{\bot}\to E}{\neg\neg\phi}\to I^1}{\phi}RAA.$$

$\wedge E2$. is similar except using $\vee I2$, also proved in part 1., in place of $\vee I1$.

$\top$. Remember that $\top := \neg\perp$. We find

$$\frac{[\perp]^1}{\top} \to I^1.$$

$\leftrightarrow I$. Since $\phi \leftrightarrow \varphi := (\phi \to \varphi) \wedge (\varphi \to \phi)$, this is just a special case of $\wedge I$.

$\leftrightarrow E1$. Just a special case of $\wedge E1$.

$\leftrightarrow E2$. Just a special case of $\wedge E2$. □

There is an important question looking at the logical structure of $\mathscr{L}_C$. There is a symbolic way $\to$ of expressing logical implication $\vdash$ in the language. That is to say that if we find $\vdash \phi \to \varphi$, that is the same as finding $\phi \vdash \varphi$. That is nothing new. But with the metalinguistic symbol $\dashv\vdash$ of logical equivalence between formula vectors, it should be that $\dashv\vdash$ behaves nicely with $\leftrightarrow$, which is shown in this corollary.

Corollary 4.1.7. *Let ϕ, φ be $\mathcal{L}_C$-formulas.*

1. $\phi \vdash \varphi$ if and only if $\vdash \phi \to \varphi$,

2. $\phi \dashv\vdash \varphi$ if and only if $\vdash \phi \leftrightarrow \varphi$.

Proof. For part 1, we have $\phi \vdash \varphi \implies \vdash \phi \to \varphi$ from $\to I$ and the converse from $\to E$. For part 2, remember that $\phi \leftrightarrow \varphi := (\phi \to \varphi) \wedge (\varphi \to \phi)$. If $\phi \dashv\vdash \varphi$, then by part 1., we have $\vdash \phi \to \varphi$, and $\vdash \varphi \to \phi$, so

$$\frac{\phi \to \varphi \quad \varphi \to \phi}{\phi \leftrightarrow \varphi} \wedge I.$$

Conversely, if $\vdash \phi \leftrightarrow \varphi$, we find by $\wedge E1$ and $\wedge E2$ that $\vdash \phi \to \varphi$ and $\vdash \varphi \to \phi$. Part 1. completes our proof. □

4.2 Soundness and Completeness with Ordinary Classical Propositional Logic

Definition 4.2.1. The **Ordinary Propositional Language** denoted $\mathcal{L}_O$ is defined as follows

1. The class *Atom.* contains an element p_i, called a **Propositional Variable**, for each $i \geqslant 1$.

2. The connectives in $\mathcal{L}_O$ include the constants $\bot, \top$, the unary connective $\neg$, and the binary connectives $\to, \wedge, \vee$.

Ordinary classical propositional logic $\mathscr{L}_{CO}$ shall be defined as the logic under the language $\mathcal{L}_O$ with all the inference maps **Definition 4.1.2**. $\triangle$

Lemma 4.2.2.

1. $\neg\phi \dashv\vdash \phi \to\bot$*,*

2. $\phi \vee \varphi \dashv\vdash \neg\phi \to \varphi$*,*

3. $\phi \wedge \varphi \dashv\vdash \neg(\neg\phi \vee \neg\varphi)$*,*

4. $\top \dashv\vdash \neg\bot$*,*

5. $\phi \leftrightarrow \varphi \dashv\vdash (\phi \to \varphi) \wedge (\varphi \to \phi)$

Proof. Note that **Corollary 4.1.7** holds in $\mathscr{L}_{CO}$ since it contains the same rules, so to prove anything of the form $\varphi \dashv\vdash \phi$, it shall suffice to prove $\vdash \varphi \leftrightarrow \phi$.

1.

$$\dfrac{\dfrac{\dfrac{\dfrac{[\phi]^1 \quad [\neg\phi]^2}{\bot}\,\neg E,}{\phi\to\bot}\to I^1}{\neg\phi\to(\phi\to\bot)}\to I^2 \qquad \dfrac{\dfrac{\dfrac{[\phi]^3 \quad [\phi\to\bot]^4}{\bot}\to E}{\neg\phi}\,\neg I^3}{(\phi\to\bot)\to\neg\phi}\to I^4}{\neg\phi\leftrightarrow(\phi\to\bot)}\leftrightarrow I.$$

2.

$$\dfrac{\phi\vee\varphi \qquad \dfrac{[\phi]^3 \quad [\neg\phi]^1}{\neg\phi\to\varphi}\to I^1 \qquad \dfrac{[\varphi]^3}{\neg\phi\to\varphi}\to I^2}{\neg\phi\to\varphi}\vee E^3,$$

so we find that $\phi \vee \varphi \vdash \neg\phi \to \varphi$.

$$\dfrac{\dfrac{}{\phi\vee\neg\phi}\,LEM \qquad \dfrac{[\phi]^1}{\phi\vee\varphi}\vee I1 \qquad \dfrac{\dfrac{[\neg\phi]^1 \quad \neg\phi\to\varphi}{\varphi}\to E}{\phi\vee\varphi}\vee I2}{\phi\vee\varphi}\vee E^1,$$

so we find that $\neg\phi \to \varphi \vdash \phi \vee \varphi$.

3.

$$\dfrac{[\neg\phi \vee \neg\varphi]^2 \quad \dfrac{[\neg\phi]^1 \quad \dfrac{[\phi \wedge \varphi]^3}{\phi}\wedge E1}{\bot}\neg E \qquad \dfrac{[\neg\varphi]^1 \quad \dfrac{\phi \wedge \varphi}{\varphi}\wedge E2}{\bot}\neg E}{\dfrac{\bot}{\neg(\neg\phi \vee \neg\varphi)}\neg I^2}\vee E^3$$

so $\phi \wedge \varphi \vdash \neg(\neg\phi \vee \neg\varphi)$.

$$\dfrac{\dfrac{}{\phi \vee \neg\phi}LEM \qquad \dfrac{\dfrac{\dfrac{\dfrac{[\phi]^2 \quad [\varphi]^1}{\phi \wedge \varphi}\wedge I \quad [\neg(\phi \wedge \varphi)]^3}{\bot}\neg E}{\neg\varphi}\neg I^1}{\neg\phi \vee \neg\varphi}\vee I2 \qquad \dfrac{[\neg\phi]^2}{\neg\phi \vee \neg\varphi}\vee I1}{\dfrac{\bot}{\phi \wedge \varphi}\neg I^3,}\neg E$$

so $\neg(\neg\phi \vee \neg\varphi) \vdash \phi \wedge \varphi$.

4.

$$\dfrac{\dfrac{\dfrac{}{\top}\top}{\neg\bot \to \top}\to I^1 \quad \dfrac{[\bot]^2}{\top \to \neg\bot}\to I^3}{\top \leftrightarrow \neg\bot}\leftrightarrow I,$$

5.

$$\dfrac{\dfrac{\phi \leftrightarrow \varphi}{\phi \to \varphi}\leftrightarrow E1 \quad \dfrac{\phi \leftrightarrow \varphi}{(\phi \to \varphi) \wedge (\varphi \to \phi)}\wedge I,}{}$$

so $\phi \leftrightarrow \varphi \vdash (\phi \to \varphi) \wedge (\varphi \to \phi)$.

$$\dfrac{\dfrac{(\phi \to \varphi) \wedge (\varphi \to \phi)}{\phi \to \varphi}\wedge E1 \quad \dfrac{(\phi \to \varphi) \wedge (\varphi \to \phi)}{\varphi \to \phi}\wedge E2}{\phi \leftrightarrow \varphi}\leftrightarrow I,$$

so $(\phi \to \varphi) \wedge (\varphi \to \phi) \vdash \phi \leftrightarrow \varphi$. $\square$

Theorem 4.2.3. *$\mathscr{L}_C$ is deduction class equivalent with $\mathscr{L}_{CO}$.*

Proof. Let $h_V : \mathcal{L}_O \longrightarrow \mathcal{L}_C$ be a function defined by

$$h_V(\phi) = \begin{cases} \phi, & \text{if } \phi \text{ is atomic, or } \phi := \phi_1 \to \phi_2, \text{ or } \phi := \perp, \\ h_V(\phi') \to \perp, & \text{if } \phi := \neg\phi', \\ \perp \to \perp, & \text{if } \phi := \top, \\ h_V(\neg\phi_1) \to h_V(\phi_2), & \text{if } \phi := \phi_1 \vee \phi_2, \\ h_V(\neg(\neg\phi_1 \vee \neg\phi_2)), & \text{if } \phi := \phi_1 \wedge \phi_2. \end{cases}$$

We can then extend the function $H_V : Ob(C(\mathscr{L}_{CO})) \longrightarrow Ob(C(\mathscr{L}_C))$ to $H_V(\Gamma) = \langle h_V(\pi_1(\Gamma)), \ldots, h_V(\pi_{|\Gamma|}(\Gamma))\rangle$. Notice that **Lemma 4.2.2** implies that formula vectors in $\mathscr{L}_{CO}$ are logically equivalent to some formula vector in $\mathscr{L}_C$. Then since $H_V(\Gamma) \rhd \lhd \rhd \Gamma$ and $h_V(\Delta) \lhd \rhd \Delta$, we find for every edge (Δ, Γ, i) in $\mathscr{L}_{CO}$, we get $H(\Gamma) \rhd H(\Delta)$. But since $H(\Gamma), H(\Delta)$ are also objects in $C(\mathscr{L}_C)$ and all inference maps in $\mathscr{L}_{CO}$ are derivable in $\mathscr{L}_C$, we find $H(\Gamma) \gg H(\Delta)$. Then choose arrows $H(\Gamma) \overset{f_e}{\mapsto} H(\Delta)$ in $C(\mathscr{L}_C)$, for each edge $e = (\Gamma, \Delta, i)$, and define $H_E : G(\mathscr{L}_{CO})(E) \longrightarrow Ar(C(\mathscr{L}_C))$. We find $H : G(\mathscr{L}_{CO}) \longrightarrow C(\mathscr{L}_C)$ form a graph homomorphism. Note that it is surjective on vertices.
Then there exists a cartesian functor $\mathcal{H} : G(\mathscr{L}_{CO}) \longrightarrow G(\mathscr{L}_C))$. Define $\mathcal{H}^* : Ar(G(\mathscr{L}_{CO})/ \ll\gg) \longrightarrow Ar(G(\mathscr{L}_C)/ \lhd \rhd))$ by $\mathcal{H}^*([f]) = [\mathcal{H}(f)]$. It is easy to verify that this is a well-defined functor. Moreover, since $[\Gamma] \neq [\Delta]$ implies $\mathcal{H}^*_{Ob.}([\Gamma]) \neq \mathcal{H}^*_{Ob.}([\Delta])$, we find H^* is injective on vertices, as well as surjective, which completes our proof. □

4.3 Classical Semantics

In this section shall quickly define a classical valuation logic that turns out to be the one desired that is complete with $\mathscr{L}_C$, and derive a few of its essential properties.

Definition 4.3.1. Let ϕ, φ be $\mathcal{L}_C$-formulas. A mapping $v : Prop \longrightarrow \{0, 1\}$ is a classical valuation, and part of the valuation class Val_C if

1. $v(\perp) = 0$,

2. $v(\phi \to \varphi) = 1$ if and ony if $v(\phi) = 0$ or $v(\varphi) = 1$.

The logic $\mathscr{L}_{VC}$ on $\mathscr{L}_C$ generated (using **Definition 4.6.1**) by the class Val_C of classical valuation mappings shall be called the **classical propositional valuation logic**. $\triangle$

First, this lemma will derive a very important tool for constructing valuation maps.

Lemma 4.3.2. *Let* $w\colon$ *Atom.* $\longrightarrow \{0,1\}$ *be a function. Then there exists a unique extension* $v\colon$ *Prop.* $\longrightarrow \{0,1\}$ *that is a classic valuation mapping.*

Proof. For existence, define $v\colon$ *Prop.* $\longrightarrow \{0,1\}$ by

$$v(\phi) = \begin{cases} w(\phi), & \text{if } \phi \text{ is Atomic,} \\ 0, & \text{if } \phi := \perp, \\ max\{1 - v(\phi_1), v(\phi_2)\}, & \text{if } \phi := \phi_1 \to \phi_2. \end{cases}$$

It follows by straightforward structural induction on $\mathcal{L}$ that v is well-defined. It follows by definition that v is a valuation mapping. And it finally follows by the same straightforward structural induction on $\mathcal{L}$ that v is unique. $\square$

As it turns out, these two conditions in **Definition 4.3.1** are enough to derive the sufficient conditions that hold on valuation maps for $\neg, \vee, \wedge, \top$, and $\leftrightarrow$.

Proposition 4.3.3. *Let* ϕ, φ *be* $\mathcal{L}_C$*-formulas. Let* Γ *be a* $\mathcal{L}_C$*-vector. Then*

1. $v(\neg\phi) = 1$ *if and only if* $v(\phi) = 0$,

2. $v(\phi \vee \varphi) = max\{v(\phi), v(\varphi)\}$,

3. $v(\phi \wedge \varphi) = min\{v(\phi), v(\varphi)\}$,

4. $v(\top) = 1$,

5. $v(\phi \leftrightarrow \varphi) = 1$ *if and only if* $v(\phi) = v(\varphi)$.

for all valuations $v\colon Prop \longrightarrow \{0,1\}$.

Proof. 1. Our conclusion follows directly from the fact that $\neg\phi := \phi \to \perp$.

2. Note that $\phi \vee \varphi := \neg\phi \to \varphi$. If $v(\phi) = 1$, then $v(\neg\phi) = 0$, hence $v(\phi \vee \varphi) = 1$. If $v(\varphi) = 1$, then $v(\phi \vee \varphi) = 1$. If $v(\phi) = v(\varphi) = 0$, then we find $v(\neg\phi) = 1$ and $v(\varphi) = 0$, hence $v(\phi \vee \varphi) = 0$.

3. Note that $\phi \wedge \varphi := \neg(\neg\phi \vee \neg\varphi)$. If $v(\phi) = 0$ or $v(\varphi) = 0$, then $v(\neg\phi) = 1$ or $v(\neg\varphi) = 1$. It follows by part 2 of this proposition that if $v(\phi) = 0$ or $v(\varphi) = 0$, then $v(\neg\phi \vee \neg\varphi) = 1$, hence $v(\phi \wedge \varphi) = 0$.

4. Follows directly from part 1 of this proposition.

5. Note that $\phi \leftrightarrow \varphi := (\phi \to \varphi) \wedge (\varphi \to \phi)$. We find $v(\phi \leftrightarrow \varphi) = 1 \iff v(\phi \to \varphi) = v(\varphi \to \phi) = 1$. Suppose $v(\phi) = v(\varphi)$. In both the case that $v(\phi) = 0$ and $v(\phi) = 1$, we get $v(\phi \to \varphi) = v(\varphi \to \phi) = 1$.

Conversely, suppose $v(\phi \leftrightarrow \varphi) = 1$. We have two cases.

Q1. Suppose $v(\phi) = 0$. Then $v(\varphi \to \phi) = v(\phi \to \varphi) = 1$. It must be that $v(\varphi) = 0$.

Q2. Suppose $v(\phi) = 1$. By similar argument, we get $v(\varphi) = 1$. □

With the next proposition, it shall be shown that finding $\vDash \phi \to \varphi$ is equivalent to finding that $\phi \vDash \varphi$, and then the analogous principle for $\vDash \phi \leftrightarrow \varphi$ and $\phi \mathrel{=\!\!|\!\!\vDash} \varphi$, similar to what was found in the last section with $\vdash$ and $\mathscr{L}_C$.

Proposition 4.3.4. *Let ϕ, φ be $\mathcal{L}_C$-formulas, let Γ be $\mathcal{L}_C$-formulas.*

1. $\langle \phi, \varphi \rangle \mathrel{=\!\!|\!\!\vDash} \phi \wedge \varphi$,

2. $\phi \vDash \varphi$ if and only if $\vDash \phi \to \varphi$,

3. $\phi \mathrel{=\!\!|\!\!\vDash} \varphi$ if and only if $\vDash \phi \leftrightarrow \varphi$,

4. $\bot \vDash \phi$,

5. If $\langle \Gamma, \phi \rangle \vDash \varphi$, then $\Gamma \vDash \phi \to \varphi$,

6. $\langle \phi, \phi \to \varphi \rangle \vDash \varphi$,

7. If $\langle \Gamma, \phi_1 \rangle \vDash \varphi$ and $\langle \Gamma, \phi_2 \rangle \vDash \varphi$ then $\langle \Gamma, \phi_1 \vee \phi_2 \rangle \vDash \varphi$.

Proof.

1. Note that **Proposition 4.3.3** tells us $v(\phi) = v(\varphi) = 1 \iff v(\phi \wedge \varphi) = 1$.

2. Suppose $\phi \vDash \varphi$. Then $v(\phi) = 0 \implies v(\phi \to \varphi) = 1$ and $v(\phi) = 1 \implies v(\varphi) = 1 \implies v(\phi \to \varphi) = 1$.

Conversely, suppose $\vDash \phi \to \varphi$. By hypothesis, we find $v(\phi \to \varphi) \to v(\phi) = 0$ or $v(\varphi) = 1$. Then given $v(\phi) = 1$, we have $v(\varphi) = 1$.

3. Suppose $\phi \dashv\vDash \varphi$. Then by part 3, we have $\vDash \phi \to \varphi$ and $\vDash \varphi \to \phi$. Then since $\phi \leftrightarrow \varphi := (\phi \to \varphi) \wedge (\varphi \to \phi)$, so $v(\phi \leftrightarrow \varphi) = min\{v(\phi \to \varphi), v(\varphi \to \phi)\} = 1$.

Conversely, suppose $\vDash \phi \leftrightarrow \varphi$. By part 2, we have $\vDash \langle \phi \to \varphi, \varphi \to \phi \rangle$. Our conclusion then follows by part 3.

4. Trivial.

5. Follows directly from part 2.

6. Suppose v is a valuation mapping that satisfies $\langle \phi, \phi \to \varphi \rangle$. It follows from $v(\phi \to \varphi) = 1$ that either $v(\phi) = 0$ or $v(\varphi) = 1$. Since $v(\phi) = 1$, our conclusion is satisfied.

7. Suppose v is a valuation mapping that satisfies $\langle \Gamma, \phi_1 \vee \phi_2 \rangle \vDash \varphi$. Then $vmax\{v(\phi_1), v(\phi_2)\} = v(\phi_1 \vee \phi_2) = 1$. This gives us two similar cases, either of which satisfy $\langle \Gamma, \phi_1 \rangle$ and $\langle \Gamma, \phi_2 \rangle$. Either way, our conclusion follows. □

4.4 Classical Completeness and Consequences

This is the much awaited section where we show that $\mathscr{L}_C$ is sound and complete on $\mathscr{L}_{VC}$, which is alas no trivial matter thanks to the puzzle of completeness. The problem is the valuation inference map V in $\mathscr{L}_{VC}$ does not keep very good track of the inference process of $\mathscr{L}_C$. While it may be that $\mathscr{L}_{VC}$ is regarded as a logic in this paper, one must remember that $\mathscr{L}_{VC}$ is nevertheless not crafted to be deductively powerful when it comes to a sequential inference process. Rather, it "verifies truth" based on valuation maps in a one-step process that would take $\mathscr{L}_C$ an arbitrarily large finite number of steps.

So what should be done about this? The one reasonable thing there is to do: Use $\mathscr{L}_{VC}$ for its strength and not its weakness. $\mathscr{L}_{VC}$ is very good at analyzing "truth" given a valuation, so why not try to come up with a sufficient condition for completeness that involves coming up with a specific valuation. On that note, it seems more natural to prove completeness contrapositively, which is to say, proving $\Gamma \nvdash \Delta \implies \Gamma \nvDash \Delta$. Now given a pretty manageable hypothesis, all that has to be done is prove there exists a valuation mapping such that $v(\phi) = 0$, for some $\phi \preceq \Delta$. That doesn't look so bad!

It especially looks manageable when one sees that the deductive system is surprisingly very good at building bigger vectors from smaller ones that conserve a given property, and eventually from a recursive collection of consistent vectors one can make valuation mapping out of them. Actually, the standard proof for completeness in a standard treatment of first order propositional logic involves creating a "maximal consistent set" that contains Γ. Unfortunately, that can't be done with vectors since vectors are finite, and all the things one could add to Γ that would keep it consistent is infinite. However, the method can still be captured by creating a recursive sequence of consistent vectors $\Delta_0, \Delta_1, \ldots, \Delta_n, \ldots$ with $\Delta_0 = \Gamma$ and for each $n+1$, go through $Form_C$ in an enumeration $\phi_1, \phi_2, \ldots, \phi_n, \ldots$ (since there *are* countably many!), and if it's consistent, set Δ_{n+1} equal to $\langle \Delta_n, \phi_{n+1} \rangle$; otherwise, keep Δ_{n+1} the same as Δ_n. Actually, a similar idea in the standard treatment is used to show a maximal consistent set exists in the first place, but why not just make a valuation mapping out of it?

Lemma 4.4.1. *Suppose* $\Gamma \nvdash \phi$, *for some* $\mathcal{L}_C$*-formula* ϕ. *Enumerate all propositions in* $\mathcal{L}_C$ $\phi_1, \phi_2, \ldots, \phi_n, \ldots$. *For each* $n \geqslant 0$, *define a* $\mathcal{L}_C$ *vector recursively as follows*

$$\Delta_n = \begin{cases} \Gamma, & \text{if } n = 0, \\ \langle \Delta_{n-1}, \phi_n \rangle, & \text{if } \langle \Delta_{n-1}, \phi_n \rangle \nvdash \phi, \\ \langle \Delta_{n-1}, \neg\phi_n \rangle & \text{otherwise.} \end{cases}$$

Define a function $v \colon Prop \longrightarrow \{0, 1\}$ *as follows*

$$v(\phi) = \begin{cases} 1, & \text{if } \phi \preceq \Delta_n, \text{ for some } n \geqslant 0, \\ 0, & \text{otherwise.} \end{cases}$$

Then v *is a classical valuation mapping.*

Proof. We shall first prove the following claims.

Claim 1. If $\Gamma \nvdash \phi$ *and* $\Gamma \vdash \varphi$*, then* $\langle \Gamma, \varphi \rangle \nvdash \phi$.

If $\langle \Gamma, \varphi \rangle \vdash \phi$, it follows that $\Gamma \vdash \langle \Gamma, \varphi \rangle$ and $\langle \Gamma, \varphi \rangle \vdash \phi$, so we would have $\Gamma \vdash \phi$, which cannot happen.

Claim 2. If $\Delta_{n_1} \vdash \varphi$*, for some* $n_1 \geqslant 0$*, then there exists some* $n_2 \geqslant 0$ *such that* $\varphi \preceq \Delta_{n_2}$.

Suppose $\Delta_{n_1} \vdash \varphi$, for some $n_1 \geqslant 0$. Then by *Claim 1*, we find $(\Delta_{n_1}, \varphi) \nvdash \phi$. Note that $\varphi := \phi_{n_2}$, for some $n_2 \geqslant 1$. We have two cases.

W1. Suppose $n_2 \leqslant n_1$. Since $\langle \Delta_{n_2-1}, \varphi \rangle \preceq \langle \Delta_{n_1}, \varphi \rangle$, we find $\langle \Delta_{n_2-1}, \varphi \rangle = \langle \Delta_{n_2-1}, \phi_{n_2} \rangle \nvdash \phi$. It follows that $\varphi \preceq \langle \Delta_{n_2-1}, \phi_{n_2} \rangle = \Delta_{n_2}$.

W2. Suppose $n_2 > n_1$. Then $n_2 - 1 \geqslant n_1$. Then since $\Delta_{n_1} \preceq \Delta_{n_2-1}$, we have $\Delta_{n_2-1} \vdash \varphi$. It follows by *Claim 1* that $\langle \Delta_{n_2-1}, \varphi \rangle \nvdash \phi$, hence $\varphi \preceq \langle \Delta_{n_2-1}, \phi_{n_2} \rangle = \Delta_{n_2}$.

Now we shall prove the lemma. Define a function $v\colon Prop \longrightarrow \{0, 1\}$ as follows.

$$v(\phi) = \begin{cases} 1, & \text{if } \phi \preceq \Delta_n \text{, for some } n \geqslant 0, \\ 0, & \text{otherwise.} \end{cases}$$

We shall prove that this is indeed a valuation mapping in Val_C. Since $\langle \Delta_n, \bot \rangle \vdash \phi$, for all $n \geqslant 0$, we find $v(\bot) = 0$. We shall spend the remainder of this proof showing that $v(\varphi_1 \to \varphi_2) = 1 \iff v(\varphi_1) = 0$ or $v(\varphi_2) = 1$.

Suppose $v(\varphi_1 \to \varphi_2) = 1$. Then $\varphi_1 \to \varphi_2 \preceq \Delta_{n_1}$, for some $n_1 \geqslant 0$. It shall suffice to prove that $v(\varphi_1) \neq 0 \implies v(\varphi_2) = 1$. Suppose $v(\varphi_1) \neq 0$. Then $\varphi_1 \preceq \Delta_{n_2}$, for some $n_2 \geqslant 0$. It shall suffice to prove $\varphi_2 \preceq \Delta_{n_3}$, for some $n_3 \geqslant 0$. Note that since $\langle \varphi_1, \varphi_1 \to \varphi_2 \rangle \preceq \Delta_{max\{n_1,n_2\}}$, we find $\Delta_{max\{n_1,n_2\}} \vdash \varphi_2$ by $\to E$. Then our conclusion follows by *Claim 2*.

For the converse, it shall suffice by *Claim 2* to prove that $\Delta_m \vdash \varphi_1 \to \varphi_2$, for some $m \geqslant 0$. We have two cases.

R1. Suppose $v(\varphi_1) = 0$. Note that $\varphi_1 := \phi_m$, for some $m \geqslant 1$. We find $\phi_m \npreceq \Delta_m$. Then $\langle \Delta_{m-1}, \phi_m \rangle \vdash \phi$, so $\Delta_m = \langle \Delta_{m-1}, \neg\phi_m \rangle$. It follows from $\vee I1$ that $\Delta_m \vdash \varphi_1 \to \varphi_2$.

R2. Suppose $v(\varphi) = 1$. Then $\varphi \preceq \Delta_m$, for some $m \geqslant 1$. We find $\Delta_m \vdash \varphi_1 \rightarrow \varphi_2$ by $\rightarrow I$. □

Theorem 4.4.2. *$\mathscr{L}_C$ is sound and complete on $\mathscr{L}_{VC}$.*

Proof. For soundness, **Proposition 4.3.4** confirms that every inference map is derivable in $\mathscr{L}_{VC}$.

For completeness, suppose $\Gamma \nvdash \Delta$. Then $\Gamma \nvdash \phi$ for some $\phi := \pi_q(\Delta)$, for some $1 \leqslant q \leqslant |\Delta|$. Using **Lemma 4.4.1**, we get a valuation mapping v such that $v(\pi_k(\Gamma)) = 1$ but $v(\phi) = 0$, hence $\Gamma \nvDash \Delta$. □

Corollary 4.4.3. *Let Γ, Δ by $\mathcal{L}_C$-formula vectors. If $\Gamma \nvDash \Delta$, then $\Gamma \nvdash \Delta$.*

Corollary 4.4.4. *$Vars$ is consistent (and hence so is $\mathscr{L}_C$). Every atomic variable, and its negation, is underivable.*

Proposition 4.4.5. *Let ϕ, φ be $\mathcal{L}_C$-formulas, let Γ be $\mathcal{L}_C$-formulas.*

1. $\phi \vee \varphi \nvdash \phi \rightarrow \neg\varphi$ and $\phi \vee \varphi \nvdash \varphi \rightarrow \neg\phi$,

2. $\phi \rightarrow \varphi \nvdash \varphi \rightarrow \phi$,

3. $\phi \rightarrow \varphi \nvdash \neg\phi \rightarrow \neg\varphi$.

5
Sequent Calculus

Now this chapter gets to a rather different way of doing logic, which is sequent calculus. It is a very exotic system and much more difficult/counterintuitive to work with, but the system is overall a better way of doing deductions, since one can look at more than one propositions and one does not have the awkward mechanic of hypothesizing. Additionally, it lays out a straightforward system of doing logic on a collection of propositions as opposed to ordingary "natural deduction", which is really only good for deriving single propositions.

The first section will be about constructing the language on which inferences are made. It turns out to be sufficient in the context of the deductive category to do deductions not *directly* on the propositional language but rather on a metalanguage for reasons that will be discussed in that section. The second section defines intuitionistic sequent calculus in an analogous way to chapter 5, where only three connectives $\to, \bot, \wedge$ are used and the rest of the connectives, and inference maps with them, are derived. The third section shall be devoted to the cut rule, and proving the surprising result that it can be derived from the other inference maps. The fourth section derives some important properties of intuitionistic sequent calculus that gives an idea as to what the system really means in the context of propositional logic. Finally, the fifth section shows an easy semantic view of sequent calculus that is complete, and moreover that intuitionistic sequent calculus is sound and complete with intuitionistic propositional calculus.

5.1 The Metalanguage and Sequents

With the way sequent calculus is presented, it calls for a new order of language since the formal deductive system of logic that has been derived in this paper does inferences directly *on* a language, and the objects which Gentzen style inference operates on are different than typical propositions, and for that matter lists of them. This world of inference rather operates on metalinguistic relations of $\Gamma \vdash \Delta$. So in order for the definitions of deduction to carry over in this instance, it is imperitive that this "metalinguistic" format be defined as a formal language itself. The definition below does exactly that.

Definition 5.1.1. Given a language $\mathcal{L}$, a **Sequent Metalanguage** $\mathcal{M}$ is a language with a single binary relation symbol $\vdash$. To represent a $\mathcal{M}$-formula, we shall write $(\Gamma \vdash \Delta)$ for any two $\mathcal{L}$-vectors Γ and Δ. The terms in the language, i.e. the $\mathcal{L}$-vectors, shall be referred to as the **Sequents**. $\triangle$

Remark 5.1.2. Although it is standard to write "$\Gamma \vdash \Delta$" in most sequent calculus text to essentially say "Δ is derivable by Γ", we shall forgo standard notation in the name of "good syntax". Instead we write "$(\Gamma \vdash \Delta)$" to denote the specific $\mathcal{M}$-formula that our definition represents. To talk about derivability in $\mathcal{M}$, we shall continue with our common practice in Chapter 3 and write $\mathbb{A} \gg \mathbb{B}$, where $\mathbb{A}$ and $\mathbb{B}$ are $\mathcal{M}$-vectors, that is to say $\mathbb{A} = \langle(\Gamma_1 \vdash \Delta_1), \ldots, (\Gamma_n \vdash \Delta_n)\rangle$ and $\mathbb{B} = \langle(\Phi_1 \vdash \Psi_1), \ldots, (\Phi_m \vdash \Psi_m)\rangle$, where $\Gamma_i, \Delta_i, \Phi_i, \Psi_i$ are sequents for every i. $\Diamond$

One thing that will be important later down the road of sequent calculus is a notion of interchangeability of a propositional symbol, because although the logical deduction is really done on the metalanguage, one is really interested in how the propositions interrelate to each other. The notion of propositional equivalance in sequent calculus then is interchangeability where given the sequent $\Gamma_1, \theta, \Gamma_2$, one can substitute θ for θ' and vice versa. The exact definition is provided below.

Definition 5.1.3. In a sequent metalanguage $\mathcal{M}$, if a two formula θ, φ are such that $(\Gamma \vdash \Delta_1, \theta, \Delta_2) \lll\ggg (\Gamma \vdash \Delta_1, \varphi, \Delta_2)$ and $(\Phi_1, \theta, \Phi_2 \vdash \Psi) \lll\ggg (\Phi_1, \varphi, \Phi_2 \vdash \Psi)$, for every sequent $\Gamma, \Delta_1, \Delta_2, \Phi_1, \Phi_2, \Psi$, then we shall call θ is **interchangeable** with φ. Define the binary relation $\leftrightharpoons$ on $Form$ such that $\theta \leftrightharpoons \varphi$ if and only if θ is interchangeable with φ. $\triangle$

5.2 Intuitionistic Sequent Logic Definition

This section will be about sequent calculus of the intuitionistic variety, using the intuitionistic language defined in Chapter 5. Remember that this language uses only the $\rightarrow, \bot, \wedge$ connectives in order to again keep the axioms at a minimum, as shall be defined here

Definition 5.2.1. The **Intuitionistic Propositional Language** denoted $\mathcal{L}_I$ is defined as follows

1. The class $Atom$. contains an element p_i, called a **Propositional Variable**, for each $i \geqslant 1$.

2. The connectives in $\mathcal{L}_I$ include the constant $\bot$ and the binary connectives $\rightarrow, \wedge$.

Furthermore, we shall further define the connectives $\neg, \vee, \wedge, \leftrightarrow, \top$ as shorthand notation for the following:

1. $\neg\phi := \phi \rightarrow \bot$

2. $\phi \vee \varphi := \neg\neg\neg\phi \rightarrow \varphi$,

3. $\phi \leftrightarrow \varphi := (\phi \rightarrow \varphi) \wedge (\varphi \rightarrow \phi)$,

4. $\top := \neg \bot$. $\triangle$

Remark 5.2.2. While everything seems eloquent with having intuitionistic logic defined in terms of three connectives, it is not 100 % certain that doing so is possible without not catching the full scope of intutionistic logic. It is the writer's opinion, and a lot of research that unfortunately did not make it to this paper, that suggests that indeed it does. So it will be an assumption that will be made, alas. For a standard inquiry on intuitionistic logic in general, Chapter 5 of Van Dalen's introductory logic text is a great resource for the basics. [3] To absorb what is going on with Getzen Calculus, an adequate resource for that is Girard's *Proofs and Types.* [5] $\Diamond$

The rest of this section is devoted to deriving the rest of the more practical and versatile inference maps to work with when it comes to doing these proofs in a more efficient manner, as well as the missing inference rules for connectives $\neg, \vee, \top$.

Definition 5.2.3. Let θ, φ be propositional variables and $\Gamma, \Delta, \Phi, \Psi$ be sequents. **Intuitionistic Sequent Calculus** $\mathscr{L}_S$ is a logic with the propositional sequent metalanguage with the following rules of inference.

1. **Identity**

$$\frac{}{(\theta \vdash \theta)}\, I$$

2. **Exchange**

$$\frac{(\Gamma, \theta, \varphi, \Phi \vdash \Delta)}{(\Gamma, \varphi, \theta, \Phi \vdash \Delta)}\, \mathcal{L}X \quad \frac{(\Gamma, \vdash \Psi, \theta, \varphi, \Delta)}{(\Gamma, \vdash \Psi, \varphi, \theta, \Delta)}\, \mathcal{R}X$$

3. **Weakening**

$$\frac{(\Gamma \vdash \Delta)}{(\Gamma, \theta \vdash \Delta)}\, \mathcal{L}W \quad \frac{(\Gamma \vdash \Delta)}{(\Gamma \vdash \theta, \Delta)}\, \mathcal{R}W$$

4. **Contraction**

$$\frac{(\Gamma, \theta, \theta \vdash \Delta)}{(\Gamma, \theta \vdash \Delta)}\, \mathcal{L}C \quad \frac{(\Gamma \vdash \theta, \theta, \Delta)}{(\Gamma \vdash \theta, \Delta)}\, \mathcal{R}C$$

5. **Impication**

$$\frac{(\Gamma \vdash \theta, \Delta) \quad (\Phi, \varphi \vdash \Psi)}{(\Gamma, \Phi, \theta \to \varphi \vdash \Delta, \Psi)}\mathcal{L}{\to} \qquad \frac{(\Gamma, \theta \vdash \varphi, \Delta)}{(\Gamma \vdash \theta \to \varphi, \Delta)}\mathcal{R}{\to}$$

6. **Falsum**

$$\frac{}{(\bot \vdash \quad)}\bot$$

7. **Conjunction**

$$\frac{(\Gamma, \theta \vdash \Delta)}{(\Gamma, \theta \wedge \varphi \vdash \Delta)}\mathcal{L}1\wedge \quad \frac{(\Delta, \varphi \vdash \Delta)}{(\Gamma, \theta \wedge \varphi \vdash \Delta)}\mathcal{L}2\wedge$$
$$\frac{(\Gamma \vdash \theta, \Delta) \quad (\Phi \vdash \varphi, \Psi)}{(\Gamma, \Phi \vdash \theta \wedge \varphi, \Delta, \Psi)}\mathcal{R}\wedge$$

△

Definition 5.2.4. Important Inference maps to be aware of (that shall be show later in this section can be derived from $\mathscr{L}_S$ as they are defined) are as follows

1. **Cut**

$$\frac{(\Gamma \vdash \theta, \Delta) \quad (\Phi, \theta \vdash \Psi)}{(\Gamma, \Phi \vdash \Delta, \Psi)}Cut,$$

2. **Negation**

$$\frac{(\Gamma, \theta \vdash \Delta)}{(\Gamma \vdash \neg\theta, \Delta)}\mathcal{L}\neg \quad \frac{(\Gamma \vdash \theta, \Delta)}{(\Gamma, \neg\theta \vdash \Delta)}\mathcal{R}\neg$$

3. **Top**

$$\frac{}{(\quad \vdash \top)}\top$$

4. **Disjunction**

$$\frac{(\Gamma, \theta \vdash \Delta) \quad (\Delta, \varphi \vdash \Psi)}{(\Gamma, \Phi, \theta \vee \varphi \vdash \Delta, \Psi)}\mathcal{L}\vee$$
$$\frac{(\Gamma \vdash \theta, \Delta)}{(\Gamma \vdash \theta \vee \varphi, \Delta)}\mathcal{R}1\vee \quad \frac{(\Gamma \vdash \varphi, \Delta)}{(\Gamma \vdash \theta \vee \varphi, \Delta)}\mathcal{R}2\vee$$

△

What makes derivations in this language rather tedious first of all is the fact that the exchange rule does not explicitly allow one to rearrange sequents in any order all in one step. However, given that the exchange rule allows simply for one two switch the position of two terms in the sequent, one can derive the fact that seqents can be rearranged in *any* order corresponding to *any* permutation, as this next proposition shows. A basic understanding of permutations is in order from then on, since arbitrary permutations will be placed on sequents a lot to signify that the order does not matter. It should also be noted that the permutations placed on sequents from then on will be done rather informally, as in the exact logistics of the domain and codomain of the permutation will be implied from the fact that a permutation σ (the often-used permutation letter of choice) is being operated specifically on an arbitrary sequent Δ to get a rearranged sequent $\sigma(\Delta)$.

Note that a lot of work will be done with permutations from here on out and somewhat of a background in algebra is assumed. For more information on the basics, however, looking through section 1.3 Dummit and Foote's Algebra text is recommended. [4]

Proposition 5.2.5. *Let Γ, Δ be vectors in $\mathcal{L}_I$-vectors. Let σ be a permutation. Then the inference maps:*

$$\frac{(\Gamma \vdash \Delta)}{(\sigma(\Gamma) \vdash \Delta)}\,\mathcal{L}X+ \qquad \frac{(\Gamma \vdash \Delta)}{(\Gamma \vdash \sigma(\Delta))}\,\mathcal{R}X+$$

are derivable in $\mathscr{L}_S$.

Proof. For a given permutation σ, note that it can be expressed as a finite number of transpositions $\tau_1\tau_2\ldots\tau_n$ for $n \geqslant 1$. If $\mathcal{L}X+$ applies to any transposition, than we can apply $\mathcal{L}X+$ any finite number of times for τ_i, for each $1 \leqslant i \leqslant n$, to get $\mathcal{L}X+$, for any permutation $\sigma = \tau_1\tau_2\ldots\tau_n$. It shall suffice to prove $\mathcal{L}X+$ applies to transpositions, and let $\mathcal{R}+$ follow from similarity.

Let τ be a transposition. We can express any permutation τ' as (j, k), where $j \neq k$ and $1 \leqslant j, k \leqslant |\Gamma|$, and $\tau(\pi_i(\Gamma)) = \pi_j(\Gamma)$ and $\tau(\pi_j(\Gamma)) = \pi_i(\Gamma)$. Let $a, b \geqslant 1$ such that

$\tau = (a,b)$. Without loss of generality, suppose $a < b$. Then we find $\tau = (a,a+1)(a+1,a+2)\dots(a+(b-a-1),b)$. For each $1 \leqslant r \leqslant b-a$, we find

$$(a+r-1,a+r)(\phi_1,\dots,\phi_n) := \phi_1,\dots,\phi_{a+r-2},\phi_{a+r},\phi_{a+r-1},\phi_{a+r+1},\dots,\phi_n,$$

for any order $n \geqslant b-a$ sequent $\phi_1,\dots,\phi_n$. It follows that for each $\Phi := \phi_1,\dots,\phi_n$ and $1 \leqslant r \leqslant b-a$, we get

$$\frac{(\Phi \vdash \Psi)}{((a+r-1,a+r)(\Phi) \vdash \Psi)}\,\mathcal{L}X.$$

Our conclusion then follows by repeated use of $\mathcal{L}X$ on each $(a+r-1,a+r)$ on the result after it, for each $1 \leqslant r \leqslant b-a$. □

Essentially, one has shown that the order of sequents are *invariant*, which is to say that a sequent Δ and $\sigma(\Delta)$ for any permutation σ are logically equivalent. This result shall be reflected in the next corollary which allows one to be very liberal with the order at which connective operations are applied.

Corollary 5.2.6. *The following inference maps are derivable in $\mathscr{L}_S$. Let $\sigma_1,\sigma_2,\sigma_3$ be permutations*

1.

$$\frac{(\Gamma \vdash \Delta)}{(\sigma_1(\Gamma,\theta) \vdash \Delta)}\,\mathcal{L}GW \qquad \frac{(\Gamma \vdash \Delta)}{(\Gamma \vdash \sigma_1(\theta,\Delta))}\,\mathcal{R}GW$$

2.

$$\frac{(\sigma_1(\Gamma,\theta,\theta) \vdash \Delta)}{(\sigma_2(\Gamma,\theta) \vdash \Delta)}\,\mathcal{L}GC \qquad \frac{(\Gamma \vdash \sigma_1(\theta,\theta,\Delta))}{(\Gamma \vdash \sigma_2(\theta,\Delta))}\,\mathcal{R}GC$$

3.

$$\frac{(\Gamma \vdash \sigma_1(\Delta,\theta)) \quad (\sigma_2(\Phi,\varphi) \vdash \Psi)}{(\sigma_3(\Gamma,\Phi,\theta \to \varphi) \vdash \Delta,\Psi)}\,\mathcal{L}G\!\to \qquad \frac{(\sigma_1(\Gamma,\theta) \vdash \sigma_2(\varphi,\Delta))}{(\Gamma \vdash \sigma_3(\theta \to \varphi,\Delta))}\,\mathcal{R}G\!\to$$

4.

$$\frac{(\sigma_1(\Gamma, \theta) \vdash \Delta)}{(\sigma_1(\Gamma, \theta \wedge \varphi) \vdash \Delta)}\,\mathcal{L}G1\wedge \quad \frac{(\sigma_1(\Gamma, \varphi) \vdash \Delta)}{(\sigma_1(\Gamma, \theta \wedge \varphi) \vdash \Delta)}\,\mathcal{L}G2\wedge$$

$$\frac{(\Gamma \vdash \sigma_1(\Delta, \theta)) \quad (\Phi \vdash \sigma_2(\varphi, \Psi))}{(\Gamma, \Phi \vdash \sigma_3(\theta \wedge \varphi, \Delta, \Psi))}\,\mathcal{R}G\wedge$$

Proof. The main insight here is that for every permutation σ, there exists an inverse permutation σ^{-1} such that $\sigma^{-1}\sigma = \sigma\sigma^{-1} = id$. □

Yet another way one can generalize inference maps is not just through repeated use of exchange, but also repeated use of weakening and contraction. The generalized inference rules that result from this observation are given in the following proposition.

Proposition 5.2.7. *Let σ be a permutation. The inference rules*

1.

$$\frac{(\Gamma \vdash \Delta)}{(\sigma(\Gamma, \Theta) \vdash \Delta)}\,\mathcal{L}GW+ \quad \frac{(\Gamma \vdash \Delta)}{(\Gamma \vdash \sigma(\Theta, \Delta))}\,\mathcal{R}GW+$$

2.

$$\frac{(\sigma(\Phi, \Gamma, \Gamma) \vdash \Delta)}{(\Phi, \Gamma \vdash \Delta)}\,\mathcal{L}GC+ \quad \frac{(\Gamma \vdash \sigma(\Delta, \Delta, \Psi))}{(\Gamma \vdash \Delta, \Psi)}\,\mathcal{R}GC+$$

3.

$$\frac{}{(\bot \vdash \Theta)}\,\bot+$$

are derivable in $\mathscr{L}_S$.

Proof.

1. $\mathcal{L}GW+$ and $\mathcal{R}GW+$ follow from repeated use of $\mathcal{L}W$ and $\mathcal{R}W$ respectively, followed by $\mathcal{L}X+$ and $\mathcal{R}X+$ respectively.

2. $\mathcal{L}GC+$ and $\mathcal{R}GC+$ follow from repeated use of $\mathcal{L}C$ and $\mathcal{R}C$ respectively, followed by $\mathcal{L}X+$ and $\mathcal{R}X+$ respectively.

3. Follows directly from $\mathcal{R}W+$. □

This brings light to a new insight, which is that intuitionistic sequent calculus the way that it is presented in the initial definition is a sound and completete system of logic that starts with these generalized rules. This will be very useful when doing proofs by structural induction, which will hapen very soon.

Corollary 5.2.8. *Let $\mathscr{L}_{GS}$ be a logic with the language as the sequent metalanguage $\mathcal{M}_{\mathcal{L}_I}$ of $\mathcal{L}_I$ and the inference rules I, $\mathcal{L}X+$, $\mathcal{R}X+$, $\mathcal{L}GW$, $\mathcal{R}GW$, $\mathcal{L}GC$, $\mathcal{R}GC$, $\mathcal{L}G\rightarrow$, $\mathcal{R}G\rightarrow$, $\perp$, $\mathcal{L}G\wedge$, and $\mathcal{R}G\wedge$. $\mathscr{L}_{GS}$ is a complete logic to $\mathscr{L}_S$.*

Now is the point where further theoretical insight takes some very advanced machinery, which turns out to be the *Cut* rule. Most texts utilize the *Cut* rule as an axiom to sequent calculus since it is essential to deriving theorems in any sort of reasonable way. However, it is a rather surprising result that actually applied to all provable sequent formulas in sequent calculus, one can derive *Cut* from simply the other axiomized maps in intuitionistic sequent calculus. What the rule is exactly is provided in the definition, however there is a more general version *GCut* that in sequent calculus is logically equivalent to *Cut*, which for practical purposes is not only better to use but actually easier to prove.

Theorem 5.2.9. *the inference map*

$$\frac{(\Gamma \vdash \sigma_1(\theta, \Delta)) \quad (\sigma_2(\Phi, \theta) \vdash \Psi)}{(\Gamma, \Phi \vdash \Delta, \Psi)}\, GCut,$$

for permutations σ_1, σ_2 is derivable in intuitionistic propositional calculus.

Since the proof of this theorem is a rather involved laborious process that takes sophisticated structural inductive techniques and a lot of cases, this proof shall be put off until the very next section–one that is entirely devoted to the *Cut* map and the proof to deriving it.

This final proposition does the job of deriving the inferece maps for the other connectives. Given the definition of these other connectives in terms of the three $\to, \bot, \wedge$, it terms out that one can derive all the others, which is definitely a good pitstop result to be in to untimately figure out that $\mathscr{L}_S$ is sound and complete with $\mathscr{L}_I$.

Proposition 5.2.10. *$\mathcal{L}\neg$, $\mathcal{R}\neg$, $\top$, $\mathcal{L}\vee$, $\mathcal{R}1\vee$, $\mathcal{R}2\vee$ are derivable in $\mathscr{L}_S$.*

Proof. $\mathcal{L}\neg$

$$\cfrac{(\Gamma \vdash \theta, \Delta) \qquad \cfrac{\cfrac{}{(\theta \vdash \theta)}\,I \qquad \cfrac{}{(\bot \vdash \quad)}\,\bot}{(\theta, \neg\theta \vdash \quad)}\,\mathcal{L}\to}{(\Gamma, \neg\theta \vdash \Delta)}\,GCut.$$

$\mathcal{R}\neg$

$$\cfrac{\cfrac{\cfrac{\cfrac{}{(\theta \vdash \theta)}\,I}{(\theta \vdash \bot, \theta}\,\mathcal{R}W}{(\quad \vdash \neg\theta, \theta)}\,\mathcal{R}\to \qquad (\Gamma, \theta \vdash \Delta)}{(\Gamma \vdash \neg\theta, \Delta)}\,GCut.$$

$\top$ Using $\mathcal{R}\neg$, we get

$$\cfrac{\cfrac{}{(\bot \vdash \quad)}\,\bot}{(\quad \vdash \top)}\,\mathcal{R}\neg.$$

$\mathcal{L}\vee$ Using $\mathcal{L}\neg$, we get

$$\cfrac{\cfrac{\cfrac{\cfrac{\Gamma, \theta \vdash \Delta}{\Gamma \vdash \neg\theta, \Delta}\,\mathcal{R}\neg}{\Gamma, \neg\neg\theta \vdash, \Delta}\,\mathcal{L}\neg}{\Gamma \vdash \neg\neg\neg\theta, \Delta}\,\mathcal{R}\neg \qquad \Phi, \varphi \vdash \Psi}{\Gamma, \phi \vee \varphi, \Phi \vdash \Delta, \Psi}\,\mathcal{L}\to .$$

$\mathcal{R}1\vee$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Gamma \vdash \theta, \Delta}{\Gamma \vdash \theta, \varphi, \Delta}\,\mathcal{R}GW}{\Gamma, \neg\theta \vdash \varphi, \Delta}\,\mathcal{L}\neg}{\Gamma \vdash \neg\neg\theta, \varphi, \Delta}\,\mathcal{R}\neg}{\Gamma, \neg\neg\neg\theta \vdash \varphi, \Delta}\,\mathcal{L}\neg}{\Gamma \vdash \theta \vee \varphi, \Delta}\,\mathcal{R}\rightarrow .$$

$\mathcal{R}2\vee$ Similar to $\mathcal{R}1\vee$. □

5.3 The Cut Rule

When one first hears the result that the *Cut* rule is derivable in $\mathscr{L}_S$, it is easy to think that this is an absurd notion that is easy to find some sort of counterexample. After all, no rule takes a proposition and eliminates them, except for the very specific case of the contraction rule. But the *Cut* rule really happens as a result of atomically derivable formulas being limited to only $(\theta \vdash \theta)$ and $(\bot \vdash \;)$, and it just happening that from the structurally inductive ground up, given that one can prove $\Gamma \vdash \theta, \Delta$ and $\Phi, \theta \vdash \Psi$, one can *always* find a way to prove $\Gamma, \Phi \vdash \Delta, \Psi$ using the original inference map axioms of $\mathscr{L}_S$.

As stated before this theorem really seeks to prove a more general result *GCut*, which is a more general and easier to prove result because it gives a more powerful inductive hypothesis to work with (that one gets for free from the base case), and proving *GCut* involves structural induction on the left formula. Where it gets tricky, however, is when one is to prove that *GCut* works for $\theta := \varphi_1 \rightarrow \varphi_2$ or $\theta := \varphi_1 \wedge \varphi_2$ with no straightforward way to backtrack and cut eliminate φ_1 and φ_2 since the right side has not been broken down. What then is in order is when proving specifically $\mathcal{R} \rightarrow$ and $\mathcal{L}\wedge$ is doing such by structural induction on the right hand formula.

Thankfully, this proof is made a little similar by the fact that there are a lot of similar cases, and also by the idea of doing structural induction on $\mathscr{L}_{GS}$ instead of $\mathscr{L}_S$ because remember those two systems are sound and complete on each other. Now the rest of this section will be the theorem restated again and its proof.

Theorem 5.3.1. *the inference map*

$$\frac{(\Gamma \vdash \sigma_1(\theta, \Delta)) \quad (\sigma_2(\Phi, \theta) \vdash \Psi)}{(\Gamma, \Phi \vdash \Delta, \Psi)} GCut,$$

for permutations σ_1, σ_2 is derivable in intuitionistic propositional calculus.

Proof. It shall suffice to prove our result on the logic $\mathscr{L}_{GS}$ in **Corollary 6.2.6**, since it is sound with $\mathscr{L}_S$. Let $(\Gamma \vdash \sigma_1(\theta, \Delta))$ and $(\sigma_2(\Phi, \theta) \vdash \Psi)$ be $\mathcal{M}$-formulae. We shall proceed by structural induction on by $\mathscr{L}_{GS}$. We shall prove $GCut$ by structural induction on $(\Gamma \vdash \sigma_1(\theta, \Delta))$ based on all derivable formulae. For our atomic case, we have two cases.

I Suppose $(\Gamma \vdash \sigma_1(\theta, \Delta)) := (\theta \vdash \theta)$

$\bot$ Since $|\sigma_1(\theta, \Delta))| \geqslant 1$, we cannot have $(\Gamma \vdash \sigma_1(\theta, \Delta)) := (\bot \vdash \quad)$. We are done.

For the inductive step, given $n \geqslant 1$, suppose $\gg_{n+1} (\Gamma \vdash \sigma_1(\theta, \Delta))$ and we can apply $Gcut$ with $(\sigma_2(\Phi, \theta) \vdash \Psi)$ to all order n derivable $\mathcal{M}_{\mathcal{L}_I}$-formula. We have ten cases.

$\mathcal{L}X+$ Suppose $(\Gamma \vdash (\sigma_1(\theta, \Delta)) := \sigma'(\Gamma') \vdash \sigma_1(\theta, \Delta)$ and $\gg_n \Gamma' \vdash \sigma_1(\theta, \Delta)$, where σ' is a permutation. We have

$$\cfrac{\cfrac{(\Gamma' \vdash \sigma_1(\theta, \Delta)) \quad (\sigma_2(\Phi, \theta) \vdash \Psi)}{(\Gamma', \Phi \vdash \Delta, \Psi)} GCut}{(\Gamma, \Phi \vdash \Delta, \Psi)} \mathcal{L}X+.$$

$\mathcal{R}X+$ Suppose $(\Gamma \vdash \sigma_1(\theta, \Delta)) := (\Gamma \vdash \sigma'(\theta, \Delta))$ and $\gg_n (\Gamma \vdash \sigma'(\theta, \Delta))$, where σ' is a permutation. We have

$$\cfrac{\cfrac{(\Gamma' \vdash \sigma'(\theta, \Delta))) \quad (\sigma_2(\Phi, \theta) \vdash \Psi)}{(\Gamma, \Phi \vdash \Delta', \Psi)} GCut}{(\Gamma, \Phi \vdash \Delta, \Psi)} \mathcal{R}X+.$$

$\mathcal{L}GW$ Suppose $(\Gamma \vdash \sigma_1(\theta, \Delta)) := \sigma'(\Gamma', \gamma) \vdash \sigma_1(\theta, \Delta)$ and $\gg_n (\Gamma' \vdash \sigma_1(\theta, \Delta))$, where σ' is a permutation. We have

$$\dfrac{\dfrac{(\Gamma' \vdash \sigma_1(\theta, \Delta)) \quad (\sigma_2(\Phi, \theta) \vdash \Psi)}{(\Gamma', \Phi \vdash \Delta, \Psi)}\, GCut}{(\Gamma, \Phi \vdash \Delta, \Psi)}\, \mathcal{R}GW.$$

$\mathcal{R}GW$ Suppose $(\Gamma \vdash \sigma_1(\theta, \Delta)) := (\Gamma \vdash \sigma'(\varphi, \Delta'))$ and $\gg_n (\Gamma \vdash \Delta')$, where σ' is a permutation. If φ is not θ, we find $\Delta' := \sigma''(\theta, \Delta'')$, for some sequent Δ'' and permutation σ'', and we have

$$\dfrac{\dfrac{(\Gamma \vdash \Delta') \quad (\sigma_2(\Phi, \theta) \vdash \Psi)}{(\Gamma, \Phi \vdash \Delta'', \Psi)}\, GCut}{(\Gamma, \Phi \vdash \Delta, \Psi)}\, \mathcal{L}GW.$$

Otherwise, if $\varphi := \theta$, we find $\Delta := \sigma''(\Delta')$, for some permutation σ'', and we have

$$\dfrac{\dfrac{(\Gamma \vdash \Delta')}{(\Gamma, \Phi \vdash \Delta')}\, \mathcal{L}W}{(\Gamma, \Phi \vdash \Delta, \Psi)}\, \mathcal{R}GW.$$

$\mathcal{L}GC$ Suppose $(\Gamma \vdash \sigma_1(\theta, \Delta)) := (\sigma'(\Gamma', \gamma, \gamma) \vdash \sigma_1(\theta, \Delta))$ and $\gg_n \sigma''(\Gamma', \gamma) \vdash \sigma_1(\theta, \Delta)$, where σ' and σ'' are permutations. We find

$$\dfrac{\dfrac{(\sigma'(\Gamma', \gamma, \gamma) \vdash \sigma_1(\theta, \Delta)) \quad (\sigma_2(\Phi, \theta) \vdash \Psi)}{(\Gamma', \gamma, \gamma, \Phi \vdash \Delta, \Psi)}\, GCut}{(\Gamma, \Phi \vdash \Delta, \Psi)}\, \mathcal{L}GC.$$

$\mathcal{R}GC$ Suppose $(\Gamma \vdash \sigma_1(\theta, \Delta)) := (\Gamma \vdash \sigma'(\varphi, \varphi, \Delta))$ and $\gg_n \Gamma \vdash \sigma''(\varphi, \Delta')$, where σ' and σ'' are permutations. If φ is not θ, we find $\Delta' := \sigma'''(\theta, \Delta'')$, for some sequent Δ'' and permutation σ''', and we have

$$\dfrac{\dfrac{(\Gamma \vdash \sigma''(\varphi, \varphi, \Delta')) \quad (\sigma_2(\Phi, \theta) \vdash \Psi)}{(\Gamma, \Phi \vdash \varphi, \varphi, \Delta'', \Psi)}\, GCut}{(\Gamma, \Phi \vdash \Delta, \Psi)}\, \mathcal{R}GC.$$

Suppose $\varphi := \theta$. Throughout the proof, we have actually proven a more general hypothesis that follows directly from our current hypothesis; this is only just the first instance where I use it (hence I bring it up now, since you have the motivation). This hypothesis is: *GCut* may be applied with $(\sigma_2(\Phi, \theta) \vdash \Psi)$ and the result $(\Gamma, \Phi \vdash \Delta, \Psi)$ after using $GCut$ on $(\sigma_2(\Phi, \theta) \vdash \Psi)$ (should any θ exist in Γ, Φ). This certainly occurs in our base case since in the I case $(\Gamma, \Phi \vdash \Delta, \Psi) := (\theta \vdash \theta)$, which is just $(\Gamma \vdash \sigma_1(\theta, \Delta))$ again, and in the $\perp$ case we cannot satisfy our given hypothesis. In our inductive step (in every case), this just follows simply through deriving $(\Gamma, \Phi \vdash \Delta, \Psi)$ entirely through inductively hypothesized formulas and inference maps that we woud then have established to be closed under our inductive step (so proving the more general hypothesis would follow from similarity of proving the less general one).

We find that

$$\dfrac{\dfrac{\dfrac{\dfrac{(\Gamma \vdash \sigma''(\varphi, \varphi, \Delta')) \quad (\sigma_2(\Phi, \theta) \vdash \Psi)}{(\Gamma, \Phi \vdash \theta, \Delta', \Psi)}\, GCut \qquad (\sigma_2(\Phi, \theta) \vdash \Psi)}{(\Gamma, \Phi, \Phi \vdash \Delta, \Psi, \Psi)}\, GCut.}{(\Gamma, \Phi \vdash \Delta, \Psi, \Psi)}\, \mathcal{L}GC+}{(\Gamma, \Phi \vdash \Delta, \Psi)}\, \mathcal{R}GC+$$

$\mathcal{L}G \rightarrow$ Suppose $\varphi := \varphi_1 \rightarrow \varphi_2$ and $(\Gamma \vdash \sigma_1(\theta, \Delta)) := (\sigma'(\Gamma_1, \Gamma_2, \varphi) \vdash \sigma_1(\theta, \Delta_1, \Delta_2))$ and $\gg_n ((\Gamma_1 \vdash \sigma'(\varphi_1, \theta, \Delta_1)), (\sigma''(\Gamma_2, \varphi_1) \vdash \Delta_2))$, where σ' and σ'' are permutations. We find that

$$\dfrac{\dfrac{(\Gamma_1 \vdash \sigma'(\varphi_1, \theta, \Delta_1)) \quad (\sigma_2(\Phi, \theta) \vdash \Psi)}{(\Gamma_1, \Phi \vdash \varphi_1, \Delta_1, \Psi)}\, GCut \qquad (\sigma''(\Gamma_2, \varphi_2) \vdash \Delta_2)}{(\Gamma, \Phi \vdash \Delta, \Psi)}\, \mathcal{L}G \rightarrow .$$

$\mathcal{R}G \rightarrow$ Suppose $\varphi := \varphi_1 \rightarrow \varphi_2$ and $(\Gamma \vdash \sigma_1(\theta, \Delta)) := (\Gamma \vdash \sigma'(\varphi, \Delta'))$ and $\gg_n (\sigma''(\Gamma, \varphi_1) \vdash \sigma'''(\varphi_2, \Delta'))$, where $\sigma', \sigma'', \sigma'''$ are permutations. If φ is not θ, then $\Delta' := \sigma_M(\Delta'', \theta)$, for some permutation σ_M, so we have

$$\dfrac{\dfrac{(\sigma''(\Gamma, \varphi_1) \vdash \sigma'''(\varphi_2, \Delta')) \quad (\sigma_2(\Phi, \theta) \vdash \Psi)}{(\Gamma, \varphi_1, \Phi \vdash \varphi_2, \Delta'', \Psi)}\, GCut}{(\Gamma, \Phi \vdash \Delta, \Psi)}\, \mathcal{R} \rightarrow .$$

Now suppose $\varphi := \theta$. We shall proceed by structural induction on $(\sigma_2(\Phi, \theta) \vdash \Psi)$ (in other words the right side). Note that the subcases $\mathcal{R}G \rightarrow: I$, $\mathcal{R}G \rightarrow: \perp$ work similarly to our base cases done earlier. For the inductive step, we find $\mathcal{R}G \rightarrow: \mathcal{L}X$, $\mathcal{R}G \rightarrow: \mathcal{R}X$, $\mathcal{R}G \rightarrow: \mathcal{L}GW$, $\mathcal{R}G \rightarrow: \mathcal{R}GW$, $\mathcal{R}G \rightarrow: \mathcal{L}GC$, $\mathcal{R}G \rightarrow: \mathcal{R}GC$ follows similarly to our previous cases. The remaining subcases include $\mathcal{R}G \rightarrow: \mathcal{L}G \rightarrow$, $\mathcal{R}G \rightarrow: \mathcal{R}G \rightarrow$, $\mathcal{R}G \rightarrow: \mathcal{L}G\wedge$, $\mathcal{R}G \rightarrow: \mathcal{L}G\wedge$.

$\mathcal{R}G \rightarrow: \mathcal{L}G \rightarrow$ Suppose $\$:= \$_1 \rightarrow \$_2$ and $(\sigma_2(\Phi,\theta) \vdash \Psi) := (\sigma^*(\Phi_1,\Phi_2,\$) \vdash \Psi_1,\Psi_2)$ and $\gg_n ((\Phi_1 \vdash \sigma^{**}(\$_1,\Psi_1)), (\sigma^{***}(\Phi_2,\$_2) \vdash \Psi_2))$, where $\sigma^*,\sigma^{**},\sigma^{***}$ are permutations. If $\$$ is not θ, then we are left with a case similar to $\mathcal{L}G \rightarrow$ when φ was not θ. Suppose $\$:= \theta$. Then $\$_1 := \varphi_1$ and $\$_2 := \varphi_2$. We find

$$\dfrac{\dfrac{(\sigma''(\Gamma,\varphi_1) \vdash \sigma'''(\varphi_2,\Delta')) \quad ((\Phi_1 \vdash \sigma^{**}(\$_1,\Psi_1))}{(\Gamma,\Phi_1 \vdash \sigma'''(\varphi_2,\Delta'),\Psi_1)}\,GCut \qquad (\sigma^{***}(\Phi_2,\$_2) \vdash \Psi_2)}{\Gamma,\Phi \vdash \Delta,\Psi}\,GCut.$$

$\mathcal{R}G \rightarrow: \mathcal{R}G \rightarrow$ Suppose $\$:= \$_1 \rightarrow \$_2$ and $(\sigma_2(\Phi,\theta) \vdash \Psi) := (\sigma_2(\Phi,\theta) \vdash \sigma^*(\$,\Psi'))$ and $\gg_n (\sigma^{**}(\Phi,\theta,\$_1) \vdash \sigma^{***}(\$_2,\Psi'))$, where $\sigma^*,\sigma^{**},\sigma^{***}$ are permutations. Then

$$\dfrac{\dfrac{(\Gamma \vdash \sigma_1(\theta,\Delta)) \quad (\sigma^{**}(\Phi,\theta,\$_1) \vdash \sigma^{***}(\$_2,\Psi'))}{(\Gamma,\Phi,\$_1 \vdash \Delta,\$_2,\Psi')}\,GCut}{(\Gamma,\Phi \vdash \Delta,\Psi)}\,\mathcal{R}G \rightarrow .$$

$\mathcal{R}G \rightarrow: \mathcal{L}G \wedge 1$ Suppose $\$:= \$_1 \wedge \$_2$ and $(\sigma_2(\Phi,\theta) \vdash \Psi) := (\sigma_2(\Phi',\$,\theta) \vdash \Psi$ and $\gg_n (\sigma^*(\Phi',\$_1,\theta) \vdash \Psi$, where σ^* is a permutation. Then

$$\dfrac{\dfrac{(\Gamma \vdash \sigma_1(\theta,\Delta)) \quad (\sigma^*(\Phi',\$_1,\theta) \vdash \Psi)}{(\Gamma,\Phi',\$_1 \vdash \Delta,\Psi)}\,Gcut}{(\Gamma,\Phi \vdash \Delta,\Psi)}\,\mathcal{L}G \wedge 1.$$

$\mathcal{R}G \rightarrow: \mathcal{L}G \wedge 2$ Similar to $\mathcal{R}G \rightarrow: \mathcal{L}G \wedge 1$, except using $\$_2$ and sub-$\mathcal{L}G \wedge 2$ in place of $\$_1$ and sub-$\mathcal{L}G \wedge 1$, respectively.

$\mathcal{R}G \rightarrow: \mathcal{R}G\wedge$ Suppose $\$:= \$_1 \wedge \$_2$ and $(\sigma_2(\Phi,\theta) \vdash \Psi) := (\Phi_1,\Phi_2) \vdash \sigma^*(\Psi_1,\Psi_2,\$)$ and $\gg_n ((\Phi_1 \vdash \sigma^{**}(\Psi_1,\$_1),(\Phi_2 \vdash \sigma^{***}(\Psi_2,\$_2))$, where $\sigma^*,\sigma^{**},\sigma^{***}$ is a permutation. Without loss of generality, suppose $\theta \preceq \Phi_1$. Then $\Phi_1 := \sigma_C(\Phi_1',\theta)$, for some permutatin σ_C and

$$\dfrac{\dfrac{(\Gamma \vdash \sigma_1(\theta,\Delta)) \quad (\Phi_1 \vdash \sigma^{**}(\Psi_1,\$_1)}{\Gamma,\Phi_1' \vdash \Delta,\Psi_1,\$_1}\,GCut \qquad (\Phi_2 \vdash \sigma^{***}(\Psi_2,\$_2)}{(\Gamma,\Phi \vdash \Delta,\Psi)}\,\mathcal{R}G \wedge .$$

$\mathcal{L}G \wedge 1$ Suppose $\varphi := \varphi_1 \wedge \varphi_2$ and $(\Gamma \vdash \sigma_1(\theta, \Delta) := (\sigma'(\Gamma', \varphi) \vdash \sigma_1(\theta, \Delta)$ and $\gg_n$ $(\sigma''(\Gamma', \varphi_1) \vdash \Delta)$, where σ', σ'' are permutations. Follows similarly from the subcase $\mathcal{R}G \rightarrow$: $\mathcal{L}G \wedge 1$.

$\mathcal{L}G \wedge 2$ Similar to $\mathcal{R}G \rightarrow$: $\mathcal{L}G \wedge 1$.

$\mathcal{R}G\wedge$ Suppose $\varphi := \varphi_1 \wedge \varphi_2$ and $(\Gamma \vdash \sigma_1(\theta, \Delta)) := (\Gamma_1, \Gamma_2 \vdash \sigma'(\varphi, \Delta_1, \Delta_2))$ and $\gg_n$ $((\Gamma_1 \vdash \sigma''(\varphi_1, \Delta_1)), (\Gamma_2 \vdash \sigma'''(\varphi_2, \Delta_2)))$, where $\sigma', \sigma'', \sigma'''$ are permutations. If φ is not θ, then we have a case similar to sub-$\mathcal{R}G\wedge$ of $\mathcal{R} \rightarrow$.

Suppose $\varphi := \theta$. We shall prove this result via structural induction on $(\sigma_2(\Phi, \theta) \vdash \Psi)$. All subcases except $\mathcal{R}G\wedge : \mathcal{L}G \wedge 1$ and $\mathcal{R}G\wedge : \mathcal{L}G \wedge 2$ follow similarly the previous cases or subcases of $\mathcal{R}G \rightarrow$, so it shall suffice to prove those subcases.

$\mathcal{R}G\wedge : \mathcal{L}G \wedge 1$ Suppose $\$:= \$_1 \wedge \$_2$ and $(\sigma_2(\Phi, \theta) \vdash \Psi) := (\sigma_2(\Phi', \$) \vdash \Psi)$ and $\gg_n$ $((\sigma^*(\Phi', \$_1) \vdash \Psi)$, where σ^* is a permutation. If $\$$ is not θ, then this case is similar to $\mathcal{R}G \rightarrow$: $\mathcal{L}G \wedge 1$. Suppose $\$:= \theta$. We find $\$_1 := \varphi_1$, so

$$\cfrac{\cfrac{\cfrac{(\Gamma_1 \vdash \sigma''(\varphi_1, \Delta_1)) \quad ((\sigma^*(\Phi', \$_1) \vdash \Psi)}{(\Gamma_1, \Phi \vdash \Delta_1, \Psi)}\,GCut}{(\Gamma, \Phi \vdash \Delta_1, \Psi)}\,\mathcal{L}GW+}{(\Gamma, \Phi \vdash \Delta, \Psi)}\,\mathcal{R}GW+.$$

$\mathcal{R}G\wedge : \mathcal{L}G \wedge 2$ Similar to $\mathcal{R}G\wedge : \mathcal{L}G \wedge 1$.

□

5.4 Properties of Intuitionistic Sequent Logic

This section deals with some important properties of $\mathscr{L}_S$, including many properties to do with propositions that are interchangable. Interchangability of θ with φ turns out to be equivalent to proving the formula $(\ \vdash \theta \leftrightarrow \varphi)$. The other crucial find is the property that $\wedge$ is a connective that corresponds in a way to commas between propositions on the left side of $\vdash$ and $\vee$ is a connective that corresponds in a way to commas between propositions on the right side of $\vdash$. In other words, from $(\sigma_1(\Gamma, \theta, \varphi) \vdash \Delta)$ one can prove

$(\sigma_2(\Gamma, \theta \wedge \varphi) \vdash \Delta)$ and vice versa for permutations σ_1, σ_2, and a similar principle applies to $\vee$. Moreover, when combining sequents in $\mathscr{L}_S$ with $\mathcal{L}\vee$ or $\mathcal{R}\wedge$, this logically equivalently corresponds to a product arrangement of the sequent broken down by the propositions connected by $\vee$ or $\wedge$ in the fashion given in the proposition below.

Proposition 5.4.1.

1. $\langle(\Gamma \vdash \sigma_1(\Delta, \theta)), (\Gamma \vdash \sigma_2(\Delta, \varphi))\rangle \ll\gg (\Gamma \vdash \sigma_3(\Delta, \theta \wedge \varphi))$*;*

2. $(\Gamma \vdash \sigma_1(\Delta, \theta, \varphi)) \ll\gg (\Gamma \vdash \sigma_2(\Delta, \theta \vee \varphi))$*;*

3. $(\sigma_1(\Gamma, \theta, \varphi) \vdash \Delta) \ll\gg (\sigma_2(\Gamma, \theta \wedge \varphi) \vdash \Delta)$*.*

4. $\langle(\sigma_1(\Gamma, \theta) \vdash \Delta), (\sigma_2(\Gamma, \varphi) \vdash \Delta)\rangle \ll\gg (\sigma_3(\Gamma, \theta \vee \varphi) \vdash \Delta)$*.*

Proof. 1. We find

$$\cfrac{\cfrac{\cfrac{(\Gamma \vdash \sigma_1(\Delta, \theta)) \quad (\Gamma \vdash \sigma_2(\Delta, \varphi))}{(\Gamma, \Gamma \vdash \Delta, \Delta, \theta \wedge \varphi)}\,\mathcal{R}G\wedge}{(\Gamma \vdash \Delta, \Delta, \theta \wedge \varphi)}\,\mathcal{L}GC+}{(\Gamma \vdash \sigma_3(\theta \wedge \varphi, \Delta))}\,\mathcal{R}X+$$

and

$$\cfrac{(\sigma_3(\Gamma \vdash \Delta, \theta \wedge \varphi)) \quad \cfrac{(\theta \vdash \theta)}{(\theta \wedge \varphi \vdash \theta)}\,\mathcal{L}\wedge 1}{(\Gamma \vdash \sigma_1(\theta, \Delta))}\,\mathcal{R}X+ \qquad \cfrac{(\sigma_3(\Gamma \vdash \Delta, \theta \wedge \varphi)) \quad \cfrac{(\varphi \vdash \varphi)}{(\theta \wedge \varphi \vdash \varphi)}\,\mathcal{L}\wedge 2}{(\Gamma \vdash \sigma_2(\varphi, \Delta))}\,\mathcal{R}X+ .$$

2. We find

$$\cfrac{\cfrac{\cfrac{(\Gamma \vdash \sigma_1(\Delta, \theta, \varphi)) \quad \cfrac{(\theta \vdash \theta)}{(\theta \vdash \theta \vee \varphi)}\,\mathcal{R}\vee 1}{(\Gamma \vdash \varphi, \Delta, \theta \vee \varphi)}\,GCut \quad \cfrac{(\varphi \vdash \varphi)}{(\varphi \vdash \theta \vee \varphi)}\,\mathcal{R}\vee 2}{(\Gamma \vdash \Delta, \theta \vee \varphi, \theta \vee \varphi)}\,GCut}{(\Gamma \vdash \sigma_2(\theta \vee \varphi, \Delta))}\,\mathcal{R}GC,$$

and

$$\cfrac{\cfrac{(\Gamma \vdash \sigma_2(\theta \vee \varphi, \Delta)) \quad \cfrac{(\theta \vdash \theta) \quad (\varphi \vdash \varphi)}{(\theta \vee \varphi \vdash \theta, \varphi)}\,\mathcal{L}\vee}{(\Gamma \vdash \Delta, \theta, \varphi)}\,GCut}{(\Gamma \vdash \sigma_1(\theta, \varphi, \Delta))}\,\mathcal{R}X+ .$$

3. and 4. we will let follow from similarity to 2. and 1. respectively. □

This next lemma shows the analogue in $\mathscr{L}_S$ of $\to E$ is derivable in the system, which further establishes the equivalence between showing $\theta \Bumpeq \varphi$ and $\gg (\quad \vdash \theta \leftrightarrow \varphi)$.

Lemma 5.4.2.

1. The inference map

$$\frac{(\Gamma \vdash \theta, \Delta) \quad (\Phi \vdash \theta \to \varphi, \Psi)}{(\Gamma, \Phi \vdash \varphi, \Delta, \Psi)} \to E,$$

is derivable in $\mathscr{L}_S$.

2. $\gg (\quad \vdash \theta \to \varphi)$ if and only if for all sequents Γ, Δ we have $(\Gamma \vdash \theta, \Delta \gg \Gamma \vdash \varphi, \Delta)$.

3. $\gg (\quad \vdash \theta \leftrightarrow \varphi)$ if and only if $\theta \Bumpeq \varphi$.

Proof. 1.

$$\cfrac{\Phi \vdash \theta \to \varphi, \Psi \qquad \cfrac{\cfrac{}{(\varphi \vdash \varphi)}\,I \qquad (\Gamma \vdash \theta, \Delta)}{\Gamma, \theta \to \varphi \vdash \varphi, \Delta}\,\mathcal{L}\to}{(\Gamma, \Phi \vdash \varphi, \Delta, \Psi)}\,GCut.$$

2. Suppose $(\Gamma \vdash \theta, \Delta \gg \Gamma \vdash \varphi, \Delta)$, for all sequents Γ, Δ. Then we have $\gg \theta \vdash \varphi$ and our conclusion follows from $\mathcal{R} \to$. Conversely, suppose $\gg (\quad \vdash \theta \to \varphi)$. Let Γ, Δ be sequents. By $\to E$, which we proved in part 1 is a derivable in $\mathscr{L}_S$, we get

$$\cfrac{\cfrac{\cfrac{\cfrac{}{(\theta \vdash \theta)}\,I \qquad (\quad \vdash \theta \to \varphi)}{\theta \vdash \varphi}\to E}{\Gamma, \theta \vdash \varphi}\,\mathcal{L}GW+}{\Gamma, \theta \vdash \varphi, \Delta}\,\mathcal{R}GW+.$$

3. Follows directly from part 2. □

Next is this lemma, which shows some of the formulas that are interchangeable in $\mathscr{L}_S$. Do note that the proofs for some of these results involve rather sophisticated Getzen derivations, but are definitely worth it to see some insights to the system, as well as prove a very central theorem in this chapter.

Lemma 5.4.3.

1. $\theta \vee (\varphi \wedge \lambda) \Bumpeq (\theta \vee \varphi) \wedge (\theta \vee \lambda)$ *and* $\theta \wedge (\varphi \vee \lambda) \Bumpeq (\theta \wedge \varphi) \vee (\theta \wedge \lambda)$

2. $\theta \wedge \top \Bumpeq \theta$ *and* $\theta \vee \bot \Bumpeq \theta$*;*

3. $\theta \wedge \bot \Bumpeq \bot$ *and* $\theta \vee \top \Bumpeq \top$*;*

4. $(\theta \wedge \varphi) \wedge \lambda \Bumpeq \theta \wedge (\varphi \wedge \lambda)$ *and* $(\theta \vee \varphi) \vee \lambda \Bumpeq \theta \vee (\varphi \vee \lambda)$

Proof. It shall suffice by **Lemma 5.4.2 (3)** to prove $\gg (\ \ \vdash \theta_1 \leftrightarrow \theta_2)$ in order to prove $\theta_1 \Bumpeq \theta_2$, for any propositions θ_1, θ_2.

1. We find

$$\begin{prooftree}
\AxiomC{}
\RightLabel{I}
\UnaryInfC{$(\theta \vdash \theta)$}
\RightLabel{$\mathcal{R} \vee 1$}
\UnaryInfC{$(\theta \vdash \theta \vee \varphi)$}
\AxiomC{}
\RightLabel{I}
\UnaryInfC{$(\theta \vdash \theta)$}
\RightLabel{$\mathcal{R} \vee 1$}
\UnaryInfC{$(\theta \vdash \theta \vee \lambda)$}
\RightLabel{$\mathcal{R}\wedge$}
\BinaryInfC{$(\theta, \theta \vdash (\theta \vee \varphi) \wedge (\theta \vee \lambda))$}
\RightLabel{$\mathcal{L}C$}
\UnaryInfC{$(\theta \vdash (\theta \vee \varphi) \wedge (\theta \vee \lambda))$}
\AxiomC{}
\RightLabel{I}
\UnaryInfC{$(\varphi \vdash \varphi)$}
\RightLabel{$\mathcal{L} \wedge 1$}
\UnaryInfC{$(\varphi \wedge \lambda \vdash \varphi)$}
\RightLabel{$\mathcal{R} \vee 2$}
\UnaryInfC{$(\varphi \wedge \lambda \vdash \theta \vee \varphi)$}
\AxiomC{}
\RightLabel{I}
\UnaryInfC{$(\lambda \vdash \lambda)$}
\RightLabel{$\mathcal{L} \wedge 2$}
\UnaryInfC{$(\varphi \wedge \lambda \vdash \lambda)$}
\RightLabel{$\mathcal{R} \vee 2$}
\UnaryInfC{$(\varphi \wedge \lambda \vdash \theta \vee \lambda)$}
\RightLabel{$\mathcal{R}\wedge$}
\BinaryInfC{$(\varphi \wedge \lambda, \varphi \wedge \lambda \vdash (\theta \vee \varphi) \wedge (\theta \vee \lambda))$}
\RightLabel{$\mathcal{L}C$}
\UnaryInfC{$(\varphi \wedge \lambda \vdash (\theta \vee \varphi) \wedge (\theta \vee \lambda))$}
\RightLabel{$\mathcal{L}\vee$}
\BinaryInfC{$(\theta \vee (\varphi \wedge \lambda) \vdash (\theta \vee \varphi) \wedge (\theta \vee \lambda))$}
\RightLabel{$\mathcal{L} \rightarrow,$}
\UnaryInfC{$(\ \ \vdash (\theta \vee (\varphi \wedge \lambda)) \rightarrow ((\theta \vee \varphi) \wedge (\theta \vee \lambda)))$}
\end{prooftree}$$

and

$$\begin{prooftree}
\AxiomC{}
\RightLabel{I}
\UnaryInfC{$(\varphi \vdash \varphi)$}
\AxiomC{}
\RightLabel{I}
\UnaryInfC{$(\theta \vdash \theta)$}
\AxiomC{}
\RightLabel{I}
\UnaryInfC{$(\lambda \vdash \lambda)$}
\RightLabel{$\mathcal{L}\vee$}
\BinaryInfC{$(\theta \vee \lambda \vdash \theta, \lambda)$}
\RightLabel{$\mathcal{R}G\wedge$}
\BinaryInfC{$(\theta \vee \lambda, \varphi \vdash \theta, \lambda \wedge \varphi)$}
\RightLabel{$\mathcal{L}G \wedge 2.$}
\UnaryInfC{$((\theta \vee \varphi) \wedge (\theta \vee \lambda), \varphi \vdash \theta, \lambda \wedge \varphi)$}
\end{prooftree}$$

It follows by **Proposition 5.4.1 (2)**, we find $\gg ((\theta \vee \varphi) \wedge (\theta \vee \lambda), \varphi \vdash \theta \vee (\lambda \wedge \varphi))$. We find

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{(\theta \vdash \theta)}\,I \qquad \dfrac{}{(\varphi \vdash \varphi)}\,I}{(\theta \vee \varphi \vdash \theta, \varphi)}\,\mathcal{L}\vee}{((\theta \vee \varphi) \wedge (\theta \vee \lambda) \vdash \theta, \varphi)}\,\mathcal{L}\wedge 2}{((\theta \vee \varphi) \wedge (\theta \vee \lambda) \vdash \theta \vee (\lambda \wedge \varphi), \varphi)}\,\mathcal{R}\vee 1 \qquad ((\theta \vee \varphi) \wedge (\theta \vee \lambda), \varphi \vdash \theta \vee (\lambda \wedge \varphi))}{((\theta \vee \varphi) \wedge (\theta \vee \lambda), (\theta \vee \varphi) \wedge (\theta \vee \lambda) \vdash \theta \vee (\lambda \wedge \varphi), \theta \vee (\lambda \wedge \varphi))}\,GCut}{((\theta \vee \varphi) \wedge (\theta \vee \lambda) \vdash \theta \vee (\lambda \wedge \varphi), \theta \vee (\lambda \wedge \varphi))}\,\mathcal{L}C}{((\theta \vee \varphi) \wedge (\theta \vee \lambda) \vdash \theta \vee (\lambda \wedge \varphi))}\,\mathcal{R}C}{(\quad \vdash (\theta \vee (\lambda \wedge \varphi)) \rightarrow ((\theta \vee \varphi) \wedge (\theta \vee \lambda)))}\,\mathcal{R}\rightarrow$$

Using $\mathcal{R}\wedge$ on $\langle(\quad \vdash (\theta \vee (\lambda \wedge \varphi)) \rightarrow ((\theta \vee \varphi) \wedge (\theta \vee \lambda))), (\quad \vdash (\theta \vee (\varphi \wedge \lambda)) \rightarrow ((\theta \vee \varphi) \wedge (\theta \vee \lambda)))\rangle$, we conclude $(\quad \vdash (\theta \vee (\varphi \wedge \lambda)) \leftrightarrow ((\theta \vee \varphi) \wedge (\theta \vee \lambda)))$.

Showing $\theta \wedge (\varphi \vee \lambda) \Bumpeq (\theta \wedge \varphi) \vee (\theta \wedge \lambda)$ is (tediously) similar using $\vee$ inferences in place of $\wedge$ inferences and vice versa.

2.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{(\theta \vdash \theta)}\,I}{(\theta \wedge \top \vdash \theta)}\,\mathcal{L}\wedge 1}{(\quad \vdash (\theta \wedge \top) \rightarrow \theta)}\,\mathcal{R}\rightarrow \qquad \dfrac{\dfrac{\dfrac{}{(\theta \vdash \theta)}\,I \quad \dfrac{}{(\quad \vdash \top)}\,\top}{\theta \vdash \theta \wedge \top}\,\mathcal{R}\wedge}{(\quad \vdash \theta \rightarrow (\theta \wedge \top))}\,\mathcal{R}\rightarrow}{(\quad \vdash (\theta \wedge \top) \leftrightarrow \theta)}\,\mathcal{R}\wedge\,.$$

We get $\theta \vee \bot \Bumpeq \theta$ by a similar derivation.

3.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{(\bot \vdash \bot)}\,\bot +}{(\theta \wedge \bot \vdash \bot)}\,\mathcal{R}\rightarrow}{(\quad \vdash (\theta \wedge \bot) \rightarrow \theta)} \qquad \dfrac{\dfrac{}{(\bot \vdash \theta \wedge \bot)}\,\bot +}{(\quad \vdash \theta \rightarrow (\theta \wedge \bot))}\,\mathcal{R}\rightarrow}{(\quad \vdash (\theta \wedge \bot) \leftrightarrow \theta)}\,\mathcal{R}\wedge\,.$$

We get $\theta \vee \top \Bumpeq \top$ by a similar derivation.

4. We find

$$\dfrac{\dfrac{}{(\theta \vdash \theta)}\,I \qquad \dfrac{\dfrac{}{(\varphi \vdash \varphi)}\,I \quad \dfrac{}{(\lambda \vdash \lambda)}\,I}{(\varphi, \lambda \vdash \varphi \wedge \lambda)}\,\mathcal{R}\wedge}{(\theta, \varphi, \lambda \vdash \theta \wedge (\varphi \wedge \lambda))}\,\mathcal{R}\wedge\,.$$

By repeated use of **Proposition 5.4.1 (2), (3)**, we get $\gg ((\theta \wedge \varphi) \wedge \lambda \vdash \theta \wedge (\varphi \wedge \lambda))$, so using $\mathcal{R}\rightarrow$, we get $\gg (\quad \vdash ((\theta \wedge \varphi) \wedge \lambda) \rightarrow (\theta \wedge (\varphi \wedge \lambda))$. Furthermore, we find

$$\dfrac{\dfrac{\dfrac{}{(\theta \vdash \theta)}\,I \quad \dfrac{}{(\varphi \vdash \varphi)}\,I}{(\theta, \varphi \vdash \theta \wedge \varphi)}\,\mathcal{R}\wedge \qquad \dfrac{}{(\lambda \vdash \lambda)}\,I}{(\theta, \varphi, \lambda \vdash (\theta \wedge \varphi) \wedge \lambda)}\,\mathcal{R}\wedge\,.$$

It follows that $\gg (\quad \vdash (\theta \wedge (\varphi \wedge \lambda)) \rightarrow ((\theta \wedge \varphi) \wedge \lambda)$, hence we get our desired result of $\gg (\quad \vdash ((\theta \wedge \varphi) \wedge \lambda) \leftrightarrow (\theta \wedge (\varphi \wedge \lambda))$, which proves that $(\theta \wedge \varphi) \wedge \lambda \mathrel{\ooalign{$\rightleftharpoons$}} \theta \wedge (\varphi \wedge \lambda)$ proving $(\theta \wedge \varphi) \wedge \lambda \rightleftharpoons \theta \wedge (\varphi \wedge \lambda)$ involves similar derivations using $\vee$ inferences in place of $\wedge$ inferences. $\square$

The last statement of the above lemma has some major use in the highlight theorem in this section to come. It basically shows that $\wedge$ and $\vee$ works associatively in a chain of propositions (as one would expect them to). It allows one to talk about propositions like $\phi_1 \wedge \ldots \wedge \phi_n$ and $\varphi_1 \vee \ldots \vee \varphi_m$ very eloquently without *really* having to care about the order at which the parantheses go since they end up all being interchangeable anyway. It will then be not terrible abuse of notation to talk about $\phi_1 \wedge \ldots \wedge \phi_n$ and $\varphi_1 \vee \ldots \vee \varphi_m$ like it was one proposition and not really an interchangeability-class of propositions.

So finally is the theorem that will allow one to convert all sequents into one big propositions, which in the next section will allow one to do powerful things like...embed $\mathscr{L}_S$ into $\mathscr{L}_I$ in order to establish that they are sound and complete with respect to one another.

Theorem 5.4.4. *Let $\phi_1, \ldots, \phi_n$ and $\varphi_1, \ldots, \varphi_m$ be sequents.* $(\phi_1, \ldots, \phi_n \vdash \varphi_1, \ldots, \varphi_m) \ll\gg (\quad \vdash (\phi_1 \wedge \ldots \wedge \phi_n) \rightarrow (\varphi_1 \vee \ldots \vee \varphi_m))$.

Proof. We find $(\phi_1, \ldots, \phi_n \vdash \varphi_1, \ldots, \varphi_m) \gg (\phi_1 \wedge \ldots \wedge \phi_n \vdash \varphi_1 \vee \ldots \vee \varphi_m)$ by repeated use of **Proposition 6.4.1 (2) (3)**, and $(\phi_1, \ldots, \phi_n \vdash \varphi_1, \ldots, \varphi_m) \gg (\ \vdash (\phi_1 \wedge \ldots \wedge \phi_n) \to (\varphi_1 \vee \ldots \vee \varphi_m))$ follows from $\mathcal{R} \to$.

Note that

$$\dfrac{\dfrac{\dfrac{}{(\phi_1 \vdash \phi_1)}\,I \quad \dfrac{}{(\phi_2 \vdash \phi_2)}\,I}{(\phi_1, \phi_2 \vdash \phi_1 \wedge \phi_2)}\,\mathcal{R}\wedge \qquad \dfrac{}{(\phi_3 \vdash \phi_3)}\,I}{\vdots}\,\mathcal{R}\wedge$$

$$\dfrac{(\phi_1, \ldots, \phi_{n-1} \vdash \phi_1 \wedge \ldots \wedge \phi_{n-1}) \qquad \dfrac{}{(\phi_n \vdash \phi_n)}\,I}{(\phi_1, \ldots, \phi_n \vdash \phi_1 \wedge \ldots \wedge \phi_n)}\,\mathcal{R}\wedge,$$

and a similar repetition of I and $\mathcal{L}\vee$ gives us $(\varphi_1 \vee \ldots \vee \varphi_m \vdash \varphi_1, \ldots, \varphi_m)$. We conclude for $\Phi := \phi_1, \ldots, \phi_n$ and $\Psi := \varphi_1, \ldots \varphi_n$ that

$$\dfrac{\dfrac{(\Phi \vdash \phi_1 \wedge \ldots \wedge \phi_n) \quad (\varphi_1 \vee \ldots \vee \varphi_m \vdash \Psi)}{(\Phi, (\phi_1 \wedge \ldots \wedge \phi_n) \to (\varphi_1 \vee \ldots \vee \varphi_m) \vdash \Psi}\,\mathcal{L}\to \qquad (\ \vdash (\phi_1 \wedge \ldots \wedge \phi_n) \to (\varphi_1 \vee \ldots \vee \varphi_m))}{\phi_1, \ldots, \phi_n \vdash \varphi_1, \ldots, \varphi_n}\,Cut.$$

□

6 Conclusion and Future Work

It was a real honor to be able to inquire in the field of logic and try to make what may have been a contribution however small. No reader should be entirely satisfied with what was presented. For although this is the last chapter, this is by no means the end when it comes to the direction that future research could go. In fact, there was a lot of ideas that unfortunately did not make it to this paper since they were not polished at this time. The end chapter will touch on a few of these sorts of ideas, hopefully clearing up some mystery while also bringing some inspiring questions to the reader.

6.1 Fixing Mistakes and Tightening Rigor

By no means were the proofs, as well as the general writing, very pollished. Some proofs for some theorems don't exist and there are a few mistakes in the proofs that unfortunately the writer had no time to remedy (although hopefully the all the proofs still communicate the "right idea"). It is in the writer's best interest to fix this up in the future.

Moreover, commutative diagrams would have been very nice.

6.2 Exploring More on "Deduction Class Equivalence"

"Deduction class equivalence" was a very late epiphany, and going back to the hastiness of some proofs, there is a lot that one might be able to tweek about Deduction Class Equivalence. The first thing one might wonder about is whether it is possibly still too strong of a condition, or even too weak of one. The second thing is working on the framework used to derive it. There was a lot of good machinery used that emphasized graph/category theory, but it might have not been very efficient. Finally, this treatment of equivalence in no way looked at natural transformations between functors and analyzed equivalence *up to natural isomorphism.*

6.3 Generalizing for Infinite Formula-Vectors

One interesting limitation of this theory is that the collections of formulas were limited to finite collections. There is definitely the machinery out there to extend the objects to infinite formula vectors. However, doing so might complicate the inductive process for a little while, especially if the logic in question was not compact, although most were.

6.4 Generalizing Consistency and Satisfiability

In case one didn't notice, not a peep was mentioned involving "consistency" and "satisfiability". It would definitely be a good idea to also extend a general notion of consistency among inference maps, and also of "satisfiability" where it fits, such as the valuation logic that was mentioned.

6.5 Exploring Intuitionistic Logic

This is a big one. While the Gentzen Calculus went fairly deep into the theory, intuitionistic logic in general has a propositional form whose theory was unmentioned due to the chapter being made on it not being quick enough. It is a rather unfortunate scenario and all apologies to any reader who was looking forward to a lot of intuitionistic logic. Here are few definitions involving intuitionistic systems that the writer had, for which he proposed is deduction class equivalent with the standard notion of intuitionistic logic. And again the language will be presented here for reference

Definition 6.5.1. The **Intuitionistic Propositional Language** denoted $\mathcal{L}_I$ is defined as follows

1. The class *Atom.* contains an element p_i, called a **Propositional Variable**, for each $i \geqslant 1$.

2. The connectives in $\mathcal{L}_I$ include the constant $\perp$ and the binary connectives $\rightarrow, \wedge$.

Furthermore, we shall further define the connectives $\neg, \vee, \wedge, \leftrightarrow, \top$ as shorthand notation for the following:

1. $\neg\phi := \phi \rightarrow\perp$

2. $\phi \vee \varphi := \neg\neg\neg\phi \rightarrow \varphi$,

3. $\phi \leftrightarrow \varphi := (\phi \rightarrow \varphi) \wedge (\varphi \rightarrow \phi)$,

4. $\top := \neg \perp$. △

Definition 6.5.2. The Logic $\mathscr{L}_I = \langle \mathcal{L}_I, \{\rightarrow I, \rightarrow E, \wedge I, \wedge E1, \wedge E2, \vee E, \perp\}\rangle$ is defined as the **intutionistic propositional logic**

The Logic $\mathscr{L}_{IO}$ called the **intuitionistic propositional ordinary logic** defined on the language $\mathcal{L}_O$ (as it was defined before) and has all inference maps previously mentioned in $\mathscr{L}_{CO}$, except *LEM* and *RAA*. △

It is a future work goal to show that these two logics are deduction class equivalent.

As for the semantics, an attempt at a valuation logic, i.e. a better suited truth table was proposed. Essentially the idea is weakening the requirements for the valuation mapping so that there are more of them, and moreover the conditions that assign truth are more complicated.

Definition 6.5.3. Let ϕ, φ be $\mathcal{L}_C$-formulas. A mapping $v\colon Prop \longrightarrow \{0,1\}$ is an intuitionistic valuation, and part of the valuation class Val_I, if

1. $v(\bot) = 0$,

2. $v(\phi \rightarrow \phi) = 1$,

3. if $v(\phi \rightarrow \varphi) = 1$, then $v(\phi) = 0$ or $v(\varphi) = 1$,

4. if $v(\neg\phi) = 0$ or $v(\varphi) = 1$, then $v(\neg\phi \rightarrow \varphi) = 1$,

5. $v(\phi \wedge \varphi) = min\{v(\phi), v(\varphi)\}$.

The logic $\mathscr{L}_{VI}$ on $\mathscr{L}_C$ generated by the class Val_I of of intuitionistic valuation mappings shall be called the **intuitionistic propositional valuation logic**. $\triangle$

Interestingly, a lot of the proofs in chapter 4 were designed to be extended more generally to intuitionistic logic. So it is a good sign for future endeavors, particularly with $\mathscr{L}_{VI}$.

6.6 Exploring Linear Logic

Believe it or not, this project was inspired by something that was not even close to making it to the paper. And that is linear logic. Some definitions I had shall be provided here

Definition 6.6.1. We shall define the linear language $\mathcal{L}_L$ as follows.

1. The class *Atom.* contains a positive variable p_i and negative variable $p_i^\perp$ for each $i \in \mathbb{N}$. By definition, we have $p_i^{\perp\perp} = p_i$, for each $i \in \mathbb{N}$. We shall call the operation $^\perp$ **Linear Negation**. Note that $^\perp$ is *not* a connective; rather is a convention for how atomic variables interact in the language.

2. The connectives in $\mathcal{L}_L$ include the binary connectives $\otimes, ⅋, \&, \oplus$ and the unary connectives $?, !$, and the constant proposition symbol $\top$. We shall call the $\otimes$ connective the **Tensor Product**, the ⅋ connective the **Tensor Sum**, the & connective the **Direct Product**, the $\oplus$ connective the **Direct Sum**, the ? and ! connectives the **"why not"** and **"of course" exponentials** respectively, and the $\top$ symbol the **Additive Truth**. Additionally, we shall extend linear negation symbol $^\perp$ to denote the following for any $\mathcal{L}_L$-formulas θ, φ:

1. $\theta^\perp \otimes \varphi^\perp := (\theta ⅋ \varphi)^\perp$,

2. $\theta^\perp ⅋ \varphi^\perp := (\theta \otimes \varphi)^\perp$,

3. $\theta^\perp \& \varphi^\perp := (\theta \oplus \varphi)^\perp$,

4. $\theta^\perp \oplus \varphi^\perp := (\theta \& \varphi)^\perp$,

5. $?\theta^\perp := (!\theta)^\perp$,

6. $!\theta^\perp := (?\theta)^\perp$. △

Definition 6.6.2. We shall define a logic $\mathscr{L}_L$ which uses the sequent metalanguage $\mathcal{M}_L$ of $\mathcal{L}_L$ with the relation symbol $\vdash$. The rules of inference shall be as follows for any propositional variables θ, φ and any sequents $\Gamma, \Delta, \Phi, \Psi$:

1. **Negative Interchangeability**

$$\frac{\theta, \Gamma \vdash \Delta}{\Gamma \vdash \theta^\perp, \Delta}\ {}^{\perp}1 \qquad \frac{\Gamma \vdash \theta, \Delta}{\theta^\perp, \Gamma \vdash \Delta}\ {}^{\perp}2$$

2. **Identity**

$$\frac{}{\vdash \theta^\perp, \theta}\ I$$

3. **Exchange**

$$\frac{\vdash \Gamma, \theta, \varphi, \Delta}{\vdash \Gamma, \varphi, \theta, \Delta}\ X$$

4. **Multiplicative Rules**

$$\frac{\vdash \theta, \varphi, \Gamma}{\vdash \theta \parr \varphi, \Gamma} \parr \qquad \frac{\vdash \theta, \Gamma \quad \vdash \varphi, \Delta}{\vdash \theta \otimes \varphi, \Gamma, \Delta} \otimes$$

5. **Additive Rules**

$$\frac{\vdash \theta, \Gamma \quad \vdash \varphi, \Gamma}{\vdash \theta \mathbin{\&} \varphi, \Gamma} \&$$
$$\frac{\vdash \theta, \Gamma}{\vdash \theta \oplus \varphi, \Gamma} 1\oplus \qquad \frac{\vdash \varphi, \Gamma}{\vdash \theta \oplus \varphi, \Gamma} 2\oplus$$
$$\frac{}{\vdash \top, \Gamma} \top$$

6. **Exponential Rules**

$$\frac{\vdash \theta, ?\Gamma}{\vdash !\theta, ?\Gamma} ! \qquad \frac{\vdash \Gamma}{\vdash ?\theta, \Gamma} \mathcal{W}? \qquad \frac{\vdash ?\theta, ?\theta, \Gamma}{\vdash ?\theta, \Gamma} \mathcal{C}? \qquad \frac{\vdash \theta, \Gamma}{\vdash ?\theta, \Gamma} \mathcal{D}?$$

△

And here, the writer stayed true to his style of defining the language in the fewest connectives possible and deriving the others.

Definition 6.6.3. We shall further introduce the symbols $\multimap, \bot, \mathbf{1}, \mathbf{0}$ to denote the following:

1. $\theta \multimap \varphi := \theta^{\bot} \parr \varphi$,

2. $\mathbf{1} := !\top$, $\bot := \mathbf{1}^{\bot}$,

3. $\mathbf{0} := \top^{\bot}$,

where θ and φ are $\mathcal{L}_L$ formulas. △

In short, linear logic has *a lot* going on, that even experts in the field don't completely understand. Looking at the inferences in terms of its meta-sequent language the way that was done with intuitionistic sequents alas did not behave nicely with this metaframework. However, a valuation class that is complete with this logic is in the works. But since that would be cutting short everything else in this project that was already cut short, the exploration of whether that would work was quickly abandoned.

There was quite a bit of inquiry of linear logic in this project that unfortunately didn't make it to this paper. But hopefully, future inquiry will be made

6.7 Exploring More Languages

The penultimate aspect of future work is that this gave quite a few examples of languages in the text, but alas there was not a lot of languages that were explored in full depth. In the future, hopefully more languages will be explored.

6.8 Exploring Predicate Logic

Lastly, how exactly this works for predicate logic really is a question that is the basis for a lot of research. Particularly, how would the so-called "logical axioms" fall into the mix, as that definitely has its impact on what it means for two logics to be deduction class equivalent, and for that matter, sound and complete. And how would quantifiers be incorporated in the languages and logics that do? Moreover, how could valuations be applied in a way that makes a system of logic that corresponds to possible models of a language. I went about trying that in on my example of classic predicate semantics. But that by no means does it *complete* justice.

Bibliography

[1] Ethan D. Bloch, *The Real Numbers and Real Analysis*, Springer, London, 2011.

[2] ———, *Proofs and Fundamentals–Second Edition*, Springer, New York, 2011.

[3] Dirk van Dalen, *Logic and Structure–Fourth Edition*, Springer-Verlag, Germany, 2004.

[4] David S. Dummit and Richard M. Foote, *Abstract Algebra–Third Edition*, John Wiley and Sons, United States, 2004.

[5] Jean-Yves Girard, *Proofs and Types*, Cambridge University Press, Cambridge, UK, 1989.

[6] J. Lambek and P.J. Scott, *Introduction to Higher Order Categorical Logic*, Cambridge University Press, Cambridge, UK, 1986.

[7] Saunders Mac Lane, *Categories for the Working Mathematician*, Springer-Verlag, New York, 1971.

[8] Robin J. Wilson, *Introduction to Graph Theory–Third Edition*, Longman, New York, 1985.

[9] The Free Encyclopedia Wikipedia, *Von NeumannBernaysGdel set theory*, `https://en.wikipedia.org/wiki/Von_Neumann-Bernays-Godel_set_theory`.