Implicit Programming and the Logic of Constructible Duality

By

Anna L. Patterson

B.S., Washington University, 1987

Thesis

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign,1998

Urbana, Illinois

ABSTRACT

We present an investigation of duality in the traditional logical manner. We extend Nelson's symmetrization of intuitionistic logic, constructible falsity, to a self-dual logic — constructible duality. We develop a self-dual model by considering an interval of worlds in an intuitionistic Kripke model. The duality arises through how we judge truth and falsity. Truth is judged forward in the Kripke model, as in intuitionistic logic, while falsity is judged backwards.

We develop a self-dual algebra such that every point in the algebra is representable by some formula in the logic. This algebra arises as an instantiation of a Heyting algebra into several categorical constructions. In particular, we show that this algebra is an instantiation of the **Chu** construction applied to a Heyting algebra, the second Dialectica construction applied to a Heyting algebra, and as an algebra for the study of recursion and corecursion. Thus the algebra provides a common base for these constructions, and suggests itself as an important part of any constructive logical treatment of duality.

Implicit programming is suggested as a new paradigm for computing with constructible duality as its formal system. We show that all the operators that have computable least fixed points are definable explicitly and all operators with computable optimal fixed points are definable implicitly within constructible duality.

Implicit programming adds a novel definitional mechanism that allows functions to be defined implicitly. This new programming feature is especially useful for programming with co-recursively defined data-types such as circular lists.

DEDICATION

To my cousin Jordan Lackey (1963-1995) whose courage with AIDS was an inspiration.

ACKNOWLEDGMENTS

In way of background, my first course-work was performed at the University of Illinois Champaign-Urbana and after passing the PhD admissions exam I promptly took a two year intermission. After deciding to return, Gul Agha, my first supervisor, was on sabbatical. I was invited to spend that time at Stanford by Carolyn Talcott. The following summer I started working for Vaughan Pratt who became my advisor.

The people that funded my graduate work, (in order of contribution): Vaughan Pratt, John McCarthy, Gul Agha, Mike Genesereth, Carolyn Talcott and Natarajan Shankar, deserve first mention.

I would like to thank my advisor Vaughan Pratt for teaching me to dream and to strive for theorems beyond my current abilities so that I would have a constant rewarding struggle to learn more. I would like to thank Carolyn Talcott for inviting me to Stanford and letting me share her office whilst tirelessly answering questions. I cannot thank Grisha Mints enough for teaching me most of the logic I know.

My family, Patricia and John Patterson, Kent Patterson, Gail Brown and Pollyanna Askew have been my constant source of support, inspiration and love.

I'd like to thank Rob van Glabbeek for listening to practice talks, reading papers and particularly his help on the bisimulation papers. I'd like to thank John McCarthy for his many gifts of books such as Birkhoff's lattice theory and for kibitzing.

I'd like to thank Gul Agha, Herbert Edelsbrunner and Uday Reddy for their work on my committee.

Thanks for the stimulating environment, support and help by Vaughan's summer visitors and students, Gordon Plotkin, Dusko Pavlovic, Carolyn Brown, Rick Blute, Hongde Hu and Vineet Gupta.

My thanks to Sol Feferman for his encouragement during my logic seminars and lunch presentations that he attended, Natarajan Shankar and Pat Lincoln for the same at SRI lunches and Johan van Benthem for his infectious enthusiasm and for organizing the inspiring CSLI Workshops.

My thanks to Ian Mason and Albert Visser for early feedback on a variety of papers; David Cyrluk for teaching me PVS; John Mitchell and Valeria de Paiva for encouragement; Matthew Hennessy for pointing out what a bad idea my first thesis topic was; Robin Milner for pointing out what a poorly formed idea the second thesis topic was[1]; Dave Liu for early involvement; T.K. Lakshman my U. of I. mentor.

My thanks to my supportive and fun friends, Ulricke Axen, Charles Lewis, Kathleen Fisher, Caroline Hayes, Michael Dorniech, Chris Fisher, Chris Houck, Viviana Bono, Neil Crellin, Stuart Cheshire, Pavani Diwanji and E. LaFonte; and administrative staff Barb Cicone, Laura Kenny, Julie Legg and Maria Bharwada.

Lastly, but not at all least, I would like to thank Tom Costello for his love, support, encouragement, reading papers, co-authoring papers, reading drafts of the thesis, listening to pre-practice talks and cuddles.

[1] This is my third thesis topic.

PREFACE

> "It is an old maxim of mine that when you have excluded the impossible, whatever remains, however improbable, must be the truth." by Sherlock Holmes in *The Adventure of the Beryl Coronet* by Sir Arthur Conan Doyle

We present a semantics for programming languages which constructs outputs from inputs based on the maxim above. Usually, a program calculates an output from an input by solving successively smaller problems. To instead return the unique output for each input, we need a gestalt programming method. We need a mechanism for determining for each input the implicit output of a program. We do this by eliminating the impossible outputs and then returning whatever remains. To carry out this eliminative reasoning, we need a notion of an impossible computation. We study a negation for computation and propose a logical framework which calculates the *implicitly* defined output for each input.

Logical reasoning starts from some basic facts (or propositions) and from these facts draws conclusions through sound inferences. A sound inference is drawing conclusions that necessarily follow from the premises. Underlying this treatment is the notion of inference. We must reason about what is true given that something else is true. Why we can say that anything is "true" has been an argument of great debate through the ages. Some philosophers will not accept *observable facts* as "true" but for the moment let us say that "snow is white" is true because it is an observable fact. This brings up what we do with propositions that are contrary to observable fact such as "snow is black". We call such propositions "false". In fact, in a quite real sense we can be more sure of the falsity of "snow is black" than the truth of "snow is white" because we have seen snow that is not black on the other hand we have not seen all of the snow.

We can build a bias into our language by choosing some predicate to be primitive and another to be derived. If the natural numbers had only the predicate $<$, such that $4 < 5$ were "true", then $5 < 4$ must be "false". Our choice of predicates can build bias into our view of truth and falsity because we could have easily chosen $>$ to be primitive and have $<$ be derived. Therefore, we have some sort of symmetry about what we can chose as elementary.

These conclusions that we draw from "true" or "false" propositions can be seen as a formal argument. These formal arguments have their roots in ancient Greece. However, more than formal arguments, these proofs themselves can be seen as a method for constructing an argument. This view of proofs dates from early geometers drawing figures in the sand. An early "proof" was a formal argument given by a geometrical construction. Logic had as its underpinnings the notion of a construction.

Brouwer (Brouwer 1928) argued for the removal of any non-constructive proofs from logic. Intuitionistic (or constructive) logic is the formalization of this idea by Heyting(Heyting 1966). Intuitionistic logic has as its basis a constructive proof, or a constructive demonstration. Since we can make a construction which assures us that 4 is less than 5 and another construction that assures us that 3 is less than 4 then we can combine these constructions to to get a construction that 3 is less than 5. In general if because of a fact A we should believe a fact B, then Brouwer argues this should mean that given the proof of A we should be able to construct the proof of B. The notion that a construction fails is that the success of a construction would lead to absurdity or a contradiction with a known truth. Absurdity then is either a unreasonable or unsound conclusion.

Once we take on the Brouwerian view that proofs should be constructions, both negation and "falsity" disappear because absurdity is not the same thing as demonstratively false. This is because a construction leading to a contradiction does not mean that we can provide a counterexample. Let us look at a simple example of this phenomena. If a proof in geometry that two points are equal leads to a contradiction, does that mean we know the location of each of the points so that we can draw a line which connects these two distinct points? The answer with intuitionistic logic is no. The only negation offered us is the failure of a construction.

In intuitionistic logic we have taken "true" to be primitive as well as "absurdity". This has its roots in Aristotle (Smith 1996) who argues that the correct distinction to be made in an argument is the distinction between *substance* and *accident* instead of "truth" and "falsity". We have "proofs" and "proofs from absurdity" as the basic notions. Thus, a "proof leading to absurdity" is a derived notion of falsity and the only one afforded us in intuitionistic logic.

Negative information can be just as constructive as positive information. We can make a construction which assures us that 5 is not less than 4 and another construction that assures us that 4 is not less than 3 then we can combine these constructions to to get a construction that 5 is not less than 3. In general if because of a counterexample to A we should believe a counterexample B, then given the counterexample of A we should be able to construct the counterexample of B. Thus the correct way to use negative information in a constructive setting would be to do the "opposite" or "backward" construction in some way.

In logic we can see this "running backwards" in the contraposition of a logical statement. That is if the truth of B follows from the truth of A then the falsity of A follows from the falsity of B. This is the simplest sense in which negative information can be seen as reversing a construction.

A better analogy would be from Lewis Carroll's "Through the Looking Glass" (Carroll 1872), "what is left is now right", "what is up is now down" and reasoning runs backwards in that the "sentencing is before the trial". We saw a glimpse that falsity will run backwards by looking at the contrapositive of a formula. However, reasoning with just falsity requires reasoning in Carroll's mirror world and brings up a central issue in this thesis – the notion of duality. This mirror world of Carroll's is what we will call reasoning in the dual world.

Now we consider duality in one of its simplest forms that is interchanging points with lines. In Euclid's elements book IV (Euclid 300 BC Translation 1908), inscribing and circumscribing polygons were regarded as dual operations and beyond Proposition 11 and 12, circumscribing a polygon was simply a corollary to inscribing the polygon. The dual construction was obtained from the original. It seems then that it is the construction itself that seems to stand alone without regard to whether it inscribes a polygon or whether it is used to create the dual construction which circumscribes a polygon.

Archimedes used this construction/dual construction approach to approximate the perimeter of a unit circle to determine the number π. The inscribed polygon gives a lower bound and the circumscribed polygon gives an upper bound. In the limit, as we add more sides to our polygon the perimeter of the inscribed polygon approaches the perimeter of the circle and similarly for the circumscribed polygon. In the limit, either one is a perfect approximation. As an approximation technique knowing both bounds is better than knowing only one. We use this example to motivate the use of constructive reasoning in logic with both *truth* and *falsity*.

Consider the following statements, "the value of x is 5", "the value of x is *not* 5", and "x being 5 is absurd". The statement x is *not* 5 means that we know the value of x and it is not

5. The last statement means that if x were 5, then we could derive a contradiction. This last (intuitionistic) negation does not need an actual value of x.

Again, is there a need for any more? Should we be allowed to mix these notions? Should we allow constructions to interact with their duals?

Consider the "Hound of the Baskervilles" by Sir Arthur Conan Doyle (Doyle 1902). This is a Sherlock Holmes mystery. Like most mysteries there is a crime, in this case it is a murder. As with many mysteries, it is full of curses and superstitions on which the ill fate of Sir Charles was blamed. However, Sherlock Holmes began a list of suspects. Knowing that someone hated Sir Charles could land them on the suspect list. (Something true leading to another truth.) Knowing someone was not at the murder scene would remove them from the list. (Something false leading to another falsity.) How can one know that someone was not at the murder scene? How can one make such a negative inference? In this case, as in many, the existence of an alibi. The heir had an excellent alibi since he lived in Canada and was still in Canada at the time of the murder and the murder took place in England. Therefore the existence of an alibi (a truth) leads to the conclusion that the heir should not be on the suspect list (a falsity). Finally, should a falsity beget a truth? The key to this particular mystery lies in the fact that "the dog did not bark". This absence of fact led Sherlock Holmes to know the killer was someone that the dog knew and led him to know that Mr. Stapleton was the murderer.

Lastly and even more controversially, should the presence of a fact force the absence of that fact in the dual? The fact that Alice exists both in the world and the mirror world in "Through the Looking Glass" should stupefy us. Do opposites always nullify each other? We cannot think of a proposition being simultaneously true and false, but can we think of a property and its opposite being simultaneously present? The following is a famous example of such a state.

Title: For Terena	Translation
Odi et amo,	I hate her and I love her too,
quare id faciam,	you ask how this could be,
fortasse requiris nescio,	I only know that it is true,
sed fieri sentio et excrucior	and bitter agony

Catullus(c82-c54B.C.)

However, more concretely let us return to projective geometry where a counterexample to two points being equal is a line. Consider a square; the negation of every line is a point and the negation of every point is a line. This turns the square into its logical and geometric dual the diamond. This transformation preserves ratios thus the square is projectively equivalent to its dual. The presence of a property does not force the absence of that property in the dual because some structures are self-dual. Reasoning with and about self-dual structures is the motivation for the logic of constructible duality.

TABLE OF CONTENTS

Chapter

1 Introduction . . . 1
- 1.1 Negation as Reversing Direction . . . 4
- 1.2 Interval-Models . . . 5
- 1.3 $H \times H^{op}$ . . . 6
- 1.4 Relationship to Categories and Linear Logic . . . 9
- 1.5 Implicit Programming . . . 11
- 1.6 Summary . . . 15

2 Intuitionistic Logic — Technical Background . . . 16
- 2.1 Introduction . . . 16
- 2.2 Axiomatics of Intuitionistic Logic . . . 17
  - 2.2.1 Hilbert Rules . . . 18
  - 2.2.2 Gentzen Rules . . . 18
  - 2.2.3 Natural Deduction Rules . . . 19
- 2.3 Discussion about Intuitionistic Logic . . . 20
- 2.4 Kripke Structures for Intuitionistic Logic . . . 21
- 2.5 First Order Intuitionistic Logic . . . 24
- 2.6 Discussion about Intuitionistic Model Theory . . . 25
- 2.7 Heyting Algebras for Intuitionistic Logic . . . 27
- 2.8 Computational Interpretation . . . 29
  - 2.8.1 Curry-Howard for Intuitionistic Logic . . . 30
  - 2.8.2 Operational Semantics . . . 35

3 Constructivity with Negative Information . . . 38
- 3.1 Summary . . . 38
- 3.2 History . . . 39
  - 3.2.1 Axiomatics . . . 39
  - 3.2.2 Models . . . 39
  - 3.2.3 Algebras . . . 40
- 3.3 Interval Kripke Structures for Constructible Falsity . . . 42
  - 3.3.1 Completeness . . . 47
- 3.4 Alternate Formalization . . . 53
- 3.5 Elementary Equivalence with Partial World Models . . . 54
  - 3.5.1 From Partial World Models to Interval Models . . . 55
  - 3.5.2 From Intervals to Partial Worlds . . . 56

3.6 First Order Case 58

4 Constructible Duality 62
4.1 Summary 62
4.2 Introduction 63
4.3 Propositional Intuitionistic Logic with Constants 63
4.4 Adding a Duality Operator 64
4.5 Kripke Structures for Propositional Intuitionistic Logic 65
4.6 Interval Kripke Structures for Constructible Duality 67
4.6.1 Completeness 69
4.7 Alternate Formalization 74
4.8 First Order Case 76
4.9 Comparison with Relevance $*$-Models 79
4.9.1 Logical Comparison to CD 81
4.9.2 Model Theoretic Comparison to CD 82

5 Algebra 84
5.1 Introduction 84
5.2 An Algebra for Constructible Duality 85
5.3 Restriction to an Algebra for N 92
5.4 Significance and Relevance 93

6 Relationship to Linear Logic 95
6.1 Summary 95
6.2 Background 95
6.3 Logics and their Algebras 96
6.3.1 Classical Logic 97
6.4 The Story of the Exponentials 97
6.4.1 Adding *Overdefined* 99
6.4.2 Adding *Undefined* 99
6.4.3 Completing the Algebra $H \times H^{op}$ 100
6.5 A New Exponential 102
6.6 The *Undefined* Exponential 104
6.7 Linear Logic 106
6.8 Relationship to the Dialectica 116
6.9 Conclusion 118

7 Implicit Programming 119
7.1 Fixed Points 120
7.1.1 Optimal Fixed Points 120
7.2 Basic Recursion Theory 123
7.2.1 Fixed Points 124
7.3 The Curry-Howard-De Bruijn Isomorphism 127
7.3.1 Definition of Formulas as Types in Constructible Duality 128
7.3.2 About the Axiomatization 134
7.3.3 Operational Semantics 138
7.4 Strong Normalization 145

7.5 Constructivity of Axiomatization . . . 151
7.6 Translations . . . 155
7.6.1 Discussion . . . 166
7.7 Examples . . . 167
7.8 Conclusion . . . 169

8 Technical Summary . . . 171
8.1 Constructive Duality . . . 171
8.2 The Model Theory . . . 172
8.3 The Exponentials . . . 173
8.4 The Algebra . . . 173
8.5 Linear Logic . . . 173
8.6 Computational Interpretation . . . 174
8.7 Implicit Programming . . . 174

Bibliography . . . 175

Vita . . . 183

Chapter 1

Introduction

Theoretical computer science has strong ties to constructive logic. The best known relationship is probably the Curry-Howard isomorphism, which relates computations in the λ calculus to certain transformation of proofs called cut-eliminations. These two research programs developed independently—the λ calculus was developed by Church and his students Kleene and Rosser, originally to provide a new bases for logic, but later it was mainly applied in the study of computation, meanwhile cut-elimination was developed by Gentzen to prove the consistency of arithmetic by finitary means. Thus both strands were developed by logicians. However, they developed along very different paths. The ideas of the λ calculus were adopted by computer science early on in the design of languages. This introduced the idea of defining functions by recursion, and computing the values of functions by expanding out their definitions, using a stack. Almost all computer languages have adopted recursion as a basic way of defining functions, and have used the idea of evaluating functions by expanding their definitions using a stack.

The tool of cut-elimination was used in a very different domain. As well as its use in analyzing the consistency of logics, it became an important tool in understanding logics. Cut-elimination shows that a certain rule, Gentzen's Cut rule, can be omitted without changing the derivable theorems. The Cut rule is the only rule that "forgets" information. Often if a proof is carried out using just the other rules, then a counter-model of an underivable formula can be read off from a failed proof of the formula. Kripke models have a form which is easily read off from such a failed proof and are the basic tool in the construction of models for non-classical logics.

In this thesis we re-examine a notion that was present earlier in one of these traditions. Negation, or the idea that for every property there is an opposite, appears in early constructive logic. Both Heyting, and later Nelson considered a constructive negation.

Strangely however, computer science did not embrace constructive negation. The reason for this is computer science's emphasis on constructivity. Computer users are not interested in knowing that a variable *doesn't* have a value, they wish to know what the value *is*.

It might seem that negation does not have a role in constructing values, the business of computer science. This thesis will deny this notion, showing that negation is a valuable tool for constructing values. However, to explain how this comes about, we first need to better understand negation from the logical point of view.

Once we better understand the logical properties of negation, we shall be able to understand how it is used to compute values. Understanding proofs helps us in understanding programs because of the following correspondence. Certain logics have the property that whenever a value can be shown to exist, the value itself can be produced (by cut-elimination). The standard logic with this computational interpretation is intuitionistic logic. In this thesis, we shall approximate computation with intuitionistic logic because of this computational, or constructive nature of intuitionistic proofs.

However, intuitionistic logic does not have the usual notion of falsity. Rather, it only allows for statements being true or contradictory. This is why we need to extend our logical understanding of negation before we consider its applications to computer science.

We begin our study by presenting a proof theory for a constructive negation, called constructible falsity (Nelson 1949), that seamlessly extends the ideas of intuitionistic logic to allow for constructions of counterexamples. This provides a notion of computing a counterexample and of computing with negative information.

Then we consider a new model theory of constructible falsity. Although there were previous models of constructible falsity, they did not have the inherent symmetry of the logic. We provide an interval-model for constructible falsity for which we show that Nelson's proof theory is both sound and complete. Furthermore, the new interval-models we propose have the same symmetries as Nelson's logic.

The next tool we use to understand negation is algebra. We look at the syntactically generated term model for intuitionistic logic. These are Heyting algebras H. Again intervals formed over Heyting algebras provides a sound and complete algebra for constructible falsity. These interval-algebras give a sort of syntactic model for constructible falsity — but not quite. We notice that the structure of these interval-algebras are not fully fleshed out. There are some intervals which are not generated by any term in the language of constructible falsity.

We cover the space by adding to the logic of constructible falsity exponential operators which defines the logic of constructible duality. Adding these exponential operations is equivalent to adding two constants.

Finally, we study the relationship our newly proposed logic has to linear logic, a logic which has recently generated strong interest among theoretical computer scientists. We study the relationship between constructible duality and linear logic and study the relationship between the algebra of constructible duality and other models studied in computer science.

Once we have understood negation logically, all that remains is to apply our insight to computer science.

We develop a new programming language paradigm that depends upon a negation for computations, implicit programming, and show its benefits. It allows us to more naturally represent certain programs that previous methods. This is because implicit programming corresponds to a different fixed point than the widely accepted previous approach. Often, implicit programming corresponds to the greatest fixed point (when one exists) or the optimal fixed point (the greatest consistent fixed point). These solutions are often more natural, and more informative than the least fixed point.

The least fixed point can be understood as deriving the value of a function from the values of other functions. Our new approach calculates the value of a function from facts about what the values other functions **don't** have. Thus, our reasoning uses negative information rather than positive information. Capturing this negative information needs a negation.

This new paradigm, implicit programming, adds a new tool to the programmer, as it gives another way of defining functions, rather than saying what the value must be, they can say what the value should not be. For some problems, especially those with cycles, this is more natural. Cyclic lists are different, when there is a finite path on which they differ, it is not so natural to state when they are equal.

1.1 Negation as Reversing Direction

We first consider one of the very basic properties of negation, it reverses direction.

A statement in classical propositional logic such as $\neg p \vee \neg q$ can be evaluated by knowing the truth value of p and the truth value of q. For a propositional statement to be valid, it must be true under all assignments of values to the propositional atoms. This means that the statement is true in all contexts or universally true.

Some statements have a weaker dependence on absolute truth. They allow for some truths to be "unknown", but once we have established that they are true, they remain true for evermore. If we prove a theorem, it stays true, we need not worry that further developments will deny it. The statements of intuitionistic logic have some of this flavor. The model theory by Kripke (1965) models how mathematicians gather theorems over time; it builds into the logic this belief that statements, once validated, will never be denied.

The classical statement reflects all points in time, whereas the intuitionistic statement reflects the state of affairs now and in the future.

Therefore the semantics of intuitionistic logic captures the tendency of facts to accrue over time. It does this by having more than one *world*. The standard semantics of intuitionistic logic is given in terms of an order on states of the world. If a formula is **true** then it is true at the current state of the world and in all future states. If a formula is not true, then it may become true in some future state of the world e.g. just because the question of whether MELL is decidable or not (the fragment of linear logic containing the multiplicatives and the exponentials) is not known now (on October 20, 1997), does not mean that it will never be known.

The only notion of intuitionistic **falsity** is that some statement leads to contradiction. This is different from having a counterexample of a statement. This intuitionistic negation merely states that a statement could never be true in the future, not that we will ever be able to firmly establish that it is not the case.

If we look at the kind of statements that can be considered in intuitionistic systems, we see that they are statements that persist. "Napoleon died", "the Dodo is extinct", and "Fermat's last theorem was proved" are all of this type. If we look that the negation of statements of this type we see they do not persist into the future. There was a time when Napoleon had not died, the Dodo was alive, and Fermat's last theorem was an open question, but these propositions are no longer the case. Therefore, this reading of counterexamples suggests that they do not persist into future, because we grow in knowledge. However, as we grow in knowledge, we shrink in counter-examples, thus counterexamples persist into the past.

1.2 Interval-Models

These intuitions of counterexamples persisting into the past led us to model counterexamples, or the negations of intuitionistic propositions, as looking backwards within models for intuitionistic logic. However, if we look for both truths and counterexamples in the same state of the world, then at every state either MELL is decidable or we must have a counterexample that MELL is not decidable. To get the indeterminacy afforded us in intuitionistic logic we need to separate the point from which we judge truth from the point where we judge falsity.

We recall that a counterexample tells us about the past, while an intuitionistic statement tells us about the future. Thus, if we wish to have both facts and counterexamples about the same time, we must judge our counterexamples from a time in its future, and our facts from a time in its past.

Using this intuition that falsity should be judged from picking a world in the future and looking backwards, while truth, as is normal is judged forwards, we generate a new model theory for constructible falsity, denoted N. Earlier, Thomason (Thomason 1969) proposed an ordering on worlds with three truth values **true, false** and **undecided** for which he shows that N is sound and complete, c.f. similar models by (Gurevich 1977, Routley 1975). Models with multiple viewpoints are used in relevant logics to avoid the rule of weakening c.f. (Hautamäki 1982) for a generalization.

We give an ordering on worlds having only two truth values **true** and **false** for which we show that N is sound and complete. Truth is defined on intervals (w_t, w_f) such that w_f is accessible from w_t. Truth is judged from one end w_t, falsity from the other w_f. A formula φ holds at an interval if and only if it holds at all proper subintervals. We interpret strong negation as reversing the accessibility relation and reversing the role of **true** and **false**.

There is an obvious three valued model for N. In this three valued model, the third value is assigned to those propositional atoms which are assigned 0 at w_t and 1 at w_f. We show these interval models give the proper meaning to the proof theory of N in that the proof theory is also *sound* and *complete* with respect to these interval models.

To establish completeness, we actually show that any set of formulas which is formally consistent in the deductive system of N is satisfied by some countably infinite structure in which any underivable formula in N is refuted. In fact, the model which refutes every underivable formula for positive intuitionistic logic is in some sense the model which refutes every underivable formula in N.

Since both the three valued models and the interval models are sound and complete, it suggests that there must be a translation between them. We present such a translation and show that it preserves validity.

1.3 $H \times H^{op}$

A canonical model for a proof theory is a model which satisfies all valid formulas and refutes every invalid one. We can re-use the canonical model for positive intuitionistic logic to obtain a canonical model for N.

These canonical models are suggestive of free algebras. Free algebras are structures generated from a set of equations which break every equation which is not imposed as a consequence of the original set of equations. One way of generating a free algebra is to consider all expressions that are generated by a rule set. If a logic contains the $\wedge$ expression then $p \wedge q$ will be generated as well as $q \wedge p$. However in most logics these mean the same thing. A free algebra identifies two expressions which "mean the same thing". More formally, when φ implies ψ and ψ implies φ, written $\varphi \equiv \psi$, then we say that they are equivalent. Identifying equivalent expressions abstracts away from the details of the expressions and concentrates more on the structure.

However, algebras cannot just identify things which happen to agree in some contexts "The lawn is wet only when it is raining" may be the case for some environmentally-friendly communities and in other communities people water their lawn to keep it green year round. Therefore, this statement may be true locally, but not globally. In an algebra, we only want to identify expressions which are equivalent in all contexts i.e.have the *congruence* property.

Interderivability in N is not a congruence. This means it is more difficult to construct an algebra. Two formulas in N can only be identified in an algebra if they are *true* under the same conditions and *false* under the same conditions. This means that the congruence relation for the algebra of N can be made from $\varphi \equiv \psi$ and $\neg\varphi \equiv \neg\psi$, where $\varphi \equiv \psi$ means that from the truth of φ you can prove the truth of ψ, i.e. $\varphi \supset \psi$, and vice versa (Rasiowa 1974, Vakarelov 1977).

The algebra for N was first constructed by (Rasiowa 1974), who called it a quasi-pseudo boolean algebra, later dubbed N-lattices after Nelson. Later authors viewed this as a set of restrictions on two copies of the intuitionistic algebra H (pseudo boolean or Heyting algebras) (Vakarelov 1977, Sendlewski 1984, Sendlewski 1989, Sendlewski 1995). Roughly, we can think of one copy of H encoding what things are true when other things are true, and the other copy as encoding what things are false when other things are false. Then we need restrictions to glue these together, because when p is true, we don't want to glue this to a state of the world where p is false. If the restrictions go too far we wind up with symmetrization by Rauszer (Rauszer 1975, Rauszer 1977b, Rauszer 1977a) who introduces a negation $-$, but in her logic $A \vee -A$ is valid. Without refuting the law of the excluded middle, a logic cannot be constructive because proofs of existence will not construct their witnessing value.

What we did was notice that the restrictions (technically forms of boolean congruences) that Vakarelov, Sendlewski and Czelakowski (Vakarelov 1977, Sendlewski 1995, Czelakowski 1975) studied could be eliminated by using the interval idea again. Instead of two copies of the rules, one for truth and one for falsity, and a restriction to make them match up, we use one copy of the system for truth and its opposite for falsity. This gives us the structure of a Heyting algebra paired with its opposite — $H \times H^{op}$ where H is Heyting algebra.

However, sometimes we would like to be able to separate out concerns. Sometimes we want to extract information about just the truths and other times the falsities. If $H \times H^{op}$ is an algebra for a theory of interest in constructible duality, then the projection operation mapping $\pi_i\colon H \times H^{op} \mapsto H$ would provide us the structure for truth in that theory and $\pi_2\colon H \times H^{op} \mapsto H^{op}$ for falsity.

A product structure usually admits projection operations, an ungluing. If $H \times H^{op}$ is some algebra, we would like to get both H and H^{op}. In the picture below, we show a particular H and its opposite H^{op} as well as its product $H \times H^{op}$.

$$
\begin{tikzcd}
1 & 0 & & (1,0) & \\
0 \arrow[u] & 1 \arrow[u] & (1,1) \arrow[ur] & H \times H^{op} & (0,0) \arrow[ul] \\
H & H^{op} & & & \\
 & & & (0,1) \arrow[uul, "!"] \arrow[uur] &
\end{tikzcd}
$$

Any H has both a bottom element 0 for **false** and a top element 1 for **true**, therefore there are four possible projection operators.

$$
\begin{aligned}
&!\ H \times H^{op} &&\mapsto H \times 1 &&\cong H \\
&?\ H \times H^{op} &&\mapsto 1 \times H^{op} &&\cong H^{op} \\
&¿\ H \times H^{op} &&\mapsto H \times 0 &&\cong H \\
&¡\ H \times H^{op} &&\mapsto 0 \times H^{op} &&\cong H^{op}
\end{aligned}
$$

Applying, ! to $H \times H^{op}$ in our picture above gives us the structure $H \times 1$ annotated on the above diagram. The other exponentials behave as projections.

The first two projections have the same structure as exponentials in linear logic. We add these exponentials to constructible falsity to yield constructible duality. These exponentials enable constructible duality to represent every point in the space of every $H \times H^{op}$.

It may seem radical to add exponentials to our language, but how fundamental an operation is it? The linear logic-like exponentials can be obtained by adding the unit to the adjoint of the ordering operator in our pair algebra. This constant is called *overdefined*. The second pair of exponentials come from our other new constant *undefined* that is related to the third truth value in the Thomason-Gurevich models.

This algebraic structure $H \times H^{op}$ for any H is self-dual and represents a constructive logic, hence the name constructible duality.

1.4 Relationship to Categories and Linear Logic

Duality, symmetry, and categorical opposites play an increasing role in our understanding of fundamental structural properties of computation and have come to the forefront of logical investigations in computer science. Linear logic(Girard 1987), Co-recursion(Pitts 1994), the μ-calculus(Kozen 1983, Pratt 1980), and self-dual categories(Barr 1979, Pratt 1995) have attracted much attention. Here we consider many of these same structures, but we started from a logical perspective, where duality appears in the form of an operator, constructive negation and two constants, undefined and overdefined.

Recent investigations have focussed on powerful categorical tools, that provide great expressive power, such as the **Chu** construction (Barr 1979, Pratt 1995), the dialectica (de Paiva 1989a) (de Paiva 1989b) (de Paiva and Hyland 1993) categories, toposes (Klunder 1992) and special final coalgebra (Barwise and Moss 1996) (Pitts 1994) (Moss and Danner 1996) (Freyd 1990, Fiore 1996) constructions.

Chu spaces (Pratt 1995, Gupta 1994) have been studied as models of computation because they can describe topologies, event structures (Nielsen et al. 1981) (Glabbeek and Plotkin 1995), other models of concurrency and a model of linear logic (Pratt 1997). We show that our pair algebra $H \times H^{op}$ is isomorphic to applying the **Chu** construction to H and its terminal object.

$H \times H^{op}$ is also a special subcategory of the dialectica category **G**. The dialectica interpretation by Gödel gave semantics to intuitionistic mathematics in terms of higher order functionals. The **G** category is a simplification of this idea. The dialectica category **G** was shown to be a model of linear logic(de Paiva and Hyland 1993). Further a simple subcategory of **G** instantiated with H is isomorphic to $H \times H^{op}$.

This provides evidence that the algebra for constructible duality might be a model of linear logic with mix. We verify this conjecture directly in the thesis using (Seely 1989).

Further there is a correspondence in that any linear logic proof can be translated to a proof in constructible duality using the following table, where the calligraphic letters on the right indicate that the translation needs to be applied to those subterms.

Linear Logic	Constructible Duality
$\mathbf{0}$	**false**
$\top$	**true**
$\mathbf{1}$	**overdefined**
$\perp$	**overdefined**
$A \multimap B$	$\mathcal{A} \supset \mathcal{B} \ \wedge\ \neg\mathcal{B} \supset \neg\mathcal{A}$
$A \otimes B$	$\neg(\mathcal{A} \supset \neg\mathcal{B} \ \wedge\ \mathcal{B} \supset \neg\mathcal{A})$
$A ⅋ B$	$\neg\mathcal{A} \supset \mathcal{B} \ \wedge\ \neg\mathcal{B} \supset \mathcal{A}$
$A \& B$	$\mathcal{A} \wedge \mathcal{B}$
$A \oplus B$	$\mathcal{A} \vee \mathcal{B}$
$A^{\perp}$	$\neg\mathcal{A}$
$!A$	$\mathcal{A} \wedge$ **overdefined**
$?A$	$\mathcal{A} \vee$ **overdefined**

The table above represents a sound translation given from linear logic to constructible duality. In fact if A is *linear or balanced* multiplicative formulas then A is provable in linear logic if and only if $\mathcal{A}$ is provable in constructible duality. Furthermore, if A and $\mathcal{A}$ has a proof it is unique. This means that both linear logic and constructible duality will have the same coherence theorem but will differ in their extensions. Whereas linear logic does not extend to full completeness for any known category, constructible duality is fully complete for $C \times C^{op}$ where C is a bicartesian closed category which is fully complete for intuitionistic logic, c.f. (Cubric 1994) for such categories. Note: Given a bicartesian closed category, if we identify any two morphisms $f, g\colon A \to B$ between objects A and B then the bicartesian closed category becomes a Heyting algebra i.e. a Heyting algebra is the posetal version of a bicartesian closed category.

These comparisons are best expressed by examining the exponentials in constructible duality and the translations between the various logics given below.

$$
\begin{tikzcd}
& \text{Linear Logic} \arrow[dl, "\text{Some Structural Rules}"'] \arrow[dd, "!"] \\
\text{Constructible Duality} \arrow[d, shift right, "?"', "¡"] \arrow[dr, leftrightarrow, "!", "¿"'] & \\
\text{Dual Int. Logic} \arrow[u, hook, shift right] & \text{Intuitionistic Logic} \arrow[d, "\sim\sim"'] \\
& \text{Classical Logic}
\end{tikzcd}
$$

This diagram is not meant as a commuting digram but rather to show how one logic can be embedded in another, either by inclusion or some sort of ! translation.

1.5 Implicit Programming

Having better understood the role negation plays in constructive logic, we are ready to apply it to computer science.

We model recursive programs with the logic of constructible duality, *CD*, and obtain different solutions to recursive programs than traditional methods. This is because traditional methods only think in terms of positive information, while we also capture the negative information that current approaches discard.

If we define a partial function by an equation, such as,

$$\forall x. f(x) = 0 * f(x)$$

then this statement contains both positive and negative facts. To see this more clearly we translate this into classical logic using the graph of f, which we denote R_f,

$$\forall x, y.[R_f(x, y) \leftrightarrow \exists z.(R_f(x, z) \wedge y = 0 * z)].$$

If we translate this classical statement into constructible duality, we get four implications,

$$\left.\begin{array}{l}\forall x,y.[R_f(x,y) \supset \exists z.(R_f(x,z) \wedge y = 0 * z)] \\ \forall x,y.[\exists z.(R_f(x,z) \wedge y = 0 * z \supset R_f(x,y))]\end{array}\right\} \quad \text{positive}_1$$
$$\left.\begin{array}{l}\forall x,y.[\neg R_f(x,y) \supset \neg(\exists z.R_f(x,z) \wedge y = 0 * z)] \\ \forall x,y.[\neg(\exists z.R_f(x,z) \wedge y = 0 * z) \supset \neg R_f(x,y)]\end{array}\right\} \quad \text{negative}_1.$$

The statement we are interested in inferring is $f(x) = 0$. We write this in classical logic as,

$$\forall x, y.R_f(x,y) \rightarrow y = 0.$$

We translate this into constructible duality as

$$\begin{array}{ll}\forall x,y.R_f(x,y) \supset y = 0 & \text{positive}_2 \\ \forall x,y.\neg(y = 0) \supset \neg R_f(x,y) & \text{negative}_2.\end{array}$$

It is notable that the positive part of the statement of $\forall x.f(x) = 0 * f(x)$ in CD does not imply the positive part of the statement of $f(x) = 0$ in CD (i.e. $\text{positive}_1 \not\supset \text{positive}_2$), but the negative part of the statement of $\forall x.f(x) = 0 * f(x)$ does imply the negative part of the statement of $f(x) = 0$ i.e. $\text{negative}_1 \supset \text{negative}_2$.

Thus the negative information about recursive definitions can have content that is not present in the purely positive part of the definition. It is this extra negative information that implicit programming uses.

Recursion is an important way of specifying functions in programming languages. It corresponds to choosing a function to be the least fixed point of a recursive operator. The key insight that makes recursive programs useful is that this least fixed point is always computable, for a good introductory text c.f. (Cutland 1980). As is familiar in mathematics, functions can be defined by giving equations that the function satisfies. A popular example of this is the factorial function.

$$f(n+1) = (n+1) * f(n), \qquad f(0) = 1$$

There is only one solution for this function over the natural numbers. In this example, $f(2)$ can be calculated, through reducing the problem through the sequence $f(2) = 2 * f(1) = 2 * (1 * f(0)) = 2 * (1 * 1) = 2 * 1 = 2$ to a series of simpler problems.

This method works very naturally for functions on inductively defined data structures, that is, data structures that are defined to be the smallest set containing certain basic values, and are closed under certain operations. The natural numbers are the smallest set that contains 0, and is closed under the successor operation. The definition of the natural numbers and the definition of factorial are interpreted in a manner which guarantees a unique solution.

However, there is another natural class of data structures, those defined as the largest set of objects satisfying certain conditions. An example of structures defined in this way are lists in Lisp. Lisp contains a certain set of data structures called atoms. A list is a data structure that is either the empty list, NIL, or consists of a head, which is a list or an atom, and a tail, which is a list. From a list l, the head of the list is obtained through the operation $car(l)$ and the tail through $cdr(l)$. The set of lists is not the smallest set obeying these rules. For example, there is a list, Ω_1, which has as its head the atom NIL, and as its tail Ω_1 i.e. Ω_1 is (NIL, Ω_1).

Programs defined by least fixed points are not natural for data structures defined as the largest set of objects satisfying certain criteria. To see why, we look at the function that replaces every NIL by '(NIL).

```
(DEFUN DOUBLE-NILS (X)
       (COND ((EQ X NIL) '(NIL))
             ((ATOM X) X)
             (T (CONS (DOUBLE-NILS (CAR X))
                      (DOUBLE-NILS (CDR X))))))
```

If we try to calculate the value of (DOUBLE-NILS Ω_1), we find that we fall into an infinite regress. When calculation is performed via expanding the definition, the function does not terminate. However, there is a unique answer satisfying the above definition. It is the function that doubles every NIL i.e. '(NIL) := '(NIL NIL) in any list. Let us call Ω_2, $('(\text{NIL}), \Omega_2)$. The result of calling (DOUBLE-NILS Ω_1) should be Ω_2.

Is there a plausible interpretation? One possible answer concerns how we choose what function we will use when we are given a defining equation that has more than one possible solution. The solution we choose when we attempt to reduce the problem to simpler problems is the least defined function. That is, it is the function that is defined in as few places as possible. For a large class of equations, this *least fixed point* is guaranteed, but may not be the desired answer. An alternate solution, that is more natural when we look at data structures like lists, is to choose the function that returns an answer whenever it is defined uniquely. Under suitable conditions this unique answer is guaranteed to exist and be computable.

Naturally, we would like to be able to use both definitional mechanisms when we program. It would be unfortunate if we had to give up explicit definitions to get implicit definitions. To this end we propose a framework that captures both kinds of definitions, and even allows us to mix definitions (within reason).

The key point in extending languages to allow *implicit programs* as well as *explicit*, is to notice that computable implicit programs can be characterized "negatively". We cannot use the standard notion of implicit definition, as this gives far too many functions, not just the computable ones. For this reason a new form of definition, *constructive implicit definition* is suggested which uses our negation for computations.

To show that we can extract values from these proofs of implicit definitions, we introduce an axiomatization of Heyting arithmetic, that does not have decided equality, but which adds certain axioms, and a predecessor. We show that this is still complete for equations and inequations.

We then introduce a lambda calculus, whose operational semantics exactly mirrors the normalization of our proofs. We show that the operational semantics is weakly confluent and strongly normalizing.

We then show how to derive an operator from a set of equations. We finally show that from a proof of a strictly positive statement (i.e. a proof the function is defined) the value of the least fixed point of this operator can be extracted, while a proof of a strictly negative statement (i.e. a proof the function is defined uniquely/implicitly) the value of the optimal fixed point can be extracted.

This makes precise the claim that implicit programming can use the extra negative information that is in recursive definitions.

The ! exponential projects onto the strictly positive information giving rise to an alternate *positive* translation from a set of recursive equations. Thus ! gives rise to the least fixed point because the value of the least fixed point can be extracted from a proof of the positive translation of a set of recursive equations. The ¡ exponential projects onto the strictly negative information giving rise to an alternate *negative* translation; thus ¡ gives rise to the greatest fixed point.

1.6 Summary

In this thesis, we develop a new semantics for constructible falsity, that relies on duality. We carry this idea over to algebras, to find a new more natural algebra for constructible falsity, free of the cumbersome restrictions previously proposed. We extend the space of algebras and define exponentials which match the expansion. We call constructible falsity with exponentials the logic constructible duality because the negation of constructible falsity now acts as a duality operator.

We show that the algebra of constructible duality arises naturally in many categorical structures studied in computer science. This algebraic structure captures the essence of duality common to all these proposed frameworks.

This notion of constructive duality sheds light not just on the logical foundations of duality in computer science, but also on practical concerns. We show that this new constructive system allows a natural specification of programs. We can define a program implicitly using this logic, a form of definition that is more natural for many purposes.

Chapter 2

Intuitionistic Logic — Technical Background

This chapter contains the proof theory of intuitionistic logic, a model theory by Kripke and an algebraic treatment by Heyting. Lastly, we present the most important relationship of intuitionistic logic as it relates to computer science — the Curry-Howard-deBruijn isomorphism between reductions in typed-lambda calculus and proof normalization in intuitionistic logic.

All of these notions and definitions are important background as much of our work extends these definitions and often uses the same structures. In particular, the axiomatics of constructible duality extends that of intuitionistic logic, the model theory is based around intervals in Kripke's models of intuitionistic logic and the algebra is based around pairs of Heyting algebras. For this reason, we review some of the basic notions surrounding intuitionistic logic.

First, we give the informal account of the logic called the BHK (Brouwer - Heyting - Kolmogorov) interpretation of intuitionistic logic for those that may need intuition. The discussion following definitions attempts to give some intuition following the definitions of the axiomatics, the model theory and the algebra respectively.

2.1 Introduction

A constructive notion of logic was introduced by Brouwer(Brouwer 1928) who argued that current methods of reasoning in mathematics in the nineteenth century were invalid. He most famously rejected the principal of the excluded middle. Instead he claimed that the basis for mathematics was constructions. This led to the Brouwer-Heyting-Kolmogorov interpretation of logic. In Brouwerian logic the law of the excluded middle is invalid, that is $A \vee \neg A$ can be refuted. He argued that A should really stand for a "proof of A" and $\neg A$ should stand for a proof of contradiction from the assumption A. He also argued that implication should be viewed as a construction, that is $A \supset B$ should be a construction of B given A. This is not the same a classical implication, where $A \to B$ is equivalent to $\neg A \vee B$.

The Brouwer-Heyting-Kolmogorov(Kolmogorov 1932b, Heyting 1966, Kolmogorov 1932a) interpretation of intuitionistic logic introduces the notion of "realizability" or constructivity. If A implies B is to be valid constructively, then a proof of B must be a construction (or realizable) from a proof of A.

The term which we call the "realization" is some function which produces a proof. The "realization" is the construction. Let φ be a formula in logic. We write $c\,\mathbf{r}\,\varphi$ to denote "the function c which *realizes* φ". This realization of intuitionistic logic satisfies the following

$$
\begin{array}{lcl}
c\,\mathbf{r}\,(\varphi \wedge \psi) & \Longleftrightarrow & c \text{ is a pair } c = \langle a_0, a_1 \rangle \text{ where } a_0\,\mathbf{r}\,\varphi \text{ and } a_1 \mathbf{r} \psi \\
c\,\mathbf{r}\,(\varphi \vee \psi) & \Longleftrightarrow & c = k_0\,a \text{ and } a\,\mathbf{r}\,\varphi \text{ or } c = k_1\,a \text{ and } a \mathbf{r} \psi \\
c\,\mathbf{r}\,(\varphi \supset \psi) & \Longleftrightarrow & c \text{ is a function which for every } d \text{ if } d\,\mathbf{r}\varphi \text{ then } c(d)\,\mathbf{r}\psi \\
\text{not } c\,\mathbf{r}\,\mathbf{false} & &
\end{array}
$$

where k is the injector function. If k_0 is present then the realization is a proof of φ and if k_1 is present indicates that the realization is a proof of ψ. Therefore, not only does a proof of a disjunction carry with it a proof of one of φ or ψ, but the realization indicates which of these two formulas is proved. Further, the last clause not $c\,\mathbf{r}\,\mathbf{false}$ states that there is no function which realizes falsity.

2.2 Axiomatics of Intuitionistic Logic

Formulas $A, B, \ldots$ are formed from propositional atoms drawn from a set of propositional atoms P, by means of the binary connectives $\wedge$, $\vee$, and $\supset$, and the unary connective $\sim$.

2.2.1 Hilbert Rules

The Hilbert style rules for propositional intuitionistic logic (Kleene - Introduction to Meta-Mathematics p82 (Kleene 1952)) are repeated below where we use $\sim$ to represent intuitionistic negation.

$$
\begin{array}{lll}
A \supset (B \supset A) & (A \supset B) \supset ((A \supset (B \supset C)) \supset (A \supset C)) & \dfrac{A, A \supset B}{B} \\
A \wedge B \supset A & A \wedge B \supset B & A \supset (B \supset A \wedge B) \\
A \supset (A \vee B) & B \supset (A \vee B) & (A \supset C) \supset ((B \supset C) \supset (A \vee B \supset C)) \\
\sim \sim A \supset A & & (A \supset B) \supset ((A \supset \sim B) \supset \sim A)
\end{array}
$$

2.2.2 Gentzen Rules

The multiple-conclusion Gentzen sequent System for Propositional Intuitionistic Logic (Beth 1952)

Gentzen System Sequents $\Gamma \Rightarrow \Delta$ should be read $\wedge\Gamma \supset \vee\Delta$, where the set $\Gamma \cup \Delta$ is a finite set of formulas, Γ, A is the set Γ with the single formula A added. The rules are divided into left and right rules according to whether the main operator is introduced on the left or on the right (respectively).

Axioms $(1)\Gamma, A \Rightarrow \Delta, A$

Rules of Inference

$$
\begin{array}{llll}
\Rightarrow \wedge & \dfrac{\Gamma \Rightarrow \Delta, A \quad \Gamma \Rightarrow \Delta, B}{\Gamma \Rightarrow \Delta, A \wedge B} & \wedge \Rightarrow & \dfrac{A, B, \Gamma \Rightarrow \Delta}{A \wedge B, \Gamma \Rightarrow \Delta} \\[2ex]
\vee \Rightarrow & \dfrac{A, \Gamma \Rightarrow \Delta \quad B, \Gamma \Rightarrow \Delta}{A \vee B, \Gamma \Rightarrow \Delta} & \Rightarrow \vee & \dfrac{\Gamma \Rightarrow \Delta, A, B}{\Gamma \Rightarrow \Delta, A \vee B} \\[2ex]
\supset\Rightarrow & \dfrac{A \supset B, \Gamma \Rightarrow \Delta, A \quad B, \Gamma \Rightarrow \Delta}{A \supset B, \Gamma \Rightarrow \Delta} & \Rightarrow\supset & \dfrac{A, \Gamma \Rightarrow B}{\Gamma \Rightarrow \Delta, A \supset B}
\end{array}
$$

$$
\begin{array}{llll}
\sim\Rightarrow & \dfrac{\sim A, \Gamma \Rightarrow \Delta, \varphi}{\sim A, \Gamma \Rightarrow \Delta} & \Rightarrow\sim & \dfrac{, \Gamma \Rightarrow}{\Gamma \Rightarrow \Delta, \sim}
\end{array}
$$

The boxed rules do not appear in the positive fragment of intuitionistic logic, denoted I_p. All rules are invertible, i.e. the premises also follow from their conclusions, except, $\supset\Rightarrow$ and $\supset\sim$.

2.2.3 Natural Deduction Rules

The Natural Deduction System for Propositional Intuitionistic Logic (also developed by Gentzen)

Gentzen natural deduction sequents $\Gamma \Rightarrow A$ should be read from the open assumptions in Γ we derive A, where the multiset Γ is a finite list of formulas, Γ, A is the set Γ with the single formula A added. The rules are divided into introduction and elimination rules according to whether the main operator is being introduced or eliminated (respectively). A natural deduction proof must have an empty set of open assumptions i.e. $\Gamma = \emptyset$.

Axioms $(1) A \Rightarrow A$

Rules of Inference

$\wedge$ Intro $\quad \dfrac{\Gamma \Rightarrow A \quad \Gamma' \Rightarrow B}{\Gamma, \Gamma' \Rightarrow A \wedge B}$ $\qquad$ $\wedge$ Elim $\quad \dfrac{\Gamma \Rightarrow A_1 \wedge A_2}{\Gamma \Rightarrow A_i} \quad i \in \{1,2\}$

$\vee$ Intro $\quad \dfrac{\Gamma \Rightarrow A_i}{\Gamma \Rightarrow A_1 \vee A_2} \quad i \in \{1,2\}$ $\qquad$ $\vee$ Elim $\quad \dfrac{\Gamma \Rightarrow A_1 \vee A_2 \quad A_1, \Delta \Rightarrow C \quad A_2, \Sigma \Rightarrow C}{\Gamma, \Delta, \Sigma \Rightarrow C}$

$\supset$ Intro $\quad \dfrac{A, \Gamma \Rightarrow B}{\Gamma \Rightarrow A \supset B}$ $\qquad$ $\supset$ Elim $\quad \dfrac{\Sigma \Rightarrow A \quad \Gamma \Rightarrow A \supset B}{\Gamma, \Sigma \rightarrow B}$

$\mathbf{0}$ $\quad \dfrac{\Gamma \Rightarrow \mathbf{0}}{\Gamma \Rightarrow A}$

It is traditional in natural deduction to rewrite $\sim A$ as $A \supset \mathbf{0}$ and then add the rule for $\mathbf{0}$ where $\mathbf{0}$ represents falsity.

In this logic $\wedge$ distributes over $\vee$ and $\vee$ distributes over $\wedge$. That is we have $a \vee (b \wedge c) \equiv (a \vee b) \wedge (a \vee c)$ and also $a \wedge (b \vee c) \equiv (a \wedge b) \vee (a \wedge c)$.

Mints (Mints 1990) proved using realizability that these three systems were all equivalent. Gentzen (Gentzen 1969) proved that the Hilbert rules were equivalent to his single-occurrence Gentzen rules and he also proved that the Hilbert rules were equivalent to his natural deduction rules. Dummett (Dummett 1977) proved that multiple occurrence rules were equivalent to Genzten's single occurrence rules. Finally, Mints (Mints 1985) proved that this particular multiple occurrence system was equivalent to Gentzen's system and that these rules have the added advantage that they can be read as a tableau for theorem proving by changing the system of (Beth 1952).

2.3 Discussion about Intuitionistic Logic

Intuitionistic logic has three major changes from Boolean logic. Intuitionistic logic changes the material implication present in Boolean logic to arrive an the notion of "proof transformation". If a formula φ intuitionistically implies ψ then a proof of φ can be transformed into a proof of ψ. In other words a proof of ψ can be constructed given the proof of φ. A "realization" of a proof of $\varphi \supset \psi$ must have as its input a proof of φ and produce as its output a proof of ψ. Therefore a "realization" of the proof of $\varphi \supset \psi$ is some function which given a proof produces another proof. This is the reason that in the BHK (Brouwer-Heyting-Kolmogorov) interpretation, the rule for implication is

$$c\,\mathbf{r}\,(\varphi \supset \psi) \iff c \text{ is a function which for every } d \text{ if } d\,\mathbf{r}\varphi \text{ then } c(d)\,\mathbf{r}\psi$$

where we write $c\,\mathbf{r}\,\varphi$ to denote c *realizes* φ.

The next major change is the treatment of disjunction. Constructivity requires that whenever a formula is valid, then it must be possible to construct a proof of that formula. However, constructive disjunction requires that if $\varphi \vee \psi$ is valid then it must be possible to construct a proof for at least one of φ or ψ. However the BHK interpretation takes this idea one step further. Whenever $\varphi \vee \psi$ is valid then not only must there be a proof of one of φ or ψ, but there must be an indication of which of these disjuncts has a proof. This is the reason that in the BHK interpretation that the rule for disjunction has an injection function k and is

$$c\,\mathbf{r}\,(\varphi \vee \psi) \iff c = k_0\,a \text{ and } a\,\mathbf{r}\,\varphi \text{ or } c = k_1\,a \text{ and } a\mathbf{r}\psi$$

where k_0 and k_1 are annotations based on the syntactic structure which indicate whether it is φ or ψ for which there is a proof. This k injector function plays the same role as disjoint union where $k_0 a$ would be $\langle a, 0\rangle$ and $k_1 a$ would be $\langle a, 1\rangle$ however it is traditional to use k.

The last major change is the treatment of negation. This change to negation arrives at the notion of "failure of every construction". This provides a negation as failure interpretation for negation in intuitionistic logic. We see that $\sim \varphi$ is encoded as $\varphi \supset 0$. Using a combination of the BHK rules for implication and absurdity, 0, we can understand their interpretation of negation. The two rules are

$$c\,\mathbf{r}\,(\varphi \supset \psi) \iff c \text{ is a function which for every } d \text{ if } d\,\mathbf{r}\varphi \text{ then } c(d)\,\mathbf{r}\psi \text{ and not } c\,\mathbf{r}\,0.$$

Therefore the derived BHK rule for the intuitionistic negation of φ, denoted $\sim \varphi$, is

$$c\,\mathbf{r}\,(\varphi \supset 0) \iff c \text{ is a function which for every } d \text{ if } d\,\mathbf{r}\varphi \text{ then } c(d)\,\mathbf{r}0.$$

Therefore if c realizes $\varphi \supset 0$ then c must be function which for every proof of φ produces a proof of 0. However, there is no proof of 0. This means that for c to realize $\varphi \supset 0$ there must be no proof of φ. Therefore, the failure of our logic to find a proof for φ is exactly intuitionistic negation.

We have seen that there are some differences between Boolean logic and intuitionistic logic. However, there are a few important distinctions to mention in terms of the consequences of these two logics. Intuitionistic logic does not validate either the law of the excluded middle $\varphi \vee \sim \varphi$ nor does it validate the law of the weak excluded middle $\sim \varphi \vee \sim\sim \varphi$. Boolean logic validates both of these formulas $\varphi \vee \neg\varphi$ and $\neg\varphi \vee \neg\neg\varphi$, where $\neg$ is Boolean negation.

2.4 Kripke Structures for Intuitionistic Logic

We are about to define the semantics for propositional N in terms of intervals over Kripke structures for propositional intuitionistic logic. Therefore, first we review some definitions and results for intuitionistic logic. In particular, our completeness asserts that any underivable formula is refutable in a canonical model. In fact, the canonical refutation Kripke structure $\mathcal{K}$ for positive intuitionistic logic is the canonical refutation Kripke structure for N, when we use

N's interval satisfiability relation over $\mathcal{K}$. Thus, we provide a brief review of the definition of $\mathcal{K}$ as well as the definition of an intuitionistic Kripke model.

Definition 1 *A Kripke structure for propositional intuitionistic logic over a set P of propositional atoms is a triple, $\langle W, R, V\rangle$, where W is a non-empty set (the set of* worlds*), R is a binary reflexive and transitive relation on W and V is a function assigning to every world a set of propositional atoms satisfying for every two worlds w, w', $wRw' \to V(w) \subseteq V(w')$.*

Thus, V is monotonic on propositional atoms.

Definition 2 *The satisfiability relation $\models_K$ on a Kripke structure $K = \langle W, R, V\rangle$ over propositional atoms P is a relation between worlds and propositional formulas. We define this relation inductively over the structure of a formula. For any $w \in W$,*

$$\begin{array}{lcr}
w \models_K p & \equiv & p \in V(w) \\
w \models_K \varphi \wedge \psi & \equiv & w \models_K \varphi \textit{ and } w \models_K \psi \\
w \models_K \varphi \vee \psi & \equiv & w \models_K \varphi \textit{ or } w \models_K \psi \\
w \models_K \ \sim \varphi & \equiv & \forall w' \in W.\ w\ R\ w' \textit{ implies } w' \not\models_K \varphi \\
w \models_K \varphi \supset \psi & \equiv & \forall w' \in W.\ w\ R\ w' \textit{ and } w' \models_K \varphi \textit{ implies } w' \models_K \psi
\end{array}$$

where the meta-logic of all definitions is classical logic.

We say that a formula φ is satisfied at a world w in a structure K if $w \models_K \varphi$.

Definition 3 *Let $K = \langle W, R, V\rangle$ be a propositional Kripke structure. A propositional formula φ is* satisfied *in K if and only if $\exists w \in W.\ w \models_K \varphi$. A propositional formula φ is* valid in K *if and only if $\forall w \in W.\ w \models_K \varphi$. A propositional formula φ is* valid *if and only if φ is valid in all propositional Kripke structures.*

Intuitionistic logic is sound and complete with respect to $\models_K$ However, instead of the completeness relying on a construction of a countermodel for every non-valid formula, there is a *canonical* model $\mathcal{K}$ in which every non-valid formula is falsified. We shall reuse $\mathcal{K}$ for our completeness result of N, hence we repeat the construction here.

A Gentzen sequent was defined such that $\Gamma \cup \Delta$ was a finite set; an *infinite* sequent is where this union is an infinite set i.e. one or both of Γ, Δ is infinite.

Definition 4 *Let $\Gamma \Rightarrow \Delta$ be an infinite sequent made from formulas in propositional intuitionistic logic. It is derivable if and only if $\Gamma' \Rightarrow \Delta'$ for some finite $\Gamma' \subset \Gamma, \Delta' \subset \Delta$.*

$\Gamma \Rightarrow \Delta$ *is* complete *if it is underivable and for any formula φ in propositional intuitionistic logic $\varphi \in \Gamma \cup \Delta$.*

This definition can be easily modified to define infinite and complete sequents for any logic. In subsequent chapters, for constructible falsity and constructible duality, we will use the terms infinite and complete sequents to mean a complete sequent for their respective logics.

Lemma 5 *Every finite underivable sequent can be extended to a complete sequent.*

The lemma above ensures that the following definition has a non-empty set of worlds.

Definition 6 *Let $\mathcal{K} = \langle W_k, R_k, V_k \rangle$ denote the canonical refutation model for I_p. We have*

1. *W_k as the set of all $\Gamma \Rightarrow \Delta$ that are* complete *sequents,*
2. *$\Gamma_i \Rightarrow \Delta_i \; R_k \; \Gamma_j \Rightarrow \Delta_j$ if and only if $\Gamma_i \subset \Gamma_j$, and*
3. *$p \in V_k(\Gamma_i \Rightarrow \Delta_i)$ if and only if $p \in \Gamma_i$.*

Note: Item 2 builds the monotonicity condition into R_k and also implies that $\Delta_j \subset \Delta_i$.

Theorem 7 *(Fitting 1991) If $\Gamma \Rightarrow \Delta$ is a complete sequent in I_p then in $\mathcal{K}$, $\Gamma \Rightarrow \Delta \not\models_{\mathcal{K}} \Gamma \Rightarrow \Delta$.*

The proof of this theorem relies on the notion of saturatedness. With these definitions and lemmas many cases in N, will also follow.

Definition 8 Saturated Sequent *A complete sequent $\Gamma \Rightarrow \Delta$ for I_p is saturated when it satisfies for all φ, ψ,*

Rule	*Sequent*		*Saturatedness*
$\Rightarrow \wedge$	$\varphi \wedge \psi \in \Delta$	*implies*	$\varphi \in \Delta$ *or* $\psi \in \Delta$
$\wedge \Rightarrow$	$\varphi \wedge \psi \in \Gamma$	*implies*	$\varphi \in \Gamma$ *and* $\psi \in \Gamma$
$\Rightarrow \vee$	$\varphi \vee \psi \in \Delta$	*implies*	$\varphi \in \Delta$ *and* $\psi \in \Delta$
$\vee \Rightarrow$	$\varphi \vee \psi \in \Gamma$	*implies*	$\varphi \in \Gamma$ *or* $\psi \in \Gamma$
$\supset\Rightarrow$	$\varphi \supset \psi \in \Gamma$	*implies*	$\varphi \in \Delta$ *or* $\psi \in \Gamma$

Definition 9 Saturated Set of Sequents *A set of complete sequents $\mathcal{X}$ is saturated if every sequent $s \in \mathcal{X}$ is saturated and if $s \in \mathcal{X}$ where $s = \Gamma \Rightarrow \Delta, \varphi \supset \psi$ then there is an $s' \in \mathcal{X}$ such that $s' = \Gamma' \Rightarrow \Delta'$ where $\Gamma \cup \{\varphi\} \subset \Gamma'$ and $\psi \in \Delta'$.*

Lemma 10 *The set of complete sequents for I_p is saturated.*

This means somewhere in the set of complete sequents we analyze the non-invertible rule and in each complete sequent we analyze all of the invertible rules.

2.5 First Order Intuitionistic Logic

We extend the language to include countably many bound variables, here denoted $x, y, z..$ and countably many free variables, here denoted $a, b, c..$ for each natural n, we have countably many predicates of arity n, denoted $P^n, Q^n, ...$

Terms are constructed from individuum variables and parameters. Atomic formulas are of the form $P^n(t_1, \ldots, t_n)$ where $t_i, 1 \leq i \leq n$ are terms. Formulas are constructed via atomic formulas via $\wedge, \vee, \sim, \exists, \forall$.

$\varphi[t/x]$ is the result of substituting t for all occurrences of x in φ. If the variable is evident, we simply denote this as $\varphi[t]$. In most cases t will denote a ground terms, that is one which contains parameters but not variables; thus collisions are impossible.

We add the following rules to those of propositional I to yield the first order system.

Rules of Inference

$$\Rightarrow\exists \quad \frac{\Gamma \Rightarrow \Delta, \exists x.\varphi, \varphi[t]}{\Gamma \Rightarrow \Delta, \exists x.\varphi} \qquad \exists\Rightarrow \quad \frac{\varphi[e], \Gamma \Rightarrow \Delta}{\exists x.\varphi, \Gamma \Rightarrow \Delta}$$

$$\Rightarrow\forall \quad \frac{\Gamma \Rightarrow \varphi[e]}{\Gamma \Rightarrow \Delta, \forall x.\varphi} \qquad \forall\Rightarrow \quad \frac{\varphi[t], \forall x.\varphi, \Gamma \Rightarrow \Delta}{\forall x.\varphi, \Gamma \Rightarrow \Delta}$$

Where t is a ground term, e is a new parameter (eigenvariable) therefore does not occur in $\Gamma \cup \Delta$. Note that $\Rightarrow \forall$ is not invertible.

The following definition is standard for first order intuitionistic logic.

Definition 11 First Order Model *A Kripke model for first order intuitionistic logic over universe U is a four-tuple $K = \langle W, R, V, D \rangle$ where W is a set, and $R \subseteq W \times W$ is a reflexive transitive relation and*

1. *D is a function from W to a subset of U satisfying wRw' implies $D(w) \subseteq D(w')$, and*

2. *V is a function from worlds W and n-ary predicates P^n to $(D(w))^n \to \{0,1\})$ satisfying the monotonicity condition that if wRw' then $V(w',P)$ is an extension of $V(w,P)$ i.e. for all $x_1, \ldots x_n \in D(w)$, $V(w,P)(x_1,\ldots,x_n)$ implies $V(w',p)(x_1,\ldots,x_n)$.*

Definition 12 *A first-order model for I is a first-order Kripke structure $M = \langle W, R, V, D\rangle$ with the entailment relation $\models_M$ is defined by induction on φ. For any $w \in W$.*

$$
\begin{array}{lllr}
A & w \models_M A(x_1,\ldots,x_n) & \equiv & V(w,A)(x_1,\ldots,x_n) = 1 \\
 & & & \textit{for all } x_1,\ldots,x_n \in D(w) \\
\varphi \wedge \psi & w \models_M \varphi \wedge \psi & \equiv & w \models_M \varphi \textit{ and } w \models_M \psi \\
\varphi \vee \psi & w \models_M \varphi \vee \psi & \equiv & w \models_M \varphi \textit{ or } w \models_M \psi \\
\varphi \supset \psi & w \models_M \varphi \supset \psi & \equiv & \forall w' \in W. w\ R\ w' \\
 & & & w' \models_M \varphi \textit{ implies } w' \models_M \psi \\
\forall x.\varphi(x) & w \models_M \forall x.\varphi(x) & \equiv & \forall w' \in W. w\ R\ w' \\
 & & & \textit{for every } d \in D(w')\ \ w' \models_M \varphi[d] \\
\exists x.\varphi(x) & w \models_M \exists x.\varphi(x) & \equiv & \textit{for some } d \in D(w)\ \ w \models_M \varphi[d]
\end{array}
$$

Definition 13 *Let $K = \langle W, R, V, D\rangle$ be a first-order Kripke structure. A first-order formula φ is* satisfied *in K if and only if $\exists w \in W.\ w \models_K \varphi$. A first-order formula φ is* valid in *K if and only if $\forall w \in W.\ w \models_K \varphi$. A first-order formula φ is* valid *if and only if φ is valid in all first-order Kripke structures.*

Theorem 14 Soundness and Completeness *A first-order intuitionistic formula φ is derivable if and only if it is valid in all first-order models for I.*

2.6 Discussion about Intuitionistic Model Theory

We shall first review some basic ideas from Boolean logic's model theory and then present an overview of the differences between this model theory and intuitionistic model theory.

The model theory for propositional Boolean logic is well known(Enderton 1972). It is founded on the idea of a valuation function. A model of Boolean logic using a set of propositional atoms is a function from P to the set containing "true" and "false" called a valuation function. This valuation function $V: P \rightarrow \{$ "true", "false" $\}$ is a function to a 2-valued set and can be viewed as a characteristic function for a set. The valuation function, or set of "true" propositional atoms, determines the theory of the classical model. This means that the valuation function determines the truth of formulas like $p \vee q$ where both $p, q \in P$. If $V(p) =$ "true" or $V(q) =$ "true" then we will call this formula true in the model of which V is the valuation function.

A model theory for propositional intuitionistic logic is a partial order $\leq$ over a set of worlds labeled with models for propositional Boolean logic. This means that a model for propositional intuitionistic logic is a partial order where each point in the partial order is labeled with a valuation function. However, the labeling of the partial order cannot be arbitrary because the range of the valuation functions must be related. If $w \leq w'$, then the range of the valuation function for w must be a subset of the range of the valuation function for w'. This is the basis for Kripke's model for intuitionistic logic(Kripke 1965).

The points in the partial ordering are called worlds by Kripke and moving upwards in the partial order (that is passing from w to w' when $w \leq w'$) represents a new stage of knowledge. A new stage of knowledge is viewed by Kripke as a stage where we have more information. Therefore, when we pass from w to w' we have learned the truth of new propositional atoms and formulae and we have a modified view of the partial order. Namely when we pass from w to w' the set of accessible worlds from w' is a subset of the worlds accessible from w. Therefore, not only are their possibly new propositional atoms true, but there is a different set of worlds accessible from w' then there was from w.

Model theoretically intuitionistic implication is the biggest change over classical implication. Model theoretically an intuitionistic implication is "true" or "satisfied" at a world in the partial order whenever the classical implication is true in all accessible worlds. That is if $\varphi \supset \psi$ is satisfied at a world w_0 then $\varphi \rightarrow \psi$ must be satisfied at any world w such that $w_0 \leq w$ using the valuation function of w. We use $\varphi \rightarrow \psi$ to denote Boolean implication.

Let us look at an example. We draw the Kripke model by annotating the valuation function (or the set of propositional atoms true) at that world and drawing arcs to denote the partial ordering among worlds.

$$\begin{tikzcd}
w_1\ p & & w_2\ \emptyset \\
& w_0\ \emptyset \arrow[ul] \arrow[ur] &
\end{tikzcd}$$

At w_0 there are no propositional atoms true, but when we pass to w_1 we have "learned" that p is true. This means that w_1 has many new consequences that w_0 did not have. When we pass from w_0 to w_2, we have not "learned" any new propositional atoms, but we have learned a new formula namely $\sim p$ is true at the world w_2 and it is not at w_0. This is the structure that refutes the law of the weak excluded middle $\sim p \vee \sim\sim p$.

2.7 Heyting Algebras for Intuitionistic Logic

This section will show that Heyting algebras are algebras for propositional intuitionistic logic. This was shown by McKinsey and Tarski (1946) using a result of Birkhoff (1933) which showed that topologically distributive lattices are pseudo-complemented. This particular presentation follows Fitting (1969).

Definition 15 *In a distributive lattice $\mathcal{X} = (X, \leq, \sqcap, \sqcup)$ with meet, $\sqcap$ and join, $\sqcup$, the* pseudo-complement *of x* relative *to x', denoted $x \supset x'$ is the element* $\max\{y | x \sqcap y \leq x'\}$.

A lattice is called *relatively pseudo-complemented* if every pair of elements has a relative pseudo-complement. If a relatively pseudo-complemented lattice has a bottom element 0, then the pseudo-complement of x relative to 0 is simply called the *pseudo-complement* and is denoted $-x$. Relatively pseudo-complemented lattices with a bottom element 0, i.e. $\forall x \in X. 0 \leq x$, are called pseudo-complemented lattices.

Definition 16 *A* Heyting *algebra is a pseudo-complemented distributive lattice. Note: -0 is denoted* 1.

Definition 17 *Let $\mathcal{X}$ be a Heyting algebra and P a set of propositional atoms. Given any function $h: P \to \mathcal{X}$, we extend h to all formulas φ of intuitionistic logic as follows:*

$$\begin{array}{rclcrcl}
h(\alpha \wedge \beta) & = & h(\alpha) \sqcap h(\beta) & \qquad & h(\alpha \vee \beta) & = & h(\alpha) \sqcup h(\beta) \\
h(\alpha \supset \beta) & = & h(\alpha) \supset h(\beta) & \qquad & h(\sim \alpha) & = & -h(\alpha)
\end{array}$$

Definition 18 *A formula φ is* satisfied *in $\mathcal{X}$, a Heyting algebra, if and only if $h(\varphi) = 1$. A formula φ is* valid in $\mathcal{X}$ *if and only if for all $h: P \to \mathcal{X}$, $h(\varphi) = 1$. A formula φ is* algebraically valid *if and only if for all Heyting algebras $\mathcal{X}$ and for all $h: P \to \mathcal{X}$, $h(\varphi) = 1$.*

Theorem 19 Soundness *Every derivable sequent in intuitionistic logic is algebraically valid.*

We can prove completeness of the algebraic semantics analogously to using canonical Kripke models by using a canonical valuation map on the free algebra. This method uses H_f the free Heyting algebra over the set of generators P for the theory of intuitionistic logic with atoms P.

Theorem 20 (Algebraic Completeness) *Every non-valid formula in the language of intuitionistic logic with the set P of propositional atoms is refuted in the free Heyting algebra over P.*

This obviously gives us a standard completeness result which says that every non-valid formula is refuted in some Heyting algebra namely the free one which is unique up to isomorphism. However, we can also obtain completeness by translating the Kripke models for intuitionistic logic to a Heyting algebra.

Lindenbaum created a construction which transformed any Kripke model into its Heyting algebra representation which created the H_f above.

Beth and Fitting created a construction which also transformed a Kripke model into its Heyting algebra representation but their construction also has backwards construction from a Heyting algebra to a Kripke model. This backwards construction is not the inverse. However this backward construction does preserve disjunctions correctly. We present it because it is easier for most people to construct the Kripke model counterexamples to theorems rather than the algebraic counterexamples and this procedure gives an easy way of translating between those representations.

Definition 21 *Let $\langle W, R, V\rangle$ be a Kripke model. Let $Fil(\langle W, R\rangle)$ denote set of of all R-upward-closed subsets of $\langle W, R\rangle$ ordered by subset inclusion, that is each $W' \in Fil(\langle W, R\rangle)$ satisfies $(w \in W' \wedge wRw')$ implies $w' \in W'$.*

All R-downward-closed subsets of a Kripke structure $\langle W, R\rangle$ are denoted $Idl(\langle W, R\rangle)$, and $Idl(\langle W, R\rangle) \cong Fil(\langle W, R\rangle)^{op}$.

Lemma 22 *$Fil((W,R))$ ordered under subset inclusion is a Heyting algebra where the lattice meet is intersection and the lattice join is union.*

The relative pseudo-complement of W_1 relative to W_2, $W_1, W_2 \in Fil(\langle W, R\rangle)$, denoted $W_1 \supset W_2$, is the largest R-upward-closed subset of $(W - W_1) \cup W_2$.

The following is an analogue to Birkhoff-Stone duality (Birkhoff 1967).

Lemma 23 Beth-Fitting *Let $K = \langle W, R, V\rangle$ be a Kripke model for intuitionistic logic over P. We define $h\colon P \to Fil(\langle W, R\rangle)$ as $h(p) = \{w \mid p \in V(w)\}$. This assignment extends to all intuitionistic formulas φ such that $h(\varphi) = \{w \mid w \models_K \varphi\}$.*

In particular if φ is valid in the model then it must be true at all worlds and $h(\varphi) = 1 = W$. The backwards construction mapping all proper *prime* filters F over a Heyting algebra to Kripke worlds where subset inclusion of filters $F\,R\,F' \iff F \subseteq F'$ defines the accessibility relation and where the assignment $p \in V(F) \iff h(p) \in F$ extends to all formulas (Fitting 1969).

Corollary 24 Completeness *A formula is derivable in intuitionistic logic* if and only if *it is algebraically valid.*

If $K_B = \langle W, R, V\rangle$ is a Beth model for intuitionistic logic, then $H = Fil(\langle W, R\rangle)$ is the Heyting algebra representing that model. The backwards construction takes *every* filter in H to be a world in a Beth model (where accessibility and valuation are as above) and provides both an analogue of Stone duality (Stone 1936) and an instance of the Yoneda embedding (Yoneda 1961, Yoneda 1962).

2.8 Computational Interpretation

Intuitionistic logic has a well known computational interpretation in terms of the lambda-calculus by means of the Curry-Howard isomorphism. Curry developed the connection between cut and function application in the typed lambda-calculus and Howard developed the interpretation between the introduction of an implication and lambda-abstraction. Together these two results are called the Curry-Howard isomorphism because it forms an isomorphism between intuitionistic formulas and typed lambda-calculus terms. This result was independently discovered by deBruijn in the AUTOMATH project (de Bruijn 1970). This isomorphism is really an isomorphism between cut-elimination (or proof normalization) and lambda-calculus reductions. Steps in proof normalization are isomorphic to typed-lambda calculus reductions.

Proof normalization in a certain natural deduction system is isomorphic to typed lambda-calculus reductions. This interpretation of lambda-terms as formulae necessarily contains the notion of proof equivalence through lambda-term equivalence. The Curry-Howard isomorphism needs as its basis a proof system which will allow for a description of proof equivalences. Proofs in the natural deduction systems can be seen as manipulation of an equivalence class of proofs. This is the reason for using the natural deduction system when proving "proofs as computation" results for logics.

2.8.1 Curry-Howard for Intuitionistic Logic

We now define a *formulas as types* formalism for constructible duality and show that proof normalization corresponds to the operational semantics of an extension of the lambda-calculus.

The terms are taken from a typed lambda calculus.

Definition 25 (Value terms) *The set of value terms is constructed from a set of* value variables V, *the constant* 0 *(representing the natural number 0), and the successor function* S.

$$v_t = 0|u||S(v_t) \qquad \textit{Value Terms,}$$

where $u \in V$.

We use u, v, x, y, z to denote value variables, that is elements of V. We use v_t, n, m to denote variables for value terms; also of course $0, S(0)$ and $S(n)$ with the notation $S^n(0)$ meaning $\overbrace{S(\ldots S(0))}^{n}$.

Definition 26 (The set of simple types) *The set of* simple types $\mathcal{T}$ *is constructed from a set of* predicate constants $\mathbf{P}$, *each with an associated arity* n, *and value terms, by means of a type forming operations.*

- *if* $p \in \mathbf{P}$ *and has arity* n, *then* p *applied to any* n *value terms belongs to* $\mathcal{T}$
- *if* $A, B \in \mathcal{T}$ *then* $A \supset B \in \mathcal{T}$

- *if* $A, B \in \mathcal{T}$ *then* $A \wedge B \in \mathcal{T}$
- *if* $A, B \in \mathcal{T}$ *then* $A \vee B \in \mathcal{T}$
- *if* $A \in \mathcal{T}$ *then* $\exists v.A \in \mathcal{T}$ *for all* $v \in V$
- *if* $A \in \mathcal{T}$ *then* $\forall v.A \in \mathcal{T}$ *for all* $v \in V$.

These simple types correspond to the formulas of intuitionistic logic (minus those for intuitionistic negation or falsity).

We assume there are countably many term variables X *of each type* A. *We write* X^A, Y^A, ..., *and we also use the* $X\colon A$ *notation. For distinct types* A *and* B *the corresponding sets of individual variables are disjoint.*

The expressions (or terms) of our lambda-calculus λ_H *for Heyting's formalization of intuitionistic logic are:*

$v_t = 0 \vert u \vert S(v_t)$	*Value Terms*
$a = X \mid \lambda X.a \mid (aa') \mid \boldsymbol{\lambda} v.a \mid \boldsymbol{(} av_t \boldsymbol{)} \mid \langle a, a' \rangle \mid \pi_i a \mid k_i a \mid D_{X,Y}(a, a_1, a_2) \mid \boldsymbol{\langle} v_t \ a \boldsymbol{\rangle} \mid E_{v,X}(a, a')$	*Terms*

We use $a, b, c, \ldots$, with primes and indices, for terms. We use π to denoted the projection function and k_i for the injection function, abstraction of terms $\lambda X.a$, abstraction of value terms, $\boldsymbol{\lambda} v.a$, and their respective applications (aa'), and $\boldsymbol{(}av_t\boldsymbol{)}$. We also have the disjunction D and existential E.

There are some restrictions on forming terms. Namely, in $E_{v,X}(a, b)$, v must not occur in the type of b, nor in the type of any unbound term variable, save X. Further, X may not occur in a. In $D_{X^A,Y^B}(a^{A \vee B}, b^C, c^C)$ a contains neither X nor Y. Also, in $\boldsymbol{(}a^{\forall x.A(x)} v_t\boldsymbol{)}$ no value variable in v_t may occur bound in the type of a, $\forall x.A(x)$. Finally, some typing rules must also be obeyed. Terms can only be formed from terms with types as given in the table below.

Term	Type
$(b^{A\supset B}\ a^A)$	B
$\langle a^A, b^B\rangle$	$A \wedge B$
$\lambda X^A.b^B$	$A \supset B$
$\pi_1 c^{A\wedge B}$	A
$\pi_2 c^{A\wedge B}$	B
$D_{X^A,Y^B}(a^{A\vee B}, b^C, c^C)$	C
$k_1 c^A$	$A \vee B$
$k_2 c^B$	$A \vee B$
$E_{v_1,X^{A[v/1/v]}}(a^{\exists v.A(v)}, c^C)$	C
$\boldsymbol{\langle} v_t, a^{A[v_t/v]}\boldsymbol{\rangle}$	$\exists v.A$
$\boldsymbol{\lambda} v.a^A$	$\forall v.A$
$\mathbf{(}a^{\forall v.A}\ v_t\mathbf{)}$	$A[v_t/v]$

The free value variables $FV_v(a)$ and free term variables $FV_t(a)$ of a term a, are defined inductively by the following table.

Term	Free value variables	Free term variables
0	$\emptyset$	$\emptyset$
v	$\{v\}$	$\emptyset$
$S(v_t)$	$FV_v(v_t)$	$\emptyset$
X	$\emptyset$	$\{X\}$
$(b^{A\supset B}\ a^A)$	$FV_v(b) \cup FV_v(a)$	$FV_t(b) \cup FV_t(a)$
$\langle a^A, b^B\rangle$	$FV_v(b) \cup FV_v(a)$	$FV_t(b) \cup FV_t(a)$
$\lambda X^A.b^B$	$FV_v(b)$	$FV_t(b) - \{X\}$
$\pi_1 c^{A\wedge B}$	$FV_v(c)$	$FV_t(c)$
$\pi_2 c^{A\wedge B}$	$FV_v(c)$	$FV_t(c)$
$D_{X,Y}(a^A, b^B, c^C)$	$FV_v(a) \cup FV_v(b) \cup FV_v(c)$	$FV_t(a) \cup ((FV_t(b) \cup FV_t(c)) - \{X, Y\})$
$k_1 c^A$	$FV_v(c)$	$FV_t(c)$
$k_2 c^B$	$FV_v(c)$	$FV_t(c)$
$E_{v_1,Z}(a^{\exists v.A(v)}, c^C)$	$FV_v(a) \cup (FV_v(c) - \{v_1\})$	$FV_t(a) \cup (FV_t(c) - \{Z\})$
$\langle v_t, a^A\rangle$	$FV_v(a) \cup FV_v(v_t)$	$FV_t(a)$
$\boldsymbol{\lambda} v.a^A$	$FV_v(a) - \{v\}$	$FV_t(a)$
$(a^{\forall v.A}\ v_t)$	$FV_v(a) \cup FV_v(v_t)$	$FV_t(a)$

Substitution is defined as replacing the free occurrences of the variable by the substituted term. Our substitution convention will be that $b[x/y]$ means that x is substituted for y everywhere y is free in b.

We associate terms with proof rules in the following way.

The term assignments below are annotated Gentzen natural deduction sequents where $\Gamma \Rightarrow a{:}\,A$ should be read as, from the open assumptions in Γ, we derive A with proof a, where the multiset Γ is a finite list of formulas, and Γ, A is the set Γ with the single formula A added. The rules are divided into introduction and elimination rules according to whether the main operator is being introduced or eliminated (respectively). A proof $\Gamma \Rightarrow C$ of a formula C must have an empty set of open assumptions i.e. $\Gamma = \emptyset$.

The rules below are the natural deduction rules (from the background on intuitionistic logic chapter) followed by the rule for recursion and rules for equality and inequality.

When reading the term calculus consider a term $b{:}\,B$ with $FV_t(b) = \{a_1^{A_1} \ldots a_n^{A_n}\}$ as representing the deduction of $A_1, \ldots A_n, \Gamma \Rightarrow B$ for arbitrary Γ.

$$\textbf{Axiom} \quad \Gamma, X{:}\,A \Rightarrow X{:}\,A$$

2.8.1.1 Propositional Intuitionistic Logic Rules

$$\wedge\ \text{Intro}\quad \frac{\Gamma \Rightarrow a{:}\,A \quad \Gamma' \Rightarrow b{:}\,B}{\Gamma, \Gamma' \Rightarrow \langle a, b\rangle{:}\,A \wedge B} \qquad \wedge\ \text{Elim}\quad \frac{\Gamma \Rightarrow a{:}\,A_1 \wedge A_2}{\Gamma \Rightarrow \pi_i a{:}\,A_i}\ i \in \{1,2\}$$

$$\supset\ \text{Intro}\quad \frac{X{:}\,A, \Gamma \Rightarrow b{:}\,B}{\Gamma \Rightarrow \lambda X^A.b{:}\,A \supset B} \qquad \supset\ \text{Elim}\quad \frac{\Sigma \Rightarrow a{:}\,A \quad \Gamma \Rightarrow b{:}\,A \supset B}{\Gamma, \Sigma \Rightarrow (b\ \ a){:}\,B}$$

$$\vee\ \text{Intro}\quad \frac{\Gamma \Rightarrow b{:}\,A_i}{\Gamma \Rightarrow k_i\ b{:}\,A_1 \vee A_2}\ i \in \{1,2\}$$

$$\vee\ \text{Elim}\quad \frac{\Sigma \Rightarrow a{:}\,A \vee B \quad \Gamma, X{:}\,A \Rightarrow c{:}\,C \quad \Gamma', Y{:}\,B \Rightarrow c'{:}\,C}{\Gamma, \Gamma', \Sigma \Rightarrow D_{X,Y}(a^{A\vee B}, c^C, c'^C){:}\,C}$$

To these rules we add the quantifier rules.

2.8.1.2 First-Order Intuitionistic Logic Rules

$$\forall\ \text{Elim}\quad \frac{\Gamma \Rightarrow a{:}\,\forall v.A}{\Gamma \Rightarrow (a\ \ n){:}\,A[n/v]} \qquad \forall\ \text{Intro}\quad \frac{\Gamma \Rightarrow a{:}\,A}{\Gamma \Rightarrow \boldsymbol{\lambda} v.a{:}\,\forall v.A}$$

If Γ does not contain v free

$$\exists\ \text{Elim}\quad \frac{\Gamma \Rightarrow a{:}\,\exists v.A, \quad \Delta, X{:}\,A[u/v] \Rightarrow c{:}\,C}{\Gamma, \Delta \Rightarrow E_{u,X}(a, c){:}\,C} \qquad \exists\ \text{Intro}\quad \frac{\Gamma \Rightarrow a{:}\,A[n/v]}{\Gamma \Rightarrow \langle n, a\rangle{:}\,\exists v.A}$$

If u is new in $\Delta \cup \{C\}$

A term assignment $T(d)$ to a natural deduction d is defined as follows. If d is an axiom $\Gamma, a{:}\,A \Rightarrow a{:}\,A$ then $T(d) = a$.

If d ends in a rule, $\dfrac{\Gamma_1 \Rightarrow a_1{:}\,A_1, \ldots, \Gamma_n \Rightarrow a_n{:}\,A_n}{\Phi(\Gamma_1, \ldots, \Gamma_n) \Rightarrow f(a_1, \ldots, a_n){:}\,\phi(A_1, \ldots, A_n)}$, and the terms assigned to the n subproofs are $b_1, \ldots, b_n$, where b_i is the term for $\Gamma_i \Rightarrow a_i{:}\,A_i$ then $T(d) = f(b_1, \ldots, b_n)$ where Φ is the transformation of premises, f is the transformation of terms and ϕ is the transformation of types.

2.8.2 Operational Semantics

The operational semantics is given by the following term conversion.

Definition 27 *Our one-step reductions of the λ_I calculus is given below.*

	Intuitionistic Logic Reductions		
$\mathbf{P_i}$	$\pi_i(\langle a_1, a_2\rangle) \rhd a_i$	**L**	$(\lambda X.a\ b) \rhd a[b/X]$
V	$(\boldsymbol{\lambda} v.a\ n) \rhd a[n/v]$	**D**	$D_{X_1,X_2}(k_i\ a, b_1, b_2) \rhd b_i[a/X_i]$
E	$E_{v,Y}(\langle\!\langle n, a\rangle\!\rangle, b) \rhd b[n/v, a/Y]$		

Note: We are not considering either permutation conversions nor simplification conversions, nor intuitionistic falsity conversions.

Definition 28 *If $t \rhd s$ then we say that t* contracts *to s; t is called a* redex *or a* contractible *term.*

The replacement of t by s is called a contraction *(of t). We write $t \rhd s$ to mean the result of applying a contraction and $t \rhd_* s$ to mean that there is a sequence of contractions from t ending in s.*

Definition 29 *A* reduction sequence *is a (finite or infinite) sequence of pairs $(t_0, \delta_0), (t_1, \delta_1) \ldots$ where δ_i is an occurrence of a redex in t_i and $t_i \rhd t_{i+1}$ via the contraction of δ_i for all i. This sequence is usually written as $t_0 \rhd^{\delta_0} t_1 \rhd^{\delta_1} \ldots$.*

Finite reduction sequences are partially ordered by their prefix relation. The collection of all reduction sequences starting from t is the reduction tree *(and is actually a tree).*

We generate an equivalence relation from the contraction relation and denote this as $t =_{\rhd} s$ which means either $t \rhd_* s$ or $s \rhd_* t$.

Definition 30 *A term t is in* normal form *if t does not contain a redex. We say that t* has a normal form *is there is a normal s such that $t \rhd_* s$.*

Definition 31 *A term is* strongly normalizing *if its reduction tree is finite.*

Definition 32 *A relation R is said to be* confluent *or have the* Church-Rosser *property if whenever tRs_1 and tRs_2 then there exists a u such that s_1Ru and s_2Ru. A relation R is said to* *be* weakly confluent *or have the* weak Church-Rosser *property if whenever* tRs_1 *and* tRs_2 *then there exists a* u *such that* s_1R_*u *and* s_2R_*u *where* R_* *is the reflexive transitive closure of* R.

2.8.2.1 Confluent and Strongly Normalizing

These two lemmas are standard and can be found in c.f. (Troelstra and Schwichtenberg 1996) pg. 11.

Lemma 33 *For a confluent reduction relation* $\rhd$, *the normal forms of terms are unique. Furthermore, if* $\rhd$ *is a confluent reduction relation then we have:* $t =_{\rhd} t' \iff$ *there is a term* s *such that* $t \rhd_* s$ *and* $t' \rhd_* s$.

Lemma 34 *Let* $\succ_*$ *the reflexive transitive closure of* $\succ$ *and let* $\succ$ *be weakly confluent.*

1. *The normal form with respect to* $\succ$ *of a strongly normalizing term* t *is unique.*
2. *If all terms are strongly normalizing with respect to* $\succ$ *then the relation* $\succ_*$ *is confluent.*

Lemma 35 (Substitutivity) *For the reduction relation* $\rhd$, *we have if* $t \rhd s$ *then* $t[a/X] \rhd s[a/X]$.

Theorem 36 *The reduction relation* $\rhd$ *for* λ_I *is weakly confluent, strongly normalizing and* $\rhd_*$ *is confluent.*

2.8.2.2 Proof Normalization

The Curry-Howard-De Bruijn isomorphism states that proof normalization in propositional intuitionistic logic is isomorphic to reductions in the typed lambda calculus.

We now give the proof normalization rules.

Definition 37 (Proof Normalization/Detour Contractions)

$\wedge$ *Contraction* $\quad \dfrac{\dfrac{\Gamma_1 \Rightarrow a_1 \colon A_1 \quad \Gamma_2 \Rightarrow a_2 \colon A_2}{\Gamma_1, \Gamma_2 \Rightarrow \langle a_1, a_2\rangle \colon A_1 \wedge A_2}}{\Gamma_1, \Gamma_2 \Rightarrow \pi_i(\langle a_1, a_2\rangle) \colon A_i \quad i \in \{1,2\}} \quad$ *conv* $\quad \Gamma_i \Rightarrow a_i \colon A_i \quad i \in \{1,2\}$

$\vee$ *Contraction* $\quad \dfrac{\dfrac{\Gamma \Rightarrow a_i \colon A_i}{\Gamma \Rightarrow k_i \ a_i \colon A_1 \vee A_2} \quad \begin{array}{c} X_1 \colon A_1, \Delta_1 \Rightarrow c_1 \colon C \\ X_2 \colon A_2, \Delta_2 \Rightarrow c_2 \colon C \end{array}}{\Gamma, \Delta_1, \Delta_2 \Rightarrow D_{X_1, X_2}(k_i \ a_i, c_1, c_2) \colon C} \quad$ *conv* $\quad \Gamma, \Delta_i \Rightarrow c_i[a_i/X_i] \colon C$

$\supset$ *Contraction* $\quad \dfrac{\dfrac{X \colon A, \Gamma \Rightarrow b \colon B}{\Gamma \Rightarrow \lambda X.b \colon A \supset B} \quad \Sigma \Rightarrow a \colon A}{\Gamma, \Sigma \Rightarrow (\lambda X.b \ a) \colon B} \quad$ *conv* $\quad \Gamma, \Sigma \Rightarrow b[a/X] \colon B$

$\forall$ *Contraction* $\quad \dfrac{\dfrac{\Gamma \Rightarrow a \colon A}{\Gamma \Rightarrow \boldsymbol{\lambda} v.a \colon \forall v.A}}{\Gamma \Rightarrow (\boldsymbol{\lambda} v.a \ \ n) \colon A[n/v]} \quad$ *conv* $\quad \Gamma \Rightarrow a[n/v] \colon A[n/v]$

$\exists$ *Contraction* $\quad \dfrac{\dfrac{\Gamma \Rightarrow a \colon A[n/v]}{\Gamma \Rightarrow \langle n, a\rangle \colon \exists v.A} \quad \Delta, X \colon A[u/v] \Rightarrow c \colon C}{\Gamma, \Delta \Rightarrow E_{u,X}(\langle n, a\rangle, c) \colon C} \quad$ *conv* $\quad \Gamma \Rightarrow c[n/u, a/X] \colon C$

2.8.2.3 Curry-Howard-deBruijn

We say that d conv d' when a sequence of contractions applied to d produce d'.

Theorem 38 Curry-Howard

1. *For every term t there is a deduction d such that $t = T(d)$ and d is unique up to trivial transformations.*

2. *$T(d) \rhd_* T(d')$ implies that d conv$_*$ d'.*

3. *d conv$_*$ d' implies that $T(d) \rhd_* T(d')$.*

4. *d is normal if and only if $T(d)$ is normal.*

Chapter 3

Constructivity with Negative Information

3.1 Summary

Constructive mathematics preserves the symmetry of Boolean logic's two truth values when it allows a proposition to be undecided: "could be either true or false". Intuitionistic logic destroys this symmetry by focusing on truth and offering only a nonconstructive "negation as failure", $\sim p = p \supset 0$. Nelson's system N of constructible falsity restores symmetry to intuitionistic logic, I, with an involutory *strong* negation $\neg p$ asserting the actual falsity of p. Thomason proposed a Kripke structure interpretation with three truth values for which he shows that N is sound and complete.

We use Kripke models with two truth values for which we show that N is sound and complete. Satisfaction is judged from an interval (w_t, w_f) in Kripke structures where w_f is accessible from w_t. A formula φ holds at an interval if and only if it holds at all non-trivial subintervals. Truth is judged from one end w_t, falsity from the other w_f. We interpret strong negation as dualizing the Kripke structure with respect to both truth and accessibility.

This model theory illuminates the close connection between intuitionistic logic and constructible falsity. Furthermore, the dualities that appear in the logic now manifest themselves in the semantics.

3.2 History

3.2.1 Axiomatics

Nelson (1949) axiomatized Constructible Falsity, later dubbed N, using Hilbert style rules and studied P (Positive) and N (Negative) realizability for his system. The goal of his work was to better capture constructive mathematics. Vorob'ev (1952) and Fitch (1954)independently axiomatized systems that were theoremwise equivalent to N.

The Gentzen treatment of N started with Lopez-Escobar(López-Escobar 1972) who in 1972 developed a single conclusion Gentzen system for N. Then Zaslavskiĭ(1973) used a Gentzen style axiomatization of a symmetric treatment of intuitionistic logic and showed that it was equivalent to N. Gurevich (1977) developed a multiple conclusion Gentzen system for N. Nelson and Almukdad(1984) in 1984 presented a Gentzen multiple conclusion system which incorporated the structural rules.

In other arenas, Goranko (1985), studied interpolation theorems for N. Akama (1988) considered cut-elimination for a re-axiomatization of N called GS. Tanaka (1988) pointed out an error in Akama's arguments for GS. Cut-elimination was already considered by Nelson and Fitch.

Another, proposed symmetrization of intuitionistic logic (Rauszer 1975, Rauszer 1977b) was to add dual operators to those of intuitionistic logic. The dual negation in Rauszer's system satisfies the law of the excluded middle and hence is not a constructive negation in the sense that is does not satisfy the disjunction property. Rauszer's symmetrizations differ greatly from Nelson's and Zaslavskiĭ's symmetrizations of intuitionistic logic which kept the disjunction property in the presence of the additional negation.

Recent uses for N (Fitting 1991, Pearce and Wansing 1992, Wagner 1991, Wansing 1993) have also been found in the realm of logic programming using explicitly "false" facts.

3.2.2 Models

The first model theory for N was developed by Thomason(Thomason 1969), who used a 3-valued Kripke model to give a sound and complete model of propositional N. Thomason considered constant domains in the first-order case for N. Gurevich(Gurevich 1977), used a 3-valued Kripke model for N but extended the result to handle non-constant domains thereby capturing the full first-order version of N. Fidel(Fidel 1979) independently extended Thomason's result similar to Gurevich.

In this chapter, we consider a new model, not based around partial worlds (or 3-valued valuation functions) but instead the "undecided" value or the representation of "uncertainty" is captured through uncertainty of location within a standard Kripke model. This region of uncertainty we call an interval.

Since the models of intuitionistic logic are extended via the examination of intervals in the models, this model theory gives an immediate model theoretic proof that N is a conservative extension of intuitionistic logic.

3.2.3 Algebras

The algebra for N was first constructed by (Rasiowa 1974), who called it a quasi-pseudo boolean algebra, later dubbed N-lattices after Nelson. Klunder studied a topos-based approach to N-lattices (Klunder 1992). Later authors viewed Rasiowa's algebra as a set of restrictions on two copies of an intuitionistic algebra H (pseudo boolean or Heyting algebras) (Vakarelov 1977, Sendlewski 1984, Sendlewski 1989, Sendlewski 1995, Czelakowski 1975). Roughly, one copy of H encoded what things are true when other things are true, and the other copy encoded what things are false when other things are false. To generate an algebra for N, restrictions are needed when gluing these two copies together, because when a propositional atom p is true in point, you don't want to glue this to a point where p is false. If the restrictions go too far you wind up with symmetrization by Rauszer (Rauszer 1975, Rauszer 1977b, Rauszer 1977a) who introduces a negation $-$, but in her logic $A \vee -A$ is valid.

Ginsberg and Fitting (Ginsberg 1988, Fitting 1991) independently studied topological bilattices of which Vakarelov's and our pair algebras are a special case. A topological bilattice is a the product of two topological lattices or two Heyting algebras. In the pair algebra, the points of the two Heyting algebras are the same, therefore results found in (Dold 1992, Fitting 1988, Fitting 1991) can be adapted to either Vakarelov's algebra or our pair algebra.

What we do here is complement the recent ideas of the algebraicists by studying how their separation of truth and falsity and their re-use of the intuitionistic structure can manifest itself in the model theory. Further, we consider that since falsity is the dual of truth perhaps we can reason about counterexamples in the dual structure or by reasoning backwards along the accessibility relation. Herein we provide a model theoretic account of that interpretation which is sound and complete for constructible falsity.

The logic of N is a conservative extension of intuitionistic logic. We do not repeat the Gentzen rules for propositional intuitionistic logic here; they are in the previous chapter.

The syntax of N is that atomic formulas are elements drawn from a set of propositional atoms; a formula of N is either an atom, or the conjunction $\wedge$ of two formulas, the disjunction $\vee$ of two formulas, the intuitionistic negation $\sim$ of a formula, the *strong* negation $\neg$ of a formula, or an implication $\supset$ parameterized by two formulas. A truth value of 1 represents true and 0 represents false.

Gentzen System Sequents $\Gamma \Rightarrow \Delta$ should be read $\wedge\Gamma \supset \vee\Delta$, where the set $\Gamma \cup \Delta$ is a finite set of formulas, Γ, φ is the set Γ with the single formula φ added. The rules are divided into left and right rules according to whether the main operator is introduced on the left or on the right (respectively).

Axiom

$$(2)\varphi, \neg\varphi, \Gamma \Rightarrow \Delta$$

Rules of Inference

$$\neg\neg \Rightarrow \quad \frac{\varphi, \Gamma \Rightarrow \Delta}{\neg\neg\varphi, \Gamma \Rightarrow \Delta} \qquad \Rightarrow \neg\neg \quad \frac{\Gamma \Rightarrow \Delta, \varphi}{\Gamma \Rightarrow \Delta, \neg\neg\varphi}$$

$$\neg\wedge \Rightarrow \quad \frac{\neg\varphi, \Gamma \Rightarrow \Delta \quad \neg\psi, \Gamma \Rightarrow \Delta}{\Gamma, \neg(\varphi \wedge \psi) \Rightarrow \Delta} \qquad \Rightarrow \neg\wedge \quad \frac{\Gamma \Rightarrow \Delta, \neg\varphi, \neg\psi}{\Gamma \Rightarrow \Delta, \neg(\varphi \wedge \psi)}$$

$$\neg\vee \Rightarrow \quad \frac{\neg\varphi, \neg\psi, \Gamma \Rightarrow \Delta}{\neg(\varphi \vee \psi), \Gamma \Rightarrow \Delta} \qquad \Rightarrow \neg\vee \quad \frac{\Gamma \Rightarrow \Delta, \neg\varphi \quad \Gamma \Rightarrow \Delta, \neg\psi}{\Gamma \Rightarrow \Delta, \neg(\varphi \vee \psi)}$$

$$\neg\supset \Rightarrow \quad \frac{\varphi, \neg\psi, \Gamma \Rightarrow \Delta}{\neg(\varphi \supset \psi), \Gamma \Rightarrow \Delta} \qquad \Rightarrow \neg\supset \quad \frac{\Gamma \Rightarrow \Delta, \varphi \quad \Gamma \Rightarrow \Delta, \neg\psi}{\Gamma \Rightarrow \Delta, \neg(\varphi \supset \psi)}$$

All of these rules are invertible. Further $\varphi \vee \neg\varphi$ is not valid in this logic even though $\neg(\varphi \wedge \psi) \equiv \neg\varphi \vee \neg\psi$. We use the equivalent representation of intuitionistic negation as $\sim \varphi = \varphi \supset \neg\varphi$ rather than having intuitionistic negation as primitive.

3.3 Interval Kripke Structures for Constructible Falsity

Thomason proposed a Kripke structure interpretation with three truth values for which he shows that N is sound and complete. The model lacks certain symmetries that appear in the logic and furthermore moves away from standard Kripke structures, to use partial valuations or 3-valued models.

We propose a new model for N which sheds light on its relationship to intuitionistic logic and gives an immediate proof of the conservitivity of N over positive intuitionistic logic.

Now we define the dual of a Kripke structure.

Definition 39 *Let $M = \langle W, R, V \rangle$ be a propositional intuitionistic model over P. We define $M^{\perp} = \langle W, R^{\cup}, \bar{V} \rangle$ where $R^{\cup}$ is the inverse of R and for all $w \in W$. $\bar{V}(w) = P - V(w)$.*

Definition 40 *An interval model for N is a propositional intuitionistic model $M = \langle W, R, V \rangle$ over P from Definition 1 with the entailment relation $\models_M$ for truth and $\Vert\!\models_{M^{\perp}}$ for falsity defined by recursion on φ. For any $w_p, w_n \in W$. such that $w_p\ Rw_n$ and $w_p \neq w_n$.*

$$
\begin{array}{lllr}
p & (w_p, w_n) \models_M p & \equiv & p \in V(w_p) \\
\varphi \wedge \psi & (w_p, w_n) \models_M \varphi \wedge \psi & \equiv & (w_p, w_n) \models_M \varphi \textit{ and } (w_p, w_n) \models_M \psi \\
\varphi \vee \psi & (w_p, w_n) \models_M \varphi \vee \psi & \equiv & (w_p, w_n) \models_M \varphi \textit{ or } (w_p, w_n) \models_M \psi \\
\varphi \supset \psi & (w_p, w_n) \models_M \varphi \supset \psi & \equiv & \forall w', w'' \in W.w' \neq w'' \wedge w_p\ R\ w'\ R\ w''\ R\ w_n \wedge \\
 & & & (w', w'') \models_M \varphi \textit{ implies } (w', w'') \models_M \psi \\
\neg\varphi & (w_p, w_n) \models_M \neg\varphi & \equiv & (w_n, w_p) \Vert\!\models_{M^{\perp}} \varphi \\
p & (w_n, w_p) \Vert\!\models_{M^{\perp}} p & \equiv & p \notin V(w_n) \\
\varphi \wedge \psi & (w_n, w_p) \Vert\!\models_{M^{\perp}} \varphi \wedge \psi & \equiv & (w_n, w_p) \Vert\!\models_{M^{\perp}} \varphi \textit{ or } (w_n, w_p) \Vert\!\models_{M^{\perp}} \psi \\
\varphi \vee \psi & (w_n, w_p) \Vert\!\models_{M^{\perp}} \varphi \vee \psi & \equiv & (w_n, w_p) \Vert\!\models_{M^{\perp}} \varphi \textit{ and } (w_n, w_p) \Vert\!\models_{M^{\perp}} \psi \\
\varphi \supset \psi & (w_n, w_p) \Vert\!\models_{M^{\perp}} \varphi \supset \psi & \equiv & (w_p, w_n) \models_M \varphi \textit{ and } (w_n, w_p) \Vert\!\models_{M^{\perp}} \psi \\
\neg\varphi & (w_n, w_p) \Vert\!\models_{M^{\perp}} \neg\varphi & \equiv & (w_p, w_n) \models_M \varphi
\end{array}
$$

Definition 41 *Let $M = \langle W, R, V \rangle$ be a propositional Kripke structure. A formula φ of propositional N is* satisfied *in M if and only if $\exists w_p, w_n \in W$. $w_p \neq w_n$ and $w_p R w_n$ and $(w_p, w_n) \models_M \varphi$. A propositional formula φ is* valid *in M if and only if $\forall w_p, w_n \in W$. $w_p \neq$*

w_n *and* $w_p R w_n$ *and* $(w_p, w_n) \models_M \varphi$. *A propositional* N *formula* φ *is* valid *if and only if* φ *is valid in all propositional Kripke structures.*

Lemma 42 Monotonicity *Let* $M = \langle W, R, V \rangle$ *be a propositional intuitionistic model over* P. *Then we have for all* $w_p, w', w'', w_n \in W$. *such that* $w_p \, R \, w' \, R \, w'' \, R \, w_n$ *and* $w_p \neq w_n$ *and* $w' \neq w''$, *and for all formula* φ *of* N,

Truth $(w_p, w_n) \models_M \varphi$ *implies* $(w', w'') \models_M \varphi$, *and*

Falsity $(w_n, w_p) \Vdash_{M^\perp} \varphi$ *implies* $(w'', w') \Vdash_{M^\perp} \varphi$.

Proof: By induction on the structure of φ.

- Basis p **Truth**: $(w_p, w_n) \models_M p \rightarrow (w', w'') \models_M p$. This follows from the monotonicity of V in definition of M.

  Basis p **Falsity**: $(w_n, w_p) \Vdash_{M^\perp} p \rightarrow (w'', w') \Vdash_{M^\perp} p$. This follows from the monotonicity of V in the definition of M and $w_p R w_n$.

- $\varphi \wedge \psi$ **Truth**: $(w_p, w_n) \models_M \varphi \wedge \psi \rightarrow (w', w'') \models_M \varphi \wedge \psi$.

  By the induction hypothesis $(w_p, w_n) \models_M \varphi \rightarrow (w', w'') \models_M \varphi$ and also $(w_p, w_n) \models_M \psi \rightarrow (w', w'') \models_M \psi$

  Thus $(w', w'') \models_M \varphi \wedge \psi$ by the induction hypothesis and the definition of $\models_M$.

  $\varphi \wedge \psi$ **Falsity**: $(w_n, w_p) \Vdash_{M^\perp} \varphi \wedge \psi \rightarrow (w'', w') \Vdash_{M^\perp} \varphi \wedge \psi$.

  By the induction hypothesis $(w_n, w_p) \Vdash_{M^\perp} \varphi \rightarrow (w'', w') \Vdash_{M^\perp} \varphi$ or $(w_n, w_p) \Vdash_{M^\perp} \psi \rightarrow (w'', w') \Vdash_{M^\perp} \psi$

  Thus $(w'', w') \Vdash_{M^\perp} \varphi \wedge \psi$ by the induction hypothesis and the definition of $\Vdash_{M^\perp}$.

- $\varphi \vee \psi$ **Truth**: $(w_p, w_n) \models_M \varphi \vee \psi \rightarrow (w', w'') \models_M \varphi \vee \psi$.

  By the induction hypothesis $(w_p, w_n) \models_M \varphi \rightarrow (w', w'') \models_M \varphi$ or $(w_p, w_n) \models_M \psi \rightarrow (w', w'') \models_M \psi$

  Thus $(w', w'') \models_M \varphi \vee \psi$ by the induction hypothesis and the definition of $\models_M$.

  $\varphi \vee \psi$ **Falsity**: $(w_n, w_p) \Vdash_{M^\perp} \varphi \vee \psi \rightarrow (w'', w') \Vdash_{M^\perp} \varphi \vee \psi$.

  By the induction hypothesis $(w_n, w_p) \Vdash_{M^\perp} \varphi \rightarrow (w'', w') \Vdash_{M^\perp} \varphi$ and $(w_n, w_p) \Vdash_{M^\perp} \psi \rightarrow (w'', w') \Vdash_{M^\perp} \psi$

Thus $(w'', w')\|\!\!=_{M^\perp}\varphi \vee \psi$ by the induction hypothesis and the definition of $\|\!\!=_{M^\perp}$.

- $\varphi \supset \psi$ **Truth**: $(w_p, w_n) \models_M \varphi \supset \psi \rightarrow (w', w'') \models_M \varphi \supset \psi$.

  We assume that $\forall w_1, w_2.w_pRw_1Rw_2Rw_n.\ (w_1, w_2) \models_M \varphi \rightarrow (w_1, w_2) \models_M \psi$.

  Thus by transitivity of R and monotonicity $(w', w'') \models_M \varphi \supset \psi$. All this case says is that since $\varphi \rightarrow \psi$ in all intervals, then this must be the case for (w', w'').

  $\varphi \supset \psi$ **Falsity**: $(w_n, w_p)\|\!\!=_{M^\perp}\varphi \supset \psi \rightarrow (w'', w')\|\!\!=_{M^\perp}\varphi \supset \psi$.

  By the induction hypotheses $(w_p, w_n) \models_M \varphi \rightarrow (w', w'') \models_M \varphi$ and $(w_n, w_p)\|\!\!=_{M^\perp}\psi \rightarrow (w'', w')\|\!\!=_{M^\perp}\psi$

  Thus $(w'', w')\|\!\!=_{M^\perp}\varphi \supset \psi$ by the induction hypothesis and the definition of $\|\!\!=_{M^\perp}$ and the fact that w_pRw_n.

- $\neg\varphi$ **Truth**: $(w_p, w_n) \models_M \neg\varphi \rightarrow (w', w'') \models_M \neg\varphi$. This can be rewritten as $(w_n, w_p)\|\!\!=_{M^\perp} \varphi \rightarrow (w'', w')\ \|\!\!=_{M^\perp}\varphi$. However, this is the induction hypothesis for monotonicity of falsity.

  $\neg\varphi$ **Falsity**: $(w_n, w_p)\|\!\!=_{M^\perp}\neg\varphi \rightarrow (w'', w')\|\!\!=_{M^\perp}\neg\varphi$. This can be rewritten as $(w_p, w_n) \models_M \varphi \rightarrow (w', w'') \models_M \varphi$. However, this is the induction hypothesis for monotonicity of truth.

∎

Theorem 43 Soundness

All rules of N are sound.

That is if M is any Kripke structure, then $\forall Rule \in N.\ \forall w_n, w_p \in W.\ w_pRw_n \wedge w_n \neq w_p$.

$$(w_p, w_n) \models_M \mathit{Antecedent}(\mathit{Rule})\ \mathit{implies}\ (w_p, w_n) \models_M \mathit{Succedent}(\mathit{Rule})$$

Proof: We need to consider all rules and show that when the antecedent is true, the succedent is true. To show $(w_p, w_n) \models_M \Gamma \Rightarrow \Delta$ we must show $(w_p, w_n) \models_m \wedge\Gamma \supset \vee\Delta$.

$\rightarrow \wedge\frac{\Gamma\Rightarrow\Delta,\varphi \quad \Gamma\Rightarrow\Delta,\psi}{\Gamma\Rightarrow\Delta,\varphi\wedge\psi}$ Assume $(w_p, w_n) \models_M \Gamma \Rightarrow \Delta, \varphi$ and also that $(w_p, w_n) \models_M \Gamma \Rightarrow \Delta, \varphi$.

By monotonicity our hypotheses hold for all subintervals w', w'' i.e. $w_pRw'Rw''Rw_n$ where $w' \neq w''$. Then our cases become $\forall w_1, w_2.w_pRw_1Rw_2Rw_n$ where $w_1 \neq w_2$,

- $(w_1, w_2) \models_M \vee\Delta$, then we are trivially finished.
- **not** $(w_1, w_2) \models_M \wedge\Gamma$, then we are trivially finished.
- $(w_1, w_2) \models_M \wedge\Gamma$ **and not** $(w_1, w_2) \models_M \vee\Delta$. From these hypothesis we then conclude $(w_1, w_2) \models_M \varphi$ and $(w_1, w_2) \models_M \psi$, thus we have $(w_1, w_2) \models_m \varphi \wedge \psi$.

$\wedge \rightarrow \frac{\varphi,\psi,\Gamma\Rightarrow\Delta}{\varphi\wedge\psi,\Gamma\Rightarrow\Delta}$ Assume $(w_p, w_n) \models_M \varphi, \psi, \Gamma \Rightarrow \Delta$.

By monotonicity our hypothesis holds for all subintervals, then our cases become $\forall w_1, w_2$. $w_p R w_1 R w_2 R w_n$ where $w_1 \neq w_2$,

- $(w_1, w_2) \models_M \vee\Delta$, then we are trivially finished.
- **not** $(w_1, w_2) \models_M \wedge\Gamma$, then we are trivially finished.
- **not** $(w_1, w_2) \models_M \varphi$, then we are trivially finished.
- **not** $(w_1, w_2) \models_M \psi$, then we are trivially finished.
- $(w_1, w_2) \models_M \varphi$ **and** $(w_1, w_2) \models_M \psi$ **and** $(w_1, w_2) \models_M \Gamma$ From these hypothesis we can conclude $(w_1, w_2) \models_M \varphi \wedge \psi, \Gamma \Rightarrow \Delta$.

$\vee \rightarrow \frac{\varphi,\Gamma\Rightarrow\Delta \quad \psi,\Gamma\Rightarrow\Delta}{\varphi\vee\psi,\Gamma\Rightarrow\Delta}$ This case is similar to $\wedge \rightarrow$.

$\rightarrow \vee \frac{\Gamma\Rightarrow\Delta,\varphi,\psi}{\Gamma\Rightarrow\Delta,\varphi\vee\psi}$ This case is similar to $\rightarrow \wedge$.

$\supset\rightarrow \frac{\varphi\supset\psi,\Gamma\Rightarrow\Delta,\varphi \quad \psi,\Gamma\Rightarrow\Delta}{\varphi\supset\psi,\Gamma\Rightarrow\Delta}$ Assume that $(w_p, w_n) \models_M \varphi \supset \psi, \Gamma \Rightarrow \Delta, \varphi$ and also that $(w_p, w_n) \models_M \psi, \Gamma \Rightarrow \Delta$.

By monotonicity our hypothesis holds for all subintervals then our cases become $\forall w_1, w_2$. $w_p R w_1 R w_2 R w_n$ where $w_1 \neq w_2$,

- $(w_1, w_2) \models_M \vee\Delta$, then we are trivially finished.
- **not** $(w_1, w_2) \models_M \wedge\Gamma$, then we are trivially finished.
- **not** $(w_1, w_2) \models_M \varphi \supset \psi$, then we are trivially finished.
- $(w_1, w_2) \models_M \varphi \supset \psi$**, and** $(w_1, w_2) \models_M \Gamma$.

  From these hypothesis we can conclude $(w_1, w_2) \models_M \varphi$ where we use our first hypothesis to conclude that $(w_1, w_2) \models_M \psi$ and thus $(w_1, w_2) \models_M \Delta$. From this information we conclude that $(w_1, w_2) \models_M \varphi \supset \psi, \Gamma \Rightarrow \Delta$.

$\rightarrow\supset \frac{\varphi,\Gamma\Rightarrow\psi}{\Gamma\Rightarrow\Delta,\varphi\supset\psi}$ $(w_p, w_n) \models_M \varphi, \Gamma \Rightarrow \varphi$.

By monotonicity our hypothesis holds for all subintervals then our cases become $\forall w_1, w_2$. $w_p R w_1 R w_2 R w_n$ where $w_1 \neq w_2$,

- $\circ$ $(w_1, w_2) \models_M \psi$, then we are trivially finished.
- $\circ$ **not** $(w_1, w_2) \models_M \wedge\Gamma$, then we are trivially finished.
- $\circ$ **not** $(w_1, w_2) \models_M \varphi$, then we are trivially finished.
- $\circ$ $(w_1, w_2) \models_M \varphi$ **and also that** $(w_1, w_2) \models_M \Gamma$, then $(w_1, w_2) \models_M \psi$, thus $(w_1, w_2) \models \varphi \supset \psi$. Since R is transitive and our hypotheses we arrive at $(w_p, w_n) \models_M \Gamma \Rightarrow \Delta, \varphi \supset \psi$

$\neg\wedge\rightarrow \frac{\Gamma,\neg\varphi\Rightarrow\Delta \quad \Gamma,\neg\psi\Rightarrow\Delta}{\Gamma,\neg(\varphi\wedge\psi)\Rightarrow\Delta}$ This case is similar to $\wedge\rightarrow$.

$\rightarrow\neg\wedge \frac{\Gamma\Rightarrow\Delta,\neg\varphi,\neg\psi}{\Gamma\Rightarrow\Delta,\neg(\varphi\wedge\psi)}$ This case is similar to $\rightarrow\wedge$.

$\rightarrow\neg\vee \frac{\Gamma\Rightarrow\Delta,\neg\varphi \quad \Gamma\Rightarrow\Delta,\neg\psi}{\Gamma\Rightarrow\Delta,\neg(\varphi\vee\psi)}$ This case is similar to $\wedge\rightarrow$.

$\neg\vee\rightarrow \frac{\neg\varphi,\neg\psi,\Gamma\Rightarrow\Delta}{\neg(\varphi\vee\psi),\Gamma\Rightarrow\Delta}$ This case is similar to $\wedge\rightarrow$.

$\rightarrow\neg\supset \frac{\Gamma\Rightarrow\Delta,\varphi \quad \Gamma\Rightarrow\Delta,\neg\psi}{\Gamma\Rightarrow\Delta,\neg(\varphi\supset\psi)}$ This case is similar to $\rightarrow\wedge$. This case is similar to $\rightarrow\wedge$.

$\neg\supset\rightarrow \frac{\varphi,\neg\psi,\Gamma\Rightarrow\Delta}{\Gamma,\neg(\varphi\supset\psi)\Rightarrow\Delta}$ This case is similar to $\wedge\rightarrow$.

$\neg\neg\rightarrow \frac{\varphi,\Gamma\Rightarrow\Delta}{\neg\neg\varphi,\Gamma\Rightarrow\Delta}$ Assume that $(w_p, w_n) \models_M \varphi, \Gamma \Rightarrow \Delta$.

By monotonicity our hypothesis holds for all subintervals then our cases become $\forall w_1, w_2$. $w_p R w_1 R w_2 R w_n$ where $w_1 \neq w_2$,

- $\circ$ $(w_1, w_2) \models_M \Delta$, then we are trivially finished.
- $\circ$ **not** $(w_1, w_2) \models_M \wedge\Gamma$, then we are trivially finished.
- $\circ$ $(w_1, w_2) \models_M \varphi$ **and also that** $(w_1, w_2) \models_M \Gamma$, then $(w_1, w_2) \models_M \Delta$, thus we need to show that $(w_1, w_2) \models \neg\neg\varphi$ which is simply $(w_2, w_1) \| \models_{M^\perp} \neg\varphi$ which in turn is $(w_1, w_2) \models_M \varphi$ which was our hypothesis, thus $(w_p, w_n) \models_M \neg\neg\varphi, \Gamma \Rightarrow \Delta$.

$\rightarrow\neg\neg \frac{\Gamma\Rightarrow\Delta,\varphi}{\Gamma\Rightarrow\Delta,\neg\neg\varphi}$ Assume that $(w_p, w_n) \models_M \Gamma \Rightarrow \Delta, \varphi$.

By monotonicity our hypothesis holds for all subintervals then our cases become $\forall w_1, w_2$. $w_p R w_1 R w_2 R w_n$ where $w_1 \neq w_2$,

∘ $(w_1, w_2) \models_M \Delta$, then we are trivially finished.

∘ **not** $(w_1, w_2) \models_M \wedge\Gamma$, then we are trivially finished.

∘ $(w_1, w_2) \models_M \Gamma$ **and not** $(w_1, w_2) \models_M \Delta$, then we have $(w_1, w_2) \models_M \varphi$. From this we need to show that $(w_1, w_2) \models \neg\neg\varphi$ which is simply $(w_2, w_1) \| \models_{M^\perp} \neg\varphi$ which in turn is $(w_1, w_2) \models_M \varphi$ which was our hypothesis. Thus $(w_p, w_n) \models_M \Gamma \Rightarrow \Delta, \neg\neg\varphi$.

∎

Lemma 44 Cut *The cut rule* $\dfrac{\Gamma \Rightarrow \Delta, \varphi \quad \varphi, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta}$ *is sound.*

Assume that $(w_p, w_n) \models_M \Gamma \Rightarrow \Delta, \varphi$. By monotonicity our hypothesis holds for all subintervals then our cases become $\forall w_1, w_2. w_p R w_1 R w_2 R w_n$ where $w_1 \neq w_2$,

∘ $(w_1, w_2) \models_M \Delta$, then we are trivially finished.

∘ **not** $(w_1, w_2) \models_M \wedge\Gamma$, then we are trivially finished.

∘ $(w_1, w_2) \models_M \Gamma$ **and not** $(w_1, w_2) \models_M \varphi$, then we have $(w_1, w_2) \models_M \Delta$ and thus (w_p, w_n) $\models_M \Gamma \Rightarrow \Delta$.

∘ $(w_1, w_2) \models_M \Gamma$ **and** $(w_1, w_2) \models_M \varphi$, then we have that $(w_1, w_2) \models_M \Delta$ from the right hand premise and thus $(w_p, w_n) \models_M \Gamma \Rightarrow \Delta$.

Contraction and Weakening are sound.

3.3.1 Completeness

Our completeness theorem in this section asserts that any underivable formula is refutable in a canonical model.

The argument in the section is summarized as follows. Any complete underivable sequent $\Gamma \Rightarrow \Delta$ in N can be syntactically partitioned into $\Gamma_p, \Gamma_n, \Gamma_o \Rightarrow \Delta_p, \Delta_n, \Delta_o$ where each φ in $\Gamma_p \cup \Delta_p$ is in I_p, and each φ in $\Gamma_n \cup \Delta_n$ is a formula of the form $\neg\psi$ where ψ is in I_p. Then $\Gamma_p \Rightarrow \Delta_p$ is a complete sequent for I_p; hence $\Gamma_p \Rightarrow \Delta_p$ is a world in $\mathcal{K}$. The formula-wise contrapositive of $\Gamma_n \Rightarrow \Delta_n$, denoted $\neg\Delta_n \Rightarrow \neg\Gamma_n$, is also a world in K. These are the two worlds in $\mathcal{K}$ that refute $\Gamma \Rightarrow \Delta$.

Definition 45 *Let $\Gamma \Rightarrow \Delta$ be an infinite sequent in N. Then $\Gamma \Rightarrow \Delta$ is* derivable *in N if there exist finite subsets $\Gamma_f \subset \Gamma$ and $\Delta_f \subset \Delta$ such that $\Gamma_f \Rightarrow \Delta_f$ is derivable in N respectively.*

Definition 46 *We say that a sequent in N, $\Gamma \Rightarrow \Delta$ is* falsified *in a structure M if there exist two worlds $w_p, w_n \in W$ such that $w_p \; R \; w_n$, $w_p \neq w_n$, $(w_p, w_n) \models_M \wedge\Gamma$, and $(w_p, w_n) \not\models_M \vee\Delta$.*

Definition 47 *Let Θ be a set of formulas. We define $\neg\Theta = \{\neg\varphi \mid \varphi \in \Theta \subseteq I_p\}$ and we overload that $\neg\Theta$ is also $\neg\Theta = \{\varphi \mid \neg\varphi \in \Theta \subseteq \neg I_p\}$.*

Lemma 48 *If Θ is a set of formulas of N, then Θ can be syntactically partitioned into $\Theta_p \cup \Theta_n \cup \Theta_o$ where Θ_p is a set of formulas in I_p, Θ_n is a set of formulas in $\neg I_p$ and Θ_o is disjoint from $I_p \cup \neg I_p$.*

Lemma 49 *For any complete sequent $\Gamma_p, \Gamma_n, \Gamma_o \Rightarrow \Delta_p, \Delta_n, \Delta_o$ for N, $\Gamma_p \Rightarrow \Delta_p$ is a complete sequent for positive propositional intuitionistic logic I_p.*

Proof: $\Gamma \Rightarrow \Delta$ is underivable, thus every subset of $\Gamma \Rightarrow \Delta$ is underivable hence $\Gamma_p \Rightarrow \Delta_p$ is underivable.

To see that $\Gamma_p \Rightarrow \Delta_p$ is complete in I_p, we know that $I_p \subsetneq N$ and that $\Gamma_p, \Gamma_n, \Gamma_o \Rightarrow \Delta_p, \Delta_n, \Delta_o$ is complete for N. Therefore, every formula in I_p occurs somewhere in $\Gamma_p, \Gamma_n, \Gamma_o \Rightarrow \Delta_p, \Delta_n, \Delta_o$. By the syntactic form of the formulas in I_p they can only occur in $\Gamma_p \cup \Delta_p$. ∎

Corollary 50 *$\Gamma_p \Rightarrow \Delta_p$ is a world in $\mathcal{K}$.*

Lemma 51 *For any complete sequent $\Gamma_p, \Gamma_n, \Gamma_o \Rightarrow \Delta_p, \Delta_n, \Delta_o$ for N, $\neg\Delta_n \Rightarrow \neg\Gamma_n$ is a complete sequent for I_p i.e. $\neg\Delta_n \Rightarrow \neg\Gamma_n$ is a world in $\mathcal{K}$.*

Proof: The arguments for the completeness of $\Gamma_n \Rightarrow \Delta_n$ for $\neg I_p$ are the same as for $\Gamma_p \Rightarrow \Delta_p$ for I_p. By the syntactic structure $\neg\Delta_n \Rightarrow \neg\Gamma_n$ could only be derivable if $\Gamma_n \Rightarrow \Delta_n$ were derivable; $\Gamma_n \Rightarrow \Delta_n$ is not derivable. Further $\neg\Delta_n \Rightarrow \neg\Gamma_n$ is complete for I_p because $\Gamma_n \Rightarrow \Delta_n$ is complete for $\neg I_p$. ∎

The lemma states that it is possible to use $\Gamma_p \Rightarrow \Delta_p$ and $\neg\Delta_n \Rightarrow \neg\Gamma_n$ as the two worlds in $\mathcal{K}$ at which to judge a formula in N because they form an interval.

Lemma 52 *Let $\Gamma_p, \Gamma_n, \Gamma_o \Rightarrow \Delta_p, \Delta_n, \Delta_o$ be any complete sequent for N where $\Gamma_p \neq \neg\Delta_n$. Then the world $\neg\Delta_n \Rightarrow \neg\Gamma_n$ is accessible from $\Gamma_p \Rightarrow \Delta_p$ in $\mathcal{K}$.*

Proof: We must check $\Gamma_p \subset \neg\Delta_n$ to ensure that $(\Gamma_p \Rightarrow \Delta_p)\ R_k\ (\neg\Delta_n \Rightarrow \neg\Gamma_n)$ in $\mathcal{K}$.

Every formula $\varphi \in \Gamma_p$ is satisfied, thus $\neg\varphi \in \Delta_n$ and thus $\varphi \in \neg\Delta_n$; therefore $\Gamma_p \subseteq \neg\Delta_n$. Our hypothesis states that $\Gamma_p \neq \neg\Delta_n$; hence $\Gamma_p \subset \neg\Delta_n$. ∎

In the technical phase of the argument we show that for a given complete sequent $\Gamma \Rightarrow \Delta$, $\Gamma_p \Rightarrow \Delta_p$ and $\neg\Delta_n \Rightarrow \neg\Gamma_n$ refutes $\Gamma \Rightarrow \Delta$ in $\mathcal{K}$. We use saturatedness of sequents in order to obtain this result. Since N only adds invertible rules, we must check that any N-sequent is saturated for invertible rules.

Lemma 53 *Any complete sequent $\Gamma \Rightarrow \Delta$ for N is saturated. That is it satisfies Definition 81 and also satisfies for all $\varphi, \psi \in N$,*

Rule	*Sequent*		*Saturatedness*
$\neg\wedge \Rightarrow$	$\neg(\varphi \wedge \psi) \in \Gamma$	*implies*	$\neg\varphi \in \Gamma$ *or* $\neg\psi \in \Gamma$
$\Rightarrow \neg\vee$	$\neg(\varphi \vee \psi) \in \Delta$	*implies*	$\neg\varphi \in \Delta$ *or* $\neg\psi \in \Delta$
$\neg\vee \Rightarrow$	$\neg(\varphi \vee \psi) \in \Gamma$	*implies*	$\neg\varphi \in \Gamma$ *and* $\neg\psi \in \Gamma$
$\Rightarrow \neg \supset$	$\neg(\varphi \supset \psi) \in \Delta$	*implies*	$\varphi \in \Delta$ *or* $\neg\psi \in \Delta$
$\neg \supset\Rightarrow$	$\neg(\varphi \supset \psi) \in \Gamma$	*implies*	$\varphi \in \Gamma$ *and* $\neg\psi \in \Gamma$
$\Rightarrow \neg\neg$	$\varphi \in \Delta$	*implies*	$\neg\neg\varphi \in \Delta$
$\neg\neg \Rightarrow$	$\varphi \in \Gamma$	*implies*	$\neg\neg\varphi \in \Gamma$

Lemma 54 *The set of complete sequents for N is saturated.*

Outline of Proof: The only non-invertible rule is implication on the right hand side, hence Lemma 10 and Lemma 53 suffice. ∎

Now armed with the information that each world in $\mathcal{K}$ is saturated and that the set of worlds in $\mathcal{K}$ is saturated for I_p and that each complete N sequent $\Gamma \Rightarrow \Delta$ is saturated, we show that the interval $(\Gamma_p \Rightarrow \Delta_p, \neg\Delta_n \Rightarrow \neg\Gamma_n)$ in $\mathcal{K}$ falsifies the sequent $\Gamma \Rightarrow \Delta$ where $\Gamma_p \cup \Delta_p \cup \Gamma_n \cup \Delta_n \subsetneq \Gamma \cup \Delta$.

We do this in three lemmas:

1. The interval $(\Gamma_p \Rightarrow \Delta_p, \neg\Delta_n \Rightarrow \neg\Gamma_n)$ refutes $\Gamma_p \Rightarrow \Delta_p$.

2. The interval $(\Gamma_p \Rightarrow \Delta_p, \neg\Delta_n \Rightarrow \neg\Gamma_n)$ refutes $\Gamma_n \Rightarrow \Delta_n$.

3. This suffices to refute $\Gamma \Rightarrow \Delta$.

This lemma states that the interval refutes the positive part of the sequent.

Lemma 55 *For all complete sequents $\Gamma_p, \Gamma_n, \Gamma_o \Rightarrow \Delta_p, \Delta_n, \Delta_o$ in N such that $\Gamma_p \neq \neg\Delta_n$, the pair $(\Gamma_p \Rightarrow \Delta_p, \neg\Delta_n \Rightarrow \neg\Gamma_n)$ falsifies $\Gamma_p \Rightarrow \Delta_p$ in $\mathcal{K}$.*

Outline of Proof: By Lemma 52 we may use $(\Gamma_p \Rightarrow \Delta_p, \neg\Delta_n \Rightarrow \neg\Gamma_n)$ as our pair of worlds.

We need simultaneous induction on the positive formula θ such that for every $\theta \in \Gamma_p$ $(\Gamma_p \Rightarrow \Delta_p, \neg\Delta_n \Rightarrow \neg\Gamma_n) \models_K \theta$ and for every $\theta \in \Delta_p$, $(\Gamma_p \Rightarrow \Delta_p, \neg\Delta_n \Rightarrow \neg\Gamma_n) \not\models_K \theta$.

We use $\Gamma_p \Rightarrow \Delta_p \not\models_{\mathcal{K}} \Gamma_p \Rightarrow \Delta_p$ from Theorem 80 for all cases except implication. Since we are looking at subintervals, saturation of the set of complete sequents from Lemma 10 is not enough. We must check that $\neg\Delta_n \Rightarrow \neg\Gamma_n$ is *high* enough above the positive world to satisfy the needed hypotheses in Lemma 10.

$\theta = \varphi \supset \psi$: If $\theta \in \Gamma_p$, then by saturation, monotonicity and the definition of a positive formula $\psi \in \Gamma'$ or $\varphi \in \Delta'$ for every $\Gamma' \Rightarrow \Delta'$ above $\Gamma_p \Rightarrow \Delta_p$ in $\mathcal{K}$. By Lemma 52, $\neg\Delta_n \Rightarrow \neg\Gamma_n$ is above $\Gamma \Rightarrow \Delta$ in $\mathcal{K}$, thus $\varphi \in \neg\Gamma_n$ or $\psi \in \neg\Delta_n$. Hence at every subinterval, $(\Gamma_p \Rightarrow \Delta_p, \neg\Delta_n \Rightarrow \neg\Gamma_n) \models_K \varphi \supset \psi$ as needed.

If $\theta \in \Delta_p$ then by saturation Lemma 10 and the definition of a positive formula we have have some $\Gamma' \Rightarrow \Delta'$ reachable from $\Gamma_p \Rightarrow \Delta_p$ where $\varphi \in \Gamma'$ and $\psi \in \Delta'$.

To ensure that $\Gamma' \Rightarrow \Delta'$ happens in the interval, we use the monotonicity built into R_k and show that (1) $\varphi \in \neg\Delta_n$ and (2) that ψ is in Δ_p.

(1) $\neg\varphi$ cannot occur in Γ_n without satisfying θ, thus $\neg\varphi$ must be in Δ_n, i.e. $\varphi \in \neg\Delta_n$.

(2) ψ cannot be in Γ_p without satisfying θ, therefore $\psi \in \Delta_p$.

So the bottom of the interval has ψ in the RHS of the sequent, and the top of the interval has φ in the LHS of the sequent, so we use Lemma 10 and the monotonicity built into R_k to conclude that there must be a world $\Gamma' \Rightarrow \Delta'$ in the interval such that $\varphi \in \Gamma'$ and $\psi \in \Delta'$. One such world is enough to refute $\varphi \supset \psi$ over the entire interval hence $(\Gamma_p \Rightarrow \Delta_p, \neg\Delta_n \Rightarrow \neg\Gamma_n) \not\models_K \varphi \supset \psi$. ∎

This lemma states that the constructed pair of worlds refutes the negative formulas in the sequent.

Lemma 56 *For all complete sequents* $\Gamma_p, \Gamma_n, \Gamma_o \Rightarrow \Delta_p, \Delta_n, \Delta_o$ *for* N *such that* $\Gamma_p \neq \neg\Delta_n$*, the pair* $(\Gamma_p \Rightarrow \Delta_p, \neg\Delta_n \Rightarrow \neg\Gamma_n)$ *falsifies* $\Gamma_n \Rightarrow \Delta_n$ *in* $\mathcal{K}$*.*

Proof: We use simultaneous induction on the negative formulas θ such that for every $\theta \in \Gamma_n$ $(\Gamma_p \Rightarrow \Delta_p, \neg\Delta_n \Rightarrow \neg\Gamma_n) \models_K \theta$ and for every $\theta \in \Delta_n$ we have that $(\Gamma_p \Rightarrow \Delta_p, \neg\Delta_n \Rightarrow \neg\Gamma_n) \not\models_K \theta$.

We use $\neg\Delta_n \Rightarrow \neg\Gamma_n \not\models_K \neg\Delta_n \Rightarrow \neg\Gamma_n$ from Theorem 80 for all cases except implication. The negation of $\varphi \supset \psi$ is $\varphi \wedge \neg\psi$, therefore wherever $\neg(\varphi \supset \psi)$ appears in the sequent so does $\varphi \wedge \neg\psi$. We apply saturation and Theorem 80 for $\neg\psi$, and Lemma 55 for φ. ∎

This lemma states that whenever you refute the wholly positive formulas and the wholly negative formulas then you must refute the mixed formulas, hence refuting the entire sequent. Remember intuitionistic negation is considered as $\sim \varphi = \varphi \supset \neg\varphi$ therefore intuitionistic negations occur in $\Gamma_o \cup \Delta_o$.

Theorem 57 *For all complete sequents* $\Gamma \Rightarrow \Delta = \Gamma_p, \Gamma_n, \Gamma_o \Rightarrow \Delta_p, \Delta_n, \Delta_o$ *in* N *such that* $\Gamma_p \neq \neg\Delta_n$ *every pair of worlds which falsifies* $\Gamma_p, \Gamma_n \Rightarrow \Delta_p, \Delta_n$ *falsifies* $\Gamma \Rightarrow \Delta$*.*

Proof: By Lemmas 55 and 56, we only need to show that every formula in Γ_o is satisfied and every formula in Δ_o is refuted. To do this we use structural induction for formulas θ in $\Gamma_o \cup \Delta_o$. This proof only uses saturatedness and the induction hypotheses again except for implication in Δ_o but since this is the most important lemma, we go through it.

$\theta = p$ **base:** These formulas cannot be in $\Gamma_o \cup \Delta_o$.

$\theta = \varphi \wedge \psi$ **:** By expanding the definition of a mixed formula into cases, φ or ψ can be mixed. When this is the case we use saturation and the induction hypotheses for the mixed formula. Otherwise, assume (without loss of generality) φ is positive and ψ is negative. By saturation φ and ψ are in $\Gamma_p \cup \Gamma_n \cup \Delta_p \cup \Delta_n$ which by Lemmas 55 and 56 are refuted.

$\theta = \varphi \vee \psi$ **:** This case is very similar to the $\wedge$ case above.

$\theta = \varphi \supset \psi$ **:** If $\theta \in \Gamma_o$, then by examining the cases and using both saturation and the induction hypothesis, we are finished.

If $\theta \in \Delta_o$, the case where both are mixed needs the induction hypothesis and saturation.

Assume (without loss of generality) that φ is positive and ψ is negative. Let $\psi = \neg\psi'$.

We must show that there is a $\Gamma' \Rightarrow \Delta'$ happens in the interval, such that by monotonicity of R_k we need (1) that $\varphi \in \neg\Delta_n$ and (2) that ψ is in Δ_n.

(1) $\neg\varphi$ cannot occur in Γ_n without satisfying θ, thus $\neg\varphi$ must be in Δ_n, i.e. $\varphi \in \neg\Delta_n$.

(2) If ψ cannot be in Γ_n without satisfying θ, therefore $\psi \in \Delta_n$.

Then $\varphi \wedge \psi' \in \neg\Delta_n$ thus refuting $\varphi \supset \neg\psi'$ in the interval.

The case where φ is negative and ψ is positive is similar; and the case where both are mixed needs the induction hypothesis and saturation.

$\theta = \neg\varphi$ **:** For this case to be a mixed formula φ then φ is mixed, so we use the induction hypothesis.

∎

The following theorem is completeness for complete underivable sequents.

Theorem 58 *Let $\Gamma_p, \Gamma_n, \Gamma_o \Rightarrow \Delta_p, \Delta_n, \Delta_o$, be a complete sequent for N, then the pair $(\Gamma_p \Rightarrow \Delta_p, \neg\Delta_n \Rightarrow \neg\Gamma_n)$ such that $\Gamma_p \neq \neg\Delta_n$ falsifies $\Gamma \Rightarrow \Delta$ in $\mathcal{K}$.*

Proof: By Lemma 52 we have constructed two worlds satisfying the reachability condition. We now must verify that $(\Gamma_p \Rightarrow \Delta_p, \neg\Delta_n \Rightarrow \neg\Gamma_n)$ falsifies $\Gamma \Rightarrow \Delta$. This pair of worlds falsifies $\Gamma_p, \Gamma_n \Rightarrow \Delta_p, \Delta_n$ by application of Lemma 55 and Lemma 56. This suffices by application of Theorem 57. ∎

The following corollary asserts completeness for underivable sequents.

Corollary 59 *If $\Gamma' \Rightarrow \Delta'$ is a finite underivable sequent in N, then it can be extended to a complete sequent $\Gamma_p, \Gamma_n, \Gamma_o \Rightarrow \Delta_p, \Delta_n, \Delta_o$ such that $\Gamma_p \neq \neg\Delta_n$ and $(\Gamma_p \Rightarrow \Delta_p, \neg\Delta_n \Rightarrow \neg\Gamma_n)$ falsifies $\Gamma \Rightarrow \Delta$.*

Proof: Every finite underivable sequent $\Gamma' \Rightarrow \Delta'$ can be extended to a complete underivable sequent $\Gamma \Rightarrow \Delta$ such that $\Gamma_p \neq \neg\Delta_n$, where we can apply Theorem 58. ∎

3.4 Alternate Formalization

Lemma 60 *Every formula φ in N is equivalent to a formula, say φ_{atom}, where $\neg$ only scopes propositional atoms.*

Outline of Proof: By examining each formula φ, and noting the following equivalences of interderivability in N.

$$
\begin{aligned}
\neg(\varphi \wedge \psi) &\equiv \neg\varphi \vee \neg\psi \\
\neg(\varphi \vee \psi) &\equiv \neg\varphi \wedge \neg\psi \\
\neg(\varphi \supset \psi) &\equiv \varphi \wedge \neg\psi \\
\neg(\neg\varphi) &\equiv \varphi
\end{aligned}
$$

By structural induction using these equivalences we derive an equivalent formula φ_{atom} where negation occurs only outside propositional atoms. ∎

If φ is a formula where negations only scope propositional atoms then, φ is in *positive normal form*. A positive normal form entailment relation for N is defined over a propositional intuitionistic model.

Definition 61 *Let $A = \langle W, R, V\rangle$ be an intuitionistic propositional Kripke model over P. We define $\models_A$ inductively over formulas φ in positive normal form. For any $w_p, w_n \in W$ such that $w_p R w_n$ and $w_p \neq w_n$,*

$$
\begin{aligned}
(w_p, w_n) \models_A p &= V(w_p, p) = 1 \\
(w_p, w_n) \models_A \varphi \wedge \psi &= (w_p, w_n) \models_A \varphi \textit{ and } (w_p, w_n) \models_A \psi \\
(w_p, w_n) \models_A \varphi \vee \psi &= (w_p, w_n) \models_A \varphi \textit{ or } (w_p, w_n) \models_A \psi \\
(w_p, w_n) \models_A \varphi \supset \psi &= \forall w', w'' \in W. w' \neq w'' \wedge w_p\, R\, w'\, R\, w''\, R\, w_n \wedge \\
&\qquad (w', w'') \models_A \varphi \rightarrow (w', w'') \models_A \psi \\
(w_p, w_n) \models_A \neg p &= V(w_n, p) = 0
\end{aligned}
$$

Theorem 62 *When φ is logically equivalent to a formula in positive normal form ψ and $\langle W, R, V\rangle$ is a Kripke model, then $\forall w_p, w_n \in W$ such that $w_p R w_n$ and $w_p \neq w_n$,*

$$(w_p, w_n) \models_A \psi \textit{ if and only if } (w_p, w_n) \models_M \varphi$$

where $\models_A$ is from Definition 61 and $\models_M$ is from Definition 40.

3.5 Elementary Equivalence with Partial World Models

Thomason (Thomason 1969) and Gurevich (Gurevich 1977) have model theories for N which use partial function from worlds to sets of propositional atoms. We can consider this equivalently as assigning one of three truth values to propositional atoms. We will define a transformation from these 3-valued models to interval model and conversely which preserve satisfiability and validity.

Definition 63 Partial World Model *A partial world model $T = \langle W, R, V\rangle$ over a set of atoms P is a set W, a binary reflexive transitive relation $R \subseteq W \times W$, and valuation function $V\colon W \times P \to \{0, 1, \bot\}$, such that $\forall w, w' \in W.\ \ \forall p \in P.$ if wRw' then if $V(w,p) = 1$ then $V(w',p) = 1$ and whenever $V(w,p) = 0$ then $V(w',p) = 0$.*

Then the partial *valuation function extends to all formula as follows: If φ is a formula in intuitionistic logic then $V(w,\varphi) = 1$ when $w \models_{T'} \varphi$ as in Definition 2 where T' is T where $\bot$ is identified with* 0, *otherwise*

$$\begin{array}{lcl}
V(w,\neg\varphi) = 1 & \iff & V(w,\varphi) = 0 \\
V(w,\neg\varphi) = 0 & \iff & V(w,\varphi) = 1 \\
V(w,\varphi\wedge\psi) = 0 & \iff & V(w,\varphi) = 0 \textit{ or } V(w,\psi) = 0 \\
V(w,\varphi\vee\psi) = 0 & \iff & V(w,\varphi) = 0 \textit{ and } V(w,\psi) = 0 \\
V(w,\varphi\supset\psi) = 0 & \iff & V(w,\varphi) = 1 \textit{ and } V(w,\psi) = 0
\end{array}$$

For ease of conversion of notation let the following shorthand denote the partial valuation function $V_i(w) = \{p \mid V(w,p) = i \text{ where } i \in \{0, 1, \bot\}\}$,

For the theorem that for every interval-model there is an elementarily equivalent partial world model and vice versa we will only consider finite models with antisymmetric accessibility relations. This class of models suffice by the finite model property and because every model is elementarily equivalent to one with an antisymmetric accessibility relation.

3.5.1 From Partial World Models to Interval Models

Lemma 64 *Let $T = \langle W, R, V\rangle$ be a finite partial world model with R antisymmetric and let E be the set of maximal worlds i.e. $E \subseteq W = \{w | \neg(\exists w'.w' \neq w \text{ and } wRw')\}$ and $R\rfloor_E = \{(w, e) \mid wRe \wedge e \in E\}$.*

Let $F(T) = M = \langle W', R', V'\rangle$ be our interval model where the items in the triple are defined as follows:

$$
\begin{array}{lll}
W' & = & (W - E) \cup (E \times \{0, 1\}) \quad \textit{Maximal worlds split into two worlds} \\
R' & = & R - R\rfloor_E \cup \{(a, (e, 0)) \mid aR\rfloor_E\, e \text{ and } a \notin E\} \cup \{(a, (e, 1)) \mid aR\rfloor_E\, e \text{ and } a \notin E\} \\
 & & \cup\{((e, 0), (e, 1)) \mid e \in E\} \cup \{((e, 0), (e, 0)) \mid e \in E\} \cup \{((e, 1), (e, 1)) \mid e \in E\} \\
V'(w) & = & \begin{cases} V_1(w) & \text{when } w \in W - E \\ V_1(e) & \text{when } w = (e, 0) \text{ where } e \in E \\ V_1(e) \cup V_\perp(e) & \text{when } w = (e, 1) \text{ where } e \in E \end{cases}
\end{array}
$$

$$
\begin{array}{llll}
\textit{Then } \forall w \in E, & V(w, \varphi) = 1 & \textit{if and only if} & ((w, 0), (w, 1)) \models_M \varphi, \textit{ and} \\
 & V(w, \varphi) = 0 & \textit{if and only if} & ((w, 0), (w, 1)) \models_M \neg\varphi,
\end{array}
$$

and

$$
\begin{array}{llll}
\forall w \in W - E & V(w, \varphi) = 1 & \textit{if and only if} & \forall w, w' \in W'.w \neq w' \wedge wR'w' \to (w, w') \models_M \varphi, \\
\textit{and} & V(w, \varphi) = 0 & \textit{if and only if} & \forall w, w \in W'.w \neq w' \wedge wR'w' \to (w, w') \models_M \neg\varphi.
\end{array}
$$

where $\to$ is classical implication.

Outline of Proof: M is a Kripke structure. R' is reflexive and transitive by its construction from R and V' is monotonic because V is monotonic and the assignment of atoms to $(e, 1)$ is a superset of the assignment to $(e, 0)$.

Now to show the alignment of satisfiability between T and M we use structural induction on formulas and the following simplification: for a formula to have the valuation $\perp$ in T, the formula must be $\perp$ or 0 in every subinterval in M. We identified $\perp$ with 0 in all worlds outside

E, hence for this lemma we need only explore worlds e in E. This is a non-trivial case because T is finite and E is non-empty.

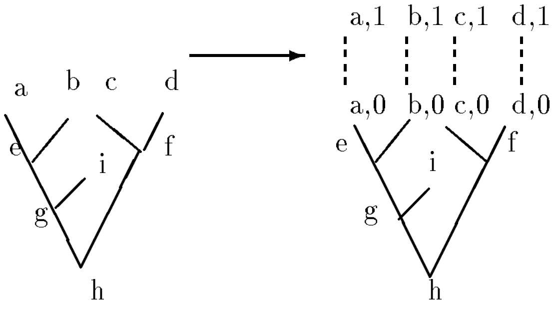
The base case follows from the assignment of propositional atoms. We only show the positive implication case. The other cases follow immediately by the induction hypotheses.

$\varphi \supset \psi$ $\quad V(e, \varphi \supset \psi) = 1$ if and only if $((e, 0), (e, 1)) \models_M \varphi \supset \psi$.

Expanding the definition of V we need the implication to be true in all proper subintervals.

There are no proper subintervals of $((e, 0), (e, 1))$, thus $((e, 0), (e, 1)) \models_M \varphi \supset \psi$, simplifies to the condition $((e, 0), (e, 1)) \models_M \varphi \rightarrow ((e, 0), (e, 1)) \models_M \psi$. This is simply, $V(e, \varphi) = 1 \rightarrow V(e, \psi) = 1$ as needed. ∎

The crux of the transformation is that we split the top worlds, a, into two worlds $(a, 0)$ and $(a, 1)$ where the $(a, 0)$ world equates $\perp$ with False, 0, and the $(a, 1)$ world equates $\perp$ with True, i.e. 1. The transformation from a partial world model to an interval model is shown pictorially below.



Conditions for Valuations at World d above

$V(d, p) = 1$ implies that $V'((d, 0), p) = 1$ and $V'((d, 1), p) = 1$

$V(d, p) = 0$ implies that $V'((d, 0), p) = 0$ and $V'((d, 1), p) = 0$

$V(d, p) = \perp$ implies that $V'((d, 0), p) = 0$ and $V'((d, 1), p) = 1$

Corollary 65 *For all formulas φ in N, φ is valid in a finite partial world model T with antisymmetric accessibility relation if and only if φ is valid in the interval model $F(T)$.*

3.5.2 From Intervals to Partial Worlds

Definition 66 *Let $R: W \times W \rightarrow 2$ be a relation. Then we define a subinterval ordering among related objects in R, denoted $R_>$, if $wRrRr'Rw'$ and $w \neq w'$ and $r \neq r'$ then $(w, w')R_>(r, r')$.*

Lemma 67 *Let $M = \langle W', R', V' \rangle$ be a propositional model for N where R' is antisymmetric, then we define a triple $\langle W, R, V \rangle = G(M) = T$, as follows:*

W	$=$	$R' - \{(w, w) \mid w \in W'\}$	*Intervals in M are points in T.*
R	$=$	$R'_{>}$	*Subinterval ordering becomes accessibility.*
$V((w, w'))$	$=$	$V'_1(w)$	*The true atoms are given from w*
$\bar{V}((w, w'))$	$=$	$V'_0(w')$	*The false atoms are given from w'*

then for all formulas φ and $\forall (w, w') \in W$.

$$(w, w') \models_M \varphi \textit{ if and only if } V((w, w'), \varphi) = 1, \textit{ and}$$

$$(w, w') \models_M \neg\varphi \textit{ if and only if } V((w, w'), \varphi) = 0.$$

Proof: The construction takes pairs of related worlds in M as one world in T. Again, formally we would have to show that T is a partial world model, but this is evident leaving the only item to prove is that this truth preserving mapping for propositional atoms extends to all formulas. We do this by induction on the formula θ. The basis follows by construction and all other cases except positive implication follow immediately from the induction hypotheses.

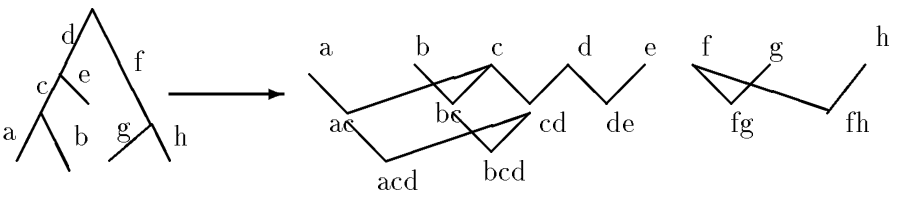
$\varphi \supset \psi$ $\quad (w_1, w_2) \models_M \varphi \supset \psi \to V((w_1, w_2), \varphi \supset \psi) = 1.$

Expanding the definition of $\models_M$ we are given that $\forall w'_1, w'_2 \in W'. w_1\ R'\ w'_1\ R'\ w'_1\ R'\ w_2 \wedge (w'_1, w'_2) \models_M \varphi \to (w'_1, w'_2) \models_M \psi$.

Then the induction hypothesis says that $V((w'_1, w'_2), \varphi) = 1 \to V((w'_1, w'_2), \psi) = 1$, and since $w_1\ R'\ w'_1\ R'\ w'_1\ R'\ w_2$, then $(w_1, w_2) R'_{>} (w'_1, w'_2)$ which says that $(w_1, w_2) R (w'_1, w'_2)$.

Since (w'_1, w'_2) was just a representative from the quantification then we have $\forall (w'_1, w'_2) \in W$. $(w_1, w_2) R (w'_1, w'_2) \wedge V((w'_1, w'_2), \varphi) = 1 \to V((w'_1, w'_2), \psi) = 1$ which is just the definition of $V((w_1, w_2), \varphi \supset \psi) = 1$. ∎

For the transformation from an interval model to a partial world model, we show this pictorially and note that the names of the partial worlds will be given in terms of composition of intervals.



Conditions for Valuations at Arc h.

$$h\left\{\begin{array}{l} p=1 \\ \\ p=1 \end{array}\right. \Rightarrow V(h,p)=1 \qquad h\left\{\begin{array}{l} p=0 \\ \\ p=0 \end{array}\right. \Rightarrow V(h,p)=0 \qquad h\left\{\begin{array}{l} p=1 \\ \\ p=0 \end{array}\right. \Rightarrow V(h,p)=\perp$$

Corollary 68 *For all formulas φ in N, φ is valid in an interval model M with antisymmetric accessibility relation if and only if φ is valid in the partial world model $G(M)$.*

Theorem 69 *For all formulas φ in N, φ is valid in all finite interval models with antisymmetric accessibility relation if and only if φ is valid in all finite partial world models with antisymmetric accessibility relation.*

Proof: Use Corollary 68 and Corollary 65. ∎

3.6 First Order Case

Thomason proposed a Kripke structure interpretation with three truth values which assumed constant domains. Gurevich proposed a similar three valued model but captured non-constant domains. Our proposal also capture non-constant domains and hence all of first-order N.

We extend the language to include countably many bound variables, here denoted $x, y, z..$ and countably many free variables, here denoted $a, b, c..$ for each natural n, we have countably many predicates of arity n, denoted $P^n, Q^n, ...$

Terms are constructed from individuum variables and parameters. Atomic formulas are of the form $P^n(t_1, \ldots, t_n)$ where $t_i, 1 \leq i \leq n$ are terms. Formulas are constructed via atomic formulas via $\wedge, \vee, \sim, \exists, \forall$.

$\varphi[t/x]$ is the result of substituting t for all occurrences of x in φ. If the variable is evident, we simply denote this as $\varphi[t]$. In most cases t will denote a ground terms, that is one which contains parameters but not variables; thus collisions are impossible.

We add the following rules to those of propositional N.

Rules of Inference

$$\Rightarrow\exists \quad \frac{\Gamma \Rightarrow \Delta, \exists x.\varphi, \varphi[t]}{\Gamma \Rightarrow \Delta, \exists x.\varphi} \qquad \exists\Rightarrow \quad \frac{\varphi[b], \Gamma \Rightarrow \Delta}{\exists x.\varphi, \Gamma \Rightarrow \Delta}$$

$$\Rightarrow\forall \quad \frac{\Gamma \Rightarrow \varphi[b]}{\Gamma \Rightarrow \Delta, \forall x.\varphi} \qquad \forall\Rightarrow \quad \frac{\varphi[t], \forall x.\varphi, \Gamma \Rightarrow \Delta}{\forall x.\varphi, \Gamma \Rightarrow \Delta}$$

$$\Rightarrow\neg\exists \quad \frac{\Gamma \Rightarrow \neg\varphi[b]}{\Gamma \Rightarrow \Delta, \neg\exists x.\varphi} \qquad \neg\exists\Rightarrow \quad \frac{\neg\varphi[t], \neg\exists x.\varphi, \Gamma \Rightarrow \Delta}{\neg\exists x.\varphi, \Gamma \Rightarrow \Delta}$$

$$\Rightarrow\neg\forall \quad \frac{\Gamma \Rightarrow \Delta, \neg\varphi[t], \neg\forall x.\varphi}{\Gamma \Rightarrow \Delta, \neg\forall x.\varphi} \qquad \neg\forall\Rightarrow \quad \frac{\neg\varphi[b], \Gamma \Rightarrow \Delta}{\neg\forall x.\varphi, \Gamma \Rightarrow \Delta}$$

Where t is a ground term and b is a new parameter (eigenvariable). Note that the $\Rightarrow\forall$ and $\Rightarrow\neg\exists$ are not invertible.

Now to define models we need a notion of a domain over which a quantifier ranges.

The following definition is the standard one for first order intuitionistic logic which we gave in the previous chapter, but we are repeating it here for convenience.

Definition 70 First Order Model *A Kripke model for first order intuitionistic logic over universe U is a four-tuple $K = \langle W, R, V, D\rangle$ where W is a set, and $R \subseteq W \times W$ is a reflexive transitive relation and*

1. *D is a function from W to a subset of U satisfying wRw' implies $D(w) \subseteq D(w')$, and*
2. *V is a function from worlds W and n-ary predicates P^n to $(D(w))^n \rightarrow \{0,1\})$ satisfying the monotonicity condition that if wRw' then $V(w', P)$ is an extension of $V(w, P)$ i.e. for all $x_1, \ldots x_n \in D(w)$, $V(w,P)(x_1, \ldots, x_n)$ implies $V(w', P)(x_1, \ldots, x_n)$.*

Definition 71 *Let $M = \langle W, R, V, D\rangle$ be a first-order intuitionistic model, then we define its dual as $M^\perp = \langle W, R^\cup, \bar{V}, D\rangle$ where $R^\cup$ is the inverse of R and for all $w \in W$ $(\bar{V}(w, P^n) = (D(w))^n - V(w, P^n))$.*

With the above definition, and the caveat that all formulae judged at an interval (w, w') are to be understood as formulae in the language of interval, then all parameters of the formula are drawn from $D(w) \cap D(w')$. Because of monotonicity, the domain for the language of an interval (w, w') is simply $D(w)$. With that note and the definition of $M^{\perp}$, the first order case generalizes from the propositional case straightforwardly and handles non-constant domains.

Definition 72 *A first-order interval model for N is a first-order Kripke structure for intuitionistic logic $M = \langle W, R, V, D\rangle$ with the entailment relation $\models_M$ for truth and $\Vert\!\models_{M^{\perp}}$ for falsity defined by recursion on φ. For any $w_p, w_n \in W$. such that $w_p R w_n$ and $w_p \neq w_n$.*

$$\begin{array}{lllr}
A & (w_p, w_n) \models_M A(x_1, \ldots, x_n) & \equiv & V(w_p, A)(x_1, \ldots, x_n) = 1 \\
 & & & \textit{for } (x_1, \ldots, x_n) \in D(w_p) \\
\varphi \wedge \psi & (w_p, w_n) \models_M \varphi \wedge \psi & \equiv & (w_p, w_n) \models_M \varphi \textit{ and } (w_p, w_n) \models_M \psi \\
\varphi \vee \psi & (w_p, w_n) \models_M \varphi \vee \psi & \equiv & (w_p, w_n) \models_M \varphi \textit{ or } (w_p, w_n) \models_M \psi \\
\varphi \supset \psi & (w_p, w_n) \models_M \varphi \supset \psi & \equiv & \forall w', w'' \in W. w' \neq w'' \wedge w_p \; R \; w' \; R \; w'' \; R \; w_n \wedge \\
 & & & (w', w'') \models_M \varphi \textit{ implies } (w', w'') \models_M \psi \\
\forall x.\varphi(x) & (w_p, w_n) \models_M \forall x.\varphi(x) & \equiv & \forall w', w'' \in W. w' \neq w'' \wedge w_p \; R \; w' \; R \; w'' \; R \; w_n \wedge \\
 & & & \textit{for every } d \in D(w') \;\; (w', w'') \models_M \varphi[d] \\
\exists x.\varphi(x) & (w_p, w_n) \models_M \exists x.\varphi(x) & \equiv & \textit{for some } d \in D(w_p) \;\; (w_p, w_n) \models_M \varphi[d] \\
\neg\varphi & (w_p, w_n) \models_M \neg\varphi & \equiv & (w_n, w_p) \Vert\!\models_{M^{\perp}} \varphi
\end{array}$$

And for falsity

$$\begin{array}{lllr}
A & (w_n, w_p) \Vert\!\models_{M^{\perp}} A(x_1, \ldots, x_n) & \equiv & \bar{V}(w_n, A)(x_1, \ldots, x_n) = 1 \\
 & & & \textit{for } (x_1, \ldots, x_n) \in D(w_n) \\
\varphi \wedge \psi & (w_n, w_p) \Vert\!\models_{M^{\perp}} \varphi \wedge \psi & \equiv & (w_n, w_p) \Vert\!\models_{M^{\perp}} \varphi \textit{ or } (w_n, w_p) \Vert\!\models_{M^{\perp}} \psi \\
\varphi \vee \psi & (w_n, w_p) \Vert\!\models_{M^{\perp}} \varphi \vee \psi & \equiv & (w_n, w_p) \Vert\!\models_{M^{\perp}} \varphi \textit{ and } (w_n, w_p) \Vert\!\models_{M^{\perp}} \psi \\
\varphi \supset \psi & (w_n, w_p) \Vert\!\models_{M^{\perp}} \varphi \supset \psi & \equiv & (w_p, w_n) \models_M \varphi \textit{ and } (w_n, w_p) \Vert\!\models_{M^{\perp}} \psi \\
\forall x.\varphi(x) & (w_n, w_p) \Vert\!\models_{M^{\perp}} \forall x.\varphi(x) & \equiv & \textit{for some } d \in D(w_p) \;\; (w_n, w_p) \Vert\!\models_{M^{\perp}} \varphi[d] \\
\exists x.\varphi(x) & (w_n, w_p) \Vert\!\models_{M^{\perp}} \exists x.\varphi(x) & \equiv & \forall w', w'' \in W. w' \neq w'' \wedge w_p \; R \; w' \; R \; w'' \; R \; w_n \wedge \\
 & & & \textit{for every } d \in D(w') \;\; (w'', w') \Vert\!\models_{M^{\perp}} \varphi[d] \\
\neg\varphi & (w_n, w_p) \Vert\!\models_{M^{\perp}} \neg\varphi & \equiv & (w_p, w_n) \models_M \varphi
\end{array}$$

Definition 73 *Let $M = \langle W, R, V, D\rangle$ be a first-order Kripke structure. A formula φ of first-order N is* satisfied *in M if and only if $\exists w_p, w_n \in W$. $w_p \neq w_n$ and $w_p R w_n$ and $(w_p, w_n) \models_M \varphi$.*

A formula φ *is* valid in M *if and only if* $\forall w_p, w_n \in W.\ w_p \neq w_n$ *and* $w_p R w_n$ *and* $(w_p, w_n) \models_M \varphi$. *A first-order N formula* φ *is* valid *if and only if* φ *is valid in all first-order Kripke structures.*

Theorem 74 Soundness and Completeness *A first-order* N *formula* φ *is derivable if and only if it is valid in all first-order interval models for* N*.*

Chapter 4

Constructible Duality

4.1 Summary

Duality, symmetry, and categorical opposites play an increasing role in our understanding of fundamental structural properties of computation. Recent investigations have focussed on powerful categorical tools, that provide great expressive power, such as the **Chu**, the Dialectica and special final coalgebra constructions, in contrast to some traditional logical methods. This thesis attempts to apply old fashioned logical techniques to investigate these issues of duality.

Our study began with Nelson's logic of constructible falsity N which symmetrizes intuitionistic logic by adding an involutive constructive negation $\neg$.

We modify constructible falsity by dropping the *consistency* axiom, $\varphi \wedge \neg\varphi \supset \psi$. This omission means that every counterexample and proof have to be constructed — not merely derived by a contradiction.

To finish turning Nelson's negation into a dualizing operator, we add two independent constants $\perp$ and $\mathbf{I}$ which give rise to operators reminiscent of exponentials. These modifications to constructible falsity create a logic which is both constructive and one in which every operator has a representable dual; hence the name constructible duality CD.

We show in the next chapter that models for CD provide a model for linear logic and in this chapter we examine the relationship between CD and relevance logic.

4.2 Introduction

Duality has recently come to the forefront of logical investigations in computer science. Linear logic, corecursion, the μ-calculus, and self-dual categories have attracted much attention. Here we consider many of these same structures, but we start from a logical perspective, where duality appears in the form of an operator, constructive negation, and two constants, undefined and overdefined.

We develop a new self-dual semantics for the logic of constructible duality. Models for logics can be more than equivalence classes of formula under the derivability relation. A classic example which illustrates this difference is Kripke models for modal logic. Kripke models brought with them a philosophical viewpoint, that of "possible worlds". Judging truth from a world in a Kripke model, represents the view from that possible world. From this vantage point other viewpoints are possible when other worlds are accessible. In models of intuitionistic logic, the accessibility relation flows in the direction of time. As time progresses, more base facts may be learned. Thus a model theory may bring more than just a completeness result, it may shed light on the underlying phenomena.

To this end, we break from this philosophical viewpoint of possible worlds and we allow instead for some indeterminacy about which world is the present world. We consider that there are some facts about the world which are certainly true and some facts which are definitely false, leaving open the possibility of conflict and the possibility of indecision. This leaves us uncertain about which possible world is our exact location. This region of uncertainty, we call an interval.

This logic conservatively extends intuitionistic logic but not constructible falsity N since in N the *consistency* axiom scheme $\varphi \wedge \neg\varphi \supset \psi$ is valid. This scheme is omitted from CD, but the negation of Nelson is the inspiration for the duality operator here.

4.3 Propositional Intuitionistic Logic with Constants

The logic of constructible duality is an extension of intuitionistic logic with constants, hence we show the multiple conclusion Gentzen rules for propositional intuitionistic logic with constants.

The syntax of propositional intuitionistic logic is that atomic formulas are elements drawn from a set of propositional atoms; a formula of propositional intuitionistic logic is either an atom, a constant **0**, **1**, or the conjunction $\wedge$ of two formulas, the disjunction $\vee$ of two formulas, or an implication $\supset$ parameterized by two formulas.

Gentzen System Sequents $\Gamma \Rightarrow \Delta$ should be read $\wedge\Gamma \supset \vee\Delta$, where the set $\Gamma \cup \Delta$ is a finite set of formulas, Γ, φ is the set Γ with the single formula φ added. The rules are divided into left and right rules according to whether the main operator is introduced on the left or on the right (respectively).

Definition 75 Axiom

$$(1)\ \ \Gamma, \varphi \Rightarrow \Delta, \varphi \quad (2)\ \ \mathbf{0}, \Gamma \Rightarrow \Delta \quad (3)\ \ \Gamma \Rightarrow \Delta, \mathbf{1}$$

Rules of Inference

$$\Rightarrow\wedge \quad \frac{\Gamma \Rightarrow \Delta, \varphi \quad \Gamma \Rightarrow \Delta, \psi}{\Gamma \Rightarrow \Delta, \varphi \wedge \psi} \qquad \wedge\Rightarrow \quad \frac{\varphi, \psi, \Gamma \Rightarrow \Delta}{\varphi \wedge \psi, \Gamma \Rightarrow \Delta}$$

$$\vee\Rightarrow \quad \frac{\varphi, \Gamma \Rightarrow \Delta \quad \psi, \Gamma \Rightarrow \Delta}{\varphi \vee \psi, \Gamma \Rightarrow \Delta} \qquad \Rightarrow\vee \quad \frac{\Gamma \Rightarrow \Delta, \varphi, \psi}{\Gamma \Rightarrow \Delta, \varphi \vee \psi}$$

$$\supset\Rightarrow \quad \frac{\varphi \supset \psi, \Gamma \Rightarrow \Delta, \varphi \quad \psi, \Gamma \Rightarrow \Delta}{\varphi \supset \psi, \Gamma \Rightarrow \Delta} \qquad \Rightarrow\supset \quad \frac{\varphi, \Gamma \Rightarrow \psi}{\Gamma \Rightarrow \Delta, \varphi \supset \psi}$$

All rules are invertible, i.e. the premises also follow from their conclusions, except, $\supset\Rightarrow$. Remember that intuitionistic negation $\sim$ is defined as $\sim \varphi = \varphi \supset \mathbf{0}$. For ease of presentation, we have chosen this route to represent intuitionistic negation. We denote this system I_c, for intuitionistic logic with constants.

4.4 Adding a Duality Operator

The Gentzen system for constructible duality adds the following rules and axioms to those for intuitionistic logic with constants. These rules can also be found by extending N^- in (Nelson and Almukdad 1984) by four constants and incorporating weakening into the Gentzen system. Constructible duality contains the positive intuitionistic connectives $\wedge$, $\vee$, $\supset$ the constants for falsity **0** and truth **1** as well as the dualizing operator, $\neg$ and constants for undefined $\perp$ and overdefined **I**.

Axiom

$$(4)\Gamma, \neg\mathbf{1} \Rightarrow \Delta \quad (5)\Gamma \Rightarrow \Delta, \neg\mathbf{0} \quad (6)\ \Gamma \Rightarrow \Delta, \mathbf{I} \quad (7)\ \Gamma, \bot \Rightarrow \Delta$$

Rules of Inference

$$\neg\neg\Rightarrow \quad \frac{\varphi, \Gamma \Rightarrow \Delta}{\neg\neg\varphi, \Gamma \Rightarrow \Delta} \qquad \Rightarrow\neg\neg \quad \frac{\Gamma \Rightarrow \Delta, \varphi}{\Gamma \Rightarrow \Delta, \neg\neg\varphi}$$

$$\neg\wedge\Rightarrow \quad \frac{\neg\varphi, \Gamma \Rightarrow \Delta \quad \neg\psi, \Gamma \Rightarrow \Delta}{\Gamma, \neg(\varphi \wedge \psi) \Rightarrow \Delta} \qquad \Rightarrow\neg\wedge \quad \frac{\Gamma \Rightarrow \Delta, \neg\varphi, \neg\psi}{\Gamma \Rightarrow \Delta, \neg(\varphi \wedge \psi)}$$

$$\neg\vee\Rightarrow \quad \frac{\neg\varphi, \neg\psi, \Gamma \Rightarrow \Delta}{\neg(\varphi \vee \psi), \Gamma \Rightarrow \Delta} \qquad \Rightarrow\neg\vee \quad \frac{\Gamma \Rightarrow \Delta, \neg\varphi \quad \Gamma \Rightarrow \Delta, \neg\psi}{\Gamma \Rightarrow \Delta, \neg(\varphi \vee \psi)}$$

$$\neg\supset\Rightarrow \quad \frac{\varphi, \neg\psi, \Gamma \Rightarrow \Delta}{\neg(\varphi \supset \psi), \Gamma \Rightarrow \Delta} \qquad \Rightarrow\neg\supset \quad \frac{\Gamma \Rightarrow \Delta, \varphi \quad \Gamma \Rightarrow \Delta, \neg\psi}{\Gamma \Rightarrow \Delta, \neg(\varphi \supset \psi)}$$

$$\neg\mathbf{I}\Rightarrow \quad \frac{\mathbf{I}, \Gamma \Rightarrow \Delta}{\neg\mathbf{I}, \Gamma \Rightarrow \Delta} \qquad \Rightarrow\neg\mathbf{I} \quad \frac{\Gamma \Rightarrow \Delta, \mathbf{I}}{\Gamma \Rightarrow \Delta, \neg\mathbf{I}}$$

$$\neg\bot\Rightarrow \quad \frac{\bot, \Gamma \Rightarrow \Delta}{\neg\bot, \Gamma \Rightarrow \Delta} \qquad \Rightarrow\neg\bot \quad \frac{\Gamma \Rightarrow \Delta, \bot}{\Gamma \Rightarrow \Delta, \neg\bot}$$

All of these rules are invertible. Further $\varphi \vee \neg\varphi$ is not valid in this logic even though $\neg(\varphi \wedge \psi) \equiv \neg\varphi \vee \neg\psi$. Note also that $\varphi \vee \psi$ is definable as $\neg(\neg\varphi \wedge \neg\psi)$.

4.5 Kripke Structures for Propositional Intuitionistic Logic

We are about to define the semantics for propositional constructible duality in terms of intervals over Kripke structures for propositional intuitionistic logic with constants. Therefore, to be precise we give a quick review of how to modify the definitions and results from Chapter 2. In particular, again our completeness asserts that any underivable formula is refutable in a canonical model. In fact, the canonical refutation Kripke structure $\mathcal{K}$ for intuitionistic logic with constants is the canonical refutation Kripke structure for constructible duality, when we use constructible duality's interval satisfiability relation over $\mathcal{K}$. Thus, we provide a brief review of the definition of $\mathcal{K}$ as well as the definition of an intuitionistic Kripke model.

This definition is review.

Definition 76 *A Kripke structure for propositional intuitionistic logic over a set P of propositional atoms is a triple, $\langle W, R, V\rangle$, where W is a non-empty set (the set of* worlds*), R is a binary reflexive and transitive relation on W and V is a function assigning to every world a set of propositional atoms satisfying for every two worlds w, w', $wRw' \to V(w) \subseteq V(w')$.*

Thus, V is monotonic on propositional atoms.

This definition is the same as Definition 2 except we have shown how to evaluate the constants.

Definition 77 *The satisfiability relation $\models_K$ on a Kripke structure $K = \langle W, R, V\rangle$ over propositional atoms P is a relation between worlds and propositional formulas. We define this relation inductively over the structure of a formula. For any $w \in W$,*

$$
\begin{array}{lcr}
w \models_K \mathbf{1} & & \\
w \not\models_K \mathbf{0} & & \\
w \models_K p & \equiv & p \in V(w) \\
w \models_K \varphi \wedge \psi & \equiv & w \models_K \varphi \text{ and } w \models_K \psi \\
w \models_K \varphi \vee \psi & \equiv & w \models_K \varphi \text{ or } w \models_K \psi \\
w \models_K \varphi \supset \psi & \equiv & \forall w' \in W.\ w\ R\ w' \text{ and } w' \models_K \varphi \text{ implies } w' \models_K \psi
\end{array}
$$

where the meta-logic of all definitions is classical logic.

We say that a formula φ is satisfied at a world w in a structure K if $w \models_K \varphi$.

Definition 78 *Let $K = \langle W, R, V\rangle$ be a Kripke structure, then a formula φ is* satisfied *in K if and only if $\exists w \in W.\ w \models_K \varphi$. A formula φ is* valid *in K if and only if $\forall w \in W.\ w \models_K \varphi$. A formula φ is* valid *if and only if φ is valid in all Kripke structures.*

We again construct a *canonical* model $\mathcal{K}$ in which every non-valid formula is falsified. We shall reuse $\mathcal{K}$ for our completeness result of constructible duality, hence we repeat the construction here.

Definition 79 *Let $\mathcal{K} = \langle W_k, R_k, V_k\rangle$ denote the canonical refutation model for I_c. We have*

1. *W_k as the set of all $\Gamma \Rightarrow \Delta$ that are* complete *sequents,*
2. *$\Gamma_i \Rightarrow \Delta_i\ R_k\ \Gamma_j \Rightarrow \Delta_j$ if and only if $\Gamma_i \subset \Gamma_j$, and*

3. $p \in V_k(\Gamma_i \Rightarrow \Delta_i)$ *if and only if* $p \in \Gamma_i$*.*

Theorem 80 *If* $\Gamma \Rightarrow \Delta$ *is a complete sequent in* I_c *then in* $\mathcal{K}$*,* $\Gamma \Rightarrow \Delta \not\models_{\mathcal{K}} \Gamma \Rightarrow \Delta$*.*

The proof of this theorem relies on the notion of saturatedness which are modified below to include mention of the constants. With these definitions and lemmas many cases in constructible duality, will follow.

Definition 81 Saturated Sequent *A complete sequent* $\Gamma \Rightarrow \Delta$ *for* I_c *is saturated when it satisfies for all* φ, ψ*,* $\mathbf{1} \in \Gamma$*,* $\mathbf{0} \in \Delta$*, and*

Rule	*Sequent*		*Saturatedness*
$\Rightarrow \wedge$	$\varphi \wedge \psi \in \Delta$	*implies*	$\varphi \in \Delta$ *or* $\psi \in \Delta$
$\wedge \Rightarrow$	$\varphi \wedge \psi \in \Gamma$	*implies*	$\varphi \in \Gamma$ *and* $\psi \in \Gamma$
$\Rightarrow \vee$	$\varphi \vee \psi \in \Delta$	*implies*	$\varphi \in \Delta$ *and* $\psi \in \Delta$
$\vee \Rightarrow$	$\varphi \vee \psi \in \Gamma$	*implies*	$\varphi \in \Gamma$ *or* $\psi \in \Gamma$
$\supset\Rightarrow$	$\varphi \supset \psi \in \Gamma$	*implies*	$\varphi \in \Delta$ *or* $\psi \in \Gamma$

Definition 82 Saturated Set of Sequents *A set of complete sequents* $\mathcal{X}$ *is saturated if every sequent* $s \in \mathcal{X}$ *is saturated and if* $s \in \mathcal{X}$ *where* $s = \Gamma \Rightarrow \Delta, \varphi \supset \psi$ *then there is an* $s' \in \mathcal{X}$ *such that* $s' = \Gamma' \Rightarrow \Delta'$ *where* $\Gamma \cup \{\varphi\} \subset \Gamma'$ *and* $\psi \in \Delta'$*.*

Lemma 83 *The set of complete sequents for intuitionistic logic with constants is saturated.*

This means somewhere in the set of complete sequents we analyze the non-invertible rule and in each complete sequent we analyze all of the invertible rules.

4.6 Interval Kripke Structures for Constructible Duality

Definition 84 *Let* $R: W \times W \to \{0,1\}$ *be a relation. Then we define a subinterval ordering* $R_>$ *among related objects.*

$$(w,w')R_>(r,r') = \left\{ \begin{array}{lll} 1 & if & wRrRr'Rw' \text{ and } w \neq w' \text{ and } r \neq r' \\ 1 & if & \neg(wRw') \text{ and } wRr \text{ and } w'Rr' \\ 0 & & \text{otherwise} \end{array} \right. \left\| \begin{array}{l} \text{Proper subintervals} \\ \text{Improper subintervals} \\ \\ \end{array} \right.$$

Definition 85 *An interval model for constructible duality is a propositional intuitionistic model* $M = \langle W, R, V\rangle$ *over* P *with the entailment relation* $\models_M$ *for truth and* $\Vdash_{M^\perp}$ *for falsity defined by recursion on* φ. *For any* $w_p, w_n \in W$. *such that* $w_p \neq w_n$.

$$
\begin{array}{llcr}
\mathbf{0} & (w_p, w_n) \not\models_M \mathbf{0} & & \\
\mathbf{1} & (w_p, w_n) \models_M \mathbf{1} & & \\
\bot & (w_p, w_n) \not\models_M \bot & & \\
\mathbf{I} & (w_p, w_n) \models_M \mathbf{I} & & \\
p & (w_p, w_n) \models_M p & \equiv & p \in V(w_p) \\
\varphi \wedge \psi & (w_p, w_n) \models_M \varphi \wedge \psi & \equiv & (w_p, w_n) \models_M \varphi \textit{ and } (w_p, w_n) \models_M \psi \\
\varphi \vee \psi & (w_p, w_n) \models_M \varphi \vee \psi & \equiv & (w_p, w_n) \models_M \varphi \textit{ or } (w_p, w_n) \models_M \psi \\
\varphi \supset \psi & (w_p, w_n) \models_M \varphi \supset \psi & \equiv & \forall (w', w'') \in W \times W.(w_p, w_n) R_{>} (w', w'') \wedge \\
 & & & (w', w'') \models_M \varphi \textit{ implies } (w', w'') \models_M \psi \\
\neg\varphi & (w_p, w_n) \models_M \neg\varphi & \equiv & (w_n, w_p) \Vdash_{M^\perp} \varphi \\
\mathbf{0} & (w_n, w_p) \Vdash_{M^\perp} \mathbf{0} & & \\
\mathbf{1} & (w_n, w_p) \not\Vdash_{M^\perp} \mathbf{1} & & \\
\bot & (w_n, w_p) \not\Vdash_{M^\perp} \bot & & \\
\mathbf{I} & (w_n, w_p) \Vdash_{M^\perp} \mathbf{I} & & \\
p & (w_n, w_p) \Vdash_{M^\perp} p & \equiv & p \notin V(w_n) \\
\varphi \wedge \psi & (w_n, w_p) \Vdash_{M^\perp} \varphi \wedge \psi & \equiv & (w_n, w_p) \Vdash_{M^\perp} \varphi \textit{ or } (w_n, w_p) \Vdash_{M^\perp} \psi \\
\varphi \vee \psi & (w_n, w_p) \Vdash_{M^\perp} \varphi \vee \psi & \equiv & (w_n, w_p) \Vdash_{M^\perp} \varphi \textit{ and } (w_n, w_p) \Vdash_{M^\perp} \psi \\
\varphi \supset \psi & (w_n, w_p) \Vdash_{M^\perp} \varphi \supset \psi & \equiv & (w_p, w_n) \models_M \varphi \textit{ and } (w_n, w_p) \Vdash_{M^\perp} \psi \\
\neg\varphi & (w_n, w_p) \Vdash_{M^\perp} \neg\varphi & \equiv & (w_p, w_n) \models_M \varphi
\end{array}
$$

where we defined $\mathbb{M}^\perp$ *in the last chapter.*

Definition 86 *Let* $M = \langle W, R, V\rangle$ *be a propositional Kripke structure. A formula* φ *of propositional CD is* satisfied *in* M *if and only if* $\exists w_p, w_n \in W.\ w_p \neq w_n$ *and* $(w_p, w_n) \models_M \varphi$. *A propositional formula* φ *is* valid in M *if and only if* $\forall w_p, w_n \in W.\ w_p \neq w_n$ *and* $(w_p, w_n) \models_M \varphi$. *A propositional CD formula* φ *is* valid *if and only if* φ *is valid in all propositional Kripke structures.*

Lemma 87 Monotonicity *Let $M = \langle W, R, V\rangle$ be a propositional intuitionistic model over P. Then we have $(w_p, w_n), (w', w'') \in W \times W$. such that $(w_p, w_n)R_{>}(w', w'')$, and for all formula φ of constructible duality,*

Truth $(w_p, w_n) \models_M \varphi$ *implies* $(w', w'') \models_M \varphi$ *, and*

Falsity $(w_n, w_p) \Vdash_{M^\perp} \varphi$ *implies* $(w'', w') \Vdash_{M^\perp} \varphi$.

Proof: By straightforward induction on the structure of φ.

∎

Theorem 88 Soundness

All rules of CD are sound.

That is if M is any Kripke structure, then $\forall Rule \in$ CD. $\forall w_n, w_p \in W.\ w_n \neq w_p$.

$$(w_p, w_n) \models_M Antecedent(Rule) \text{ implies } (w_p, w_n) \models_M Succedent(Rule)$$

Outline of Proof: We need to consider all rules and show that when the antecedent is true, the succedent is true. To show $(w_p, w_n) \models_M \Gamma \Rightarrow \Delta$ we must show $(w_p, w_n) \models_m \wedge\Gamma \supset \vee\Delta$. ∎

4.6.1 Completeness

Our completeness theorem in this section asserts that any underivable formula is refutable in a canonical model. In fact, the canonical refutation model $\mathcal{K}$ for intuitionistic logic I_c is, in some sense, the canonical refutation model for constructible duality.

The argument in the section is summarized as follows. Any complete underivable sequent $\Gamma \Rightarrow \Delta$ in CD can be syntactically partitioned into $\Gamma_p, \Gamma_n, \Gamma_o \Rightarrow \Delta_p, \Delta_n, \Delta_o$ where each φ in $\Gamma_p \cup \Delta_p$ is in I_c, and each φ in $\Gamma_n \cup \Delta_n$ is a formula of the form $\neg\psi$ where ψ is in I_c. Then $\Gamma_p \Rightarrow \Delta_p$ is a complete sequent for I_c; hence $\Gamma_p \Rightarrow \Delta_p$ is a world in $\mathcal{K}$. The formula-wise contrapositive of $\Gamma_n \Rightarrow \Delta_n$, denoted $\neg\Delta_n \Rightarrow \neg\Gamma_n$, is also a world in K. These are the two worlds in $\mathcal{K}$ that refute $\Gamma \Rightarrow \Delta$.

Definition 89 *Let $\Gamma \Rightarrow \Delta$ be an infinite sequent in constructible duality then $\Gamma \Rightarrow \Delta$ is* derivable *if there exists finite subsets $\Gamma_f \subset \Gamma$ and $\Delta_f \subset \Delta$ such that $\Gamma_f \Rightarrow \Delta_f$ is derivable in constructible duality respectively.*

Definition 90 *We say that a sequent in constructible duality,* $\Gamma \Rightarrow \Delta$ *is* falsified *in a structure* M *if there exists two worlds* $w_p, w_n \in W$ *such that* $w_p \neq w_n$ *and* $(w_p, w_n) \models_M \wedge\Gamma$ *and* $(w_p, w_n) \not\models_M \vee\Delta$*.*

Definition 91 *Let* Θ *be a set of formulas. We define* $\neg\Theta = \{\neg\varphi \mid \varphi \in \Theta \subseteq I_c\}$ *and we overload that* $\neg\Theta$ *is also* $\neg\Theta = \{\varphi \mid \neg\varphi \in \Theta \subseteq \neg I_c\}$*.*

Lemma 92 *If* Θ *is a set of formulas of constructible duality, then* Θ *can be syntactically partitioned into* $\Theta_p \cup \Theta_n \cup \Theta_o$ *where* Θ_p *is a set of formulas in* I_c*,* Θ_n *is a set of formulas in* $\neg I_c$ *and* Θ_o *is disjoint from* $I_c \cup \neg I_c$*.*

Lemma 93 *For any complete sequent* $\Gamma_p, \Gamma_n, \Gamma_o \Rightarrow \Delta_p, \Delta_n, \Delta_o$ *for* CD*,* $\Gamma_p \Rightarrow \Delta_p$ *is a complete sequent for propositional intuitionistic logic with constants.*

Proof: $\Gamma \Rightarrow \Delta$ is underivable, thus every subset of $\Gamma \Rightarrow \Delta$ is underivable hence $\Gamma_p \Rightarrow \Delta_p$ is underivable.

To see that $\Gamma_p \Rightarrow \Delta_p$ is complete in I_c, we know that $I \subsetneq CD$ and that $\Gamma_p, \Gamma_n, \Gamma_o \Rightarrow \Delta_p, \Delta_n, \Delta_o$ is complete for CD. Therefore, every formula in I_c occurs somewhere in $\Gamma_p, \Gamma_n, \Gamma_o \Rightarrow \Delta_p, \Delta_n, \Delta_o$. By the syntactic form of the formulas in I_c they can only occur in $\Gamma_p \cup \Delta_p$. ∎

Corollary 94 $\Gamma_p \Rightarrow \Delta_p$ *is a world in* $\mathcal{K}$*.*

Lemma 95 *For any complete sequent* $\Gamma_p, \Gamma_n, \Gamma_o \Rightarrow \Delta_p, \Delta_n, \Delta_o$ *for* CD*,* $\neg\Delta_n \Rightarrow \neg\Gamma_n$ *is a complete sequent for* I_c *i.e.* $\neg\Delta_n \Rightarrow \neg\Gamma_n$ *is a world in* $\mathcal{K}$*.*

Proof: The arguments for the completeness of $\Gamma_n \Rightarrow \Delta_n$ for $\neg I_c$ are the same as for $\Gamma_p \Rightarrow \Delta_p$ for I_c. By the syntactic structure $\neg\Delta_n \Rightarrow \neg\Gamma_n$ could only be derivable if $\Gamma_n \Rightarrow \Delta_n$ were derivable; $\Gamma_n \Rightarrow \Delta_n$ is not derivable. Further $\neg\Delta_n \Rightarrow \neg\Gamma_n$ is complete for I_c because $\Gamma_n \Rightarrow \Delta_n$ is complete for $\neg I_c$. ∎

Now the technical phase of the argument, we show that for a given complete sequent $\Gamma \Rightarrow \Delta$, $\Gamma_p \Rightarrow \Delta_p$ and $\neg\Delta_n \Rightarrow \neg\Gamma_n$ refutes $\Gamma \Rightarrow \Delta$ in $\mathcal{K}$. We use saturatedness of sequents in order to obtain this result. Since CD only adds invertible rules, we must check that any CD-sequent is saturated for invertible rules.

Lemma 96 *Any complete sequent $\Gamma \Rightarrow \Delta$ for CD is saturated. That is it satisfies Definition 81 and also satisfies for all $\varphi, \psi \in CD$, $\neg \mathbf{0} \in \Gamma$, $\neg \mathbf{1} \in \Delta$ and*

Rule	*Sequent*		*Saturatedness*
$\neg\wedge \Rightarrow$	$\neg(\varphi \wedge \psi) \in \Gamma$	*implies*	$\neg\varphi \in \Gamma$ *or* $\neg\psi \in \Gamma$
$\Rightarrow \neg\vee$	$\neg(\varphi \vee \psi) \in \Delta$	*implies*	$\neg\varphi \in \Delta$ *or* $\neg\psi \in \Delta$
$\neg\vee \Rightarrow$	$\neg(\varphi \vee \psi) \in \Gamma$	*implies*	$\neg\varphi \in \Gamma$ *and* $\neg\psi \in \Gamma$
$\Rightarrow \neg \supset$	$\neg(\varphi \supset \psi) \in \Delta$	*implies*	$\varphi \in \Delta$ *or* $\neg\psi \in \Delta$
$\neg \supset \Rightarrow$	$\neg(\varphi \supset \psi) \in \Gamma$	*implies*	$\varphi \in \Gamma$ *and* $\neg\psi \in \Gamma$
$\Rightarrow \neg\neg$	$\varphi \in \Delta$	*implies*	$\neg\neg\varphi \in \Delta$
$\neg\neg \Rightarrow$	$\varphi \in \Gamma$	*implies*	$\neg\neg\varphi \in \Gamma$
$\Rightarrow \neg\bot$	$\bot \in \Delta$	*implies*	$\neg\bot \in \Delta$
$\neg\mathbf{I} \Rightarrow$	$\mathbf{I} \in \Gamma$	*implies*	$\neg\mathbf{I} \in \Gamma$

Lemma 97 *The set of complete sequents for CD is saturated.*

Outline of Proof: The only non-invertible rule is implication on the right hand side, hence Lemma 83 and Lemma 96 suffice. ∎

Now armed with the information that each world in $\mathcal{K}$ is saturated and that the set of worlds in $\mathcal{K}$ is saturated for I_c and that each complete CD sequent $\Gamma \Rightarrow \Delta$ is saturated, we show that the interval $(\Gamma_p \Rightarrow \Delta_p, \neg\Delta_n \Rightarrow \neg\Gamma_n)$ in $\mathcal{K}$ falsifies the sequent $\Gamma \Rightarrow \Delta$ where $\Gamma_p \cup \Delta_p \cup \Gamma_n \cup \Delta_n \subsetneq \Gamma \cup \Delta$.

We do this in three lemmas:

1. The interval $(\Gamma_p \Rightarrow \Delta_p, \neg\Delta_n \Rightarrow \neg\Gamma_n)$ refutes $\Gamma_p \Rightarrow \Delta_p$.

2. The interval $(\Gamma_p \Rightarrow \Delta_p, \neg\Delta_n \Rightarrow \neg\Gamma_n)$ refutes $\Gamma_n \Rightarrow \Delta_n$.

3. This suffices to refute $\Gamma \Rightarrow \Delta$.

This lemma states that the interval refutes the positive part of the sequent.

Lemma 98 *For all complete sequents $\Gamma_p, \Gamma_n, \Gamma_o \Rightarrow \Delta_p, \Delta_n, \Delta_o$ in CD such that $\Gamma_p \neq \neg\Delta_n$, the pair $(\Gamma_p \Rightarrow \Delta_p, \neg\Delta_n \Rightarrow \neg\Gamma_n)$ falsifies $\Gamma_p \Rightarrow \Delta_p$ in $\mathcal{K}$.*

Outline of Proof: We need simultaneous induction on the positive formula θ such that for every $\theta \in \Gamma_p$ $(\Gamma_p \Rightarrow \Delta_p, \neg\Delta_n \Rightarrow \neg\Gamma_n) \models_K \theta$ and for every $\theta \in \Delta_p$, $(\Gamma_p \Rightarrow \Delta_p, \neg\Delta_n \Rightarrow \neg\Gamma_n) \not\models_K \theta$.

We use $\Gamma_p \Rightarrow \Delta_p \not\models_{\mathcal{K}} \Gamma_p \Rightarrow \Delta_p$ from Theorem 80 for all cases except implication.

When we have an implication over a non-proper subinterval then Lemma 83 suffices because implication is judged forward from a world. If however, we are looking at subintervals, then saturation of the set of complete sequents from Lemma 83 is not enough. We must check that when we have a proper subinterval $\neg\Delta_n \Rightarrow \neg\Gamma_n$ is *high* enough above the positive world to satisfy the needed hypotheses in Lemma 83.

$\theta = \varphi \supset \psi$: If $\theta \in \Gamma_p$, then by saturation, monotonicity and the definition of a positive formula $\psi \in \Gamma'$ or $\varphi \in \Delta'$ for every $\Gamma' \Rightarrow \Delta'$ above $\Gamma_p \Rightarrow \Delta_p$ in $\mathcal{K}$. By the hypothesis that, $\neg\Delta_n \Rightarrow \neg\Gamma_n$ is above $\Gamma \Rightarrow \Delta$ in $\mathcal{K}$, thus $\varphi \in \neg\Gamma_n$ or $\psi \in \neg\Delta_n$. Hence at every subinterval, $(\Gamma_p \Rightarrow \Delta_p, \neg\Delta_n \Rightarrow \neg\Gamma_n) \models_K \varphi \supset \psi$ as needed.

If $\theta \in \Delta_p$ then by saturation Lemma 83 and the definition of a positive formula we have have some $\Gamma' \Rightarrow \Delta'$ reachable from $\Gamma_p \Rightarrow \Delta_p$ where $\varphi \in \Gamma'$ and $\psi \in \Delta'$.

To ensure that $\Gamma' \Rightarrow \Delta'$ happens in the interval, we use the monotonicity built into R_k and show that (1) $\varphi \in \neg\Delta_n$ and (2) that ψ is in Δ_p.

(1) $\neg\varphi$ cannot occur in Γ_n without satisfying θ, thus $\neg\varphi$ must be in Δ_n, i.e. $\varphi \in \neg\Delta_n$.

(2) ψ cannot be in Γ_p without satisfying θ, therefore $\psi \in \Delta_p$.

So the bottom of the interval has ψ in the RHS of the sequent, and the top of the interval has φ in the LHS of the sequent, so we use Lemma 83 and the monotonicity built into R_k to conclude that there must be a world $\Gamma' \Rightarrow \Delta'$ in the interval such that $\varphi \in \Gamma'$ and $\psi \in \Delta'$. One such world is enough to refute $\varphi \supset \psi$ over the entire interval hence $(\Gamma_p \Rightarrow \Delta_p, \neg\Delta_n \Rightarrow \neg\Gamma_n) \not\models_K \varphi \supset \psi$. ∎

This lemma states that the constructed pair of worlds refutes the negative formulas in the sequent.

Lemma 99 *For all complete sequents* $\Gamma_p, \Gamma_n, \Gamma_o \Rightarrow \Delta_p, \Delta_n, \Delta_o$ *for* CD *such that* $\Gamma_p \neq \neg\Delta_n$, *the pair* $(\Gamma_p \Rightarrow \Delta_p, \neg\Delta_n \Rightarrow \neg\Gamma_n)$ *falsifies* $\Gamma_n \Rightarrow \Delta_n$ *in* $\mathcal{K}$.

Proof: We use simultaneous induction on the negative formulas θ such that for every $\theta \in \Gamma_n$ $(\Gamma_p \Rightarrow \Delta_p, \neg\Delta_n \Rightarrow \neg\Gamma_n) \models_K \theta$ and for every $\theta \in \Delta_n$ we have that $(\Gamma_p \Rightarrow \Delta_p, \neg\Delta_n \Rightarrow \neg\Gamma_n) \not\models_K \theta$.

We use $\neg\Delta_n \Rightarrow \neg\Gamma_n \not\models_{\mathcal{K}} \neg\Delta_n \Rightarrow \neg\Gamma_n$ from Theorem 80 for all cases except implication. The negation of $\varphi \supset \psi$ is $\varphi \wedge \neg\psi$, therefore wherever $\neg(\varphi \supset \psi)$ appears in the sequent so does $\varphi \wedge \neg\psi$. We apply saturation and Theorem 80 for $\neg\psi$, and Lemma 98 for φ. ∎

This lemma states that whenever you refute the wholly positive formulas and the wholly negative formulas then you must refute the mixed formulas; hence refuting the entire sequent.

Theorem 100 *For all complete sequents $\Gamma \Rightarrow \Delta = \Gamma_p, \Gamma_n, \Gamma_o \Rightarrow \Delta_p, \Delta_n, \Delta_o$ in CD such that $\Gamma_p \neq \neg\Delta_n$ every pair of worlds which falsifies $\Gamma_p, \Gamma_n \Rightarrow \Delta_p, \Delta_n$ falsifies $\Gamma \Rightarrow \Delta$.*

Proof: By Lemmas 98 and 99, we only need to show that every formula in Γ_o is satisfied and every formula in Δ_o is refuted. To do this we use structural induction for formulas θ in $\Gamma_o \cup \Delta_o$. This proof only uses saturatedness and the induction hypotheses again except for implication in Δ_o but since this is the most important lemma, we go through it.

$\theta = \perp$**:** By axiom (7) of CD, $\perp$ can only occur in Δ_o otherwise the sequent would be derivable. The occurrence of $\perp$ on the right must be refuted as the sequent is underivable.

$\theta =$ **I:** By axiom (6) of CD, **I** can only occur on Γ_o otherwise the sequent would be derivable. The occurrence of **I** on the left must be satisfied as the sequent is underivable.

$\theta = p$ **base:** These formulas cannot be in $\Gamma_o \cup \Delta_o$.

$\theta = \varphi \wedge \psi$ **:** By expanding the definition of a mixed formula into cases, φ or ψ can be mixed. When this is the case we use saturation and the induction hypotheses for the mixed formula. Otherwise, assume (without loss of generality) φ is positive and ψ is negative. By saturation φ and ψ are in $\Gamma_p \cup \Gamma_n \cup \Delta_p \cup \Delta_n$ which by Lemmas 98 and 99 are refuted.

$\theta = \varphi \vee \psi$ **:** This case is very similar to the $\wedge$ case above.

$\theta = \varphi \supset \psi$ **:** If $\theta \in \Gamma_o$, then by examining the cases and using both saturation and the induction hypothesis, we are finished.

If $\theta \in \Delta_o$, the case where both are mixed needs the induction hypothesis and saturation.

Assume (without loss of generality) that φ is positive and ψ is negative. Let $\psi = \neg\psi'$. Again if we have a non-proper interval then the saturation Lemma 83 and the induction hypothesis suffice. Otherwise, we have a proper subinterval.

We must show that there is a $\Gamma' \Rightarrow \Delta'$ happens in the interval, such that by monotonicity of R_k we need (1) that $\varphi \in \neg\Delta_n$ and (2) that ψ is in Δ_n.

(1) $\neg\varphi$ cannot occur in Γ_n without satisfying θ, thus $\neg\varphi$ must be in Δ_n, i.e. $\varphi \in \neg\Delta_n$.

(2) If ψ cannot be in Γ_n without satisfying θ, therefore $\psi \in \Delta_n$.

Then $\varphi \wedge \psi' \in \neg\Delta_n$ thus refuting $\varphi \supset \neg\psi'$ in the interval.

The case where φ is negative and ψ is positive is similar; and the case where both are mixed needs the induction hypothesis and saturation.

$\theta = \neg\varphi$ **:** For this case to be a mixed formula φ then φ is mixed, so we use the induction hypothesis.

∎

The following theorem is completeness for complete underivable sequents.

Theorem 101 *Let $\Gamma_p, \Gamma_n, \Gamma_o \Rightarrow \Delta_p, \Delta_n, \Delta_o$, be a complete sequent for CD, then the pair $(\Gamma_p \Rightarrow \Delta_p, \neg\Delta_n \Rightarrow \neg\Gamma_n)$ such that $\Gamma_p \neq \neg\Delta_n$ falsifies $\Gamma \Rightarrow \Delta$ in $\mathcal{K}$.*

Proof: This pair of worlds falsifies $\Gamma_p, \Gamma_n \Rightarrow \Delta_p, \Delta_n$ by application of Lemma 98 and Lemma 99. This suffices by application of Theorem 100. ∎

This corollary is completeness for underivable sequents.

Corollary 102 *Let $\Gamma' \Rightarrow \Delta'$ be a finite underivable sequent in CD, then it can be extended to a complete sequent $\Gamma_p, \Gamma_n, \Gamma_o \Rightarrow \Delta_p, \Delta_n, \Delta_o$ such that $\Gamma_p \neq \neg\Delta_n$ and $(\Gamma_p \Rightarrow \Delta_p, \neg\Delta_n \Rightarrow \neg\Gamma_n)$ falsifies $\Gamma \Rightarrow \Delta$.*

Proof: For all finite underivable sequents $\Gamma' \Rightarrow \Delta'$, they can be extended to complete underivable sequents $\Gamma \Rightarrow \Delta$ such that $\Gamma_p \neq \neg\Delta_n$ where we can apply Theorem 101. ∎

4.7 Alternate Formalization

Lemma 103 *Every formula φ in CD is equivalent to a formula, say φ_{atom}, where $\neg$ only scopes propositional atoms.*

Outline of Proof: By examining each formula φ, and noting the following equivalences of interderivability in CD.

$$\begin{array}{rcl}
\neg(\mathbf{0}) & \equiv & \mathbf{1} \\
\neg(\mathbf{1}) & \equiv & \mathbf{0} \\
\neg(\mathbf{I}) & \equiv & \mathbf{I} \\
\neg(\bot) & \equiv & \bot \\
\neg(\varphi \wedge \psi) & \equiv & \neg\varphi \vee \neg\psi \\
\neg(\varphi \vee \psi) & \equiv & \neg\varphi \wedge \neg\psi \\
\neg(\varphi \supset \psi) & \equiv & \varphi \wedge \neg\psi \\
\neg(\neg\varphi) & \equiv & \varphi
\end{array}$$

By structural induction using these equivalences we derive an equivalent formula φ_{atom} where negation occurs only outside propositional atoms. ∎

If φ is a formula where negations only scope propositional atoms then, φ is in *positive constant normal form.*

Lemma 104 *Every formula φ in CD in* positive constant normal form *is equivalent to a formula, say φ_{int}, where only constants from intuitionistic logic appear.*

Outline of Proof: By examining each formula φ in positive constant normal form, and noting the following equivalences of interderivability in CD for formulas in positive constant normal form.

$$\begin{array}{rcl}
(\mathbf{I}) & \equiv & \mathbf{1} \\
(\bot) & \equiv & \mathbf{0}
\end{array}$$

By structural induction using these equivalences we derive an equivalent formula φ_{int} where negation does not occur except scoping propositional atoms nor the constants $\bot$ and $\mathbf{I}$. ∎

If φ is a formula where negations only scope propositional atoms and has constants for intuitionistic logic then, φ is in *positive normal form.* A positive normal form entailment relation for CD is defined over a propositional intuitionistic model.

Definition 105 *Let $A = \langle W, R, V\rangle$ be an intuitionistic propositional Kripke model over P. We define $\models_A$ inductively over formulas φ in positive normal form. For any $w_p, w_n \in W$ such that $w_p \neq w_n$.*

$$\begin{array}{lcr}
(w_p, w_n) \models_A \mathbf{1} & = & \\
(w_p, w_n) \not\models_A \mathbf{0} & = & \\
(w_p, w_n) \models_A p & = & V(w_p, p) = 1 \\
(w_p, w_n) \models_A \varphi \wedge \psi & = & (w_p, w_n) \models_A \varphi \textit{ and } (w_p, w_n) \models_A \psi \\
(w_p, w_n) \models_A \varphi \vee \psi & = & (w_p, w_n) \models_A \varphi \textit{ or } (w_p, w_n) \models_A \psi \\
(w_p, w_n) \models_A \varphi \supset \psi & = & \forall (w', w'') \in W \times W.(w_n, w_p) R_{>} (w', w'') \wedge \\
 & & (w', w'') \models_A \varphi \rightarrow (w', w'') \models_A \psi \\
(w_p, w_n) \models_A \neg p & = & V(w_n, p) = 0
\end{array}$$

Theorem 106 *When φ is logically equivalent to a formula in positive normal form ψ and $\langle W, R, V\rangle$ is a Kripke model, then $\forall w_p, w_n \in W$. such that $w_p R w_n$ and $w_p \neq w_n$,*

$$(w_p, w_n) \models_A \psi \textit{ if and only if } (w_p, w_n) \models_M \varphi$$

where $\models_A$ is from Definition 105 and $\models_M$ is from Definition 85.

4.8 First Order Case

Again for the first order case, we extend the language to include countably many bound variables, here denoted $x, y, z, \ldots$ and countably many free variables, here denoted $a, b, c, \ldots$ for each natural n, we have countably many predicates of arity n, denoted $P^n, Q^n, \ldots$.

Terms are constructed from individuum variables and parameters. Atomic formulas are of the form $P^n(t_1, \ldots, t_n)$ where $t_i, 1 \leq i \leq n$ are terms. Formulas are constructed via atomic formulas via $\wedge, \vee, \sim, \exists, \forall$.

$\varphi[t/x]$ is the results of substituting t for all occurrences of x in φ. If the variable is evident, we simply denote this as $\varphi[t]$. In most cases t will denote a ground terms, that is one which contains parameters but not variables; thus collisions are impossible.

We add the following rules to those of propositional CD. **Rules of Inference**

$$\Rightarrow \exists \quad \frac{\Gamma \Rightarrow \Delta, \exists x.\varphi, \varphi[t]}{\Gamma \Rightarrow \Delta, \exists x.\varphi} \qquad \exists \Rightarrow \quad \frac{\varphi[b], \Gamma \Rightarrow \Delta}{\exists x.\varphi, \Gamma \Rightarrow \Delta}$$

$$\Rightarrow \forall \quad \frac{\Gamma \Rightarrow \Delta, \varphi[b]}{\Gamma \Rightarrow \Delta, \forall x.\varphi} \qquad \forall \Rightarrow \quad \frac{\varphi[t], \forall x.\varphi, \Gamma \Rightarrow \Delta}{\forall x.\varphi, \Gamma \Rightarrow \Delta}$$

$$\Rightarrow \neg\exists \quad \frac{\Gamma \Rightarrow \Delta, \neg\varphi[b]}{\Gamma \Rightarrow \Delta, \neg\exists x.\varphi} \qquad \neg\exists \Rightarrow \quad \frac{\neg\varphi[t], \neg\exists x.\varphi, \Gamma \Rightarrow \Delta}{\neg\exists x.\varphi, \Gamma \Rightarrow \Delta}$$

$$\Rightarrow \neg\forall \quad \frac{\Gamma \Rightarrow \Delta, \neg\varphi[t], \neg\forall x.\varphi}{\Gamma \Rightarrow \Delta, \neg\forall x.\varphi} \qquad \neg\forall \Rightarrow \quad \frac{\neg\varphi[b], \Gamma \Rightarrow \Delta}{\neg\forall x.\varphi, \Gamma \Rightarrow \Delta}$$

Where t is a ground term, b is a new parameter (eigenvariable). Note that the introduction of the $\forall$ is not invertible.

Now to define models we need a notion of a domain over which a quantifier ranges.

The following definition is the standard one for first order intuitionistic logic that we gave in Chapter 2, but we are giving it here as a convenience to the reader.

Definition 107 First Order Model *A Kripke model for first order intuitionistic logic over universe U is a four-tuple $K = \langle W, R, V, D\rangle$ where W is a set, and $R \subseteq W \times W$ is a reflexive transitive relation and*

1. *D is a function from W to a subset of U satisfying wRw' implies $D(w) \subseteq D(w')$, and*

2. *V is a function from worlds W and n-ary predicates P^n to $(D(w))^n \rightarrow \{0,1\})$ satisfying the monotonicity condition that if wRw' then $V(w', P)$ is an extension of $V(w, P)$ i.e. for all $x_1, \ldots x_n \in D(w)$, $V(w, P)(x_1, \ldots, x_n)$ implies $V(w', p)(x_1, \ldots, x_n)$.*

Again we remind of this previous definition from Chapter 3.

Definition 108 *Let $M = \langle W, R, V, D\rangle$ be a first-order intuitionistic model, then we define its dual as $M^{\perp} = \langle W, R^{\cup}, \bar{V}, D\rangle$ where $R^{\cup}$ is the inverse of R and for all $w \in W$ $(\bar{V}(w, P^n) = (D(w))^n - V(w, P^n))$.*

Again all formulae judged at a pair (w, w') are to be understood as being a formula in the language of the pair then all parameters of the formula are drawn from $D(w) \cap D(w')$.

Definition 109 *A first-order interval model for CD is a first-order Kripke structure $M = \langle W, R, V, D\rangle$ with the entailment relation $\models_M$ for truth and $\Vdash_{M^\perp}$ for falsity defined by recursion on φ. For any $w_p, w_n \in W$. such that $w_p \neq w_n$.*

$\mathbf{0}$	$(w_p, w_n) \not\models_M \mathbf{0}$		
$\mathbf{1}$	$(w_p, w_n) \models_M \mathbf{1}$		
$\perp$	$(w_p, w_n) \not\models_M \perp$		
$\mathbf{I}$	$(w_p, w_n) \models_M \mathbf{I}$		
A	$(w_p, w_n) \models_M A(x_1, \ldots, x_n)$	$\equiv$	$V(w_p, A)(x_1, \ldots, x_n) = 1$ *for* $(x_1, \ldots, x_n) \in D(w_p)$
$\varphi \wedge \psi$	$(w_p, w_n) \models_M \varphi \wedge \psi$	$\equiv$	$(w_p, w_n) \models_M \varphi$ *and* $(w_p, w_n) \models_M \psi$
$\varphi \vee \psi$	$(w_p, w_n) \models_M \varphi \vee \psi$	$\equiv$	$(w_p, w_n) \models_M \varphi$ *or* $(w_p, w_n) \models_M \psi$
$\varphi \supset \psi$	$(w_p, w_n) \models_M \varphi \supset \psi$	$\equiv$	$\forall (w', w'') \in W \times W.(w_p, w_n) R_{>} (w', w'') \wedge$ $(w', w'') \models_M \varphi$ *implies* $(w', w'') \models_M \psi$
$\forall x.\varphi(x)$	$(w_p, w_n) \models_M \forall x.\varphi(x)$	$\equiv$	$\forall (w', w'') \in W \times W.(w_p, w_n) R_{>} (w', w'') \wedge$ *for every* $d \in D(w') \cap D(w'')$ $(w', w'') \models_M \varphi[d]$
$\exists x.\varphi(x)$	$(w_p, w_n) \models_M \exists x.\varphi(x)$	$\equiv$	$\forall (w', w'') \in W \times W.(w_p, w_n) R_{>} (w', w'') \wedge$ *for some* $d \in D(w') \cap D(w'')$ $(w', w'') \models_M \varphi[d]$
$\neg\varphi$	$(w_p, w_n) \models_M \neg\varphi$	$\equiv$	$(w_n, w_p) \Vdash_{M^\perp} \varphi$

And for falsity

$$
\begin{array}{llcl}
\mathbf{0} & (w_n, w_p) \Vert\!\models_{M^\perp} \mathbf{0} & & \\
\mathbf{1} & (w_n, w_p) \not\Vert\!\models_{M^\perp} \mathbf{1} & & \\
\perp & (w_n, w_p) \not\Vert\!\models_{M^\perp} \perp & & \\
\mathbf{I} & (w_n, w_p) \Vert\!\models_{M^\perp} \mathbf{I} & & \\
A & (w_n, w_p) \Vert\!\models_{M^\perp} A(x_1, \ldots, x_n) & \equiv & \bar{V}(w_n, A)(x_1, \ldots, x_n) = 1 \\
 & & & \quad \textit{for } (x_1, \ldots, x_n) \in D(w_n) \\
\varphi \wedge \psi & (w_n, w_p) \Vert\!\models_{M^\perp} \varphi \wedge \psi & \equiv & (w_n, w_p) \Vert\!\models_{M^\perp} \varphi \textit{ or } (w_n, w_p) \Vert\!\models_{M^\perp} \psi \\
\varphi \vee \psi & (w_n, w_p) \Vert\!\models_{M^\perp} \varphi \vee \psi & \equiv & (w_n, w_p) \Vert\!\models_{M^\perp} \varphi \textit{ and } (w_n, w_p) \Vert\!\models_{M^\perp} \psi \\
\varphi \supset \psi & (w_n, w_p) \Vert\!\models_{M^\perp} \varphi \supset \psi & \equiv & (w_p, w_n) \models_M \varphi \textit{ and } (w_n, w_p) \Vert\!\models_{M^\perp} \psi \\
\forall x.\varphi(x) & (w_n, w_p) \Vert\!\models_{M^\perp} \forall x.\varphi(x) & \equiv & \forall (w', w'') \in W \times W.(w_p, w_n) R_{>} (w', w'') \wedge \\
 & & & \quad \textit{for some } d \in D(w') \cap D(w'') \;\; (w'', w') \Vert\!\models_{M^\perp} \varphi[d] \\
\exists x.\varphi(x) & (w_n, w_p) \Vert\!\models_{M^\perp} \exists x.\varphi(x) & \equiv & \forall (w', w'') \in W \times W.(w_p, w_n) R_{>} (w', w'') \wedge \\
 & & & \quad \textit{for every } d \in D(w') \cap D(w'') \;\; (w'', w') \Vert\!\models_{M^\perp} \varphi[d] \\
\neg\varphi & (w_n, w_p) \Vert\!\models_{M^\perp} \neg\varphi & \equiv & (w_p, w_n) \models_M \varphi
\end{array}
$$

Definition 110 *Let $M = \langle W, R, V, D \rangle$ be a first-order Kripke structure. A formula φ of first-order CD is* satisfied *in M if and only if $\exists w_p, w_n \in W.\ w_p \neq w_n$ and $(w_p, w_n) \models_M \varphi$. A formula φ is* valid *in M if and only if $\forall w_p, w_n \in W.\ w_p \neq w_n$ and $(w_p, w_n) \models_M \varphi$. A first-order CD formula φ is* valid *if and only if φ is valid in all first-order Kripke structures.*

Theorem 111 Soundness and Completeness *A first-order CD formula φ is derivable if and only if it is valid in all first-order interval models for CD.*

4.9 Comparison with Relevance $*$-Models

Relevance logic has a model theory from the "Australian" school which bears a resemblance to the interval models which we described here. Even though there are logics such as interval-temporal logics which mention intervals in the name, the intervals in interval-temporal logic manifest themselves by logical operations not just in model theory.

Therefore, we are going to study the generality of this $*$ approach to negation. We will examine $*$ models of relevance logic and their relationship to models for Nelson's N (Patterson 1996b) and the models of CD presented herein.

The logic of relevance logic R has single conclusion Gentzen rules for its propositional system; this system can be found in (Dunn 1986). The syntax of propositional R logic is — atomic formulas are elements drawn from a set of propositional atoms; a formula of propositional intuitionistic logic is either an atom, the negation $\neg$ of a formula, the *extensional* conjunction $\wedge$ of two formulas, the *intentional* conjunction $\circ$ of two formulas, the disjunction $\vee$ of two formulas, or the *relevant* implication $\supset$ of one formula by another.

For an implication to be relevant "weakening on the left" can be used only in a restricted manner. For *CD*, classical and intuitionistic logic, whenever φ is valid, then $\psi \supset \varphi$ is valid for any ψ. However, this is just the sort of principle that is denied in relevance logic because if φ is already valid then ψ is not *relevant* to proving φ.

Since we are mainly focusing on certain similarities between the models we present the $*$ models of relevance logic following (Dunn 1986, Restall 1995).

Definition 112 A $*$ Model *A relevant model $R_M = \langle w_*, W, R, V, * \rangle$ over a set of atoms P is a set W, a distinguished start world $w_* \in W$, a ternary relation $R \subseteq W \times W \times W$, and valuation function $V{:}\, W \times P \to \{0,1\}$, such that $\forall w, w' \in W.\ \ \forall p \in P.$ if $w \leq w'$ then if $V(w,p) = 1$ then $V(w',p) = 1$ where $w \leq w' \iff R(w_*, w, w')$.*

Further R has to satisfy

Identity $\forall w \in W.R(w', w, w)$

Commutativity $\forall w_1, w_2, w_3 \in W.R(w_1, w_2, w_3)$ *implies* $R(w_2, w_1, w_3)$

Associativity $\forall w_1, w_2, w_3, w_4 \in W.\exists w.R(w_1, w_2, w)$ *and* $R(w, w_3, w_4)$ *implies* $\exists w.R(w_1, w, w_4)$ *and* $R(w_2, w_3, w)$.

Idempotence $\forall w \in W.R(w, w, w)$.

Monotony $\forall w_1, w_2, w_3 \in W.w_0 \leq w_1$ *and* $R(w_1, w_2, w_3)$ *implies* $R(w_0, w_2, w_3)$.

and $*$ has to satisfy

Period Two $\forall w \in W.w^{**} = w$

Inversion $\forall w_1, w_2, w_3 \in W.R(w_1, w_2, w_3)$ *implies* $R(w_1, w_3^*, w_2^*)$. *This implies that if* $w_1 \leq w_2$ *then* $w_2^* \leq w_1^*$. *The simpler case in terms of* $\leq$ *is called* **Order Reversal**.

We extend the valuation on propositional atoms to all formulae.

$$
\begin{array}{lcl}
V(w, \neg\varphi) = 1 & \Longleftrightarrow & V(w^*, \varphi) = 0 \\
V(w, \varphi \wedge \psi) = 1 & \Longleftrightarrow & V(w, \varphi) = 1 \textit{ and } V(w, \psi) = 1 \\
V(w, \varphi \vee \psi) = 1 & \Longleftrightarrow & V(w, \varphi) = 1 \textit{ or } V(w, \psi) = 1 \\
V(w, \varphi \supset \psi) = 1 & \Longleftrightarrow & \forall w_1, w_2 \in W.R(w, w_1, w_2).V(w_1, \varphi) = 1 \textit{ implies } V(w_2, \psi) = 1
\end{array}
$$

Since these frames have an initial world, then φ is *valid* **in** R_M iff $V(w_*, \varphi) = 1$. We say that φ is *valid* if φ is valid in all models.

Notice the *intentional* connective $\circ$ is omitted because in this model theory $\varphi \circ \psi$ is definable in terms of implication and negation that is $\varphi \circ \psi = \neg(\varphi \supset \neg\psi)$.

4.9.1 Logical Comparison to *CD*

Most of the differences between the logics occur in the treatment of negation. There are several acceptable principles about negation to be found among relevance logics: contraposition, reductio, involution of negation and DeMorgan laws. In *CD*, we have the latter two for the involutory negation $\neg$ and we have reductio for intuitionistic negation $\sim$. However, neither *CD* negation validates contraposition.

Let us examine each of these denied principles. In *CD* reductio is not valid. Below is the entire proof search tree. Since all rules have been analyzed and no proof found this is not valid.

Not Reductio: $(p \supset \neg p) \not\supset \neg p$

$$
\cfrac{\cfrac{p \supset \neg p \Rightarrow \neg p, p \qquad \neg p \Rightarrow \neg p}{p \supset \neg p \Rightarrow \neg p}}{\Rightarrow (p \supset \neg p) \supset \neg p}
$$

Secondly, contraposition, i.e. $\varphi \supset \psi \equiv \neg\psi \supset \neg\varphi$, is not valid. We show this by first looking at the valid formula $(p \wedge \sim p) \supset q$, and show that $\neg q \not\supset \neg(p \wedge \sim p)$. Remember $\sim p = p \supset 0$ and after DeMorgan application on the right we have $\neg q \Rightarrow \neg p \vee \neg(p \supset 0)$ which reduces to $\neg q \Rightarrow \neg p \vee p$. This would only be valid if the excluded middle were valid.

4.9.2 Model Theoretic Comparison to *CD*

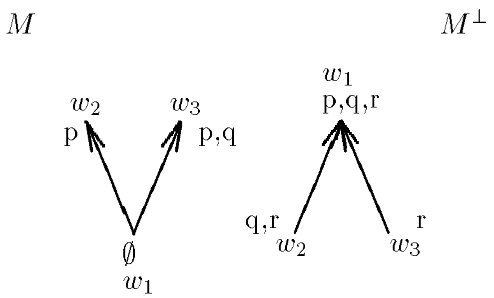
Model theoretically, $A \vee \neg A$ is valid in all models, hence should be valid axiomatically. This means that the relevant negation given by the models is non-constructive.

However, the focus of this section is to notice that negation is captured in relevance models via a $*$ operation on models of positive relevance logic. This $*$ operation gives rise to the model theory for relevance logic similarly to how applying the interval operation to models of positive intuitionistic logic yields a model for constructible falsity (Patterson 1996b), and generalized intervals on models for positive intuitionistic logic yields the model theory of N^-, and generalized intervals on models of positive intuitionistic logic with constants $\mathbf{0}$ and $\mathbf{1}$ yields the model theory for constructible duality presented herein.

We now examine how the $*$ operation can give rise to interval-models. If we started with a model for positive intuitionistic logic with constants $M = \langle W, R, V \rangle$ and added a $*$ function as any involutory order reversing function from W to W, then we would have a $*$-model $M_* = \langle W, R, V, * \rangle$. If validity in M_* were to judge formulas at all pairs (w, w^*) such that $w \neq w^*$ in M using Definition 85, then by examining all such pairs using all possible $*$ maps added to M is really just the same as looking at all non-trivial intervals on M.

This suggests a new semantics for *CD* because for a formula to be valid it needs to be valid in all models. However, adding each possible $*$ function to each positive model is the same as looking at all non-trivial intervals in all models. In this chapter we have shown that examining all non-trivial intervals over all models for positive intuitionistic logic with constants is sound and complete for *CD*.

Pictorially imagine that the $*$ function is a partial function from M to $M^{\perp}$ and back again which is order reversing and involutory.



This gives seven partial functions obeying involution and order-reversal,

1. the empty function,

2. $w_1^* = w_1$,

3. $w_2^* = w_2$,

4. $w_3^* = w_3$,

5. $w_1^* = w_2, w_2^* = w_1$,

6. $w_1^* = w_3, w_3^* = w_1$.

7. $w_2^* = w_3, w_3^* = w_2$

All such non-empty $*$ functions between M and $M^\perp$ are all pairs of worlds in M.

The similarities between the $*$ models and interval models are striking. Firstly, the $*$ operation for relevance logic is involutive and order reversing on the derived $\leq$ relation and truth is judged from w and falsity from w^*. For *CD*, the dualizing of the model M to $M^\perp$ is involutive and order reversing on R. However *CD* adds the restriction that we cannot judge truth and falsity from the same world otherwise the intuitionistic implication becomes classical.

Using this intuition we may modify the interval approach over models of positive propositional intuitionistic logic with constants to the $*$ approach to obtain models for constructible duality.

If $C = \langle W, R, V\rangle$ is a model of positive propositional intuitionistic logic with constants and $*$ obeys $\forall w \in W. w^{**} = w$ and $\forall w, w' \in W. wRw'$ implies $w'^* R w^*$ and the $*$ map is non-empty then $C = \langle W, R, V, *\rangle$ is a model of constructible duality where validity in a model is satisfaction of a formula at all pairs (w, w^*) such that $w \neq w^*$ as in Definition 85. Hence validity in all $*$ models corresponds to validity in all interval models of constructible duality because by examining all such $*$ models on C, we will eventually check all intervals on C. We have already shown that all intervals over C give a sound and complete semantics for *CD*.

The $*$ map could be suitably constrained to give $*$ models for N because in (Patterson 1996b) we consider how “proper intervals” in models of positive intuitionistic logic give a sound and complete semantics for N.

Chapter 5

Algebra

5.1 Introduction

We proposed an extension from N to constructible duality, and just showed a self-dual Kripke structure for which we show that constructible duality is sound and complete.

We now investigate the algebras of constructible duality and see that again it is intervals over Heyting algebras (or algebras for intuitionistic logic).

We show that the free algebra of constructible duality over n generators is the free algebra for intuitionistic logic over n generators times its opposite. Every point in this algebra is represented by a formula in the logic of constructible duality.

We go through the Beth-Fitting construction for the algebra of constructible duality and note its similarity to an instance of the Yoneda embedding and we define a symmetric version of Beth-Fitting construction which inter-translates the interval Kripke models and a self-dual algebra $H \times H^{op}$, where H is a Heyting algebra.

If we symmetrize the Beth construction and apply it to a interval Beth model, the symmetric Beth construction becomes an instantiation of a symmetric Yoneda embedding.

Finally, we show that this algebra is isomorphic to a $\mathbf{Chu}(H, 1)$ and is also isomorphic to a special subcategory of the G Dialectica category. This provides evidence that these models for constructible duality are models of linear logic with mix. We verify this theorem independently.

5.2 An Algebra for Constructible Duality

Rasiowa proposed a lattice structure for N in (Rasiowa 1974) and an equational definition of an algebra for N, together with a Stone-type duality theorem. However, the basic duality between positive and negative information is brought out and studied by (Vakarelov 1977). Our work can be seen as both simplifying and completing this work by adding the extra operators needed to capture all of $H \times H^{op}$.

Furthermore, we have an onto representation theorem which shows that this algebra is the Lindenbaum algebra for constructible duality.

Lemma 113 *Let $\mathcal{X}$ be a Heyting algebra and P a set of propositional atoms. Given any function $h\colon P \to \mathcal{X} \times \mathcal{X}^{op}$, we can extend it to all formula φ of constructible duality as a homomorphism as follows: Let $x, x', y, y' \in \mathcal{X}$.*

$$
\begin{array}{lllllllll}
Formulas & & & Units & & & BaseCase & & \\
h(\alpha \wedge \beta) & = & h(\alpha) \sqcap h(\beta) & h(\mathbf{1}) & = & (1,0) & (x,x') \sqcap (y,y') & = & (x \sqcap y, x' \sqcup y') \\
h(\alpha \vee \beta) & = & h(\alpha) \sqcup h(\beta) & h(\mathbf{0}) & = & (0,1) & (x,x') \sqcup (y,y') & = & (x \sqcup y, x' \sqcap y') \\
h(\alpha \supset \beta) & = & h(\alpha) \supset h(\beta) & h(\perp) & = & (0,0) & (x,x') \supset (y,y') & = & (x \supset y, x \sqcap y') \\
h(\neg\alpha) & = & \neg h(\alpha) & h(\top) & = & (1,1) & \neg(x,x') & = & (x', x)
\end{array}
$$

Definition 114 *Let $\mathcal{X}$ be a Heyting algebra. A formula φ is* valid *if and only if for all Heyting algebras $\mathcal{X}$ and for all $h\colon P \to \mathcal{X} \times \mathcal{X}^{op}$, $h(\varphi) = (1, -)$, where $-$ means any valuation.*

Theorem 115 Soundness *Every derivable sequent in constructible duality is valid.*

Proof: First we check the axioms:

$$
\begin{array}{ll}
(1)\Gamma, \varphi \Rightarrow \Delta, \varphi & h(\varphi) \sqcap h(\wedge\Gamma) \supset h(\varphi) \sqcup h(\vee\Delta) \\
& h(\varphi) \sqcap h(\wedge\Gamma) \supset h(\varphi) \supset h(\varphi) \sqcup h(\vee\Delta) \\
\\
(2)\mathbf{0}, \Gamma \Rightarrow \Delta & h(\mathbf{0}) \sqcap h(\wedge\Gamma) \supset h(\vee\Delta) \\
& (0,1) \sqcap h(\wedge\Gamma) \supset h(\vee\Delta) \\
& (0,-) \supset h(\vee\Delta) \\
& (1,-) \\
(3)\Gamma \Rightarrow \Delta, \mathbf{1} & h(\wedge\Gamma) \supset h(\vee\Delta) \sqcup h(\mathbf{1}) \\
& h(\wedge\Gamma) \supset h(\vee\Delta) \sqcup (1,0) \\
& h(\wedge\Gamma) \supset (1,-) \\
& (1,-) \\
(6)\Gamma \Rightarrow \Delta, \mathbf{I} & h(\wedge\Gamma) \supset h(\vee\Delta) \sqcup h(\mathbf{I}) \\
& h(\wedge\Gamma) \supset h(\vee\Delta) \sqcup (1,1) \\
& h(\wedge\Gamma) \supset (1,-) \\
& (1,-) \\
(7)\bot, \Gamma \Rightarrow \Delta & h(\bot) \sqcap h(\wedge\Gamma) \supset h(\vee\Delta) \\
& (0,0) \sqcap h(\wedge\Gamma) \supset h(\vee\Delta) \\
& (0,-) \supset h(\vee\Delta) \\
& (1,-)
\end{array}
$$

Axioms (4) and (5) are just as easy.

Now we check the rules:

$$\to\wedge \quad \frac{\Gamma \Rightarrow \Delta, \varphi \quad \Gamma \Rightarrow \Delta, \psi}{\Gamma \Rightarrow \Delta, \varphi \wedge \psi} \qquad \wedge\to \quad \frac{\varphi, \psi, \Gamma \Rightarrow \Delta}{\varphi \wedge \psi, \Gamma \Rightarrow \Delta}$$

$$\supset\to \quad \frac{\varphi \supset \psi, \Gamma \Rightarrow \Delta, \varphi \quad \psi, \Gamma \Rightarrow \Delta}{\varphi \supset \psi, \Gamma \Rightarrow \Delta} \qquad \to\supset \quad \frac{\varphi, \Gamma \Rightarrow \psi}{\Gamma \Rightarrow \Delta, \varphi \supset \psi}$$

$$\neg\wedge\to \quad \frac{\Gamma, \neg\varphi \Rightarrow \Delta \quad \Gamma, \neg\psi \Rightarrow \Delta}{\Gamma, \neg(\varphi \wedge \psi) \Rightarrow \Delta} \qquad \to\neg\wedge \quad \frac{\Gamma \Rightarrow \Delta, \neg\varphi, \neg\psi}{\Gamma \Rightarrow \Delta, \neg(\varphi \wedge \psi)}$$

$$\to\neg\supset \quad \frac{\Gamma \Rightarrow \Delta, \varphi \quad \Gamma \Rightarrow \Delta, \neg\psi}{\Gamma \Rightarrow \Delta, \neg(\varphi \supset \psi)} \qquad \neg\supset\to \quad \frac{\varphi, \neg\psi, \Gamma \Rightarrow \Delta}{\Gamma, \neg(\varphi \supset \psi) \Rightarrow \Delta}$$

$$\neg\neg\to \quad \frac{\varphi, \Gamma \Rightarrow \Delta}{\neg\neg\varphi, \Gamma \Rightarrow \Delta} \qquad \to\neg\neg \quad \frac{\Gamma \Rightarrow \Delta, \varphi}{\Gamma \Rightarrow \Delta, \neg\neg\varphi}$$

The rules for units $\neg\mathbf{I}\to, \to\neg\mathbf{I}, \neg\bot\to, \to\neg\bot$ are immediate from the definition of h on units.

The rules $\to\wedge, \wedge\to, \neg\wedge\to, \to\neg\wedge$ reflect the properties of lattice meet $\wedge$ and lattice join $\vee$ as well as the translation of h.

Let us examine one such rule $\neg\wedge\to$

$$\frac{\Gamma, \neg\varphi \Rightarrow \Delta \quad \Gamma, \neg\psi \Rightarrow \Delta}{\Gamma, \neg(\varphi \wedge \psi) \Rightarrow \Delta}$$

$h(\wedge\Gamma) \sqcap h(\neg\varphi) \supset h(\vee\Delta)$	and $h(\wedge\Gamma) \sqcap h(\neg\psi) \supset h(\vee\Delta)$
$h(\wedge\Gamma) \sqcap (h(\neg\varphi \sqcup h(\neg\psi)) \supset h(\vee\Delta))$	By Distributivity
$h(\wedge\Gamma) \sqcap h(\neg(\varphi \wedge \psi)) \supset h(\vee\Delta)$	By DeMorgan

The rules for implication are similarly easy:

$\supset\to$

$$\frac{\varphi \supset \psi, \Gamma \Rightarrow \Delta, \varphi \quad \psi, \Gamma \Rightarrow \Delta}{\varphi \supset \psi, \Gamma \Rightarrow \Delta}$$

$h(\wedge\Gamma) \sqcap h(\varphi \supset \psi) \supset h(\vee\Delta) \sqcup h(\varphi)$	and $h(\wedge\Gamma) \sqcap h(\psi) \supset h(\vee\Delta)$
$h(\wedge\Gamma) \sqcap h(\varphi \supset \psi) \supset h(\vee\Delta) \sqcup (h(\varphi) \sqcap h(\varphi \supset \psi)) \supset h(\vee\Delta) \sqcup h(\psi)$	
$h(\wedge\Gamma) \sqcap h(\varphi \supset \psi) \supset h(\vee\Delta) \sqcup (h(\psi) \sqcap h(\wedge\Gamma)) \supset h(\vee\Delta) \sqcap h(\vee\Delta)$	
$h(\wedge\Gamma) \sqcap h((\varphi \supset \psi)) \supset h(\vee\Delta)$	By Transitivity

$\rightarrow\supset$

$$\frac{\varphi, \Gamma \Rightarrow \psi}{\Gamma \Rightarrow \Delta, \varphi \supset \psi}$$

$$(h(\wedge\Gamma) \sqcap h(\varphi)) \supset h(\psi)$$
$$h(\wedge\Gamma) \supset h(\varphi) \supset h(\psi) \qquad \text{By ??}$$
$$h(\wedge\Gamma) \supset h(\vee\Delta) \sqcup h(\varphi \supset \psi)$$

$\rightarrow \neg \supset$

$$\frac{\Gamma \Rightarrow \Delta, \varphi \quad \Gamma \Rightarrow \Delta, \neg\psi}{\Gamma \Rightarrow \Delta, \neg(\varphi \supset \psi)}$$

$h(\wedge\Gamma) \supset (h(\vee\Delta) \sqcup h(\varphi))$	and $h(\wedge\Gamma) \supset (h(\vee\Delta) \sqcup h(\neg\psi))$
$h(\wedge\Gamma) \supset (h(\vee\Delta) \sqcup (h(\varphi)) \sqcap (\neg\psi))$	By Distributivity
$h(\wedge\Gamma) \supset (h(\vee\Delta) \sqcup ((\pi_1(h(\varphi)), \pi_2(h(\varphi))) \sqcap (\neg(\pi_1(h(\psi)), \pi_2(h(\psi))))))$	Expanding h
$h(\wedge\Gamma) \supset (h(\vee\Delta) \sqcup ((\pi_1(h(\varphi)), \pi_2(h(\varphi))) \sqcap (\pi_2(\psi), \pi_1(\psi))))$	Definition for $\neg$
$h(\wedge\Gamma) \supset (h(\vee\Delta) \sqcup (\pi_1(h(\varphi)) \wedge \pi_2(h(\psi)), \pi_2(h(\varphi)) \vee \pi_1(h(\psi))))$	Base Definition $\sqcap$
	$\supset$
$h(\wedge\Gamma) \supset (h(\vee\Delta) \sqcup (\pi_1(h(\varphi)) \wedge \pi_2(h(\psi)), \pi_1(h(\varphi)) \supset \pi_1(h(\psi))))$	Material Implies $\supset$
$h(\wedge\Gamma) \supset h(\vee\Delta) \sqcup h(\neg(\varphi \supset \psi))$By Transitivity	

$\neg \supset\rightarrow$

$$\frac{\varphi, \neg\psi, \Gamma \Rightarrow \Delta}{\Gamma, \neg(\varphi \supset \psi) \Rightarrow \Delta}$$

$$h(\varphi) \sqcap h(\neg\psi) \sqcap h(\wedge\Gamma) \supset h(\vee\Delta)$$
$$(\pi_1(h(\varphi)), \pi_2(h(\varphi))) \sqcap (\neg(\pi_1(h(\psi)), \pi_2(h(\psi)))) \sqcap h(\wedge\Gamma) \supset h(\vee\Delta) \quad \text{Expanding } h$$
$$(\pi_1(h(\varphi)) \wedge \pi_2(h(\psi)), \pi_2(h(\varphi)) \wedge \pi_1(h(\psi))) \sqcap h(\wedge\Gamma) \supset h(\vee\Delta) \qquad \text{Expanding } \sqcap \text{ and } \neg$$
$$h(\neg(\varphi \supset \psi)) \sqcap h(\wedge\Gamma) \supset h(\vee\Delta)$$

The rules $\neg\neg\rightarrow$ and $\rightarrow\neg\neg$ make $\neg\neg\varphi = \varphi$ are immediate from the base case $h(\neg(a, b)) = (b, a)$.

∎

Theorem 116 (Onto Representation) *Let $H_f \times H_f^{op}$ be the free Heyting algebra over n propositional atoms, then every (h, h') in $H_f \times H_f^{op}$ is representable by some formula of constructible duality.*

Proof: Let $\langle h, b\rangle$ and $\langle c, h'\rangle$ be any two pairs in $H_f \times H_f^{op}$. We construct $\langle h, h'\rangle$ via $((\langle h, b\rangle \wedge \langle 1, 1\rangle) \vee ((\langle c, h'\rangle \wedge \langle 0, 0\rangle)$ — this expands to $\langle h, h'\rangle = \langle h, 1\rangle \vee \langle 0, h'\rangle$.

Now we use the onto representation theorem from intuitionistic logic to Heyting algebras (Birkhoff 1933) and the onto representation theorem from Brouwerian logic to dual Heyting algebras (Birkhoff 1933) to conclude that with the simple construction above every pair of elements in $H_f \times H_f^{op}$ is constructible. ∎

In the next chapter, we obtain completeness by using the free algebra. Here to show the similarly of the technique from Chapter 2., we translate our interval Kripke models into our pair algebra.

Lemma 117 Symmetric Beth-Fitting *Let $K = \langle W, R, V\rangle$ be a Kripke structure for constructible duality. If h is any function $h\colon P \to Fil(\langle W, R\rangle) \times Idl(\langle W, R\rangle)$ such that $h(p) = \{w \mid p \in V(w)\} \times \{w \mid p \notin V(w)\}$, then h extends to all formulas φ such that*
$h(\varphi) = \{w_p \mid \exists w'_n.(w_p, w'_n) \models_K \varphi\} \times \{w_n \mid \exists w'_p.(w_n, w'_p) \| {\models_{K^\perp}} \varphi\}$

Proof: By induction on the structure of φ. Throughout this proof we will drop unnecessary mention of $w_p \neq w_n$.

- Units **1 Truth:** $h(\mathbf{1}) = (1, 0) \iff \forall w_p, w_n \in W.(w_p, w_n) \models_M \mathbf{1}$. This is so by definition.

  Units **1 Falsity:** $h(\neg\mathbf{1}) = (1, 0) \iff \forall w_p, w_n \in W.(w_n, w_p) \| {\models_{M^\perp}} \mathbf{1}$. Expanding the definitions we get, $h(\mathbf{1}) = (0, 1) \iff \forall w_p, w_n \in W.(w_n, w_p) \| {\not\models_{M^\perp}} \mathbf{1}$. This is so by definition.

- Units **0 Truth:** $h(\mathbf{0}) \neq (1, 0) \iff \forall w_p, w_n \in W.(w_p, w_n) \not\models_M \mathbf{0}$. This is so by definition.

  Units **0 Falsity:** $h(\neg\mathbf{0}) = (1, 0) \iff \forall w_p, w_n \in W.(w_n, w_p) \| {\models_{M^\perp}} \mathbf{0}$. Expanding the definitions we get, $h(\mathbf{0}) = (0, 1) \iff \forall w_p, w_n \in W.(w_n, w_p) \| {\models_{M^\perp}} \mathbf{0}$. This is so by definition.

- Units **I Truth:** $h(\mathbf{I}) = (1, -) \iff \forall w_p, w_n \in W.(w_p, w_n) \models_M \mathbf{I}$. This is so by definition.

Units **I Falsity:** $h(\neg\mathbf{I}) = (1, -) \iff \forall w_p, w_n \in W.(w_n, w_p) \Vdash_{M^\perp} \mathbf{I}$. Expanding the definitions we get, $h(\mathbf{I}) = (-, 1) \iff \forall w_p, w_n \in W.(w_n, w_p) \Vdash_{M^\perp} \mathbf{I}$. This is so by definition.

- Units $\perp$ **Truth:** $h(\perp) \neq (1, -) \iff \forall w_p, w_n \in W.(w_p, w_n) \not\models_M \perp$. This is so by definition.

  Units $\perp$ **Falsity:** $h(\neg\mathbf{0}) \neq (1, -) \iff \forall w_p, w_n \in W.(w_n, w_p) \not\Vdash_{M^\perp} \perp$. Expanding the definitions we get, $h(\perp) \neq (-, 1) \iff \forall w_p, w_n \in W.(w_n, w_p) \not\Vdash_{M^\perp} \perp$. This is so by definition.

- Basis p **Truth**: $h(p) = (1, -) \iff \forall w_p, w_n \in W.(w_p, w_n) \models_M p$. Expanding the definitions we get, $\{w \mid p \in V(w)\} = W \iff \forall w \in W.p \in V(w)$.

  Basis p **Falsity**: $h(\neg p) = (1, -) \iff \forall w_p, w_n \in W.(w_n, w_p) \Vdash_{M^\perp} p$. Expanding the definitions we get, $h(p) = (-, 1) \iff \forall w \in W.p \notin V(w)$. Expanding the left hand side we obtain $\{w \mid p \notin V(w)\} = W \iff \forall w \in W.p \notin V(w)$.

- $\varphi \wedge \psi$ **Truth**: $h(\varphi \wedge \psi) = (1, -) \iff \forall w_p, w_n \in W.\ (w_p, w_n) \models_M \varphi \wedge \psi$. Expanding on the left we get $(\pi_1(h(\varphi)), \pi_2(h(\varphi))) \sqcap (\pi_1(h(\psi)), \pi_2(h(\psi)) = (1, -) \iff \forall w_p, w_n \in W.\ (w_p, w_n) \models_M \varphi \wedge \psi$. Expanding again on the left by $\sqcap$ and on the right by $\wedge$ we get, $\pi_1(h(\varphi)) \wedge \pi_1(h(\psi)) = 1 \iff \forall w_p, w_n \in W.\ (w_p, w_n) \models_M \varphi$ and $(w_p, w_n) \models_M \psi$. Expanding on the left by the definition of π_1 and h, we get that $h(\varphi) = (1, -)$ and $h(\psi) = (1, -) \iff \forall w_p, w_n \in W.\ (w_p, w_n) \models_M \varphi$ and $(w_p, w_n) \models_M \psi$. This is true by the induction hypotheses.

  $\varphi \wedge \psi$ **Falsity**: $h(\neg(\varphi \wedge \psi)) = (1, -) \iff \forall w_p, w_n \in W.\ (w_n, w_p) \Vdash_{M^\perp} \varphi \wedge \psi$. Expanding on the left we have $h(\varphi \wedge \psi) = (-, 1) \iff \forall w_p, w_n \in W.\ (w_n, w_p) \Vdash_{M^\perp} \varphi \wedge \psi$. Expanding again on the left we get $(\pi_1(h(\varphi)), \pi_2(h(\varphi))) \sqcap (\pi_1(h(\psi)), \pi_2(h(\psi)) = (-, 1) \iff \forall w_p, w_n \in W.\ (w_n, w_p) \Vdash_{M^\perp} \varphi \wedge \psi$. Expanding on the left by $\sqcap$ and on the right by $\wedge$ we get, $\pi_2(h(\varphi)) \vee \pi_2(h(\psi)) = 1 \iff \forall w_p, w_n \in W.\ (w_n, w_p) \Vdash_{M^\perp} \varphi$ or $(w_n, w_p) \Vdash_{M^\perp} \psi$. Expanding on the left by the definition of π_2 and h, we get that $h(\varphi) = (-, 1)$ or $h(\psi) = (-, 1) \iff \forall w_p, w_n \in W.\ (w_n, w_p) \Vdash_{M^\perp} \varphi$ or $(w_n, w_p) \Vdash_{M^\perp} \psi$. This is true by the induction hypotheses.

- $\varphi \vee \psi$ **Truth**: This is similar to the falsity of $\wedge$.

  $\varphi \vee \psi$ **Falsity**: This is similar to the truth of $\wedge$.

- $\varphi \supset \psi$ **Truth**: $h(\varphi \supset \psi) = (1, -) \iff \forall w_p, w_n \in W.\ (w_p, w_n) \models_M \varphi \supset \psi$. Expanding on the left we get $(\pi_1(h(\varphi)), \pi_2(h(\varphi))) \supset (\pi_1(h(\psi)), \pi_2(h(\psi)) = (1, -) \iff \forall w_p, w_n \in W.\ (w_p, w_n) \models_M \varphi \supset \psi$. Expanding again on the left by $\supset$ we get, $\pi_1(h(\varphi)) \supset \pi_1(h(\psi)) = 1 \iff \forall w_p, w_n \in W.\ (w_p, w_n) \models_M \varphi \supset \psi$. We can expand each side once more and then apply the induction hypotheses.

  $\varphi \supset \psi$ **Falsity**: $h(\neg(\varphi \supset \psi)) = (1, -) \iff \forall w_p, w_n \in W.\ (w_n, w_p) \|\!\models_{M^\perp} \varphi \supset \psi$. Expanding on the left we have $h(\varphi \supset \psi) = (-, 1) \iff \forall w_p, w_n \in W.\ (w_n, w_p) \|\!\models_{M^\perp} \varphi \supset \psi$. Expanding again on the left we get $(\pi_1(h(\varphi)), \pi_2(h(\varphi))) \supset (\pi_1(h(\psi)), \pi_2(h(\psi)) = (-, 1) \iff \forall w_p, w_n \in W.\ (w_n, w_p) \|\!\models_{M^\perp} \varphi \supset \psi$. Expanding on the left by $\supset$ and on the right by $\supset$ we get, $\pi_1(h(\varphi)) \wedge \pi_2(h(\psi)) = 1 \iff \forall w_p, w_n \in W.\ (w_p, w_n) \models_M \varphi$ and $(w_n, w_p) \|\!\models_{M^\perp} \psi$. Expanding on the left by h and π get, $h(\varphi) = (1, -)$ and $h(\psi) = (-, 1) \iff \forall w_p, w_n \in W.\ (w_p, w_n) \models_M \varphi$ and $(w_n, w_p) \|\!\models_{M^\perp} \psi$. This is true by the induction hypotheses.

- $\neg\varphi$ **Truth**: $h(\neg\varphi) = (1, -) \iff \forall w_p, w_n \in W.\ (w_p, w_n) \models_M \neg\varphi$. Expanding both of the sides of the formula we have $h(\varphi) = (-, 1) \iff \forall w_p, w_n \in W.\ (w_n, w_p) \|\!\models_{M^\perp} \varphi$. This is just the induction hypothesis for falsity.

  $\neg\varphi$ **Falsity**: $h(\neg\neg\varphi) = (1, -) \iff \forall w_p, w_n \in W.\ (w_n, w_p) \|\!\models_{M^\perp} \neg\varphi$. Expanding $h(\neg\varphi) = (-, 1) \iff \forall w_p, w_n \in W.\ (w_p, w_n) \models_M \varphi$. However this rewrites to $h(\varphi) = (1, -) \iff \forall w_p, w_n \in W.\ (w_p, w_n) \models_M \varphi$. However, this is the induction hypothesis for truth.

∎

Lemma 118 *For any Kripke structure K, $Idl(K)$ is isomorphic to $Fil(K)^{op}$.*

The backwards construction mapping all proper *prime* filter-filter pairs over $H \times H^{op}$ to intervals in K, or $K \times K^\perp$, takes prime filter-ideal pairs in H to intervals in K preserving all formulas.

Corollary 119 Completeness *A formula is derivable in constructible duality* if and only if *it is algebraically valid.*

If $K_B = \langle W, R, V\rangle$ is a **Beth** model for intuitionistic logic, then intervals may again be used for a **Beth** model of constructible duality. The forward construction taking K_B to $Fil(\langle W, R\rangle) \times Idl(\langle W, R\rangle)$ again makes an algebra. The backwards construction takes *every* filter-ideal pair to be an interval in a Beth model for constructible duality. This construction provides an instantiation of a symmetric Yoneda embedding.

If $\mathcal{C}$ is a Cartesian closed category for which intuitionistic logic is fully-complete, then $\mathcal{C}\times\mathcal{C}^{op}$ is fully-complete for constructible duality.

5.3 Restriction to an Algebra for N

Rasiowa proposed a lattice structure for N in (Rasiowa 1974) and an equational definition of an algebra for N, together with a Stone-type duality theorem. However, the basic duality between positive and negative information is brought out and studied by (Vakarelov 1977). We present a slight modification here, studying $H \times H^{op}$ instead of (Vakarelov 1977) who used a restriction on $H\times H$. These H^2 spaces were generalized by (Sendlewski 1984, Sendlewski 1989, Sendlewski 1995) by studying boolean congruences, but here we start with $H\times H^{op}$ the algebra of constructible duality and get rid of boolean congruences to get an algebra of N.

Definition 120 *Let $\mathcal{X}$ be a Heyting algebra and P a set of propositional atoms. Given any function $h: P \to \mathcal{X}\times\mathcal{X}^{op}$ such that for all $p \in P$ $h(p)$ maps to disjoint pairs in $\mathcal{X}$, we extend it to all formula φ of constructible falsity as follows: Let $(x, x'), (y, y') \in rng(h)$.*

$$\begin{array}{lllll}
Formulas & & & & BaseCase \\
h(\alpha\wedge\beta) & = & h(\alpha)\sqcap h(\beta) & (x,x')\sqcap(y,y') & = & (x\sqcap y, x'\sqcup y') \\
h(\alpha\vee\beta) & = & h(\alpha)\sqcup h(\beta) & (x,x')\sqcup(y,y') & = & (x\sqcup y, x'\sqcap y') \\
h(\alpha\supset\beta) & = & h(\alpha)\supset h(\beta) & (x,x')\supset(y,y') & = & (x\supset y, x\sqcap y') \\
h(\neg\alpha) & = & \neg h(\alpha) & \neg(x,x') & = & (x', x)
\end{array}$$

Definition 121 *Let $\mathcal{X}$ be a Heyting algebra. A formula φ is* valid *if and only if for all Heyting algebras $\mathcal{X}$ and for all $h: P \to \mathcal{X}\times\mathcal{X}^{op}$ such that for all $p \in P$ $h(p)$ maps to disjoint pairs in $\mathcal{X}$, $h(\varphi) = (1, -)$, where $-$ means any valuation.*

Theorem 122 Soundness *Every derivable sequent in constructible falsity is valid.*

Again for completeness we could use the canonical evaluation map on the free algebra for constructible falsity over P. Whenever H_f is a free Heyting algebra over P, then disjoint pairs of $H_f \times H_f^{op}$ is the free algebra for constructible falsity. To show the similarly of the previous technique, we translate our interval Kripke models into our pair algebra.

Lemma 123 Symmetric Beth-Fitting *Let* $K = \langle W, R, V\rangle$ *be a Kripke structure for constructible falsity. If* h *is any function* $h\colon P \to Fil(\langle W, R\rangle) \times Idl(\langle W, R\rangle)$ *such that* $h(p) = \{w \mid p \in V(w)\} \times \{w \mid p \notin V(w)\}$*, then* h *extends to all formulas* φ *such that*
$h(\varphi) = \{w_p \mid \exists w'_n.(w_p, w'_n) \models_K \varphi\} \times \{w_n \mid \exists w'_p.(w_n, w'_p) \models_{K^\perp} \varphi\}$

The backwards construction mapping all proper *prime* filter-ideal pairs over $H \times H^{op}$ to intervals in K, or $K \times K^\perp$, takes prime filter-ideal pairs in H to intervals in K preserving all formulas.

Corollary 124 Completeness *A formula is derivable in constructible falsity* if and only if *it is algebraically valid.*

If $K_B = \langle W, R, V\rangle$ is a **Beth** model for intuitionistic logic, then intervals may again be used for a **Beth** model of constructible falsity. The forward construction taking K_B to $Fil(\langle W, R\rangle) \times Idl(\langle W, R\rangle)$ again makes an algebra. The backwards construction takes *every* filter-ideal pair to be an interval in a Beth model for constructible falsity. This construction provides an instantiation of a symmetric Yoneda embedding.

5.4 Significance and Relevance

Duality and symmetry are prevalent in the study of linear logic and concurrency(de Paiva 1989b, de Paiva and Hyland 1993, Gupta 1994, Pratt 1995). The duality in linear logic arises from the presence of a constructive involutory negation and the symmetric nature of proofs (Girard 1987). The μ calculus(Kozen 1983) is another natural dual logic where its negation acts as a dualizing operator, turning least fixed points to greatest fixed points. The μ-calculus is not constructive but has many of the features present in constructive duality.

Least, optimal and greatest fixed point constructions are central to computer science. The partial computable functions are naturally those defined via least fixed points on recursive operators. (Patterson and Costello 1997c) have recently proposed using combinations of computable greatest and least fixed points as a programming constructs. Pitts (1994), Milner (1991), Fiore (1996) and others have argued for the importance of corecursion, or greatest fixed points. Pitts, in (1994) stresses the need for mixing recursion and corecursion, and suggests that $C^{op} \times C$ is an important candidate for these investigations. Our algebra is the posetal case of his suggestion.

We show that the underlying structure, present in these formalisms has $H \times H^{op}$ as a simple case; hence constructible duality as a part of its reasoning machinery. We now study its relationship to linear logic formally.

Chapter 6

Relationship to Linear Logic

6.1 Summary

An exponential is a unary operator, reminiscent of a modality. We suggest that exponentials be regarded as logic transformers. Exponentials in linear logic give rise to the ! translation which details which syntactic subset of linear logic which is theoremwise equivalent to intuitionistic logic. Therefore, sometimes logics may appear as certain syntactic subsets of other logics.

In constructible duality exponential is both a *projection* (giving rise to a translation) and an *interior* operation (similar to a modality).

We show in this chapter that the algebra of constructible duality provides a model for linear logic and in that model ! is an exponential (as defined by linear logicians) a projection operator and an interior operator.

Another exponential ¡ is definable in constructible duality which has no linear logic counterpart. This is also a projection and an interior operator. Closure and interior operations can be thought of as approximations.

We begin by a light discussion about the behavior of our exponentials and conclude with the precise relationship between these logics.

6.2 Background

The most famous example of a logic transformer is the double negation translation of Gödel from intuitionistic logic to classical logic. Gödel showed that prefixing every subformula with

$\sim\sim$ embedded classical logic into intuitionistic logic. In the propositional case we can see this as *Glivenko:* A propositional formula φ is valid classically iff $\sim\sim\varphi$ is valid intuitionistically.

In this chapter, we are most concerned with logic transformers called exponentials. These share certain topological properties with other logic transformers, such as the Gödel translation above.

Before we consider the general properties that these logic transformers have, we consider the logics that we shall translate between. These are classical and intuitionistic logic, constructible duality and linear logic (Girard 1987).

We review the algebras of the first four of these logics because all of the logic transformers that we will study have simple algebraic characterizations. We then concentrate on the transformation given to us by the exponential operators. We conclude by describing all of the logic transformers topologically.

6.3 Logics and their Algebras

We present some logics and their algebras, so that we can later show how certain logic transformers can map one logic into another.

Another very useful view of a Heyting algebra is as a topology.

Definition 125 *A* topological space $\mathcal{T}_{X_{\mathcal{O}}}$ *is a pair* $(X, \mathcal{O})$ *consisting of a set* X *together with a set of subsets of* X *which are* open *sets.* $\mathcal{O}$ *is a system of* open *sets when any union of open sets is open, the intersection of two open sets is open, and* $\emptyset$ *and* X *are open.*

The interior, *Int*, of a subset Y of X is the largest open set contained in Y. The closure of a subset Y is the smallest set that contains Y and is the complement of an open set. We say an open set in $\mathcal{O}$ is *regular* in case it coincides with the interior of its own closure. We write the complement of a set Q as Q^{-} and the closure as Q'. We write $Q^{-\prime}$ as $Q^{\perp}$, thus regular open sets obey $Q = Q^{\perp\perp}$.

Proposition 126 *The open sets $\mathcal{O}$ of a non-empty topological space $\mathcal{T}_{X_{\mathcal{O}}}$ form a Heyting algebra with respect to the Heyting operations defined by*

Heyting		*Topology*	*Heyting*		*Topology*
0	$=$	$\emptyset$	1	$=$	X
$P \sqcap Q$	$=$	$P \cap Q$	$P \sqcup Q$	$=$	$P \cup Q$
$-P$	$=$	$\mathrm{Int}(P^{-})$	$P \supset Q$	$=$	$\mathrm{Int}(P^{-} \cup Q)$

6.3.1 Classical Logic

Classical propositional logic can be axiomatized by adding the axiom scheme $\sim\sim A \supset B$ to the intuitionistic rules. The algebra of classical logic are Boolean algebras. The most normal example of a Boolean algebra is the set of all subsets of a given set. We will find it more useful to consider the following algebra.

Proposition 127 *The regular open sets of a non-empty topological space $\mathcal{T}_{X_{\mathcal{O}}}$ form a Boolean algebra with respect to the distinguished Boolean elements and operations defined by*

Boolean		*Topology*	*Boolean*		*Topology*
$\mathbf{0}$	$=$	$\emptyset$	$\mathbf{1}$	$=$	X
$P \wedge Q$	$=$	$P \cap Q$	$P \vee Q$	$=$	$(P \cup Q)^{\perp\perp}$
$\neg P$	$=$	$P^{\perp}$			

Therefore, the double negation $\sim\sim$ translation of Gödel can be seen as collapsing a topology onto its regular open sets. Therefore, this Gödel translation has a nice characterization in terms of the algebras. We will study the translation that arose because of the exponentials, but first we relate the discovery of these exponentials.

6.4 The Story of the Exponentials

I'll call a logic *algebraically friendly* when the free algebras of the logics are just the term algebras. That is make the points of the algebra the equivalence classes (written $[]$) of interderivable formulas i.e. $B \in [A]$ in the algebra if and only if $A \equiv B$ in the logic. Furthermore, the ordering among the points is derived from the implication in the logic, that is $[A] \leq [B]$ just when

$A \supset B$. Both intuitionistic and classical logic are algebraically friendly. However, N and CD are not.

That is, interderivability cannot be used to create these equivalence classes to construct an algebra because their implications does not form a congruence. This means that two formulas can be equivalent in one context and not in another.

We consider an example of this phenomena common to both logics: $\neg(A \supset B) \equiv A \wedge \neg B$ where $A \equiv B$ when $A \supset B$ and $B \supset A$.

This equivalence holds but the negation of each of these formulas is not equivalent. $\neg\neg(A \supset B) \not\equiv \neg(A \wedge \neg B)$ because using the involution of negation on the left and the DeMorgan equivalence on the right we get the following inequality $A \supset B \not\equiv \neg A \vee B$. If this were an equivalence, this logic would be classical.

This causes difficulty in forming an algebra. To ameliorate this problem of a *substitutable implication* for N, $A \multimap B = A \supset B \ \wedge \ \neg B \supset \neg A$ was introduced by (Rasiowa 1974, Vakarelov 1977) where it was denoted $\Rightarrow$ which made N algebraically friendly under this new implication. This implication is substitutable and forms a congruence on equivalence class of interderivable formulae. That is, $[A]$ is the equivalence class of formulas B such that $A \multimapboth B$ and if C is derivable and contains an occurrence of A then $C[B/A]$ is derivable (where $C[B/A]$ denotes replacing every occurrence of A by B and $A \multimapboth B = A \multimap B \wedge B \multimap A$).

This is where the story digresses from previous studies. If the "ordering" on the points of an algebra is $\multimap$, what is (if any) the conjunction with respect to which we may have a deduction theorem in the algebraic setting?

We now introduce a *new*, but definable, operation, tensor,

$$A \otimes B \text{ defined as } \neg(A \multimap \neg B).$$

This new operator obeys the rule,

$$(A \otimes B) \multimap C = A \multimap (B \multimap C).$$

Thus it acts much like a conjunction for $\multimap$ and gives us our deduction rule. It turns out that in linear logic $A \otimes B = (A \multimap B^{\perp})^{\perp}$, thus our $\neg$ is acting like $^{\perp}$.

This new operator $\otimes$ acts as the conjunction for the overall ordering of the algebra. Even though $\otimes$ is definable, this operator and its interaction with $\multimap$ has not been studied. In this thesis, $\otimes$ will be of pivotal importance in our introduction of exponentials to *CD*.

First however let us define the DeMorgan dual of $\otimes$ which we denote $\oplus$. Since, we have an involutory negation $\otimes$ must have a DeMorgan dual. This DeMorgan dual defines a disjunction associated with $\multimap$. It is called or co-tensor (written $⅋$ in linear logic) as $A \oplus B = \neg(\neg A \otimes \neg B) = \neg A \multimap B$, as the De Morgan dual of $\otimes$. In linear logic $A ⅋ B = A^{\perp} \multimap B$, thus again our $\neg$ is acting like $^{\perp}$.

6.4.1 Adding *Overdefined*

Now that we have discovered our conjunction $\otimes$ with respect to which the algebraic ordering $\multimap$ is the residual, it is natural to ask whether $\otimes$ has a definable unit?
It does not.

To get the unit for $\otimes$, we added a new constant $\mathbf{I}$, in *CD*. This unit is distinct from the other constants, falsity, $\mathbf{0}$, and truth, $\mathbf{1}$. $\mathbf{I}$ is not definable in N or I_c, nor I. We can axiomatize this unit as $A \supset \mathbf{I}$, $\neg\mathbf{I} \equiv \mathbf{I}$ where A is any formula in *CD*.

The addition of $\mathbf{I}$ demands that we drop the consistency axiom $A \wedge \neg A \supset B$ because we have an object which is equal to its own negation but does not imply everything. Furthermore, this unit has an interpretation as maximally overdefined — both "true" and "false". The negation of overdefined, $\mathbf{I}$, is itself, and every formula implies it. Thus, we could not add $\mathbf{I}$ to a logic containing the axiom $A \wedge \neg A \supset B$ without trivializing the logic.

Remember the sequent rules for this new constant were:

Axiom

$$(6)\Gamma \Rightarrow \Delta, \mathbf{I}$$

$$\neg\mathbf{I} \Rightarrow \quad \frac{\mathbf{I}, \Gamma \Rightarrow \Delta}{\neg\mathbf{I}, \Gamma \Rightarrow \Delta} \qquad \Rightarrow \neg\mathbf{I} \quad \frac{\Gamma \Rightarrow \Delta, \mathbf{I}}{\Gamma \Rightarrow \Delta, \neg\mathbf{I}}$$

6.4.2 Adding *Undefined*

We also added the constant *undefined* written $\perp$ in (Vakarelov 1977) to *CD*.

This constant captures the notion of a proposition that is neither true nor false, i.e. it is *undefined*. As such, $\perp$ implies every formula and like $\mathbf{I}$ it is its own negation. However, it is distinct from $\mathbf{I}$, and the constants $\mathbf{1}$ and $\mathbf{0}$. Remember the sequent rules for this new constant were:

Axiom

$$(7)\Gamma, \perp \Rightarrow \Delta$$

$$\neg\perp \Rightarrow \quad \frac{\perp, \Gamma \Rightarrow \Delta}{\neg\perp, \Gamma \Rightarrow \Delta} \quad \Rightarrow \neg\perp \quad \frac{\Gamma \Rightarrow \Delta, \perp}{\Gamma \Rightarrow \Delta, \neg\perp}$$

6.4.3 Completing the Algebra $H \times H^{op}$

The addition of the two constants $\perp$ and $\mathbf{I}$ complete *CD*, in the sense that there is an onto representation theorem for *CD* with these additional constants with respect to $H \times H^{op}$ where H is a Heyting algebra. This naturally completes *CD* so that every point in the algebra $H \times H^{op}$ is representable as an equivalence class of interderivable formulas .

We remind the reader of this definition and theorem from the previous chapter, but here we show the new definable connectives and their algebraic representation.

Lemma 128 Representation *Let $\mathcal{X}$ be a Heyting algebra and P a set of propositional atoms. Given any function $h: P \to \mathcal{X} \times \mathcal{X}^{op}$, we extend it to all formula φ of constructible duality as follows: Let $x, x', y, y' \in \mathcal{X}$.*

$Formulas$			$Units$			$BaseCase$		
$h(\alpha \wedge \beta)$	$=$	$h(\alpha) \sqcap h(\beta)$	$h(\mathbf{1})$	$=$	$(1, 0)$	$(x, x') \sqcap (y, y')$	$=$	$(x \sqcap y, x' \sqcup y')$
$h(\alpha \vee \beta)$	$=$	$h(\alpha) \sqcup h(\beta)$	$h(\mathbf{0})$	$=$	$(0, 1)$	$(x, x') \sqcup (y, y')$	$=$	$(x \sqcup y, x' \sqcap y')$
$h(\alpha \supset \beta)$	$=$	$h(\alpha) \supset h(\beta)$	$h(\perp)$	$=$	$(0, 0)$	$(x, x') \supset (y, y')$	$=$	$(x \supset y, x \sqcap y')$
$h(\neg\alpha)$	$=$	$\neg h(\alpha)$	$h(\mathbf{I})$	$=$	$(1, 1)$	$\neg(x, x')$	$=$	(x', x)
						$(x, x') \multimap (y, y')$	$=$	$(x \supset y \wedge y' \supset x', x \wedge y')$
						$(x, x') \otimes (y, y')$	$=$	$(x \wedge y, x \supset y' \wedge y \supset x')$
						$(x, x') \oplus (y, y')$	$=$	$(x' \supset y \wedge y' \supset x, x' \wedge y')$

To review:

$A \multimap B = \neg(A \otimes \neg B) = \neg A \oplus B.$

Theorem 129 Soundness and completeness *A formula is derivable in constructible duality* if and only if *it is algebraically valid.*

Now we can give a more standard completeness proof algebraically.

Theorem 130 (Algebraic Completeness) *Every non-valid formula in the language of constructible duality with the set P of propositional atoms is refuted in $H_f \times H_f^{op}$ where H_f is the free Heyting algebra over P.*

Outline of Proof: Form the Lindenbaum algebra over P under $A \multimap\!\!\circ B$ as the equivalence class generator. This generates the free algebra from equivalence classes of interderivable formulas in constructible duality.

Separate $A \multimap B$ into its constituent parts i.e. $A \supset B$ and $\neg B \supset \neg A$. This must generate the free algebra for intuitionistic logic and the free algebra for Brouwerian logic (dual intuitionistic logic) by Theorem 116.

Then apply the Lindenbaum construction on each half which one half constructs from the free Heyting algebra a Kripke model isomorphic to $\mathcal{K}$ our canonical Kripke model and on the other half a Kripke model isomorphic to $\mathcal{K}^{\perp}$.

At this point we may use the previous model theoretic completeness theorem. ∎

This obviously gives us a standard completeness result which says that every non-valid formula is refuted in some algebra for constructible duality.

We now have a algebra for *CD* with our additional units. This algebra bears a strong resemblance to the algebra of intuitionistic logic, in fact, it is two copies of a Heyting algebra, one order reversed, producted together. Thus it clearly contains two copies of intuitionistic logic (or a copy of intuitionistic logic and a copy of Brouwerian logic (Stone 1937)).

We now show how these two copies can be extracted from constructible duality. Then we consider Girard's linear logic and in particular we discuss its exponentials which will be the tools that carry out our promised decomposition of *CD*. These exponentials allow us to provide a one-line proof that the representation lemma is onto.

6.5 A New Exponential

We now study the first exponential. It is an operator satisfying all of the rules of an exponential in linear logic. The definitions and lemmas needed to prove these statements appear in the next section. Here we study some of its properties.

This new exponential, !, is defined as $!A = A \wedge \mathbf{I}$. This operation removes the negative content from A. This is done by making the falsity of A true. Therefore, everything implies $\neg !A$. We can see this as $\vdash \phi \supset \neg !A$.

$\vdash \phi \supset \neg !A$	We get
$\vdash \phi \supset \neg(A \wedge \mathbf{I})$	from expanding the definition
$\vdash \phi \supset \neg A \vee \neg \mathbf{I}$	and this using deMorgan. Finally
$\vdash \phi \supset \neg A \vee \mathbf{I}$	follows from the
$\vdash \phi \supset \mathbf{I}$	axiom.

Intuitionistic implication is definable as $!A \multimap B$ using this new exponential and substitutable implication.

$$
\begin{aligned}
A \supset B &= \ !A \multimap B \\
A \supset B &= A \wedge \mathbf{I} \multimap B \\
A \supset B &= ((A \wedge \mathbf{I}) \supset B) \wedge \ \neg B \supset \neg(A \wedge \mathbf{I}) \\
A \supset B &= ((A \wedge \mathbf{I}) \supset B) \wedge \ \neg B \supset (\neg A \vee \neg \mathbf{I}) \\
A \supset B &= ((A \wedge \mathbf{I}) \supset B) \wedge \ \neg B \supset (\neg A \vee \mathbf{I}) \\
A \supset B &= ((A \wedge \mathbf{I}) \supset B) \qquad \mathbf{I} \text{ on the left acts like the unit for } \wedge \\
A \supset B &= A \supset B
\end{aligned}
$$

This new exponential obeys two important rules, $!A \multimap A$, that is, applying ! weakens its argument, and $!!A = !A$, applying ! twice is the same as applying it once. We prove these two properties formally.

Lemma 131 1. $!A \multimap A$ 2. $!!A = !A$

Proof:

$$1. \quad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{A, \mathbf{I} \Rightarrow A}{A \wedge \mathbf{I} \Rightarrow A}}{!A \Rightarrow A}}{\Rightarrow !A \supset A} \qquad \dfrac{\dfrac{\dfrac{\neg A \Rightarrow \neg A, \neg \mathbf{I}}{\neg A \Rightarrow \neg (A \wedge \mathbf{I})}}{\neg A \Rightarrow \neg !A}}{\Rightarrow \neg A \supset \neg !A}}{\Rightarrow !A \supset A \wedge \neg A \supset \neg !A}}{\Rightarrow !A \multimap A}$$

$$2. \quad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{A, \mathbf{I} \Rightarrow A}{A \wedge \mathbf{I} \Rightarrow A} \quad \Rightarrow \mathbf{I}}{A \wedge \mathbf{I} \Rightarrow A \wedge \mathbf{I}}}{A \wedge \mathbf{I} \Rightarrow (!A)} \qquad A \wedge \mathbf{I} \Rightarrow \mathbf{I}}{A \wedge \mathbf{I} \Rightarrow (!A) \wedge \mathbf{I}}}{A \wedge \mathbf{I} \Rightarrow !!A}}{!A \Rightarrow !!A}}{\Rightarrow !A \supset !!A} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\neg !!A \Rightarrow \neg A, \mathbf{I}}{\neg !!A \Rightarrow \neg A, \neg \mathbf{I}}}{\neg !!A \Rightarrow \neg (A \wedge \mathbf{I})}}{\neg !!A \Rightarrow \neg !A}}{\Rightarrow \neg !!A \supset \neg !A}}{\Rightarrow !A \multimap !!A} \quad \text{and 1. give us } !A = !!A$$

∎

Linear logic contains two exponentials, $!_l$ and $?_l$, which are DeMorgan duals. We also gain a dual exponential ? to our new ! defined as $?A = \neg !\neg A = A \vee \mathbf{I}$.

$$\begin{aligned} \neg !\neg A &= A \vee \mathbf{I} \\ \neg(\neg A \wedge \mathbf{I}) &= A \vee \mathbf{I} \\ \neg\neg A \vee \neg \mathbf{I} &= A \vee \mathbf{I} \\ A \vee \mathbf{I} &= A \vee \mathbf{I} \end{aligned}$$

This new exponential obeys three important rules, $A \multimap ?A$, $?A = ??A$ and $A \multimap ?B = \neg B \supset \neg A$. Therefore, ? the DeMorgan dual to ! extracts out the dual proof. Remember

$$\begin{array}{llcl} & A \multimap B & = & A \supset B \quad \wedge \quad \neg B \supset \neg A \\ \text{and} & !A \multimap B & = & A \supset B \\ \text{whereas} & A \multimap ?B & = & \qquad\qquad\quad \neg B \supset \neg A. \end{array}$$

Therefore, the ! exponential extracts out the construction of a proof from a proof (in its BHK interpretation (Heyting 1966)) and the dual exponential ? extracts out the construction of a counterexample from the construction of a counterexample.

We now consider how these new exponentials behave algebraically. We show that ! maps $H \times H^{op}$ to $H \times 1$ which is isomorphic to H. This gives algebraic justification to our previously stated intuition that ! extracts the positive information. This means that ! is a projection from the algebra of constructible duality to a Heyting algebra; thus ! projects a constructible duality to intuitionistic logic.

Similarly ? maps $H \times H^{op}$ to $1 \times H^{op}$ which is isomorphic to H^{op}. This gives algebraic justification that ? extracts the negative information. This means that ? is a projection from the algebra of constructible duality to a dual-Heyting algebra (also called a Brouwerian algebra); thus ? projects the logic of constructible duality to Brouwerian logic (McKinsey and Tarski 1944, Birkhoff 1948).

Proposition 132 *! maps $H \times H^{op}$ to $H \times 1$ and ? maps $H \times H^{op}$ to $1 \times H^{op}$.*

Proof: Let A be a point $A = \langle a_1, a_2 \rangle$ in $H \times H^{op}$. Then, by definition, $!A = A \wedge \mathbf{I}$ where $\mathbf{I} = \langle 1, 1 \rangle$. Therefore $!A = \langle a_1, a_2 \rangle \wedge \langle 1, 1 \rangle$ by the definition of $\wedge$ gives, $!A = \langle a_1 \sqcap 1, a_2 \sqcup 1 \rangle$. Since 1 is the top of the lattice, this gives, $!A = \langle a_1, 1 \rangle$. Therefore, every point $\langle a_1, a_2 \rangle$ is mapped to $\langle a_1, 1 \rangle$, as required. The ? case is similar.

Thus these new exponentials project out an intuitionistic algebra from the larger algebra of constructible duality. To project to classical logic we simply perform the ! translation followed by the $\sim\sim$ closure. Before we show that these exponentials are the same exponentials as linear logic, we consider two more exponentials. These project in another way. They give $H \times 0$ and $0 \times H^{op}$. As ! and ? came from the overdefined unit, these new exponentials come from the unit for undefined.

6.6 The *Undefined* Exponential

Remember another exponentials is in *CD* using the constant for *undefined*, $\perp$. A weak exponential ¡ is definable as $¡A = A \wedge \perp$.

This weak exponential obeys $¡A \multimap A$ and $¡¡A = A$ and has its DeMorgan dual $¿A = A \vee \perp$ where $A \multimap ¿A$ and $¿A = ¿¿A$. This DeMorgan dual ¿ of ¡ also extracts intuitionistic implication from substitutable implication $\multimap$. However, it is applied to the second argument via the rule $A \multimap ¿B$.

$$\begin{array}{rcll}
A \supset B & = & A \multimap ¿B & \\
A \supset B & = & A \multimap (B \vee \bot) & \\
A \supset B & = & (A \supset (B \vee \bot)) \wedge & \neg(B \vee \bot) \supset \neg A \\
A \supset B & = & (A \supset (B \vee \bot)) \wedge & (\neg B \wedge \neg\bot) \supset \neg A \\
A \supset B & = & (A \supset B) \wedge & (\neg B \wedge \bot) \supset \neg A \\
A \supset B & = & A \supset B &
\end{array}$$

And ¡ extracts out the dual intuitionistic proof that is $¡A \multimap B = \neg B \supset \neg A$. In summary:

$$\begin{array}{llcl}
 & A \multimap B & \equiv & A \supset B \;\wedge\; \neg B \supset \neg A \\
\text{and} & !A \multimap B & \equiv & A \supset B \\
\text{and} & A \multimap ¿B & \equiv & A \supset B \\
\text{whereas} & A \multimap ?B & \equiv & \qquad\qquad \neg B \supset \neg A \\
\text{and} & ¡A \multimap B & \equiv & \qquad\qquad \neg B \supset \neg A
\end{array}$$

All four exponentials are distinct. It is notable that $A \multimap ¿\, B$ is not strongly equivalent, i.e. $\circ\!\!-\!\!\circ$, to $!A \multimap B$ because $\neg(A \multimap ¿\, B)$ is not equivalent to $\neg(\,!A \multimap B)$.

Proposition 133 *¡ maps $H \times H^{op}$ to $0 \times H^{op}$ and ¿ maps $H \times H^{op}$ to $H \times 0$.*

Theorem 134 *Given any Heyting algebra, H and any point $\langle h, h' \rangle$ in $H \times H^{op}$ there is a formula in constructible duality which represents $\langle h, h' \rangle$.*

Proof: Pick two points $\langle h, b \rangle$ and $\langle c, h' \rangle$ of $H \times H^{op}$. We construct $\langle h, h' \rangle$ via $!\langle h, b \rangle \vee ¡\langle c, h' \rangle$ — this expands to $\langle h, 1 \rangle \vee \langle 0, h' \rangle$ and then to $\langle h, h' \rangle$.

Now we use the onto representation theorem from intuitionistic logic to Heyting algebras and the onto representation theorem from Brouwerian logic to dual Heyting algebras to conclude that with the simple construction above every pair of elements in $H \times H^{op}$ is constructible. ∎

We can put this together with the representation lemma to conclude that constructible duality has an onto representation theorem with respect to $H \times H^{op}$.

We have shown that these new operators act as projections. We now proceed to establish the claim that these projection operations are exponentials, we need to show that they satisfy those requirements. Exponentials were introduced by Girard in linear logic so for a definition of an exponential we need to review their properties in the linear logic setting.

6.7 Linear Logic

Linear logic is a resource conscious logic introduced by Girard (Girard 1987). It has as five binary operators $\multimap, \otimes, \parr, \&, \oplus$, a unary $^{\perp}$ and two unary exponentials $!_l$ and $?_l$ where $!_l A = (?_l A^{\perp})^{\perp}$. It has four units $1, 0, T, \perp$. It can be axiomatized in a Gentzen system as follows.

Bold letters denote lists of formulas. Propositions are constructed from atomic ones (letters and their negations) by applying the binary connectives, $\otimes$, $\parr$, $\otimes$, & where the linear negation of a proposition is defined as follows:

$$1^{\perp} = \perp, \quad \perp^{\perp} = 1 \quad T^{\perp} = 0, \quad 0^{\perp} = T$$
$$(a^{\perp})^{\perp} = a$$
$$(A \otimes B)^{\perp} = A^{\perp} \parr B^{\perp}, (A \parr B)^{\perp} = A^{\perp} \otimes B^{\perp},$$
$$(A \oplus B)^{\perp} = A^{\perp} \& B^{\perp}, (A \& B)^{\perp} = A^{\perp} \oplus B^{\perp},$$

Logical Axioms

$\vdash A, A^{\perp}$

Cut rule:

$$\frac{\vdash \mathbf{B}, A \quad \vdash, A^{\perp}, \mathbf{C}}{\vdash \mathbf{B}, \mathbf{C}} \text{ Cut.}$$

Exchange rule:

$$\frac{\vdash \mathbf{A}}{\vdash \mathbf{B}} \text{ EXCH,}$$

$\mathbf{B}$ is a permutation of $\mathbf{A}$.

Additive rules:

$\vdash T, \mathbf{A}$

$$\frac{\mathbf{C}, A \quad \vdash \mathbf{C}, B}{\vdash \mathbf{C}, A \& B} \&1, \frac{A, \mathbf{C}}{\mathbf{D}, A \oplus B, \mathbf{C}} 1 \oplus \frac{B, \mathbf{C}}{\mathbf{D}, A \oplus B, \mathbf{C}} 2 \oplus .$$

Multiplicative rules:

$\vdash \mathbf{1}$(axiom), $\dfrac{\vdash \mathbf{A}}{\vdash \perp, \mathbf{A}} \perp,$

$$\frac{\vdash A, \mathbf{C} \quad \vdash B, \mathbf{D}}{\vdash A \otimes B, \mathbf{C}, \mathbf{D}} \otimes, \frac{\vdash \mathbf{D}, A, B, \mathbf{C}}{\vdash \mathbf{D}, A \parr B, \mathbf{C}}$$

Exponential rules:

$$\frac{\vdash A, \mathbf{B}}{\vdash ?_l A, \mathbf{B}} D?_l \quad (\text{ dereliction}), \quad \frac{\vdash ?_l A, ?_l A, \mathbf{B}}{\vdash ?_l A, \mathbf{B}} C?_l \quad (\text{contraction}).$$

$$\frac{\vdash \mathbf{B}}{\vdash ?_l A, \mathbf{B}} W?_l \quad (\text{ weakening}), \quad \frac{\vdash A, ?_l \mathbf{B}}{\vdash !_l A, ?_l \mathbf{B}}$$

Rather than give an algebra for linear logic it is more normal to consider categorical models. These models carry with them a notion of proof identity as well as the usual ordering relation of an algebra. We shall not be interested in the conditions of proof identity in this chapter since we covered this earlier.

We consider a theorem by Seely, that provides a definition of a model for linear logic. We then consider how we can instantiate this definition with a structure that we have already considered — $H \times H^{op}$. We need $H \times H^{op}$ to satisfy the prerequisites.

Theorem 135 *(Seely) Any $*$-autonomous category with products $\times$, coproducts $+$ together with a comonad $(!, e, \delta)$ such that there are isomorphisms $!(A \times B) \cong !A \otimes !B$, and $!1 \cong \mathbf{I}$ which takes* *the comonoid structure* $(A, !: A \to 1, \Delta: A \to A \times A)$ *to the comonoid structure* $(A, e: !A \to \mathbf{I}, \delta: !A \to !A \otimes !A)$ *is a model of classical linear logic.*

We can see any Heyting algebra as a simple category where objects are points of the lattice and morphisms are $\leq$ in the lattice. The category $\mathcal{X} \times \mathcal{X}^{op}$ has as its objects pairs of objects from $\mathcal{X}$ and has as its morphisms pairs of morphisms from $\mathcal{X}$ such that $(X, X') \in \mathcal{X} \times \mathcal{X}^{op}$ has a morphism to another object (W, W') just when $(X, X'') \multimap (W, W')$ that is when and $X \leq W$ and $W' \leq X'$ in $\mathcal{X}$. We now show that this $H \times H^{op}$ is a $*$-autonomous category. We show this via the Chu construction.

The Chu construction was introduced by Chu, a student of Barr's, in the appendix of (Barr 1979), as a method of producing $*$-autonomous categories.

Definition 136 Chu *(Barr 1979)*

Let $\mathcal{V}$ *be a finitely complete symmetric monoidal closed category, and let* $\perp$ *be an object of* $\mathcal{V}$*. We define a category* $\mathbf{Chu}(\mathcal{V}, \perp)$ *such that*

- *the objects are* Chu spaces*, that is, triples* (V, V', v) *where* V, V' *are objects of* $\mathcal{V}$ *and* $v: V \otimes V' \to \perp$ *is a morphism in* $\mathcal{V}$*,*
- *a morphism from* (V, V', v) *to* (W, W', w) *is a pair of morphisms* $\langle f, g\rangle$ *from* $\mathcal{V}$*, where* $f: V \to W$ *and* $g: W' \to V'$*, satisfying the following* adjunction property

$$\begin{tikzcd}
V \otimes W' \arrow[r, "f \otimes 1_{W'}"] \arrow[d, "1_V \otimes g"'] & W \otimes W' \arrow[d, "w"] \\
V \otimes V' \arrow[r, "v"] & \perp
\end{tikzcd}$$

Barr shows that for any choice of object, $\perp$, in $\mathcal{V}$, the category $\mathbf{Chu}(\mathcal{V}, \perp)$ is a $*$-autonomous category which embeds $\mathcal{V}$ (Barr 1979). A Heyting algebra is a finitely complete symmetric monoidal closed category with product $\wedge$ and residual $\supset$, thus a candidate for the **Chu** construction.

Theorem 137 *Let* $\mathcal{X}$ *be a Heyting algebra.* $\mathbf{Chu}(\mathcal{X}, 1)$ *is isomorphic to* $\mathcal{X} \times \mathcal{X}^{op}$*.*

Outline of Proof: We construct two functors F from $\mathbf{Chu}(\mathcal{X}, 1)$ to the pair algebra $\mathcal{X} \times \mathcal{X}^{op}$ and $G: \mathcal{X} \times X^{op} \to \mathbf{Chu}(\mathcal{X}, 1)$ such that $FG = 1_{\mathcal{X} \times \mathcal{X}^{op}}$ and $GF = 1_{Chu(\mathcal{X},1)}$.

$$
\begin{array}{lcllcl}
F((X, X', x\colon X \wedge X' \to 1)) & = & (X, X') & G((X, X')) & = & (X, X', x\colon X \wedge X' \to 1) \\
F(\langle f, g\rangle) & = & \langle f, g\rangle & G(\langle f, g\rangle) & = & \langle f, g\rangle
\end{array}
$$

The fact that $FG = 1_{\mathcal{X}\times\mathcal{X}^{op}}$ and that $GF = 1_{\mathbf{Chu}(\mathcal{X},1)}$ is immediate because $x\colon X \wedge X' \to 1$ always exists and is unique in any Heyting algebra. $\blacksquare$

Thus we have shown that the algebra $H \times H^{op}$ is a $*$-autonomous category.

To instantiate $H \times H^{op}$ in Seely's theorem, we need a comonad $(!, e, \delta)$ such that there are isomorphisms $!(A \times B) \cong !A \otimes !B$, and $!1 \cong \mathbf{I}$ which takes the comonoid structure $(A, !\colon A \to 1, \Delta\colon A \to A \times A)$ to the comonoid structure $(A, e\colon !A \to \mathbf{I}, \delta\colon !A \to !A \otimes !A)$

Definition 138 *A co-monad $\langle !, \epsilon\colon ! \to id, \delta\colon ! \to !^2\rangle$ consists of an endofunctor $!\colon C \to C$ and two natural transformations $\epsilon\colon ! \to id_C$ and $\delta\colon ! \to !^2$ such that the diagrams commute*

$$
\begin{tikzcd}
! \arrow[r, "\delta"] \arrow[d, "\delta"'] & !^2 \arrow[d, "!\delta"] \\
!^2 \arrow[r, "\delta_!"] & !^3
\end{tikzcd}
\qquad
\begin{tikzcd}
 & ! \arrow[dl, "id"'] \arrow[d, "\delta"] \arrow[dr, "id"] & \\
! & !^2 \arrow[l, "!\epsilon"'] \arrow[r, "\epsilon_!"] & !
\end{tikzcd}
$$

We now show that the operator $!$, that maps $H \times H^{op}$ to itself, via the rule $!A = A \wedge \mathbf{I}$ gives rise to a co-monad.

Definition 139 *A* closure *operation on some partial order $(X, \leq)$ is an operation T satisfying $\forall x \in X$.*

1. $x \leq Tx$
2. $TTx = Tx$
3. $x \leq y$ *implies* $Tx \leq Ty$

An interior *operation has the same 2. and 3. but 1. is changed to*

1. $Tx \leq x$

Lemma 140 *In a partially ordered set (poset) $(X, \leq)$ a comonad is an interior operation.*

Lemma 141 $!A = A \wedge \mathbf{I}$

1. *is a comonad* $(!, e, id)$.
2. *such that there are isomorphisms* $!(A \times B) \cong !A \otimes !B$
3. *and* $!1 = \mathbf{I}$.

Proof: For 1. We must show that ! is an interior operator. In Lemma 131 we showed the first two requirements, here we show the third. Remember: On the the poset $H \times H^{op}$, $\leq$ is given by $\multimap$.

$$\begin{array}{lll}
(A \multimap B) & \text{implies} & !A \multimap !B \\
A \supset B \wedge \neg B \supset \neg A & \text{implies} & A \wedge \mathbf{I} \multimap B \wedge \mathbf{I} \\
A \supset B \wedge \neg B \supset \neg A & \text{implies} & A \wedge \mathbf{I} \multimap B \wedge \mathbf{I} \\
A \supset B \wedge \neg B \supset \neg A & \text{implies} & A \wedge \mathbf{I} \supset B \wedge \mathbf{I} \ \wedge \neg(B \wedge \mathbf{I}) \supset \neg(A \wedge \mathbf{I}) \\
A \supset B \wedge \neg B \supset \neg A & \text{implies} & A \supset B \wedge \mathbf{I} \ \wedge \neg B \vee \mathbf{I} \supset \neg A \vee \mathbf{I} \\
A \supset B \wedge \neg B \supset \neg A & \text{implies} & A \supset B \text{ and } A \supset \mathbf{I} \\
A \supset B \wedge \neg B \supset \neg A & \text{implies} & A \supset B
\end{array}$$

Now we use Lemma 140 to conclude ! gives rise to a comonad. It is immediately an endofunctor.

To show that there are isomorphisms $!(A \times B) \cong !A \otimes !B$ we need them to be equal in the algebra. However, $!(A \wedge B) = !A \otimes !B$ and $!1 = \mathbf{I}$ immediately by putting together the representation theorem and the projection theorem. ∎

Corollary 142 *The algebra of constructible duality is a model for classical linear logic.*

The algebra of constructible duality validates mix.

As we showed that *CD* with our additional units has a representation onto $H \times H^{op}$. This means there is a sound translation from linear logic to constructible duality. Furthermore, this translation is the one alluded to.

Theorem 143 *Every proof rule of linear logic is admissible in* N^-*, under the translation given below. Let* A, B *be formulas in linear logic.*

Linear Logic	N^-	*Linear Logic*	N^-
$\mathbf{0}$	$\mathbf{0}$	$\top$	$\mathbf{1}$
$\mathbf{1}$	$\mathbf{I}$	$\perp$	$\mathbf{I}$
$A \otimes B$	$\neg((\mathcal{A} \supset \neg\mathcal{B}) \wedge (\mathcal{B} \supset \neg\mathcal{A}))$	$A \& B$	$\mathcal{A} \wedge \mathcal{B}$
$A \parr B$	$(\neg\mathcal{A} \supset \mathcal{B}) \wedge (\neg\mathcal{B} \supset \mathcal{A})$	$A \oplus B$	$\mathcal{A} \vee \mathcal{B}$
$A \multimap B$	$(\mathcal{A} \supset \mathcal{B}) \wedge (\neg\mathcal{B} \supset \neg\mathcal{A})$	$A^{\perp}$	$\neg\mathcal{A}$
$!A$	$\mathcal{A} \wedge \mathbf{I}$	$?A$	$\mathcal{A} \vee \mathbf{I}$

where calligraphic letters indicate that the translation is recursively applied to the subformulas.

Proof: The alignment of $\neg$ and $^{\perp}$ is obvious since $\neg$ inverts the additive units and makes intuitionistic ($\wedge$ and $\vee$) and ($\otimes$ and $\parr$) De Morgan duals. The rules involving constants are also immediate.

For every rule in linear logic we give its proof in N^-. The sequent $\vdash A, A^{\perp}$ is $A \parr A^{\perp}$ which translates to $(\neg\mathcal{A} \supset \neg\mathcal{A}) \wedge (\neg\neg\mathcal{A} \supset \neg\neg\mathcal{A})$ which gives the following prooftree.

$$\frac{\neg\mathcal{A} \supset \neg A \quad \neg\neg\mathcal{A} \supset \neg\neg\mathcal{A}}{(\neg\mathcal{A} \supset \neg\mathcal{A}) \wedge (\neg\neg\mathcal{A} \supset \neg\neg\mathcal{A}).} \wedge I$$

We now assume the rules are $\parr$'s of formulas. Furthermore, every LL rule will only need the top level translation from the table so we omit mention that formula A is translated to $\mathcal{A}$. This makes the $\parr$ rule $\frac{\Gamma, A, B}{\Gamma, A \parr B}$ immediate.

First we consider the permutation rule $EXCH$ $\quad \frac{\neg\Gamma, \neg A, \neg B, \Delta}{\neg\Gamma, \neg B, \neg A, \Delta}$.

Since Γ and Δ are arbitrary, then we chose the negations so that the translation becomes $\Gamma \multimap [A \multimap (B \multimap \Delta)]$.

First we do the negative side of the multimap $\neg[A \multimap (B \multimap \Delta)] \supset \neg\Gamma$.

Since $A \multimap B = \neg B \multimap \neg A$, then we use $\dfrac{A \multimap B \quad \neg B}{\neg A}$ as an admissible rule, we call it $\supset E^*$, shorthand $*$.

$$
\begin{prooftree}
\AxiomC{$\Gamma \multimap (A \multimap (B \multimap \Delta))$}
\AxiomC{$B \wedge (A \wedge \neg\Delta)^{(i)}$}
\UnaryInfC{$A \wedge \neg\Delta$}
\UnaryInfC{A}
\AxiomC{$B \wedge (A \wedge \neg\Delta)^{(i)}$}
\UnaryInfC{$A \wedge \neg\Delta$}
\UnaryInfC{$\neg\Delta$}
\AxiomC{$B \wedge (A \wedge \neg\Delta)^{(i)}$}
\UnaryInfC{B}
\BinaryInfC{$B \wedge \neg\Delta$}
\BinaryInfC{$A \wedge (B \wedge \neg\Delta)$}
\RightLabel{$*$}
\BinaryInfC{$\neg\Gamma$}
\RightLabel{$\supset I^{(i)} \quad (1)$}
\UnaryInfC{$(B \wedge (A \wedge \neg\Delta)) \supset \neg\Gamma$}
\end{prooftree}
$$

Now the positive part of the proof. Only the $\supset I$ rules will be annotated to keep track of closing open assumptions.

$$
\begin{prooftree}
\AxiomC{$B^{(c)}$}
\AxiomC{$A^{(d)}$}
\AxiomC{$\Gamma^{(a)}$}
\AxiomC{$\Gamma \multimap (A \multimap (B \multimap \Delta))$}
\BinaryInfC{$A \multimap (B \multimap \Delta)$}
\BinaryInfC{$B \multimap \Delta$}
\BinaryInfC{Δ}
\RightLabel{$\supset I^{(d)}$}
\UnaryInfC{$A \supset \Delta$}
\AxiomC{$\Gamma^{(a)}$}
\AxiomC{$\Gamma \multimap (A \multimap (B \multimap \Delta))$}
\BinaryInfC{$A \multimap (B \multimap \Delta)$}
\AxiomC{$\neg\Delta^{(e)}$}
\AxiomC{$B^{(c)}$}
\BinaryInfC{$B \wedge \neg\Delta$}
\RightLabel{$*$}
\BinaryInfC{$\neg A$}
\RightLabel{$\supset I^{(e)}$}
\UnaryInfC{$\neg\Delta \supset \neg A$}
\BinaryInfC{$A \multimap \Delta$}
\RightLabel{$\supset I^{(c)}$}
\UnaryInfC{$B \supset (A \multimap \Delta)$}
\AxiomC{$\neg\Delta \wedge A^{(b)}$}
\UnaryInfC{A}
\AxiomC{$A \multimap (B \multimap \Delta)$}
\BinaryInfC{$B \multimap \Delta$}
\AxiomC{$\neg\Delta \wedge A^{(b)}$}
\UnaryInfC{$\neg\Delta$}
\RightLabel{$*$}
\BinaryInfC{$\neg B$}
\RightLabel{$\supset I^{(b)}$}
\UnaryInfC{$\neg\Delta \wedge A \supset \neg B$}
\BinaryInfC{$B \multimap (A \multimap \Delta)$}
\RightLabel{$\supset I^{(a)} \quad (2)$}
\UnaryInfC{$\Gamma \supset [B \multimap (A \multimap \Delta)]$}
\RightLabel{$\wedge I$ on (1) and (2) from above}
\UnaryInfC{$\Gamma \multimap [B \multimap (A \multimap \Delta)]$}
\end{prooftree}
$$

$$\text{Tensor Rule } \frac{\Gamma, A \quad \Delta, B}{\Gamma, \Delta, A \otimes B}$$

We are given $\neg\Gamma \multimap A$ and $\neg\Delta \multimap B$, but since Γ is arbitrary we will assume $\Gamma \multimap A$ and $\Delta \multimap B$ and derive $(\Gamma \parr (\Delta \multimap A \otimes B))$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Gamma \multimap A \quad \Gamma^{(a)}}{A} \quad \dfrac{\Delta \multimap B \quad \Delta^{(b)}}{B}}{A \wedge B}}{\Delta \supset A \wedge B}\supset I^{(b)} \quad \dfrac{\dfrac{\dfrac{\dfrac{\Gamma \multimap A \quad \Gamma^{(a)}}{A} \quad A \multimap \neg B^{(c)}}{\neg B} \quad \Delta \multimap B}{\neg \Delta}}{(A \multimap \neg B) \supset \neg\Delta}\supset I^{(c)}}{\Delta \multimap A \otimes B}}{\Gamma \supset [\Delta \multimap A \otimes B]}\supset I^{(a)} \quad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Delta \wedge (A \multimap \neg B)^{(d)}}{\Delta} \quad \Delta \multimap B}{B} \quad \dfrac{\Delta \wedge (A \multimap \neg B)^{(d)}}{A \multimap \neg B}}{\neg A} \quad \Gamma \multimap A}{\neg\Gamma}}{\Delta \wedge (A \multimap \neg B) \supset \neg\Gamma}\supset I^{(d)}}{\Gamma \multimap [\Delta \multimap (A \otimes B)]}$$

Additive Rules

$$\frac{\Gamma, A \quad \Gamma, B}{\Gamma, A \& B}$$

Again this means (after translation) we are given $\neg\Gamma \multimap A$ and $\neg\Gamma \multimap B$, but since Γ is arbitrary we will assume $\Gamma \multimap A$ and $\Gamma \multimap B$ and derive $\Gamma \multimap A \wedge B$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\Gamma \multimap A \quad \Gamma^{(a)}}{A} \quad \dfrac{\Gamma \multimap B \quad \Gamma^{(a)}}{B}}{A \wedge B}}{\Gamma \supset A \wedge B}\supset I^{(a)} \quad \dfrac{\dfrac{\dfrac{\Gamma \multimap A \quad \neg A}{\neg\Gamma} \quad \dfrac{\Gamma \multimap B \quad \neg B}{\neg\Gamma} \quad (\neg A \vee \neg B)^{(b)}}{\neg\Gamma}\vee E}{(\neg A \vee \neg B) \supset \neg\Gamma}\supset I^{(b)}}{\Gamma \multimap A \wedge B}$$

$$\frac{\Gamma, A}{\Gamma, A \oplus B}$$

Again this means we are given $\neg\Gamma \multimap A$ and $\neg\Gamma \multimap B$, but since Γ is arbitrary we will assume $\Gamma \multimap A$ and derive (after translation) $\Gamma \multimap A \vee B$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\Gamma \multimap A \quad \Gamma^{(a)}}{A}}{A \vee B}}{\Gamma \supset A \vee B} \supset I^{(a)} \qquad \dfrac{\dfrac{\dfrac{(\neg A \wedge \neg B)^{(c)}}{\neg A} \quad \Gamma \multimap A}{\neg\Gamma}}{(\neg A \wedge \neg B) \supset \neg\Gamma} \supset I^{(c)}}{\Gamma \multimap A \vee B}$$

The Exponential Rules

$$\frac{\Gamma, A}{\Gamma, ?A}$$

Again this means we are given $\neg\Gamma \multimap A$ and $\neg\Gamma \multimap B$, but since Γ is arbitrary we will assume $\Gamma \multimap A$ and derive (after translation) $\Gamma \multimap A \vee \mathbf{I}$, but this follows immediately from the additive proof above.

Similarly weakening for exponentials

$$\frac{B}{?A, B}$$

after translation looks like, $\dfrac{B}{A \wedge \mathbf{I} \multimap B}$

$$\dfrac{\dfrac{B}{A \wedge \mathbf{I} \supset B} \supset I \qquad \dfrac{\dfrac{\mathbf{I}}{\neg A \vee \mathbf{I}} \vee I}{\neg B \supset \neg A \vee \mathbf{I}} \supset I}{A \wedge \mathbf{I} \multimap B}$$

Similar-y the other weakening for exponentials

$$\frac{A, B}{?A, B}$$

after translation looks like, $\dfrac{A \multimap B}{A \wedge \mathbf{I} \multimap B}$. Remember $\neg\mathbf{I} \equiv I$, so we do not bother to show negations over $\mathbf{I}$.

$$\dfrac{\dfrac{\dfrac{\dfrac{(A\wedge \mathbf{I})^{(a)}}{A}\quad A\multimap B}{B}}{A\wedge\mathbf{I}\supset B}\supset I^{(a)}\qquad \dfrac{\dfrac{\mathbf{I}}{\neg A\vee \mathbf{I}}\vee I}{\neg B\supset(\neg A\vee\mathbf{I})}\supset I}{A\wedge \mathbf{I}\multimap B}$$

Lastly, contraction for exponential

$$\dfrac{?A, ?A, B}{?A, B}$$

after translation looks like $\dfrac{(A\wedge\mathbf{I})\multimap[(A\wedge\mathbf{I})\multimap B]}{(A\wedge\mathbf{I})\multimap B}$.

$$\dfrac{\dfrac{\dfrac{\dfrac{(A\wedge\mathbf{I})^{(a)}\quad (A\wedge\mathbf{I})\multimap[(A\wedge\mathbf{I})\multimap B]}{(A\wedge\mathbf{I})\multimap B}\qquad (A\wedge\mathbf{I})^{(a)}}{B}}{(A\wedge\mathbf{I})\supset B}\supset I^{(a)}\qquad \dfrac{\dfrac{\mathbf{I}}{\neg A\vee\mathbf{I}}\vee I}{\neg B\supset(\neg A\vee\mathbf{I})}\supset I}{A\wedge\mathbf{I}\multimap B}$$

MIX

$\dfrac{\Gamma, A\Delta, B}{\Gamma, \Delta, A, B}$ after translation we get from premises $\neg\Gamma \multimap A$ $\neg\Delta \multimap B$ we need $\neg\Gamma \multimap [\neg\Delta \multimap (\neg A \multimap B)]$, so after the usual messing with negations because Γ and Δ are arbitrary we get: $\dfrac{\Gamma \multimap A \quad \Delta \multimap B}{\Gamma \multimap [\Delta \multimap (\neg A \multimap B)]}$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Gamma\multimap A\quad \Gamma^{(a)}}{A}}{\neg B\supset A}\supset I\qquad \dfrac{\dfrac{\Delta\multimap B\quad \Delta^{(b)}}{B}}{\neg A\supset B}\supset I}{(\neg A\multimap B)}}{\Delta\supset(\neg A\multimap B)}\supset I^{(b)}\qquad \dfrac{\dfrac{\dfrac{(\neg A\wedge\neg B)^{(c)}}{\neg B}\quad \Delta\multimap B}{\neg\Delta}}{(\neg A\wedge\neg B)\supset\neg\Delta}\supset I^{(c)}}{\Delta\multimap(\neg A\multimap B)}}{\Gamma\supset[\Delta\multimap(\neg A\multimap B)]}\supset I^{(a)}\qquad \dfrac{\dfrac{\dfrac{(\Delta\wedge\neg A\wedge\neg B)^{(d)}}{\neg A}\quad \Gamma\multimap A}{\neg\Gamma}}{(\Delta\wedge\neg A\wedge\neg B)\supset\neg\Gamma}\supset I^{(d)}}{\Gamma\multimap[\Delta\multimap(\neg A\multimap B)]}$$

∎

Corollary 144 *1. Let A be a valid formula in linear logic, then $\mathcal{A}$ is valid in N^-.*

2. Let D be a proof of A in linear logic, then the result of replacing each occurrence of a proof rule by the corresponding proof from the admissibility Theorem 143 is a proof of $\mathcal{A}$ in N^-.

Thus every linear logic proof gives rise to a proof in N^-, therefore we have the promised interpretation of linear logic in N^-. This result lifts to proofs. This is we can translate every line of a linear logic proof with this translation and this gives a proof in *CD*.

This result can be strengthened for the multiplicatives.

Definition 145 *A formula is balanced if each propositional atom occurs at most twice and if twice with opposite sign.*

Every proof of a formula of multiplicative linear logic is an identification of a proof of a balanced formula.

Theorem 146 *If A is a balanced formula in multiplicative linear logic, then A is valid in multiplicative linear logic with mix, if and only if $\mathcal{A}$ is valid in constructible duality.*

The proof of this is in (Patterson and Costello 1997d)

In fact there is only one proof in each case, but that requires a proof of a coherence theorem for a suitable category but this is future work.

Using the results of (Cubric 1994), this logic lifts easily to a fully-complete logic for $\mathcal{C} \times \mathcal{C}^{op}$ where $\mathcal{C}$ is a bicartesian closed category. Alternately, we can use Cubric's weak Set (**Set** with a weak initial object), and use $\mathbf{Set}_w \times \mathbf{Set}_w^{op}$.

6.8 Relationship to the Dialectica

Another test of its relationship to linear logic is its relationship with the second Dialectica category. Here we show that the algebra of constructible duality is also closely related to this category.

The second Dialectica category by de Paiva is usually called $G\mathcal{C}$ (1989b) for Girard category. $G\mathcal{C}$ has been used to model Reddy's LLMS(1994) and linear logic. We give an equivalent definition only for G acting upon a Heyting algebra.

Definition 147 Dialectica Category for LL

Let $\mathcal{X}$ be a Heyting algebra. A category $G\mathcal{X}$ is defined as follows:

- *objects of $G\mathcal{X}$ are morphisms in $\mathcal{X}$ of the form $\alpha: A \to V \wedge V'$ and*
- *any morphism of $G\mathcal{X}$ from $\alpha: A \to V \wedge V'$ to $\beta: B \to W \wedge W'$ is a pair $\langle f, g\rangle$ of morphisms from $\mathcal{X}$, $f: V \to W$ and $g: W' \to V'$, such that there exists an $h: B \to A$ in $\mathcal{X}$ such that the following diagram commutes*

$$\begin{tikzcd} B \arrow[r, "h"] \arrow[dr, "\beta"] & A \arrow[r, "\alpha"] & V \wedge V' \arrow[d, "f \wedge 1_{V'}"] \\ & W \wedge W' \arrow[r, "1_W \wedge g"] & W \wedge V' \end{tikzcd}$$

We now consider a smaller category than $G\mathcal{X}$. We consider the category in which all maps come from the same object D rather than from different A's and B's.

Definition 148 *Let $G\mathcal{X}_D$ have as objects the set $D = \{\alpha \mid \alpha: D \to V \wedge V' \in G\mathcal{X} \wedge V, V' \in \mathcal{X}\}$. Let $G\mathcal{X}_D$ have as morphisms the morphisms in $G\mathcal{X}$ between objects in D.*

This diagram is the restricted morphism condition.

$$\begin{tikzcd} D \arrow[r, "\alpha"] \arrow[d, "\beta"'] & V \wedge V' \arrow[d, "f \wedge 1_{V'}"] \\ W \wedge W' \arrow[r, "1_W \wedge g"] & W \wedge V' \end{tikzcd}$$

Theorem 149 *Let $\mathcal{X}$ be a pseudo-Boolean algebra, $G\mathcal{X}_0$ is isomorphic to $\mathcal{X} \times \mathcal{X}^{op}$.*

Outline of Proof: Again we construct a functor F from $G\mathcal{X}_0$ to the pair algebra and a functor G' from the pair algebra $\mathcal{X} \times \mathcal{X}^{op}$ to $G\mathcal{X}_0$.

$$\begin{array}{lcl} F(\alpha: 0 \to V \wedge V') & = & (V, V') \\ F(\langle f, g\rangle) & = & \langle f, g\rangle \end{array} \qquad \begin{array}{lcl} G'((V, V')) & = & \alpha: 0 \to V \times V' \\ G'(\langle f, g\rangle) & = & \langle f, g\rangle \end{array}$$

Both F and G' are functors such that $FG' = 1_{\mathcal{X} \times \mathcal{X}^{op}}$ and $G'F = 1_{G\mathcal{X}_0}$. ∎

6.9 Conclusion

All of the logic transformers, !, ?, ¡, ¿ $!_l$, $?_l$ and $\sim\sim$ are closure or interior operators.

Closure and interior operations can be thought of as approximations. When the approximation is from above, like $\lceil n \rceil$, which maps every real number to the first natural greater than or equal to it, the approximation is a closure operation. $\lfloor n \rfloor$, the approximation from below it is an interior operation.

The Glivenko or Gödel $\sim\sim$ operator is a closure operation. Also Girard's *why not* $?_l$ operator is a closure operation as are our ? and ¿ operators.

The Gödel $\Box$ translation from S4 to intuitionistic logic is an interior operation. Girard's *of course* or $!_l$ and our ! and ¡ operators are also interior operations.

This shows that we can think logic transformers as approximations. Thus our projections (or exponentials) that recover consistent logics from paraconsistent logics are approximations by a consistent logic of a paraconsistent logic. Both ! and ¿ capture the notion of removing all negative information from a proposition and ? and ¡ capture the notion of removing all positive information from a proposition.

Thus we showed that we can naturally divide the exponential construction of Girard into two parts, one that deals with "linearity" or counting the occurrences of assumptions, and a second that captures the embedding of a intuitionistic logic in a paraconsistent logic.

We have four exponentials while linear logic has two and these occur in a conservative extension of *CD* called constructible duality. They are all projections in the algebra, and the first two, the overdefined exponentials, arise as Girard's exponentials in a model of linear logic. The undefined exponentials do not appear in linear logic, but they are natural in constructible duality.

Chapter 7

Implicit Programming

We present a new semantics for recursive programs, that is programs that define functions using recursive (or self-referencing) definitions (McCarthy 1963b). Rather than choose the least defined solution, we suggest that if there is a uniquely defined solution, then that solution should be chosen. We show that this uniquely defined solution corresponds to Manna and Shamir's optimal fixed point (Manna and Shamir 1976, Manna and Shamir 1977) for many natural programs.

Manna and Shamir's optimal fixed point is uncomputable in general. However, using notions from proof theory (Troelstra and Schwichtenberg 1996), we show how a constructive definition in the style of an implicit definition in CD corresponds to a computable function. We use the Curry-Howard-De Bruijn isomorphism in the first order case, and show how least fixed points correspond to proofs of existence, while constructive optimal fixed points correspond to proofs of uniqueness.

An assignment of terms to proofs gives us a lambda calculus in which these optimal fixed points are calculated. We give examples at the beginning of this chapter to motivate the use of optimal fixed points, and at the end to explain the technique we propose in more detail.

Some of the examples are functions on the natural numbers, and other examples are functions which act upon co-recursively defined data-structures, such as streams and circular lists. Implicit definitions of programs are very natural for functions whose domains include co-recursively defined data-structures (Barwise and Moss 1996, Fiore 1996, Moss and Danner 1996, Park 1983, Milner and Tofte 1991, Pitts 1994, Gordon 1995).

7.1 Fixed Points

Many programming languages are naturally characterized in terms of the least fixed point of a recursive operator (Kleene 1959, Moschovakis 1974, Moschovakis 1977, Cutland 1980).

Consider the simple example of the factorial function.

$$\begin{aligned} f(0) &= 1 \\ f(n+1) &= (n+1) * f(n) \end{aligned}$$

This is a recursive definition of the factorial function. We can make this definition into an operator, i.e. a mapping of functions from $\mathbb{N}$ to $\mathbb{N} \cup \{\perp\}$ to functions from $\mathbb{N}$ to $\mathbb{N} \cup \{\perp\}$, as follows:

$$\Phi(f) = \lambda x. \begin{cases} 1 \text{ if } x = 0 \\ (x+1) * f(x) \text{ otherwise.} \end{cases}$$

If we apply Φ to the totally undefined function $\lambda x.\perp$, we get $\lambda x.$ if $x = 0$ then 1 else $\perp$. If we apply Φ to this result, we get $\lambda x.$ if $x = 0$ then 1 else if $x = 1$ then 2 else $\perp$. What is the least defined function g which has the property that $\Phi(g) = g$? In this case, the answer is the factorial function. In general, the least fixed point of an operator is the function that is a fixed point of the operator and is undefined as often as possible.

7.1.1 Optimal Fixed Points

Other fixed points are possible, such as the optimal fixed point. The optimal fixed point is more natural than the least fixed point for certain classes of programs. We use a new example to illustrate the difference between optimal and least fixed points. We need a different example than the factorial example presented above because for primitive recursive functions, optimal fixed points agree with least fixed points.

Consider the following recursive equation.

$$f(x) = 0 * f(x).$$

There is a unique total function that satisfies this equation — the function $\lambda x.0$. However, there are many partial functions that also satisfy this equation; those are the partial functions that return 0 for some inputs. The operator corresponding to this recursive definition is

$$\Phi_0(f) = \lambda x.0 * f(x)$$

The least fixed point of this operator is $\lambda x.\bot$ — the totally undefined function. The constant zero function seems a more informative answer. The constant zero function is a fixed point,

$$\begin{aligned} \Phi(\lambda x.0) &= \lambda x.0 * (\lambda x.0)(x) \\ & \lambda x.0 * 0 \\ & \lambda x.0 \end{aligned}$$

A natural intuition is that if there is a unique answer, then the program should return that answer, rather than not return at all. The optimal fixed point of the operator Φ_0 in the example above is the constant zero function $\lambda x.0$.

We now define the notion of an implicit definition for classical logic. Let $\overline{x}$ denote a tuple of variables.

Definition 150 *Let L be a signature not mentioning the function symbols f and f'. Let $\Gamma(f)$ be a set of sentences in $L \cup \{f\}$ and let $\Gamma(f')$ be the same set of sentences replacing f by f'.*

$\Gamma(f)$ implicitly defines f *if and only if* $\Gamma(f) \cup \Gamma(f') \models \forall \bar{x}.[f(\bar{x}) = f'(\bar{x})]$.

This states that if two first-order classical logic models of Γ agree on the interpretation of the L formulas then they must agree on the interpretation of the function symbol f. This can easily be adapted to implicit definitions of relations or constants.

Now we consider implicit definitions of functions which act on co-recursive data structures. Consider list structures, as in Lisp. A natural notion of equality, (called `equal` in lisp) is (partially) axiomatized as follows.

$$l = l' \iff car(l) = car(l') \wedge cdr(l) = cdr(l').$$

This gives rise to an operator $\Phi_=$,

$$\begin{aligned}\Phi_=(R) \quad = \quad & \lambda xy.\ \text{if } atom(car(x)) \text{ or } atom(car(y)) \text{ then} \\ & car(x) = car(y) \wedge R(cdr(x), cdr(y)) \\ & \ \text{else } R(car(x), car(y)) \wedge R(cdr(x), cdr(y))\end{aligned}$$

The least fixed point of $\Phi_=$ correctly characterizes equality for lists that do not contain loops. Two cyclic lists however are not equal, even to themselves, by the least fixed point of `equal`.

Another possibility is to consider a different equivalence relation on lists: bisimilarity (Milner 1989) (Barwise and Moss 1996). The notion of defining data-types corecursively has been considered by many researchers (Gordon 1994, Gordon 1995, Pitts 1994, Freyd 1990). When data-types are defined co-recursively, bisimilarity is the notion of identity that is used. Bisimilarity is the equivalence notion given by the greatest fixed point of the defining equations for `equal`.

With bisimilarity as our notion of equivalence of lists, we can consider simple programs on lists. The following function is intended to return a list with every occurrence of a natural number replaced with the symbol '***. Perhaps we wish to remove all numbers from a document before making it public.

$$\Phi(f) = \quad \lambda x. \begin{array}{l}\text{if nat}(x) \text{ then } '***\\ \text{elseif atom}(x) \text{ then } x \\ \text{else cons}(f(car(x)), f(cdr(x)))\end{array}$$

The least fixed point of this operator Φ is defined for all lists that do not contain loops. However, for cyclic lists the least fixed point of this operator Φ is undefined. The natural answer is that every number in a list should be replaced by '***. We can see is that there is only one totally defined solution of the above definition. Further, this totally defined function is our intuitively correct answer. This function is defined implicitly and for each input returns the cyclic list which is the bisimulation class representative. We would like this implicit answer for the output of our program. We now use this intuition to define another fixed point.

7.2 Basic Recursion Theory

Definition 151 *Let $\mathcal{Y}, \mathcal{Y}'$ be two partially ordered sets, ordered by $\leq$ and $\leq_1$ respectively.*

A function $f \colon \mathcal{Y} \to \mathcal{Y}'$ is a monotone function if whenever $y \leq y'$, then $f(y) \leq_1 f(y')$.

Definition 152 *Let $N_\perp = \mathbb{N} \cup \{\perp\}$ and let $\mathcal{F}_n$ be the class of all total functions $\mathbb{N}^n \to \mathbb{N}_\perp$.*

An operator *$\Phi \colon \mathcal{F}_m \to \mathcal{F}_n$ is a total function from $\mathcal{F}_m$ to $\mathcal{F}_n$.*

We shall only be interested in monotonic operators, therefore for the remainder of the chapter an *operator* is a *monotonic operator.* Furthermore, the total functions from $\mathbb{N} \to \mathbb{N}_\perp$ we sometimes refer to as *partial* functions since we interpret $f(n) = \perp$ as f is *not defined* at n; hence partial.

Definition 153 *The information ordering $<_\perp$ on $\mathbb{N}_\perp$ is the least relation such that $\forall n \in \mathbb{N}.\ \perp <_\perp n$. The $\leq_\perp$ ordering on $\mathbb{N}_\perp$ is $\forall n, n' \in N_\perp . n <_\perp n'$ or $n = n'$.*

The meet $\sqcap_\perp$ and join $\sqcup_\perp$ on $\mathbb{N}_\perp$ is defined as $\forall n, n' \in \mathbb{N}_\perp$.

$$n \sqcap_\perp n' = \begin{cases} \textit{if } n = n' \textit{ then } n \\ \textit{else } \perp \end{cases}$$

Definition 154 *We say that one function $f' \colon \mathbb{N}^n \to \mathbb{N}_\perp$ is an extension of another function $f \colon \mathbb{N}^n \to N_\perp$, written $f \sqsubseteq f'$, if $\forall \bar{x} \in \mathbb{N}^n . f(\bar{x}) \leq_\perp f'(\bar{x})$.*

Definition 155 *We define a semi-lattice on $\mathcal{F}_m$, $L_{\mathcal{F}_m}$, by taking $\sqsubseteq$ as the ordering on points and $\sqcap$ as the pointwise non-empty meet which is $\forall \bar{x} \in \mathbb{N}^m.\ g(\bar{x}) = f(\bar{x}) \sqcap_\perp f'(\bar{x})$ where $g = f \sqcap f'$ is the meet. The bottom of the lattice is the totally undefined function $f_\perp$, there is no top of the lattice.*

Note: Johnstone (Johnstone 1982) argues that a semi-lattice with non-empty meets must have the empty meets and therefore a top element in the semi-lattice whereas we take the traditional definition and require no such top element.

Definition 156 *For any subset S of $L_{\mathcal{F}_m}$*

- *$f \in S$ is the* least element *of S if $f \sqsubseteq g$ for any $g \in S$.*
- *$f \in S$ is a* minimal element *of S if there is no $g \in S$ which satisfies $g \sqsubseteq f$, and $\neg(f \sqsubseteq g)$.*

- $f \in L_{\mathcal{F}_m}$ *is an* upper bound *of* S *if* $g \sqsubseteq f$ *for all* $g \in S$.
- $f \in L_{\mathcal{F}_m}$ *is the* least upper bound *(lub) of* S *if* f *is the least element, under* $\sqsubseteq$ *in the set of upper bounds of* S. *Note — this may not exist.*

The notions of a maximal element, *a* lower bound *and the* greatest lower bound *(glb) are defined dually.*

Note: The lub and glb of a subset S are not necessarily in S.

Definition 157 *(Kleene 1967a) If f and g are unary functions, then we write $f(x) \simeq g(x)$ if and only if*

(f is defined at x if and only if g is defined at x) and when defined their values are equal i.e.

$$f(x) \neq \bot \iff g(x) \neq \bot \quad \wedge \quad f(x) \neq \bot \textit{ implies } f(x) = g(x).$$

If we write $f(x) \simeq y$ to mean if f is defined at x it equals y.

This obviously generalizes to n-ary functions.

Definition 158 (Consistent Functions) *Two functions $f, g \in L_{\mathcal{F}_m}$ are* consistent *if for all $\overline{x} \in \mathbb{N}^m$ if $f(\overline{x}) \neq \bot$ and $g(\overline{x}) \neq \bot$ then $f(\overline{x}) = g(\overline{x})$.*

A non-empty subset S of $L_{\mathcal{F}_m}$ is consistent if every two elements in S are consistent.

Lemma 159 *A non-empty subset S of $L_{\mathcal{F}_m}$ has a lub if and only if S is consistent.*

Every nonempty subset S of $L_{\mathcal{F}_m}$ has a glb.

7.2.1 Fixed Points

Definition 160 *Let Φ be a monotonic operator on $\mathcal{F}_m$.*

A function $f \in L_{\mathcal{F}_m}$ is a fixed point *of $\Phi: \mathcal{F}_m \to \mathcal{F}_m$ if $f = \Phi(f)$.*

A function $f \in L_{\mathcal{F}_m}$ is a pre-fixedpoint *of Φ if $f \sqsubseteq \Phi(f)$.*

The set of all fixedpoints and pre-fixedpoints of Φ are denoted by $FXP(\Phi)$ and $PRE(\Phi)$ respectively.

A few consequences should be pointed out here.

Each set of points $FXP(\Phi)$, and $PRE(\Phi)$ is closed under application of Φ if Φ is monotone.

Lemma 161 *If a subset S of $PRE(\Phi)$ is consistent, then the lub of S is in $PRE(\Phi)$.*

This is a specific theorem which slightly generalizes older results, see (Birkhoff 1967) for a chapter of similar results.

Theorem 162 (Hitchcock and Park) *Let $\mathcal{Y}$ be a partially ordered set with a least element Ω and such that any totally ordered subset has a lub. Then for any monotonic mapping $\tau: \mathcal{Y} \to \mathcal{Y}$, the set of fixed points of τ contains a least element.*

We now apply this theorem to yield the following result of Manna and Shamir.

Theorem 163 (Manna and Shamir) *Let Φ be any monotonic operator $\Phi: \mathcal{F}_m \to \mathcal{F}_m$, then*

1. *$FXP(\Phi)$ contains a least element.*
2. *If $f \in PRE(\Phi)$ $(f \sqsubseteq \Phi(f))$, then the set $\{f' \mid f' \in FXP(\Phi)$ and $f \sqsubseteq f'\}$ contains a least element.*

Definition 164 (Consistent Fixed points) *A fixed point f of Φ is* FXP-consistent *if for any $f' \in FXP(\Phi)$, f and f' are consistent.*

The set of all FXP-consistent fixed points of Φ is denoted $FXPC(\Phi)$.

Lemma 165 (Least and Greatest Consistent Fixed points) *The least fixed point f of Φ is in $FXPC(\Phi)$.*

The lub of $FXPC(\Phi)$ exists.

The lub over the consistent fixed points exists by Lemma 159.

Theorem 166 (Manna and Shamir) *For a monotone operator $\Phi: \mathcal{F}_m \to \mathcal{F}_m$, $FXPC(\Phi)$ contains a greatest element.*

Proof: The lub f_l of $FXPC(\Phi)$ exists by Lemma 159. Since f_l is a lub of consistent fixed points, then $f_l \in PRE(\Phi)$ by the observations after Definition 160. Hence by Theorem 163 (2), the set $\{f' \mid f' \in FXP(\Phi)$ and $f_l \sqsubseteq f'\}$ contains a least element say f_2.

We now show that $f_2 \in FXPC(\Phi)$, implying that f_2 is the greatest function in $FXPC(\Phi)$.

Let g be any fixed point of Φ. We would like to prove that f_2 and g are consistent by showing the existence of a function f_3 such that $f_2 \sqsubseteq f_3$ and $g \sqsubseteq f_3$. The set of fixed points $S = FXPC(\Phi) \cup \{g\}$ is consistent by the definition of $FXPC(\Phi)$ and therefore again by Theorem 163 (2), there exists some $f_3 \in FXP(\Phi)$ such that (lub S) $\sqsubseteq f_3$. Thus $g \sqsubseteq f_3$ and (lub $FXPC(\Phi)$) $\sqsubseteq f_3$. Since f_2 was defined as the least fixed point such that (lub $FXPC(\Phi)$) $\sqsubseteq f_2$, we have $f_2 \sqsubseteq f_3$. ∎

These optimal fixed points were first defined by (Manna and Shamir 1976) and studied by (Manna and Shamir 1977, Gallier 1980, Chen 1996, Lassez and Maher 1985). We present the original formulation, rather than later presentation, as Manna and Shamir also studied the relationship between fixed points and equational definitions of functions.

Definition 167 (Manna and Shamir) *The* optimal fixed point f_o *of a monotonic operator* Φ *is the greatest element in* $FXPC(\Phi)$.

We know the optimal fixed point exists because of the previous proof.

Theorem 168 *Let* Φ *be a monotonic operator. If there is a greatest fixed point of* Φ*, then it is the optimal fixed point of* Φ.

Proof: The optimal fixed point of Φ, f_o is the greatest element in $FXPC(\Phi)$. If there is a unique maximal fixed point f_m, then f_m is the lub over the set of all fixed points $FIX(\Phi)$. Using Lemma 159 this means that the set of fixed points is consistent. Hence $FIX = FIXC$. Therefore, the greatest element of FIX equals the the greatest element of $FIXC$. ∎

Example: 1 *The optimal fixed point of the operator,* $\Phi(f) = \lambda x.0 * f(x)$ *is* $\lambda x.0$.

The fixed points of this operator are the functions that map a subset of the naturals to 0*, and are undefined everywhere else. Thus, there is a unique maximum fixed point, the function* $\lambda x.0$.

Proposition 169 *The optimal fixed point* f_o *of a monotonic operator* Φ *is at least as defined as the least fixed point* f_l*, i.e* $f_l \sqsubseteq f_o$.

In fact, if f_l is total, then $f_o = f_l$ since there is only one fixed point.

However, the optimal fixed point lacks one important feature of the least fixed point: *computability.* The least fixed point of a recursive operator Φ is computable: we can compute it in the following way. Suppose we wish to evaluate the least fixed point of Φ at a value v. We apply Φ to the totally undefined function $\lambda x.\bot$. As Φ is recursive, this returns a function, which we call f_1. We apply Φ to f_1 to get $f_2, f_3, \ldots, f_n \ldots$. If the least fixed point of Φ is defined at v, then for some value n, $f_n(v)$ will be defined. Thus, after a finite number of steps, we will find the value of the least fixed point of Φ at any given value v.

The optimal fixed point cannot be evaluated in this way. In fact, there is no uniform way to evaluate the optimal fixed point, as the function it defines can be uncomputable.

Theorem 170 (Manna and Shamir) *Every arithmetical function, i.e. a function that can be defined in the language of arithmetic over the natural numbers, is the optimal fixed point of a recursive operator.*

Rather than consider arbitrary optimal fixed points, we consider the functions whose outputs we can constructively prove are implicitly defined. This is a subset of the functions defined by optimal fixed points, namely those optimal fixed points that we can constructively show are unique maximal fixed points. We define this fixed point by using the relationship between constructive logic and computation.

7.3 The Curry-Howard-De Bruijn Isomorphism

We now define a *formulas as types* formalism for constructible duality and show that proof normalization corresponds to the operational semantics of an extension of the lambda-calculus.

The Curry-Howard-De Bruijn isomorphism states that proof normalization in propositional intuitionistic logic is isomorphic to reductions in the typed lambda calculus. This has been extended to Heyting arithmetic in (Troelstra 1973). We consider an extension of this isomorphism to constructible duality.

We use the logic of constructible duality in positive normal form and give an assignment of terms in a lambda calculus to proofs in this logic. These terms have a set of reductions, $\triangleright$, associated with them, shown in the following table. This is the λ_{HAZ} calculus; for Heyting arithmetic plus Z. We added the *new* Z_0 and Z_1 reductions for predecessor terms to Heyting arithmetic.

7.3.1 Definition of Formulas as Types in Constructible Duality

We give a natural deduction formulation (written sequent-style) for proofs in constructible duality mentioning only formulas in positive normal form, Lemma 106, that do not mention the constants, that is do not mention intuitionistic negation or falsity. In this formulation, we give an assignment of terms to proofs.

The terms are taken from a typed lambda calculus.

Definition 171 (Value terms) *The set of value terms is constructed from a set of* value variables V*, the constant* 0 *(representing the natural number 0), the successor function* S*, and the predecessor* P*.*

$$v_t = 0|u|P(v_t)|S(v_t) \qquad \textit{Value Terms,}$$

where $u \in V$*.*

We use u, v, x, y, z to denote value variables, that is elements of V. We use v_t, n, m to denote variables for value terms; also of course $0, S(0)$ and $S(n), P(0)$ and $P(n)$ with the notation $S^n(0)$ meaning $\overbrace{S(\ldots S(0))}^{n}$.

Definition 172 (The set of simple types) *The set of* simple types $\mathcal{T}$ *is constructed from a set of* predicate constants $\mathbf{P}$*, each with an associated arity* n*, and value terms, by means of a type forming operations.*

- *if* $p \in \mathbf{P}$ *of arity* n *and* $v_{t1}, \ldots v_{tn}$ *are value terms, then* $p(v_{t1}, \ldots v_{tn}) \in \mathcal{T}$
- *if* $p \in \mathbf{P}$ *and has arity* n*, and* $v_{t1}, \ldots v_{tn}$ *are value terms, then* $\neg p(v_{t1}, \ldots v_{tn}) \in \mathcal{T}$
- *if* $A, B \in \mathcal{T}$ *then* $A \supset B \in \mathcal{T}$
- *if* $A, B \in \mathcal{T}$ *then* $A \wedge B \in \mathcal{T}$
- *if* $A, B \in \mathcal{T}$ *then* $A \vee B \in \mathcal{T}$
- *if* $A \in \mathcal{T}$ *then* $\exists v.A \in \mathcal{T}$ *for all* $v \in V$

- *if* $A \in \mathcal{T}$ *then* $\forall v.A \in \mathcal{T}$ *for all* $v \in V$*.*

These simple types correspond to the formulas in positive normal form of constructible duality, that do not contain intuitionistic negation or falsity, from Lemma 106.

We assume there are countably many term variables X^A *of each type* A*. We write* X^A, Y^A*, …, and we also use the* $X\colon A$ *notation. For distinct types* A *and* B *the corresponding sets of term variables are disjoint.*

The expressions (or terms) of our lambda-calculus λ_{HAZ} *are:*

$a = X^A \mid \lambda X^A.a \mid (aa') \mid \boldsymbol{\lambda} v.a \mid \boldsymbol{(} av_t \boldsymbol{)} \mid \langle a, a' \rangle \mid \pi_i a \mid k_i a \mid D_{X^A, Y^B}(a, a_1, a_2) \mid \boldsymbol{\langle} v_t\ \ a \boldsymbol{\rangle} \mid E_{v, X^A}(a, a')$	*Terms*

We use $a, b, c, \ldots,$ with primes and indices, for terms. We use π to denoted the projection function and k_i for the injection function, abstraction of terms $\lambda X^A.a$, abstraction of value terms, $\boldsymbol{\lambda} v.a$, and their respective applications (aa'), and $\boldsymbol{(}av_t\boldsymbol{)}$. We also have the disjunction D and existential E.

Every well-formed term will have a formula of constructible duality in positive normal form as its type, given by the following table. Well-formed terms may only be made from sub-terms of the appropriate type.

There are some restrictions on forming well-formed terms which we show by the typing rules. First however, we define free variables of terms.

Term	Type	Discussion
X^A	A	Variable
$(b^{A \supset B}\ a^A)$	B	Application
$\langle a^A, b^B \rangle$	$A \wedge B$	Pairing
$\lambda X^A . b^B$	$A \supset B$	Abstraction
$\pi_1 c^{A \wedge B}$	A	Projection
$\pi_2 c^{A \wedge B}$	B	Projection
$D_{X^A, Y^B}(a^{A \vee B}, b^C, c^C)$	C	Descriminator for constructive $\vee$
$k_1 c^A$	$A \vee B$	Injector
$k_2 c^B$	$A \vee B$	Injector
$E_{v_1, X^{A[v_1/v]}}(a^{\exists v.A(v)}, c^C)$	C	Existential Value
$\langle v_t, a^{A[v_t/v]} \rangle$	$\exists v.A$	Constructive Existential
$\lambda v.a^A$	$\forall v.A$	Abstraction for a Value
$(a^{\forall v.A}\ v_t)$	$A[v_t/v]$	Application of a Value

The free value variables $FV_v(a)$ and free term variables $FV_t(a)$ and free assumption variables $FV_s(a)$ of a term a, are defined inductively by the following table.

Term	Free value variables	Free term variables	Free assumption variables
0	$\emptyset$	$\emptyset$	$\emptyset$
v	$\{v\}$	$\emptyset$	$\emptyset$
$S(v_t)$	$FV_v(v_t)$	$\emptyset$	$\emptyset$
$P(v_t)$	$FV_v(v_t)$	$\emptyset$	$\emptyset$
X^A	$\emptyset$	$FV_t(A)$	$\{X\}$
$(b^{A \supset B}\ a^A)$	$FV_v(b) \cup FV_v(a)$	$FV_t(b) \cup FV_t(a)$	$FV_a(b) \cup FV_a(a)$
$\langle a^A, b^B \rangle$	$FV_v(b) \cup FV_v(a)$	$FV_t(b) \cup FV_t(a)$	$FV_a(b) \cup FV_a(a)$
$\lambda X^A . b^B$	$FV_v(b) \cup FV_v(A)$	$FV_t(b) \cup FV_t(A)$	$FV_a(b) - \{X\}$
$\pi_1 c^{A \wedge B}$	$FV_v(c)$	$FV_t(c)$	$FV_a(c)$
$\pi_2 c^{A \wedge B}$	$FV_v(c)$	$FV_t(c)$	$FV_a(c)$
$D_{X,Y}(a^A, b^B, c^C)$	$FV_v(a) \cup FV_v(b) \cup FV_v(c)$	$FV_t(a) \cup FV_t(b) \cup FV_t(c)$	$FV_a(a) \cup ((FV_a(b) \cup FV_a(c)) - \{X, Y\})$
$k_1 c^A \colon A \vee B$	$FV_v(c) \cup FV_v(B)$	$FV_t(c) \cup FV_t(B)$	$FV_a(c)$
$k_2 c^B \colon A \vee B$	$FV_v(c) \cup FV_v(B)$	$FV_t(c) \cup FV_t(A)$	$FV_a(c)$
$E_{v_1, Z}(a^{\exists v.A(v)}, c^C)$	$FV_v(a) \cup (FV_v(c) - \{v_1\})$	$FV_t(a) \cup FV_t(c)$	$FV_a(a) \cup (FV_a(c) - \{Z\})$
$\langle v_t, a^A \rangle$	$FV_v(a) \cup FV_v(v_t)$	$FV_t(a)$	$FV_a(a)$
$\lambda v.a^A$	$FV_v(a) - \{v\}$	$FV_t(a)$	$FV_a(a)$
$(a^{\forall v.A}\ v_t)$	$FV_v(a) \cup FV_v(v_t)$	$FV_t(a)$	$FV_a(a)$

Substitution is defined as replacing the free occurrences of the variable by the substituted term. Our substitution convention will be that $b[x/y]$ means that x is substituted for y everywhere y is free in b.

We associate terms with proof rules in the following way.

The term assignments below are annotated Gentzen natural deduction sequents where $\Gamma \Rightarrow a{:}\,A$ should be read as, from the open variable/assumptions pairs in Γ, we derive A with proof a, where Γ is a multiset of variables annotated with formulas, (their types), and $\Gamma, X{:}\,A$ is Γ with the single pair $X{:}\,A$ added. The rules are divided into introduction and elimination rules according to whether the main operator is being introduced or eliminated (respectively). A proof $\Gamma \Rightarrow c{:}\,C$ of a formula C must have an empty set of open assumptions i.e. $\Gamma = \emptyset$.

The rules below are the natural deduction rules in sequent form (from the background on intuitionistic logic chapter) followed by the rule for recursion and rules for equality and inequality.

When reading the term calculus consider a term $b{:}\,B$ with $FV_a(b) = \{X_1^{A_1} \ldots X_n^{A_n}\}$ as representing the deduction of $X_1{:}\,A_1, \ldots X_n{:}\,A_n, \Gamma \Rightarrow b{:}\,B$ for arbitrary multiset of variable/formula pairs Γ. Thus the term $\lambda X^A.X^A$, may denote the proof $\dfrac{Y{:}\,B, X{:}\,A \Rightarrow X{:}\,A}{Y{:}\,B \Rightarrow \lambda X.X{:}\,A \supset A}$ or $\dfrac{X{:}\,A \Rightarrow X{:}\,A}{\Rightarrow \lambda X.X{:}\,A \supset A}$. Also note that [X:A] means that that assumption may be present or absent.

$$\textbf{Axiom} \quad \Gamma, X{:}\,A \Rightarrow X{:}\,A$$

7.3.1.1 Propositional Intuitionistic Logic Rules

$$\wedge\ \text{Intro} \quad \frac{\Gamma \Rightarrow a{:}\,A \quad \Gamma' \Rightarrow b{:}\,B}{\Gamma, \Gamma' \Rightarrow \langle a, b\rangle{:}\,A \wedge B} \qquad \wedge\ \text{Elim} \quad \frac{\Gamma \Rightarrow a{:}\,A_1 \wedge A_2}{\Gamma \Rightarrow \pi_i a{:}\,A_i} \; i \in \{1,2\}$$

$$\supset\ \text{Intro} \quad \frac{[X{:}\,A], \Gamma \Rightarrow b{:}\,B}{\Gamma \Rightarrow \lambda X^A.b{:}\,A \supset B} \qquad \supset\ \text{Elim} \quad \frac{\Sigma \Rightarrow a{:}\,A \quad \Gamma \Rightarrow b{:}\,A \supset B}{\Gamma, \Sigma \Rightarrow (b\ \ a){:}\,B}$$

$$\vee\ \text{Intro} \quad \frac{\Gamma \Rightarrow b{:}\,A_i}{\Gamma \Rightarrow k_i\ b{:}\,A_1 \vee A_2} \; i \in \{1,2\}$$

$$\vee\ \text{Elim}\quad \frac{\Sigma \Rightarrow a{:}\, A \vee B \quad \Gamma, [X{:}\, A] \Rightarrow c{:}\, C \quad \Gamma', [Y{:}\, B] \Rightarrow c'{:}\, C}{\Gamma, \Gamma', \Sigma \Rightarrow D_{X^A, Y^B}(a^{A\vee B}, c^C, c'^C){:}\, C}\quad X^A \notin \Sigma \cup \Gamma' \text{ and } Y^B \notin \Sigma \cup \Gamma$$

To these rules we add the quantifier rules.

7.3.1.2 First-Order Intuitionistic Logic Rules

$$\forall\ \text{Elim}\quad \frac{\Gamma \Rightarrow a{:}\, \forall v.A}{\Gamma \Rightarrow (a\ \ n){:}\, A[n/v]} \qquad \forall\ \text{Intro}\quad \frac{\Gamma \Rightarrow a{:}\, A}{\Gamma \Rightarrow \boldsymbol{\lambda} v.a{:}\, \forall v.A}$$

If Γ does not contain v free

$$\exists\ \text{Elim}\quad \frac{\Gamma \Rightarrow a{:}\, \exists v.A, \quad \Delta, [X{:}\, A[u/v]] \Rightarrow c{:}\, C}{\Gamma, \Delta \Rightarrow E_{u, X^{A[u/v]}}(a, c){:}\, C} \qquad \exists\ \text{Intro}\quad \frac{\Gamma \Rightarrow a{:}\, A[n/v]}{\Gamma \Rightarrow \langle n, a\rangle{:}\, \exists v.A}$$

If $X^{A[u/v]} \notin \Gamma$ and $u \notin \Gamma \cup \Delta \cup \{A, C\}$

The rules above give a term assignment $T(d)$ to a natural deduction d. If d is an axiom $\Gamma, X{:}\, A \Rightarrow X{:}\, A$ then $T(d) = X^A$. In the previous remark, $FV_a(X^A) = \{X^A\}$ thus the term X^A represents a deduction $X{:}\, A, \Gamma \Rightarrow X{:}\, A$ for arbitrary Γ.

If d ends in a rule, $\dfrac{\Gamma_1 \Rightarrow a_1{:}\, A_1, \ldots, \Gamma_n \Rightarrow a_n{:}\, A_n}{\Phi(\Gamma_1, \ldots, \Gamma_n) \Rightarrow f(a_1, \ldots, a_n){:}\, \phi(A_1, \ldots, A_n)}$, and the terms assigned to the n subproofs are $b_1, \ldots, b_n$, where b_i is the term for $\Gamma_i \Rightarrow a_i{:}\, A_i$ then $T(d) = f(b_1, \ldots, b_n)$ where Φ is the transformation of premises, f is the transformation of terms and ϕ is the transformation of types.

7.3.1.3 Axioms for Arithmetic

We now have some axioms for equality and inequality. We only allow equality rules for atomic formulas for simplicity. As we are now extending our proof system, we must also extend our term system. Each new axiom adds a new term, whose type is the formula of the axiom. The free value variables of these terms are the free variables of the value terms they contain, save that v is not free in $sub_{A,v}(m, n)$ or $Nsub_{A,v}(m, n)$. There are no free term variables in these axioms.

$\Rightarrow eq(n)\colon n = n$

$\Rightarrow neq(n,m)\colon n \neq m$
if n and m are distinct ground terms
or $n = S \cdots S(m)$

$\Rightarrow tr(n,m,n')\colon n = m \wedge m = n' \supset n = n'$

$\Rightarrow ntr(n,m,n')\colon n \neq m \supset n \neq n' \vee m \neq n'$

$\Rightarrow sym(n,m)\colon n = m \supset m = n$

$\Rightarrow nsym(n,m)\colon n \neq m \supset m \neq n$

$\Rightarrow sub_{A,v}(m,n)\colon (m = n \wedge A[m/v]) \supset A[n/v]$

$\Rightarrow Nsub_{A,v}(m,n)\colon A[m/v] \wedge \neg A[n/v] \supset m \neq n$

Note: In $Nsub_{A,v}$ and $sub_{A,v}$, $A[n/v]$ is a predicate constant of arity n applied to an n tuple of value terms.

We also have the axioms that characterize the natural numbers, and their associated terms, which contain no free variables, and which have as type, the formula of the axiom.

$$\Rightarrow Z\colon \forall v.(v \neq 0 \supset S(P(v)) = v)$$

$$\Rightarrow N0\colon \forall v.S(v) \neq 0$$

$$\Rightarrow C0\colon \forall u.((\forall v.S(v) \neq u) \supset u = 0)$$

$$\Rightarrow Prec\colon \forall u,v.(S(v) = S(u) \supset v = u)$$

$$\Rightarrow Succ\colon \forall u,v.(v \neq u \supset S(v) \neq S(u))$$

We also have two new rules, Z_0 and Z_1, and the associated term forming operators Z_0 and Z_1. The free term variables of $Z_i(a)$ are the free term variables of a. The free value variables of

$Z_i(a)$ are the free value variables of a.

$$Z_0 \quad \frac{\Gamma \Rightarrow a\colon A[0/v]}{\Gamma \Rightarrow Z_0(a)\colon A[P(0)/v]}$$

$$Z_1 \quad \frac{\Gamma \Rightarrow a\colon A[v_t/v]}{\Gamma \Rightarrow Z_1(a)\colon A[P(S(v_t))/v]}$$

7.3.1.4 Recursion Rule

We have an important rule, induction, and its associated recursor term $\mathbf{r}$.

$$\frac{\Gamma \Rightarrow a\colon A[0/v] \quad \Delta \Rightarrow b\colon \forall v.A \supset A[S(v)/v]}{\Gamma, \Delta \Rightarrow \mathbf{r}abn\colon A[n/v]}$$

The free term variables of $\mathbf{r}abn$ are the free term variables of a union the free term variables of b. The free value variables of $\mathbf{r}abn$ are the union of the free value variables in a, b or n.

7.3.2 About the Axiomatization

This section concerns our choice of functions and the form of these functions in the axiomatization of arithmetic that we presented.

First we note, that these axioms concerning equality and inequality do not affect the constructive nature of proofs in any way, except for the trichotomy axiom because the trichotomy axiom contains a disjunction. Therefore, much later we shall prove that proofs in CD that use this axiom still have the disjunction property. However, now we want some assurance that all inequations that are derivable are derivable from our axioms. For this we use some results of Arai (Arai 1996).

We have added the axioms for the predecessor function to the usual rules for intuitionistic arithmetic to ensure that we can derive inequations, that is $t \neq t'$, directly. The axiom Z ensures the existence of a descriminator function, and the axiom $Prec$ ensures the existence of the predecessor. Together they are sufficient to ensure that every equation that follows from a set of equations in classical logic can be calculated directly, in our constructive formalism. This follows from Arai. Primitive recursive arithmetic is just the theory of primitive recursive functions.

The exact definition can be found in section 3.10.2 in (Troelstra and van Dalen 1988), but is not important here.

Theorem 173 (Arai (Arai 1996)) *For a finite set of equations E and an equation e if $E \vdash e$ in primitive recursive arithmetic without induction then there exists a direct computation $\mathcal{D}$ of e from E in primitive recursive arithmetic without induction with the additional rules,* $\dfrac{S(t_1) = S(t_2)}{t_1 = t_2}$ *and* $\dfrac{0 = S(t)}{e}$*,. $\mathcal{D}$ is a direct computation when it is a sequence of substitutions such that every function constant occurring in $\mathcal{D}$ occurs in $E \cup \{e\}$.*

We also need the following proposition.

Proposition 174 *Every equation in variables v_1 and v_2 and the successor function S that is not of the form*

$$\begin{array}{ll} \overbrace{S \ldots S}^{n} 0 = \overbrace{S \ldots S}^{m} 0 \quad m \neq n & \\ \overbrace{S \ldots S}^{n} v_1 = \overbrace{S \ldots S}^{m} 0 \quad m < n & \quad \textit{and symmetrically} \\ \overbrace{S \ldots S}^{n} v_1 = \overbrace{S \ldots S}^{m} v_1 \quad m \neq n & \end{array}$$

is consistent with primitive recursive arithmetic.

Proof: We consider an equation not of the above form. We have four cases, the equation contains no variables, in which case it is

$$\overbrace{S \ldots S}^{n} 0 = \overbrace{S \ldots S}^{n} 0,$$

which is valid.

If we have one occurrence of a variable, we can choose it to be v_1 without loss of generality. We may also assume that it occurs on the left. This leaves equations of the form,

$$\overbrace{S \ldots S}^{n} v_1 = \overbrace{S \ldots S}^{m} 0 \quad m \geq n.$$

Setting v_1 equal to $\overbrace{S \cdots S}^{m-n} 0$, yields a valid equation, thus equations of this form are consistent.

If we have two occurrences of the same variable, then the equation must be of the form

$$\overbrace{S\ldots S}^{n}v_1 = \overbrace{S\ldots S}^{m}v_1 \quad m = n$$

which is valid.

If we have two occurrences of different variables, the equation must be of the form

$$\overbrace{S\ldots S}^{n}v_1 = \overbrace{S\ldots S}^{m}v_2$$

We may assume that $m \geq n$ without loss of generality. Then substituting v_1 with $\overbrace{S\ldots S}^{m-n}v_2$ yields a valid equation.

Thus every equation not of one of the above forms is consistent. ∎

We therefore have every valid inequation $t \neq t'$ in just the successor as an axiom. The first and third cases are handled by *neq* the second case follows from $N0$. We now consider the equations that also mention the predecessor.

Proposition 175 *Every valid inequation in the predecessor P is derivable from a valid inequation in just the successor, using Z_0, Z_1 and substitution.*

Proof: An inequation $t \neq t'$ is valid just when the equation $t = t'$ has no models. We consider our inequation $t \neq t'$ and we reduce it to an inequation $s = s'$, that contains only occurrences of P on one side of the inequation, and where every P occurs as $\overbrace{P\cdots P}^{n}v_1$.

Suppose a P occurs somewhere not in the form $\overbrace{P\cdots P}^{n}v_1$, then it occurs as $P(0)$ or $P(S(s''))$. In the first case, using Z_0 we can derive the inequation $t[P(0)/v] \neq t'$ from $t[0/v] = t'$. In the second case we can derive $t[P(S(s''))/s''] \neq t'$ from $t = t'$.

Thus we are left with the case that P occurs only as $\overbrace{P\cdots P}^{n}v_1$. We suppose that P occurs on both sides of the inequation. Then the inequation is of one of the forms, where we assume that $m \leq n$ without loss of generality.

$$t[\overbrace{P\cdots P}^{n}v_1] \neq t'[\overbrace{P\cdots P}^{m}v_1]$$
$$t[\overbrace{P\cdots P}^{n}v_1] \neq t'[\overbrace{P\cdots P}^{m}v_2]$$

In the first case, we can derive this from

$$t[\overbrace{P \cdots P}^{n-m} v_1] \neq t'[v_1],$$

by substitution.

In the second case the inequation is not valid, as can be seen by substituting in 0 for v_2 and $\overbrace{P \cdots P}^{length(t)+length(t')}$ for v_1.

∎

Therefore we can derive all valid inequations. We now show, using Arai's result that we can derive all valid sequents of the form $E, f' \Rightarrow f$, where f, f' are inequations, and E is a set of equations in the formalization of arithmetic in this chapter. Since primitive recursive arithmetic (PRA) is very close to our system, we can systematically transform its proofs into proofs in our system.

Theorem 176 *Let E be a set of equations, and f is an inequation $t \neq t'$, written $f\colon t \neq t'$, such $E \cup \{f\}$ is stated in the functions P and S, the constant 0, and variables.*

If $E \vdash f$ is valid in primitive recursive arithmetic, then one of $E \Rightarrow f$, $E - \{e\}, \neg f \Rightarrow \neg e$, for some $e \in E$ is derivable in the formalization of arithmetic in this chapter where $\neg f\colon t = t'$.

Proof: We consider the sequent $E, \neg f \vdash$. There is a direct proof of this in primitive recursive arithmetic (PRA) by Theorem 173.

We transform this PRA proof into a proof in the axiomatization of arithmetic in this chapter of one of $E \Rightarrow f$, $E - \{e\}, \neg f \Rightarrow \neg e$ by induction of the length of proofs.

If the proof ends with $\dfrac{St = St'}{t = t'}$, then we invert this step to yield $\dfrac{t \neq t'}{St' \neq St'}$ in our system.

If the proof ends with $\dfrac{t[t_1/x] = t' \quad t_1 = t_2}{t[t_2/x] = t'}$ then we invert this to, $\dfrac{t[t_2/x] \neq t' \quad t_1 = t_2}{t[t_1/x] = t'}$ if no assumption in E occurs above $t_1 = t_2$ or to $\dfrac{t[t_2/x] = t' \quad t[t_1/x] \neq t'}{t_1 \neq t_2}$ otherwise.

If the proof ends with $\dfrac{0 = St}{e}$, we invert this to the axiom, $St \neq 0$ in our system.

Thus we invert every sub-tree that does not contain just axioms. Therefore, we invert some branch completely, yielding $\neg e$ for some $e \in E$, or yielding f, as required. ∎

These Theorems justify our dropping of decided equality $t = t' \vee t \neq t'$, as we have shown that we can derive inequations from inequations directly, rather than assume an equation and derive a contradiction.

7.3.3 Operational Semantics

The operational semantics is given by the following term conversion rules where the boxed reductions are called *Heyting arithmetic reductions.*

Definition 177 *Our one-step reductions for the λ_{HAZ} is given below.*

	Intuitionistic Logic Reductions		
$\mathbf{P_i}$	$\pi_i(\langle a_1, a_2\rangle) \rhd a_i$	$\mathbf{L}$	$(\lambda X.a\ b) \rhd a[b/X]$
$\mathbf{V}$	$(\boldsymbol{\lambda} v.a\ n) \rhd a[n/v]$	$\mathbf{D}$	$D_{X_1,X_2}(k_i\ a, b_1, b_2) \rhd b_i[a/X_i]$
$\mathbf{E}$	$E_{v,Y}(\boldsymbol{\langle} n, a\boldsymbol{\rangle}, b) \rhd b[n/v, a/Y]$		
	Recursion Reductions		
$\mathbf{R0}$	$\mathbf{r}ab0 \rhd a$	$\mathbf{R}$	$\mathbf{r}abSn \rhd ((bn)\mathbf{r}abn)$

Z Reductions

$$\mathbf{Z_0}\quad \mathbf{r}ab\overbrace{P\cdots P}^{n}(0) \rhd \overbrace{Z_0\cdots Z_0}^{n}(\mathbf{r}ab0) \qquad \mathbf{Z_1}\quad \mathbf{r}abv_t[P(S(n))] \rhd Z_1(\mathbf{r}abv_t[n])$$

The Z_0 reduction is defined to introduce n Z_0's, and eliminate n occurrences of P, as it is just as awkward to define the reduction that removes one occurrence of P and adds one Z_0. This would be, $\mathbf{r}ab\overbrace{P\cdots P}^{n}(0) \rhd \overbrace{Z_0\cdots Z_0}^{n-1}(\mathbf{r}ab\overbrace{P\cdots P}^{n-1}(0)$. *We cannot write* $\mathbf{r}abP(v_t) \rhd Z_0(\mathbf{r}abv_t)$, *as the reduction should only take place when v_t is the predecessor function repeatedly applied to* 0.

Note: We are not considering either permutation conversions nor simplification conversions, nor intuitionistic falsity conversions.

Definition 178 *A* reduction sequence *is a (finite or infinite) sequence of pairs (t_0, δ_0) , (t_1, δ_1) … where δ_i is an occurrence of a redex in t_i and $t_i \rhd t_{i+1}$ via the contraction of δ_i for all i. This sequence is usually written as $t_0 \rhd^{\delta_0} t_1 \rhd^{\delta_1} \ldots$.*

Finite reduction sequences are partially ordered by their prefix relation. The collection of all reduction sequences starting from t is the reduction tree *(and is actually a tree).*

Definition 179 *If $t \rhd s$ then we say that t* contracts *to s; t is called a* redex *or a* contractible term.

The replacement of t by s is called a contraction *(of t). We write $t \rhd s$ to mean the result of applying a contraction and $t \rhd_* s$ to mean that there is a sequence of contractions from t ending in s.*

Without confusion, we will now denote $\rhd$ as the congruence closure of $\rhd$ so that $t \rhd s$ means that some sub-term of t contracts.

We generate an equivalence relation from the contraction relation and denote this as $t =_{\rhd} s$ which means either $t \rhd_* s$ or $s \rhd_* t$.

Definition 180 *A term t is in* normal form *if t does not contain a redex. We say that t* has a normal form *is there is a normal s such that $t \rhd_* s$.*

Definition 181 *A term is* strongly normalizing *if its reduction tree is finite.*

Definition 182 *A relation R is said to be* confluent *or have the* Church-Rosser *property if whenever tRs_1 and tRs_2 then there exists a u such that s_1Ru and s_2Ru. A relation R is said to be* weakly confluent *or have the* weak Church-Rosser *property if whenever tRs_1 and tRs_2 then there exists a u such that s_1R_*u and s_2R_*u where R_* is the reflexive transitive closure of R.*

7.3.3.1 Confluent and Strongly Normalizing

These two lemmas are standard and can be found in (Troelstra and Schwichtenberg 1996) pg. 11.

Lemma 183 *For a confluent reduction relation $\rhd$, the normal forms of terms are unique. Furthermore, if $\rhd$ is a confluent reduction relation then we have: $t =_{\rhd} t' \iff$ there is a term s such that $t \rhd_* s$ and $t' \rhd_* s$.*

Lemma 184 *Let $\succ_*$ be the reflexive transitive closure of the congruence closure of $\succ$ and let $\succ$ be weakly confluent.*

1. *The normal form with respect to $\succ$ of a strongly normalizing term t is unique.*

2. *If all terms are strongly normalizing with respect to $\succ$ then the relation $\succ_*$ is confluent.*

Lemma 185 (Term Substitutivity) *For the reduction relation* $\rhd$ *for* λ_{HAZ}*, we have if* $t \rhd s$ *then* $t[a/X] \rhd s[a/X]$.

Proof: We consider one Heyting reduction, and show that after substitution, the reduction may still be applied. We then consider our new reductions Z_0 and Z_1.

- $\mathbf{P_i}$.

  Let t contain $\pi_i(\langle a_1, a_2\rangle)$, that is $t = t'[\pi_i(\langle a_1, a_2\rangle)/Y]$, for some term t' with a new variable Y that occurs only once. Then $s = t'[a_i/Y]$. We now consider substituting a term a for occurrences of X. We then have, $t[a/X] = t'[a/X][\pi_i(\langle a_1[a/X], a_2[a/X]\rangle)/Y]$.

  We can assume that a does not contain Y, as we chose Y to be new. Now, the reduction $\mathbf{P_i}$ is applicable to $t[a/X]$, as $t'[a/X]$ contains an occurrence of Y, and therefore $t'[a/X][\pi_i(\langle a_1[a/X], a_2[a/X]\rangle)/Y]$ contains a term of the form $\pi_i(\langle a_1', a_2'\rangle)$. In fact, the terms a_1' and a_2' can be chosen to be $a_1[a/X]$ and $a_2[a/X]$.

- Z_0 Let t contain $\mathbf{r}ab\overbrace{P\cdots P}^{n}(0)$, that is there is a term t' with one occurrence of a new variable Y such that $t = t'[\mathbf{r}ab\overbrace{P\cdots P}^{n}(0)/Y]$. We now consider substituting a term a for occurrences of X. We then have, $t[a/X] = t'[a/X][\mathbf{r}ab\overbrace{P\cdots P}^{n}(0)/Y]$. We can assume that a does not contain Y, as we chose Y to be new. Now, the reduction $\mathbf{Z_0}$ is applicable to $t[a/X]$, as $t'[a/X]$ contains an occurrence of Y, and therefore $t[a/X] = t'[a/X][\mathbf{r}ab\overbrace{P\cdots P}^{n}(0)/Y]$ contains a term of the form $\mathbf{r}a'b'\overbrace{P\cdots P}^{n}(0)$. In fact, the terms a' and b' can be chosen to be $a'[a/X]$ and $b'[a/X]$.

- Z_1 Let t contain $\mathbf{r}abv_t[P(S(n))]$, that is there is a term t' with one occurrence of a new variable Y such that $t = t'[\mathbf{r}abv_t[P(S(n))]]$. We now consider substituting a term a for occurrences of X. We then have, $t[a/X] = t'[a/X][\mathbf{r}abv_t[P(S(n))]]$. We can assume that a does not contain Y, as we chose Y to be new. Now, the reduction $\mathbf{Z_1}$ is applicable to $t[a/X]$, as $t'[a/X]$ contains an occurrence of Y, and therefore $t[a/X] = t'[a/X][\mathbf{r}ab\ v_t[P(S(n))]]$ contains a term of the form $\mathbf{r}a'b'v_t'[P(S(n'))]$. In fact, the terms a' and b' can be chosen to be $a'[a/X]$ and $b'[a/X]$, and v_t' and n' can be taken to be v_t and n.

Thus substitution of terms for term variables preserve all reductions.

∎

Lemma 186 (Value Term Substitutivity) *For the reduction relation $\rhd$ for λ_{HAZ}, we have if $t \rhd s$ then $t[v_t/v] \rhd s[v_t/v]$.*

Proof: We consider one Heyting reduction, and show that after substitution, the reduction may still be applied. We then consider our new reductions Z_0 and Z_1.

- $\mathbf{P_i}$. Let t contain $\pi_i(\langle a_1, a_2\rangle)$, that is $t = t'[\pi_i(\langle a_1, a_2\rangle)/Y]$, for some term t' with a new variable Y. The $s = t'[a_i/Y]$. We now consider substituting a value term v_t for occurrences of v. We then have, $t[a/X] = t'[v_t/v][\pi_i(\langle a_1[v_t/v], a_2[v_t/v]\rangle)/Y]$. . Now, the reduction $\mathbf{P_i}$ is applicable to $t[v_t/v]$, as $t'[v_t/v]$ contains an occurrence of Y, and therefore $t'[v_t/v][\pi_i(\langle a_1[v_t/v], a_2[v_t/v]\rangle)/Y]$ contains a term of the form $\pi_i(\langle a'_1, a'_2\rangle)$. In fact, the terms a'_1 and a'_2 can be chosen to be $a_1[v_t/v]$ and $a_2[v_t/v]$.

- Z_0 Let t contain $\mathbf{r}ab\overbrace{P\cdots P}^{n}(0)$, that is there is a term t' with one occurrence of a new variable Y such that $t = t'[\mathbf{r}ab\overbrace{P\cdots P}^{n}(0)/Y]$. We now consider substituting a term v_t for occurrences of v. We then have, $t[v_t/v] = t'[v_t/v][\mathbf{r}ab\overbrace{P\cdots P}^{n}(0)/Y]$. Now, the reduction $\mathbf{Z_0}$ is applicable to $t[v_t/v]$, as $t'[v_t/v]$ contains an occurrence of Y, and therefore $t[v_t/v] = t'[v_t/v][\mathbf{r}ab\overbrace{P\cdots P}^{n}(0)/Y]$ contains a term of the form $\mathbf{r}a'b'\overbrace{P\cdots P}^{n}(0)$. In fact, the terms a' and b' can be chosen to be $a'[v_t/v]$ and $b'[v_t/v]$.

- Z_1 Let t contain $\mathbf{r}abv'_t[P(S(n))]$, that is there is a term t' with one occurrence of a new variable Y such that $t = t'[\mathbf{r}abv'_t[P(S(n))]/Y]$. We now consider substituting a term v_t for occurrences of v. We then have, $t[v_t/v] = t'[v_t/v][\mathbf{r}abv'_t[P(S(n))]/Y]$. Now, the reduction $\mathbf{Z_1}$ is applicable to $t[v_t/v]$, as $t'[v_t/v]$ contains an occurrence of v, and therefore $t[v_t/v] = t'[v_t/v][\mathbf{r}ab\ v'_t[P(S(n))]/Y]$ contains a term of the form $\mathbf{r}a'b'v''_t[P(S(n'))]$. In fact, the terms a' and b' can be chosen to be $a'[v_t/v]$ and $b'[v_t/v]$, where $v''_t = v'_t[v_t/v]$ and $n' = n[v_t/v]$.

Thus substitution of value terms for value variables preserve reductions.

∎

Lemma 187 *The reduction relation $\rhd$ for λ_{HAZ} is weakly confluent.*

Proof: The Heyting arithmetic reductions are weakly confluent (Troelstra 1973), and we show that the addition of the Z reductions

$$\mathbf{Z_0}\quad \mathbf{r}ab\overbrace{P\cdots P}^{n}(0) \rhd \overbrace{Z_0\cdots Z_0}^{n}(\mathbf{r}ab0) \qquad \mathbf{Z_1}\quad \mathbf{r}abv_t[P(S(n))] \rhd Z_1(\mathbf{r}abv_t[n])$$

preserves this property.

Note that for any two distinct sub-terms either one is contained in the other or they are disjoint. Therefore, we show this lemma by cases:

$\mathbf{P_i}$ We first assume $t_1 \overset{Z_0}{\rhd} t_2$ and $t_1 \overset{\mathbf{P_1}}{\rhd} t_3$ and now we exhibit the t_4 such that $t_2 \rhd_* t_4$ and $t_3 \rhd_* t_4$. Remember that $\rhd$ is the congruence closure so this one step contraction need not be at the top level.

If $t_1 \overset{Z_0}{\rhd} t_2$ then $t_1[\overbrace{Z_0\cdots Z_0}^{n}(\mathbf{r}ab0)/\mathbf{r}ab\overbrace{\mathcal{P}\cdots P}^{n}(0)] = t_2$ and $t_3 = t_1[a_1/\pi_1(\langle a_1, a_2\rangle)]$, where this substitution is read as substituting a designated occurrence of the redex term, then either

1. $\pi_1(\langle a_1, a_2\rangle)$ occurs in term a or b.
2. $\mathbf{r}ab\overbrace{\mathcal{P}\cdots P}^{n}(0)$ occurs in term a_1
3. $\mathbf{r}ab\overbrace{\mathcal{P}\cdots P}^{n}(0)$ occurs in term a_2
4. none of the above

For case (1), we may take t_4 as $t_2[a_1/\pi_1(\langle a_1, a_2\rangle)]$ as the reductions do not interfere. For case (2), we may take $t_2[a_1/\pi_1(\langle a_1, a_2\rangle)] = t_3[\overbrace{Z_0\cdots Z_0}^{n}(\mathbf{r}ab0)/\mathbf{r}ab\overbrace{\mathcal{P}\cdots P}^{n}(0)]$ as t_4. For case (3), we may take t_4 as t_3 as a_2 is discarded by the P_i reduction. For case (4), we may take $t_2[a_1/\pi_1(\langle a_1, a_2\rangle)] = t_3[\overbrace{Z_0\cdots Z_0}^{n}(\mathbf{r}ab0)/\mathbf{r}ab\overbrace{\mathcal{P}\cdots P}^{n}(0)]$ as t_4.

All other cases are similar. The $\mathbf{E}, \mathbf{R}, \mathbf{D}$ and $\mathbf{L}$ reductions can copy terms. These reductions prevent the reduction relation from being confluent.

∎

7.3.3.2 Proof Normalization

We now give the proof normalization rules.

Definition 188 (Proof Normalization/Detour Contractions)

$\mathbf{P_i}$ $\wedge$ *Contraction*

$$\frac{\dfrac{\Gamma_1 \Rightarrow a_1 \colon A_1 \quad \Gamma_2 \Rightarrow a_2 \colon A_2}{\Gamma_1, \Gamma_2 \Rightarrow \langle a_1, a_2 \rangle \colon A_1 \wedge A_2}}{\Gamma_1, \Gamma_2 \Rightarrow \pi_i(\langle a_1, a_2 \rangle) \colon A_i \quad i \in \{1,2\}} \qquad conv \quad \Gamma_i \Rightarrow a_i \colon A_i \quad i \in \{1,2\}$$

$\mathbf{D}$ $\vee$ *Contraction*

$$\frac{\dfrac{\Gamma \Rightarrow a_i \colon A_i}{\Gamma \Rightarrow k_i \ a_i \colon A_1 \vee A_2} \quad \begin{array}{c} X_1 \colon A_1, \Delta_1 \Rightarrow c_1 \colon C \\ X_2 \colon A_2, \Delta_2 \Rightarrow c_2 \colon C \end{array}}{\Gamma, \Delta_1, \Delta_2 \Rightarrow D_{X_1, X_2}(k_i \ a_i, c_1, c_2) \colon C} \qquad conv \quad \Gamma, \Delta_i \Rightarrow c_i[a_i/X_i] \colon C$$

$\mathbf{L}$ $\supset$ *Contraction*

$$\frac{\dfrac{X \colon A, \Gamma \Rightarrow b \colon B}{\Gamma \Rightarrow \lambda X.b \colon A \supset B} \quad \Sigma \Rightarrow a \colon A}{\Gamma, \Sigma \Rightarrow (\lambda X.b \ a) \colon B} \qquad conv \quad \Gamma, \Sigma \Rightarrow b[a/X] \colon B$$

$\mathbf{V}$ $\forall$ *Contraction*

$$\frac{\dfrac{\Gamma \Rightarrow a \colon A}{\Gamma \Rightarrow \boldsymbol{\lambda} v.a \colon \forall v.A}}{\Gamma \Rightarrow (\boldsymbol{\lambda} v.a \ \ n) \colon A[n/v]} \qquad conv \quad \Gamma \Rightarrow a[n/v] \colon A[n/v]$$

$\mathbf{E}$ $\exists$ *Contraction*

$$\frac{\dfrac{\Gamma \Rightarrow a \colon A[n/v]}{\Gamma \Rightarrow \langle n, a \rangle \colon \exists v.A} \quad \Delta, X \colon A[u/v] \Rightarrow c \colon C}{\Gamma, \Delta \Rightarrow E_{u,X}(\langle n, a \rangle, c) \colon C} \qquad conv \quad \Gamma \Rightarrow c[n/u, a/X] \colon C$$

$\mathbf{R}_0$ *Contraction*

$$\frac{\Gamma \Rightarrow a \colon A[0] \quad \Delta \Rightarrow b \colon \forall x.A[x] \supset A[S(x)]}{\Gamma, \Delta \Rightarrow \mathbf{r}ab0 \colon A[0]} \qquad conv \quad \Gamma \Rightarrow a \colon A[0]$$

$\mathbf{R}$ *Contraction*

$$\frac{\Gamma \Rightarrow a \colon A[0] \quad \Delta \Rightarrow b \colon \forall x.A[x] \supset A[S(x)]}{\Gamma, \Delta \Rightarrow \mathbf{r}abSn \colon A[Sn]} \qquad conv$$

$$\frac{\dfrac{\Gamma \Rightarrow a \colon A[0] \quad \Delta \Rightarrow b \colon \forall x.A[x] \supset A[S(x)]}{\Gamma, \Delta \Rightarrow \mathbf{r}abn \colon A[n]} \quad \dfrac{\Delta \Rightarrow b \colon \forall x.A[x] \supset A[Sx]}{\Delta \Rightarrow (b \ n) \colon A[n] \supset A[Sn]}}{\Gamma, \Delta \Rightarrow ((b \ n) \ \mathbf{r}abn) \colon A[Sn]}$$

$$\mathbf{Z_0}\ \textit{Conversion}\qquad \frac{\Gamma \Rightarrow a\colon A[0/v] \quad \Delta \Rightarrow b\colon \forall v.A \supset A[S(v)/v]}{\Gamma, \Delta \Rightarrow \mathbf{r}ab\,\overbrace{P \cdots P}^{n}(0)\colon A[\overbrace{P \cdots P}^{n}(0)/v]}\ \textit{conv}$$

$$\frac{\dfrac{\Gamma \Rightarrow a\colon A[0/v] \quad \Delta \Rightarrow b\colon \forall v.A \supset A[S(v)/v]}{\Gamma, \Delta \Rightarrow \mathbf{r}ab0\colon A[0/v]}}{\Gamma, \Delta \Rightarrow \overbrace{Z_0 \cdots Z_0}^{n}(\mathbf{r}ab0)\colon A[\overbrace{P \cdots P}^{n}(0)/v]}$$

$$\mathbf{Z_1}\ \textit{Conversion}\qquad \frac{\Gamma \Rightarrow a\colon A[0/v] \quad \Delta \Rightarrow b\colon \forall v.A \supset A[S(v)/v]}{\Gamma, \Delta \Rightarrow \mathbf{r}abv_t[P(S(n))]\colon A[v_t[P(S(n))]/v]}\ \textit{conv}$$

$$\frac{\dfrac{\Gamma \Rightarrow a\colon A[0/v] \quad \Delta \Rightarrow b\colon \forall v.A \supset A[S(v)/v]}{\Gamma, \Delta \Rightarrow \mathbf{r}abv_t[n]\colon A[v_t[n]/v]}}{\Gamma, \Delta \Rightarrow Z_1(\mathbf{r}abv_t[n])\colon A[v_t[P(S(n))]/v]}$$

Again from now on, we consider *conv* to be the congruence closure of the one step proof conversions above.

The Z_0 and Z_1 conversions are similar to recursion reductions, in that they do not remove a pair of rules, rather they attempt to simplify the value term that results from induction. They both push induction up the proof tree.

We can calculate with λ terms, and construct these lambda-terms from proofs and construct proofs from correctly typed lambda-terms. so we state this alignment with the Curry-Howard theorem.

7.3.3.3 Curry-Howard-De Bruijn

Theorem 189 Curry-Howard *We say that d conv d' when a sequence of contractions applied to d produce d'.*

1. *For every well-typed term t there is a deduction d such that $t = T(d)$ and d is unique up arbitrary additional weakening premises in axioms.*
2. *$T(d) \rhd T(d')$ implies that d conv d'.*
3. *d conv d' implies that $T(d) \rhd T(d')$.*
4. *d is normal if and only if $T(d)$ is normal.*

Proof: For (1), the assignment of terms to deductions is given by the sequent-style term assignments to proofs in Section 7.3.1. For (2) and (3), we use that the operational semantic reductions correspond one-to-one with the normalization steps. Lastly, both proofs contractions and term conversion terminate at the same time because of the alignment of (2) and (3). ∎

We now show that eventually we always normalize to a unique normal form.

7.4 Strong Normalization

This section follows Troelstra 1973 very closely. We first define just the notion that will make the induction go through. We identify a proof d with its term $T(d)$.

Definition 190 *A deduction d is strongly valid, $SV(d)$ if one of the following clauses applies.*

1. *The final rule is $\wedge I$, $\vee I$,$\exists I$, Z_0 or Z_1, i.e d is of the form $\langle d', d'' \rangle$, $k_i\ d'$, $\boldsymbol{\langle} n, d' \boldsymbol{\rangle}$, $Z_0(d')$, or $Z_1(d')$, and the deductions of the premises of the final rule are strongly valid, i.e d' and d'' are strongly valid.*

2. *The final rule is $\supset I$, deriving $A \supset B$, i.e. $d = \lambda X^A.{d_1}^B$, and for all strongly valid derivations d' ending in A, the proof that results from substituting the derivation d' for every open term variable with type A in d is strongly valid, i.e $d[d'/X]$ is strongly valid.*

3. *The final rule of d is $\forall I$, that is $d = \boldsymbol{\lambda} v.d'^{A[v]}$, the value variable in the premise of the rule is v, and every proof that results from replacing the value variable v with a value term v_t in the subproof d', i.e. $d'[v_t/v]$ is strongly valid.*

4. *The final rule of d is not an introduction rule, nor a Z_0 nor a Z_1 rule and*

   (a) *for every d' such that $d \rhd d'$, d' is strongly valid, or d is normal.*

   (b) *If the final rule of d is $\vee E$, that is $D_{X,Y}(d_1^{A_1 \vee A_2}, d_2, d_3)$ then the reduction tree of d_1 is finite, and d_2 and d_3 are strongly valid, and for every d' such that $d_1 \rhd^* d'$, if d'' is a subderivation above the highest occurrence of A_i traced up the derivation d' from the conclusion $A_1 \vee A_2$, then the result of replacing for every free term variable X (or Y) of type A_i by d'' in d_2 (or d_3) is strongly valid.*

(c) *If the final rule of d is $\exists E$, i.e. $E_{v,X}(d'^{\exists x.A[x]}, d'')$ the reduction tree of d' is finite, d'' is strongly valid, and if for every d_1 such that $d' \rhd^* d_1$, and for d_2 a subderivation above the highest occurrence of $A[n]$ traced up the derivation d_1 from the conclusion $\exists x.A[x]$, then the result of replacing d_2 for every free term variable X of type $A[v]$ in d'' is strongly valid.*

We then need the following extension of the Lemma 4.1.10. Troelstra pg. 288.

Lemma 191 *Let the final rule in d be an introduction rule, or Z_0 or Z_1 so that the proof d is of the form $\dfrac{\Gamma \Rightarrow d' \colon B}{\Delta \Rightarrow d \colon A}$ or $\dfrac{\Gamma \Rightarrow d' \colon B \quad \Sigma \Rightarrow d'' \colon C}{\Delta \Rightarrow d \colon A}$, and let $d, d_1, d_2, \ldots$, be a reduction sequence starting from d. Therefore, the last rule of d_i is an introduction rule, or a Z_0 or a Z_1 rule of the form $\dfrac{\Gamma \Rightarrow d_i' \colon B}{\Delta \Rightarrow d \colon A}$ or $\dfrac{\Gamma \Rightarrow d_i' \colon B \quad \Sigma \Rightarrow d_i'' \colon C}{\Delta \Rightarrow d \colon A}$ respectively. The $d', d_1', d_2' \ldots$ and $d'', d_1'', d_2'' \ldots$ are reduction sequences after omission of repetitions.*

Proof: Trivial after inspection of the various cases. ∎

Lemma 192 *Reduction preserves strong validity. If $d \rhd d'$ and d is strongly valid then d' is strongly valid*

Proof: We prove this by induction on the size of d.

If d ends in an instance of a $\wedge I$, $\vee I$, $\exists I$, $\supset I$, or $\forall I$ rule, or a Z_0 or a Z_1 rule, then we have, by the definition of strong validity, that $d = \dfrac{\Gamma_1 \Rightarrow d_1 \colon B_1 \ \ldots \Gamma_n \Rightarrow d_n \colon B_n}{\Sigma \Rightarrow d \colon A}$, where the d_i's are strongly valid. Furthermore, if $d \rhd d'$, then there are proofs d_i' such that $d_i \rhd d_i'$, and $d' = \dfrac{\Gamma_1 \Rightarrow d_1' \colon B_1 \ \ldots \Gamma_n \Rightarrow d_n' \colon B_n}{\Sigma \Rightarrow d' \colon A}$, by Lemma 191. But, by the induction hypothesis, $d_1' \ \ldots \ d_n'$ are strongly valid, thus d' is strongly valid.

If d does not end with an introduction rule, then we have by clause 4 (a) of strong validity, that for every d' such that $d \rhd d'$, d' is strongly valid, or d is normal.

Thus we have that one step reduction preserves strong validity. ∎

We prove that every strongly valid deduction has a finite reduction tree.

Lemma 193 *Every strongly valid deduction has a finite reduction tree.*

Proof: We prove this by induction on deductions.

If the final rule is an introduction rule, or a Z_1 rule or a Z_0 rule then it follows from the Lemma 191 that the induction hypothesis implies the reduction tree is finite.

If the final rule is not an introduction, then either the deduction is normal, or our induction hypothesis is $\forall d'.d \rhd d'$ implies d' has a finite reduction tree, which immediately implies the reduction tree of d is finite. ∎

Next we show that a deduction that does not end in introductions is strongly valid under substitution.

We need to first establish the following lemma.

Lemma 194 *A deduction d that does not end in an instance of an introduction rule, or Z_0 or Z_1, is strongly valid if the following conditions are satisfied.*

1. *The reduction tree of every proof of a premise is finite.*
2. *If the final rule is $\wedge E$, $\supset E$ or $\forall E$, then every proof of a premise is strongly valid.*
3. *If the final rule is induction, i.e.* $\dfrac{\Gamma \Rightarrow d' \colon A[0/v] \quad \Delta \Rightarrow d'' \colon \forall v.A \supset A[S(v)/v]}{\Gamma, \Delta \Rightarrow d \colon A[n/v]}$ *then d' is strongly valid, and for each strongly valid proof d_1 of $A[n/v]$, the proof,*
$$\dfrac{\Sigma \Rightarrow d_1 \colon A[n/v] \quad \dfrac{\Delta \Rightarrow d'' \colon \forall v.A \supset A[S(v)/v]}{\Delta \Rightarrow (n, d'') \colon A[n/v] \supset A[S(n)/v]}}{\Sigma, \Delta \Rightarrow ((n, d'')\ d_1) \colon A[S(n)/v]}$$
is strongly valid.
4. *If the final rule is $\vee E$ or $\exists E$ then the conditions in the definition of strong validity are satisfied.*

Proof: To a deduction d satisfying condition 1 (i.e finiteness of the reduction trees of its premises), we assign an induction value $(\alpha, \delta, \epsilon)$. where

- α = the number of connectives of the last line of d.
- δ = the number of nodes in the reduction trees of the proofs of the premises of the last rule.

- ϵ = the complexity of the induction term n occurring in the last rule measured in symbols, if the last rule if an induction rule, and 0 otherwise.

We assume the induction values to be ordered lexicographically.

The proof proceeds by induction on the induction value of deductions satisfying the conditions of the lemma. We have for a deduction that satisfies 1-4 of the Definition of strong validity only to verify the 4.a in the definition of strong validity, as by the fourth assumption of this lemma 4.b and 4.c hold.

If d is normal we are finished. If d is not normal, we must show that every d' such that $d \rhd^* d'$ is strongly valid.

- Case a: d' is obtained from d by replacing a *proper* redex (not the entire term) d by a contraction, or a (proper) Z_0 or a Z_1 conversion. If $d = \dfrac{\Gamma_1 \Rightarrow d_1 \colon B_1 \ \ldots \ \Gamma_i \Rightarrow d_i \colon B_i}{\Delta \Rightarrow d \colon A}$ then $d' = \dfrac{\Gamma_1 \Rightarrow d'_1 \colon B_1 \ \ldots \ \Gamma_i \Rightarrow d'_i \colon B_i}{\Delta \Rightarrow d' \colon A}$ where for some $1 \leq j \leq i$ $d_i \rhd d'_i$, and for $k \neq j$ $d_k = d'_k$. Let the induction value of d' be $(\alpha', \delta', \epsilon')$. Then $(\alpha', \delta', \epsilon') < (\alpha, \delta, \epsilon)$ since d_i is the major premise and then, if there is a reduction, $\alpha = \alpha'$ and $\delta' < \delta$.

  If d' results from a Z_0 or a Z_1 conversion, then we reduce the sum of the length of the reduction trees of the proofs of the premises of the last rule. Therefore,$\alpha = \alpha'$, and $\delta' < \delta$, as we have reduced the reduction tree by at least one step. d' satisfies the first condition of the lemma, the second and third condition are satisfied because reduction preserves strong validity from Lemma 192.

  The fourth condition is also satisfied for d'. Let d be

$$\frac{\Gamma \Rightarrow d_1 \colon B_1 \vee B_2 \quad \Delta_1, X \colon [B_1] \Rightarrow d_2 \colon A \quad \Delta_2, Y \colon [B_2] \Rightarrow d_3 \colon A}{\Gamma, \Delta_1, \Delta_2 \Rightarrow D_{X,Y}(d_1, d_2, d_3) \colon A}$$

  and let $d_2 \rhd d'_2$ and $d_3 \rhd d'_3$. Then, d'_i is strongly valid, as reductions preserve strong validity, and if $d_1 \rhd^* c$ where c contains a subderivation c' directly above the last A_i traced up through c, then, the replacement of the variable X (Y) of type B_j by c' in d_i can be reduced to the replacement of the variable X (Y) of type B_j by c' in d'_i, by one reduction. But, as from Lemma 192, one step reduction preserve strong validity, then we have shown the necessary proofs, those resulting from replacing occurrences of the assumption B_j by a particular normalized subproof of d_1 are strongly valid. Thus, we have shown that the second subpart of the fourth condition of strong normalization holds.

The third subpart is similar, i.e. $\exists E$.

- Case b: d' is a proper reduction of d or a Z conversion, but not an recursion reduction and case (a) does not apply. We first consider that d' is a proper reduction of d, then the major premises B of the last rule of d must be the conclusion of an introduction, as otherwise case (a) would apply.

  If B is a conjunction, implication, or universal quantification, then by clause two of the lemma we are given that the sub-proofs are strongly valid. The first three clauses in the definition of strong validity, respectively allow us to conclude the full derivation d is strongly valid.

  We now consider the case where we can apply a Z_0 conversion to d. That is, $d = \mathbf{r}ab\overbrace{P\ldots P}^{n}(0)$. Then $d' = \overbrace{Z_0\ldots Z_0}^{n}rab0$. From the definition of strong validity of proofs ending Z_0, it suffices to show that $rab0$ is strongly valid. This follows from the first part of case (c).

  Finally we consider the case where we can apply a Z_1 conversion to d, but no other contraction. That is, $d = \mathbf{r}abv_t[P(S(n))]$. Then $d' = Z_1(\mathbf{r}abv_t[n])$. Then, by the definition of strong validity for proof ending in Z_1, we need to show that $\mathbf{r}abv_t[n]$ is strongly valid. This follows from the second part of case (c).

- Case c: The last rule is induction (i.e. the proper reduction is induction), and case (a) does not apply. If the induction term n is 0, then by the first part of the third assumption of this lemma, the reduced proof is strongly valid. If the term is Sn, then the $d = \mathbf{r}abSn$. The proof $\mathbf{r}abn$ is strongly valid, as it has a lower induction value, $(\alpha', \delta', \epsilon')$, as $\alpha = \alpha'$ $\delta = \delta'$ and $\epsilon' + 1 = \epsilon$. Thus, by the second part of the third assumption, d is strongly valid.

This concludes the possible cases, and thus establishes the lemma. ∎

We now prove that all deductions are strongly valid under substitutions. That is, a deduction d is strongly valid under substitutions if for every substitution of terms for parameters, and each replacement of the open hypothesis in d by strongly valid deductions of those hypotheses, the resulting deduction is strongly valid.

Lemma 195 *All deductions are strongly valid under substitutions.*

Proof:

1. For axioms the lemma is trivial.
2. If the final rule is an introduction rule, or Z_0 or Z_1, then the induction step is trivial, from clauses 1-3 in the definition of strong validity.
3. If the final rule is not an introduction rule, then the induction step follows from the previous lemma as follows.

   We show that a derivation $d = \dfrac{\Gamma_1 \Rightarrow d_1 \colon B_1 \ \ldots \Gamma_n \Rightarrow d_n \colon B_n}{\Delta \Rightarrow d \colon A}$ that does not end in an introduction rule is strongly valid under substitution, if $d_1 \ \ldots d_n$ are strongly valid under substitution. Let $d*$ be any deduction obtained from d by substituting terms for parameters which are not proper parameters (that is is not used as the parameter of a $\forall I$ or a $\exists E$) in d and substituting strongly valid deductions of open hypotheses of d for these hypotheses. The d^* is of the form

   $$\frac{\Gamma_1 \Rightarrow d_1^* \colon B_1^* \ \ldots \Gamma_1 \Rightarrow d_n^* \colon B_n^*}{\Delta \Rightarrow d^* \colon A^*}.$$

   As we assumed $d_1 \ \ldots d_n$ to be strongly valid under substitution, then $d_1^* \ \ldots d_n^*$ are strongly valid. Therefore condition 2 of Lemma 194 holds. The first condition of Lemma 194 holds, as Lemma 193 proved that strongly valid proofs had finite reduction trees.

   Assumption 3 holds because of the strong validity under substitution of d_2, which yields that if the final rule is induction, the substitution of a strongly valid deduction $d' \colon A^*t$ into $d_2^* \colon \forall v.A \supset A[S(v)/v]$ is strongly valid.

   Condition 4 also holds for d^*. Assume that d_1^* reduces to $d_{1'}^*$. Since d_1^* is strongly valid, this is strongly valid, as strong validity is preserved under reductions. The subproofs of that occur just above the end formula traced upwards are strongly valid, by the definition of strong validity. Finally, our induction hypothesis gives the strong validity under substitution of d_2^*. Therefore, the replacement of assumptions in d_2^* by strongly valid proofs of those assumptions is strongly valid.

The $\vee E$ rule is similar.

∎

Theorem 196 *The lambda calculus λ_{HAZ} is strongly normalizable with respect to $\rhd$.*

Proof:

This proof follows Troelstra (Troelstra 1973) chapter IV part 1 pp. 275-298. It differs from that presentation in dropping all the falsity conversions, and the permutation conversions, and adding two new conversions Z_0 and Z_1.

Thus we needed to extend Troelstra's notion of strongly valid so that our new axioms are strongly valid. We also said that if a deduction d is strongly valid, then $\dfrac{d[0/v]}{d[P(0)/v]}$ and $\dfrac{d[v_t/v]}{d[P(S(v_t))/v]}$ are strongly valid.

We have shown by Lemma 195 that every proof is strongly valid under substitution, and by Lemma 193 that every strongly valid proof had a finite reduction tree. Combining these gives that every proof has a finite reduction tree.

∎

Theorem 197 (Confluence) *The lambda calculus λ_{HAZ} is confluent with respect to $\rhd_*$.*

Application of Lemma 184, Theorems 187 and 196.

7.5 Constructivity of Axiomatization

Given a proof that a value exists, we can extract that value uniquely from the proof. Via the Curry-Howard-De Bruijn isomorphism, we consider the proof as a program that calculates that value. The property that we can extract a value m satisfying ϕ from a proof of $\exists x.\phi(x)$ is called the *disjunction property*.

Definition 198 *The* principal formula *of an elimination rule is the succedent formula explicitly shown in the rule and containing the eliminated connective.*

The principal premise *of an elimination rule is is the one containing the principal formula.*

The main branch *of a deduction is the branch of the proof that contains the final sequent and all principal premises of consecutive elimination or Z_0 or Z_1 rules.*

Theorem 199 *Given a proof of $\Rightarrow \exists x.\phi(x)$ in the formalization of arithmetic herein, where ϕ has no free variables besides x, we can extract a number n such that $\phi(n)$.*

Further, given a proof of $\Gamma \Rightarrow \exists x.\phi(x)$, where Γ contains no existentials or disjunctions in positive position, where ϕ has no free variables besides x, we can extract a number n such that $\phi(n)$.

Proof:

First apply normalization to the proof of $\exists x.\phi(x)$. We have shown that all terms in λ_{HAZ} are strongly normalizing and by confluence the normalization of proofs terminates in a unique normal form.

We consider the last step in the normal proof $\dfrac{\mathcal{D}}{\exists x.\phi(x)}$, it is either an existential introduction, a Z_0 or Z_1 rule, or an elimination rule, or induction.

If it is an existential introduction, there is a term m with possible free variable v such that $\phi(m)$. Then $m[0/v]$ is our needed n. Every well-formed term can only contain one free variable as our only value term function is unary.

Therefore, we need to consider the five elimination rules.

If a proof ends in an elimination rule, then the main branch contains only elimination rules or Z_0 or Z_1 rules (otherwise it would not be normal).

$\wedge$ **Elimination** If the last step is an $\wedge$ elimination, then in a normal proof, it would look like

$$\frac{\Rightarrow a\colon \exists x.\phi(x) \wedge A_2}{\Rightarrow \pi_1 a\colon \exists x.\phi(x)}.$$

As the main branch contains only elimination rules, $\exists x.\phi(x) \wedge A_2$ must occur in an axiom. However, we have no axiom of this form. Thus this case cannot occur in a normalized proof.

$\supset$ **Elimination** If the last step is an $\supset$ elimination, then it would look like

$$\frac{\Rightarrow a\colon A \qquad \Rightarrow b\colon A \supset \exists x.\phi(x)}{\Rightarrow (b\ \ a)\colon \exists x.\phi(x)}$$

As the main branch contains only elimination rules, $A \supset \exists x.\phi(x)$ must occur in an axiom. However, we have no axiom of this form. Thus this case cannot occur in a normalized proof.

$\vee$ **Elimination** As the main branch contains only elimination rules, $A \vee B$ must occur in an axiom. However, we have only one axiom of this form – the trichotomy axiom. Thus the proof must look like

$$\frac{\dfrac{\Rightarrow neq(m,n)\colon m \neq n \qquad \Rightarrow ntr(m,n,m')\colon m \neq n \supset m \neq m' \vee n \neq m'}{\Rightarrow (ntr(m,n,m')\ \ neq(m,n))\colon m \neq m' \vee n \neq m'} \qquad \begin{array}{c} X\colon m \neq m' \Rightarrow c\colon \exists x.\phi(x) \\ Y\colon n \neq m' \Rightarrow c'\colon \exists x.\phi(x) \end{array}}{\Rightarrow D_{X,Y}((ntr(m,n,m')\ \ neq(m,n)), c, c')\colon \exists x.\phi(x).}$$

Where m and n are distinct. But then we can simplify this proof to

$$\Rightarrow (c[n/m'][neq(n,m)/X]\colon \exists x.\phi(x).$$

This is in normal form, as replacing a variable by a constant term $neq(n,m)$ does not allow any new reductions. We now have a proof that does not use trichotomy on the main branch. This reduces this case to the other five cases, $\supset$ $\wedge$, $\forall$, $\exists$ elimination and $\exists$ introduction.

$\forall$ **Elimination** As the main branch contains only elimination rules, $\forall y.\exists x.\phi(x)$ must occur in an axiom. However, we have no axiom of this form. Thus this case cannot occur.

$$\frac{\Rightarrow a\colon \forall v.\exists x.\phi(x)}{\Rightarrow (a\ \ n)\colon \exists x.\phi(x)[n/v]}$$

$\exists$ **Elimination** This case look as follows.

$$\frac{\Rightarrow a\colon \exists v.A, \quad X\colon A[u/v] \Rightarrow c\colon \exists x.\phi(x)}{\Rightarrow E_{u,X}(a,c)\colon \exists x.\phi(x)}$$

We note that we can only have a proof that ends in an existential elimination, if we have another smaller proof, $\Rightarrow a\colon \exists v.A$ that ends in an existential. If $\Rightarrow a\colon \exists v.A$ contains other free variables, we can replace them with 0, and still have a valid proof. Thus, we assume that we have no other free variables. Thus, we can have no proofs that have only existential eliminations on their main branch, as we have no axioms of the form $\exists v.A$.

Recursion This case looks as follows:

$$\frac{\Rightarrow \exists x.\phi(x)[0], \quad \Rightarrow \forall y.\exists x.\phi(x)[y] \supset \exists x.\phi(x)[Sy]}{\Rightarrow \exists x.\phi(x)[n]}$$

but as we have by assumption that no variables occur in the conclusion, we have that n does not contain a variable, and thus must be of the form Sm, 0, $\overbrace{P \cdots P}^{n}(0)$, or $\overbrace{P \cdots P}^{n} Sm$ for some m. The first is reducible by $\mathbf{R}$, the second by $\mathbf{R}_0$, the third by Z_0, and the fourth by Z_1. Thus there are no normal proofs of this form.

We have shown that there are no proofs that end in $\supset$, $\forall$ or $\wedge$ eliminations, nor in induction. We have shown that we can always find a proof that does not end with a $\vee$ elimination. Finally, as every proof that ends with a $\exists$ elimination must contain a subproof that ends with a different elimination, we can transform every proof of $\exists x.\phi(x)$ into one that does not end with an elimination rule.

Thus every proof $\exists x.\phi(x)$ can be transformed into one which ends in a $\exists$ introduction, or a Z_0 or Z_1 rule. The main branch cannot be a sequence of Z_0 or Z_1 rules back to an axiom, as we have no axiom with a positive occurrence of $\exists$. Therefore, we can find a sequence of Z_0 and Z_1 rules upwards from the conclusion of the proof that is terminated by an $\exists I$. We showed how to extract a value from proofs of this form, thus we can extract a value from every proof of $\exists x.\phi(x)$, by transforming it into a proof that does not end in an elimination, and extracting the value from that proof.

We note that the only assumption we needed about the axioms were that they did not contain existentials or disjunctions in positive position. Thus, the proof generalizes to the second claim of the theorem.

∎

7.6 Translations

We are now going to show how this proof system helps us with programming.

Therefore, we need a language in which to define recursive functions. We consider our language to be partial recursive equations (Feferman 1986, Kleene 1967b, McCarthy 1963a) which by the MRDP (Davis et al. 1961) theorem is Turing complete.

We give two translations from recursive equations to *CD*, one that captures the positive information (the $\exists$ translation), and another that captures the negative information (the $\forall$ translation). By reducing the Kleene normal form via a Maslov-style transformation(Maslov 1987), it suffices to define the translation for equations which have function calls at nesting depth two. We define a recursive operator over these sets of recursive equations. We align the notions of operator and model by proving that every fixed-point of the operator is a model of the recursive equations. We also show that every model of the recursive equations can be transformed into a *CD* model of our two translations. Therefore, if we can extract a value from a *CD* proof using the translations, then it is true in all *CD* models of the equations. Anything true in all *CD* models we show holds equationally and thus agrees with fixed points.

Definition 200 (Partial Recursive Equations) *A set of* partial recursive equations *on the natural numbers is a set of equations E, whose individual equations are $t = t'$ where t and t' are terms, made from a set of function symbols $\{f_i | i \in I\}$, the constant 0, the successor function S, and a countable set of variables $x_i, i \in \mathbb{N}$.*

Definition 201 *A* computation *of an equation e from a set of equations E is a proof of e from E using, reflexivity $t = t$, substitution $\dfrac{t = t'}{t[u/x] = t'[u/x]}$ where u is a term, replacement of identicals, $\dfrac{t = t' \quad t_1 = t_1'}{t[t_1/t_1'] = t'}$, and the symmetry of equality $\dfrac{t = t'}{t' = t}$ and the equations E.*

Notation: A set of partial recursive equations on the natural numbers E, implies an equation e, written $E \rightsquigarrow e$, if there is a computation of e from E. This means there is a proof of e from E.

Definition 202 *The* nesting depth *of a term is defined recursively on its structure.*

- *The nesting depth of a variable x_i is* 0*,*
- *the nesting depth of* 0 *is* 0*,*
- *the nesting depth of $S(t)$ is the nesting depth of t.*
- *The nesting depth of an n-ary function $f_i(t_1, \ldots, t_n)$ is one plus the maximum of the nesting depths of $t_1, \ldots, t_n$.*

Definition 203 *The* nesting depth *of an equation $t = t'$ is the maximum of the nesting depths of t and t'.*

The nesting depth *of a set of equations E is the maximum nesting depth of the equations in E.*

Lemma 204 (Maslov Transformation of GRE) *If E is a set of partial recursive equations on the natural numbers, then there is another set of partial recursive equations E' having nesting depth 2 such that if $E \rightsquigarrow e$, then $E' \rightsquigarrow e$ and conversely.*

Furthermore, if e has nesting depth 2, then there is a proof $E' \rightsquigarrow e$ such that every equation in the proof has at most nesting depth 2.

This transformation is standard and we can see this in the following set of equations $E = \{i(k(h(x))) = g(x), g(x) = i(j(x))\}$ we can introduce a new function symbol f such that $E' = \{i(f(x)) = z(x), f(x) = k(h(x)), z(x) = i(j(x))\}$.

This lemma justifies the use of equations with nesting depth 2, as computations with equations with greater depth can be simulated by introducing new functions. Therefore, it suffices to define the translations on sets of equations of nesting depth 2.

Definition 205 (Translations) *Let E be a set of equations where each equation is of the form $f(\bar{x}) = c$, or $f(\bar{x}) = g(\bar{x}, \bar{c})$ or $f(\bar{x}) = g(h(\bar{x}), \bar{c})$, where $\bar{x}$ is a tuple of variables, or the successor applied to variables or constants or constant terms and $\bar{c}$ is a tuple of terms.*

We define the $E^\exists$ translation of a set of equations E as follows: We introduce an $n+1$-ary relation R_g for every n-ary function symbol g, and for each equation e in E we have a formula, $e^\exists$

$$
\begin{array}{ll}
(f(\bar{x}) = c)^\exists \stackrel{def}{=} \forall \bar{x}.R_f(\bar{x}, c) & \textit{constant function} \\
(f(\bar{x}) = g(\bar{x}, \bar{c}))^\exists \stackrel{def}{=} \forall \bar{x}, z.[R_f(\bar{x}, z) \equiv R_g(\bar{x}, \bar{c}, z)] & \textit{projection} \\
(f(\bar{x}) = g(h(\bar{x}), \bar{c}))^\exists \stackrel{def}{=} \forall \bar{x}, z.(\exists y.R_g(y, \bar{c}, z) \wedge R_h(\bar{x}, y)) \supset R_f(\bar{x}, z)] & \textit{composition}
\end{array}
$$

We define another translation the $\forall$ translation, for the same equations as follows,

$$
\begin{array}{l}
(f(\bar{x}) = c)^\forall \stackrel{def}{=} \forall \bar{x}, z.[\neg R_f(\bar{x}, z) \equiv z \neq c] \\
(f(\bar{x}) = g(\bar{x}, \bar{c}))^\forall \stackrel{def}{=} \forall \bar{x}, z.[\neg R_f(\bar{x}, z) \equiv \neg R_g(\bar{x}, \bar{c}, z)] \\
(f(\bar{x}) = g(h(\bar{x}), \bar{c}))^\forall \stackrel{def}{=} \forall \bar{x}, z.[(\forall y.\neg R_g(y, \bar{c}, z) \vee \neg R_h(\bar{x}, y)) \supset \neg R_f(\bar{x}, z)]
\end{array}
$$

where $A \equiv B$ is $A \supset B \ \wedge \ B \supset A$.

Definition 206 *A* model *of a finite set of partially recursive equations E is an assignment of partial functions to function symbols in E satisfying the equations in E.*

That is if the set of equations E is the singleton equation $f(S(x)) = 1$, then E is satisfied by the total function $\lambda x.1$ or the partial function $\lambda x.$ if $x = 0$ then $\perp$ else 1. These are the two models of E. Instead of partial functions we could once again take total functions to $\mathbb{N}_\perp$.

A partial function g can be viewed as the relation R_g which is the graph of g i.e. $\forall x, y.g(x) = y \iff R_g(x, y)$. Therefore, we can construct a model of relations $\mathbb{M}_E^R$ from a model of partial functions $\mathbb{M}_E^f$ such that

$$\mathbb{M}_E^R \models R_g(n, m) \text{ if and only if } \mathbb{M}_E^f \models g(n) = m.$$

A model assigning partial functions to function symbols, can be seen as a model of the logic CD, namely those intervals models which only have two worlds each labeled with this model.

Lemma 207 *Let E be a set of partial recursive equations.*

If $\mathbb{M}_E^f$ is a model of E then $\mathbb{M}_{CD} = \langle W, R, V \rangle$ is a model of $E^\forall$ where $W = \{w_1, w_2\}$ and $R = \{(w_1, w_1), (w_1, w_2), (w_2, w_2)\}$, and $V = \lambda w.\mathbb{M}_E^R$.

If $\mathbb{M}_E^f$ is a model of E then $\mathbb{M}_{CD}$ is a model of $E^\exists$.

Lemma 208 *Let E be a set of partial recursive equations.*

- *If $E^{\exists} \Rightarrow R_f(n,m)$ in our axiomatization of mathematics then $E \vdash f(n) = m$ equationally.*
- *If $E^{\forall} \Rightarrow \neg R_f(n,m)$ in our axiomatization of mathematics then $E \vdash f(n) \neq m$ equationally.*

Proof: *If $E^{\exists} \Rightarrow R_f(n,m)$ then in all models of $E^{\exists}$, $R_f(n,m)$ is satisfied.*

Assume that $E \not\vdash f(n) = m$. Then, for some model $\mathbb{M}_E^f$ of E, $f(n) = m$ is not satisfied. Consider $\mathbb{M}_{CD}$ constructed as above from $\mathbb{M}_E^f$; $R_f(n,m)$ is not satisfied in $\mathbb{M}_{CD}$ by construction. However, $\mathbb{M}_{CD}$ is a model of $E^{\exists}$, by Lemma 207. But by assumption $R_f(n,m)$ is true in all models of $E^{\exists}$ — a contradiction. Thus $E \vdash f(n) = m$. ∎

We shall need the following facts later.

Lemma 209 *The $\forall$ translation of a set of equations does not contain an existential quantifier, nor a disjunction in positive position.*

Lemma 210 *The $\exists$ translation of a set of equations does not contain an existential quantifier in positive position, nor a disjunction.*

Consider the set of equations we had earlier for factorial. We can translate them using the $\exists$ translation to:

$$\begin{array}{ll} R_{factorial}(0,1) & A_1 \\ \forall x,y.[(\exists z.R_{factorial}(x,z) \wedge y = z * S(x)) \supset R_{factorial}(S(x),y)] & A_2 \end{array}$$

which cheats[1] a bit because we represented $*$ as a function instead of a relation.

[1] To be completely accurate we should translate the equations $0 * x = 0$, $y * S(x) = x + y * x$, $x + 0 = x$, $x + S(y) = S(x+y)$. These are translated as

$$\begin{array}{l} \forall x,y,z.(R_*(x,0,z) \equiv z = 0) \\ \forall x,y,z.((\exists u.R_*(x,y,y) \wedge R_+(x,u,z)) \supset R_*(x,S(y),z)) \\ \forall x,y,z.R_+(x,0,z) \equiv z = x \\ \forall x,y,z.((\exists u.R_*(x,y,u) \wedge S(u) = z) \supset R_+(x,S(y),z)) \end{array}$$

The above then becomes the less readable

$$\begin{array}{l} R_{factorial}(0,1) \\ \forall x,y.[\exists z.R_{factorial}(x,z) \wedge R_*(z,S(x),y) \supset R_{factorial}(S(x),y)] \end{array}$$

We can prove that $\forall x.\exists y.R_{factorial}(x,y)$, from A_1 and A_2 using the axiomatization of arithmetic in Section 7.3.1. If we consider the term associated with this proof we find that it is a function, from natural numbers, that returns a pair. The first projection of this pair is the answer that we seek. We show the proof[2] of $\forall x.\exists y.R_{factorial}(x,y)$ below.

$$
\dfrac{\dfrac{\dfrac{R(0,1)}{\exists y.R(0,y)} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{R(z,b) \quad Sz*b=Sz*b}{R(z,b)\wedge Sz*b=Sz*b}}{\exists z'.R(z,z')\wedge Sz*b=Sz*z'} \quad \dfrac{\dfrac{\forall x,y.\exists z'.R(x,z')\wedge y=Sx*z' \supset R(Sx,y)}{\forall y.\exists z'.R(z,z')\wedge y=Sz*z' \supset R(Sz,y)}}{\exists z'.R(z,z')\wedge Sz*b=Sz*z' \supset R(Sz,Sz*b)}}{R(Sz,Sz*b)}}{\exists y.R(Sz,y) \quad \exists y.R(z,y)}}{\exists y.R(Sz,y)}}{\exists y.R(z,y) \supset \exists y.R(Sz,y)}}{\forall z(\exists y.R(z,y)) \supset \exists y.R(Sz,y)}}{\exists y.R(x,y)}}{\forall x\exists y.R(x,y)}
$$

It has as its associated term, if we assign A_1 and A_2 as the terms of the two axioms,

$$
\boldsymbol{\lambda} x.\mathbf{r}\ \langle 1, A_1\rangle\ \boldsymbol{\lambda} z.\lambda m.E_{b,X}(m^{\exists y.R(z,y)}, \langle Sz*b, (((A_2\ z)\ Sz*b)\ \langle b, \langle X^{R(z,b)}, eq(Sz*b)\rangle\rangle)\rangle)\ \ x
$$

This function, applied to 2 reduces to

$$
\langle 2, (((A_2\ 1)\ 2)\ \langle 1, \langle(((A_2\ 0)\ 1)\ \langle 1, \langle A_1\ eq(1)\rangle\rangle), eq(2)\rangle\rangle)\rangle.
$$

Its first projection is 2, and thus, as was claimed, factorial$(2) = 2$.

With the translation of the equation $(f(x) = 0 * f(x))^{\exists}$, we cannot prove prove $R_f(n,0)$ for any n. This failure is because the least fixed point of this equation is undefined.

In general, given a set of equations recursively defining a function f, we can prove $R_f(n,m)$ in our axiomatization if and only if the least fixed point of the associated operator Φ of the recursive definition of f is defined at n and equal to m.

[2] As we translate $*$ into a relation, we should replace $y = Sx*z'$ by $R_*(Sx,z',y)$ in the first line, $Sz*b = Sz*z'$ by $\exists u,v.R_*(Sz,z,u)\wedge R_*(Sz,b,v)\wedge u=v$, and we should derive $\exists z'.R(z,z')\wedge\exists u,v.R_*(Sz,z,u)\wedge R_*(Sz,b,v)\wedge u = v$ rather than $\exists z'.R(z,z')\wedge Sz*b = Sz*z'$. We can prove this from $\forall z,z'.\exists u,v.R_*(z,z',u)\wedge R_*(z,z',v)\wedge u=v$, the statement that $*$ is a total function. The proof of this is simple, and does not help the understanding of the present issues, so we avoid complications by leaving $*$ a function. In general, we can leave total functions as functions.

Thus the values that we can prove are defined in the positive normal form of CD correspond to the least fixed point.

We now define how to construct an operator Φ from a set of defining equations in order to state the alignment of least-fixed points and extracted values from an existence proofs.

We take only equations which are mutually recursive in the style

$$\begin{array}{lcl} f_1(\overline{x}, 0) & = & g_0^1(\overline{x}) \\ f_1(\overline{x}, S(y)) & = & g_s^1(\overline{x}, y, f_1(\overline{x}, y)) \\ \vdots & = & \vdots \\ f_n(\overline{x}, 0) & = & g_0^n(\overline{x}) \\ f_n(\overline{x}, S(y)) & = & g_s^n(\overline{x}, y, f_n(\overline{x}, y)) \end{array}$$

where g_0^i and g_s^i are defined by composition of functions from $\{f_1, \ldots, f_n\} - \{f_i\}$ and the successor and 0.

Lemma 211 *(Kleene 1967a) Every partial recursive function can be defined by a mutually recursive set of functions.*

In this chapter, we call such definitions of partial recursive functions by mutually recursive sets of equations, mutually recursive equations.

Notation and Discussion In this chapter, we are going to consider only sets of mutually recursive equations that are *consistent* i.e. have a standard model where function symbols are interpreted as total functions on the naturals.

When the optimal fixed point is a total function, then it is the greatest fixed point. We have presented our intuitions in terms of optimal fixed points rather than in terms of greatest fixed points for several reasons.

Greatest fixed points have been considered recently, by Aczel (1988), Milner and Tofte (1991), Barwise and Moss (1996) and others (Moss and Danner 1996, Gordon 1995, Pitts 1994). Aczel considers when the greatest fixed point of an operator is a set. Milner considers how a principle of co-induction can be used to prove the consistency of functional programming languages, while Barwise and Moss consider non-wellfounded phenomena, where greatest fixed points are natural.

Rather than present our work in terms of this approach, we choose to present implicit programming in the style of Manna and Shamir's work. There are three reasons why we have chosen this manner of presentation.

1. Manna and Shamir present recursive equations and argue that traditional evaluation often gives unintuitive answers. They proposed changing how we evaluate *functions*. However, they did not propose changing the induction scheme over the natural numbers nor the program representation.

2. Work in co-induction and greatest-fixed points concerns what objects or *data-structures* exist. It suggests that new sets, that contain themselves should exist. It does not suggest a new way to evaluate functions. Our approach will not change what natural numbers exist. We do not postulate co-induction over numbers, i.e. we do not say that the co-numbers are the largest class that obeys the first three Peano axioms.

3. Least fixed points are almost universally accepted as the correct semantics for recursive programs. We develop the comparison with optimal fixed points as this is the only other approach we are aware of that suggests a different semantic interpretation of recursive functions. Other *procedural* interpretations,(e.g. lazy evaluation) have been suggested, but a direct comparison with these approaches is difficult as these approaches are motivated by procedural intuition.

We are now ready to state the correspondence between rewriting in equational logic, proofs in CD and least fixed points of operators. First however, we must generalize the notion of operator to n functions. The fixed points of an operator Φ^n are n functions such that $\Phi^n(g_1, \ldots, g_n) = g_1 \ldots g_n$.

We are going to construct an n-place operator $\Phi^n_E(f'_1, \ldots, f'_n) = h_1 \ldots h_n$ which maps n partial functions to n partial functions having the appropriate arities.

Definition 212 *Let E be a set of mutually recursive equations using symbols $\{f_1, \ldots, f_n\}$.*

For every $e \in E$ where e is $t = t'$, we define e' to be $t = t'[f'_j/f_j]$ for every $j.(1 \leq j \leq n)$. $t'[f'_j/f_j]$ denotes the substitution of the function *f'_j for the* function symbol *f_j in e. Let $E' = \bigcup_{e \in E} e'$.*

E' is a definition of a set of partial functions which we take to be the result of the operator. To elaborate: E' is a set of equations of the form:

$$\begin{array}{rcl} f_1(\overline{x}, 0) & = & g_1(\overline{x}) \\ f_1(\overline{x}, S(x)) & = & g_2(\overline{x}, x, f_1'(\overline{x}, x)) \\ & \vdots & \end{array}$$

where g_1, g_2 are made of compositions of f_i''s and successor and 0.

We now define the output functions (wlog) h_1 in terms of the given functions g_1, g_2,

$$h_1(\overline{x}, y) \stackrel{def}{=} \begin{cases} g_1(\overline{x}) & \text{if } y = 0 \\ g_2(\overline{x}, y-1, f_1'(\overline{x}, y-1)) & \text{otherwise} \end{cases}$$

Proposition 213 *If E is a finite set of mutually recursive equations, then Φ_E^n is a monotonic operator.*

In the unary case, we need to show that $f \sqsubseteq g$ implies $\Phi^1(f) \sqsubseteq \Phi^1(g)$. Our interpretations of partial functions is strict i.e. $f(\perp) = \perp$ and we constructed Φ to be an operator which mimics the computation performed equationally on mutually recursive equations.

Theorem 214 *Let E be a finite set of mutually recursive equations mentioning function symbols $\{f_1, \ldots, f_n\}$.*

$$\Phi_E^n(g_1, \ldots, g_n) = g_1 \ldots g_n \text{ if and only if } g_1 \ldots g_n \text{ are models of } E$$

That is every fixed point of Φ_E^n is a satisfying assignment to the system of equations E and vice versa.

Proof:

We first from right to left. If $g_1 \ldots g_n$ are satisfying assignments to the system of equations of E, then they are a fixed point. By definition, $g_1 \ldots g_n$ are a satisfying assignment to the equations if whenever they are substituted in for all symbols f_i, then the equations are valid. However, whenever these functions are substituted in on the left hand side, they yield $g_i = f_i'(\ldots)$. The only instantiation for values on the right-hand side that could make this equation valid is an instantiation that evaluates to g_i. But the function to which the right hand side evaluates is the definition of the output of the operator Φ_E^n. Thus the operator is the identity whenever its input is a model of the equations i.e. $g_1 \ldots g_n$ is a fixed point.

We now show from left to right. If $\Phi_E^n(g_1, \ldots, g_n) = g_1 \ldots g_n$ then $g_1, \ldots, g_n$ is a satisfying assignments to the system of equations of E. To show this it suffices to show that the instantiation of functions $g_1, \ldots, g_n$ for symbols $f_1, \ldots, f_n$ in E is valid. By definition of Φ_E^n and that assumption that g_i's are fixed points, then the instantiation of g_i's on the right yields functions which just evaluate to g_i again.

But, substituting g_i for f_i, also yields g_i on the left. Thus the instantiation of $g_1, \ldots, g_n$ for $f_1, \ldots, f_n$ in E yields valid equations, showing that $g_1, \ldots, g_n$ is a model. ∎

Theorem 215 *Let E be a consistent and ω-consistent finite set of equations mentioning function symbol f, then the least fixed point of f is computed by the term associated with a proof of*

$$\forall \bar{x} \exists y. R_f(\bar{x}, y)$$

from the sentences in $E^\exists$ using the axiomatization in Section 7.3.1.

Proof:

Lemma 210 notes that the sentences in $E^\exists$ do not contain positive occurrences of existentials or disjunctions. Thus the preconditions for Theorem 199 are satisfied. We therefore instantiate the universal and prove $\exists y. R_f(\bar{c}, y)$ and extract the value from the normalized proof by Theorem 199.

Lemma 208 says that whenever $R_f(n, m)$ is provable from the exists translation, i.e. $E^\exists \Rightarrow R_f(n, m)$, then $E \vdash f(n) = m$. However, provable corresponds to truth in all models because equational logic is complete for ground terms. E has a model by assumption so we can use Theorem 214.

Theorem 214 states that every model of the equations E corresponds to a fixed point of the operator Φ_E. The values of f that are defined in every model of E are exactly the values of f that are defined in the least model, hence defined in least fixed point of Φ_E. Therefore the extracted values are the value of the function in the least fixed point.

Did we get all values in the least fixed point, or strictly fewer? We have extracted a total function f, that agrees with the least fixed point f_l where it is defined; hence $f \sqsubseteq f_l$. Therefore, the extracted function f is the least fixed point of Φ_E. ∎

Note: For non-total least-fixed points, we need to relativize to a domain.

The usual statement that there is a unique value x that satisfies ϕ is

$$\forall x, y.[\phi(x) \wedge \phi(y) \supset x = y]$$

we can re-express this as

$$\exists x.\forall y.[x \neq y \supset \neg\phi(y)].$$

The formula implies that for any number y not equal to x, we can construct a counterexample to y being a solution of ϕ. It is this information that we need for computable optimal fixed points. As we have a logic with the disjunction property we can extract the value of x and actually compute the optimal fixed point.

However, we need the $\forall$ translation, as the $\exists$ translation of the recursive definition of factorial contains no information about when factorial is not true of an input-output pair; it is this negative information we now need.

The $\forall$ translation of $f(x) * 0 = f(x)$ is

$$\forall x, z.[\forall y.\neg R_f(x, y) \vee z \neq y * 0 \supset \neg R_f(x, z)]$$

Using the above, we can prove

$$\forall x.\exists y.\forall z.[z \neq y \supset \neg R_f(x, z)]$$

using the following proof,

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{0 = u * 0 \quad z \neq 0}{0 = u * 0 \wedge z \neq 0} \quad 0 = u * 0 \wedge z \neq 0 \supset z \neq u * 0}{z \neq u * 0}}{\neg R_f(x, z) \vee z \neq u * 0}}{\forall u. \neg R_f(x, u) \vee z \neq u * 0} \quad \dfrac{\dfrac{\forall xz.(\forall u. \neg R_f(x, u) \vee z \neq u * 0) \supset \neg R_f(x, z)}{\forall z.(\forall u. \neg R_f(x, u) \vee z \neq u * 0) \supset \neg R_f(x, z)}}{(\forall u. \neg R_f(x, u) \vee z \neq u * 0) \supset \neg R_f(x, z)}}{\neg R_f(x, z)}}{z \neq 0 \supset \neg R_f(x, z)}}{\forall z. z \neq 0 \supset \neg R_f(x, z)}}{\exists y \forall z. z \neq y \supset \neg R_f(x, z)}}{\forall x. \exists y \forall z. z \neq y \supset \neg R_f(x, z)}
$$

We can associate a term with this proof

$$\boldsymbol{\lambda} x. \boldsymbol{\langle} 0, \boldsymbol{\lambda} z. \lambda X. (\pi_2(\boldsymbol{(}(A\ x\boldsymbol{)}\ z)) \ \boldsymbol{\lambda} u. k_2\ (sub_{z \neq x, x}(0, u * 0)\ \langle \boldsymbol{(} eq_0\ u \boldsymbol{)}, X^{z \neq 0} \rangle))) \boldsymbol{\rangle}$$

where we avoid an axiomatization of multiplication be denoting $\forall x.(0 = 0 * x)$ as eq_0.

As before, to evaluate the function, we apply the term to the given value, and take the first projection. It is clear that this gives the constant zero function.

Theorem 216 *The optimal fixed point of a program f, where f is defined by a set of mutually recursive equations E, is computed by the term associated with a proof of*

$$\forall \bar{x} \exists y \forall z. z \neq y \supset \neg R_f(\bar{x}, z)$$

from the sentences in $E^{\forall}$ using our axiomatization from Section 7.3.1.

Proof:

Lemma 209 notes that the sentences in $E^{\exists}$ do not contain positive occurrences of existentials or disjunctions. Thus the preconditions for Theorem 199 are satisfied. We therefore can extract the value from the normalized proof by Theorem 199.

If $E^{\forall} \Rightarrow \forall x. x \neq m \supset \neg R_f(n, x)$, then in all models of $E^{\forall}$ this is the case.

By assumption, E has a model in which f is total and the natural numbers are standard, call it $\mathbb{M}_E^f$. From $\mathbb{M}_E^f$, we construct M_{CD} which is a model of $E^{\forall}$. Therefore, in $\mathbb{M}_{CD}$ $\forall x. x \neq m \supset \neg R_f(n, x)$ is satisfied. Therefore, M_E^f satisfies $f(n) = m$ as f is total and cannot have any other value.

By Theorem 214 the models of the equations correspond to the fixed points. Therefore, there is a fixed point f_o such that $f_o(n) = m$.

Therefore the extracted values from the proof are the values of a maximal fixed point, as they are the values drawn from a maximal (total) model.

We now show that this is the only maximal fixed point . There is no fixed point that has a different value for f at a value n than this fixed point, as we prove from the $\forall$ translation that every model does not have a different value, and the set of models agree with the set of fixed points. But, to not differ with a fixed point on all values is to be consistent with that fixed point. Thus every fixed point is consistent with our total function, and hence our function is the maximal consistent fixed point, and hence the optimal fixed point. ∎

Thus f cannot help but have y as its value when it has a value. The strong normalization property implies that when we have a constructive proof of uniqueness, then the optimal fixed point is computable, as we can effectively normalize the proof, and extract the value.

Corollary 217 *If from a set of sentences $E^\forall$ we can prove $\forall \bar{x} \exists y \forall z . z \neq y \supset \neg R_f(\bar{x}, z)$, then the optimal fixed point is computable.*

This follows because of strong normalization. Therefore, we have shown that we can extract programs that compute optimal fixed points from proofs of uniqueness, analogously to how least fixed points arise from proofs of existence.

We know that for a given function whose optimal fixed point is computable, there is an axiomatization whose proof calculates the optimal fixed point. This depends heavily on the choice of the axiomatization, in particular the strength of the induction principle.

A topic of future work is to consider the case of sets of conditional equations and sets of equations mixed with inequations.

7.6.1 Discussion

The previous two theorems consider the case where the set of recursive equations are classically consistent. If we consider the case of inconsistent equations, much less is known. If we worked in a system that contained the consistency principle of N, that is $A \wedge \neg A \supset B$, then inconsistent equations would yield all answers. In the case we have presented, analysis is more difficult as inequations need to be proved directly. For primitive recursive arithmetic Friedman has conjectured that an inequation $t \neq t'$ is provable only when there is a term t_0, such that t is $\overbrace{S \ldots S}^{m} t_0$ and $t' = \overbrace{S \ldots S}^{n} 0$ with $m > n$ or vice versa.

This means that the theory PRE can discriminate between terms only when one is 0 and the other is of the form St. Unless the equation is of the form $St = u$, where the term u occurs in t and u contains a variable, the conjecture is easily seen to hold. Also if u is a variable x, then by H. Friedman (Friedman 1975) we have $St(x) \neq x$.

This is all we know about this case.

Computer science often considers partial functions, that is programs that sometimes loop. If we know exactly when a program loops, we can be make a total program that extends it. We do this by wrapping the first program in a conditional. If the partial function f is undefined exactly when a recursive predicate P is the case, then the program $\lambda x.$ *if* $P(x)$ *then* 0 *else* $f(x)$ is total.

In this way we can reduce reasoning about partial functions, over a well defined subset of their range, to reasoning about total functions. This transformation is sometimes called relativization, analogously to relativization in logic.

7.7 Examples

We now consider two examples which show the applicability of this approach.

Example: 2 *Consider the following recursively defined function,*

$$\begin{aligned} 2 * f(0) &= g(0) \\ 2 * f(S(x)) &= g(S(x)) \\ g(0) &= f(0) \\ g(S(x)) &= f(x). \end{aligned}$$

This can more easily be written as $2 * f(x) = f(x)$*, which is in the form that the* $\exists$ *translation can be applied to.*

We cannot show,

$$\forall x \exists y. R_f(x, y),$$

from the existential translation

$$(2 * f(x) = f(x))^{\exists} \quad = \quad \forall xy.(\exists z.R_f(x, z) \wedge y = 2 * z) \supset R_f(x, y),$$

but we can prove,

$$\forall x \exists y \forall z.z \neq y \supset \neg R_f(x, z).$$

from the universal translation

$$(2 * f(x) = f(x))^{\forall} \quad = \quad \forall xy.(\forall z.\neg R_f(x, z) \vee y \neq 2 * z) \supset \neg R_f(x, y)$$

We prove this by proving,

$$\forall z.z \neq 0 \supset \neg R_f(x, z).$$

We prove this in CD using the axiomatization of arithmetic in this chapter by considering the fact that every non-zero number is the product of an odd number y and a power of 2.

$$\forall x.x \neq 0 \supset \exists yn.x = y * 2^n \wedge \forall u.y \neq 2 * u$$

Using this fact, and the assumption that $z \neq 0$, we can prove in CD, by induction on n, that $\neg R_f(x, z)$.

We now consider an example that uses co-recursively defined data-structures, e.g. lists equivalent up to bisimulation. The following is a natural program that extracts the leftmost path through a list structure. We give it as an example of how programs on co-recursive data-structures, that arise as optimal fixed points, may adjust the pointer structure. The least fixed point of this program is undefined on all lists whose leftmost branch is circular.

Example: 3 *Consider the Lisp program:*

```
(defun f(l)
  (cond ((atom l) l)
    (t (car (f l)))))
```

The above Lisp program can be written pseudo-*equationally as,*

$$f(cons(a, b)) = cons(f(a), nil)$$
$$atom(a) \rightarrow f(a) = a$$

This can be written in mutually recursive equations, over lists, as follows.

$$f(atom(a)) = a$$
$$f(cons(a, b)) = cons(f(a), nil).$$

We defined mutually recursive equations over the natural numbers, but the definition generalizes to other recursively defined data-types, such as lists.

The optimal fixed point is computable, and selects the leftmost branch. The $\forall$ translation of the above definition gives,

$$\forall xyz.(\forall u.\neg R_f(x,u) \vee cons(u,nil) \neq z) \supset \neg R_f(cons(x,y),z)$$
$$\forall xz.atom(x) \supset (\neg R_f(x,z) \equiv x \neq z)$$

From these two sentences and some basic facts about *cons* and list equality, we can prove

$$\forall x \exists y. \forall z. z \neq y \supset \neg R_f(x,z).$$

The associated lambda calculus term is a program that correctly calculates the optimal fixed point of f, that is, it correctly extracts the leftmost branch.

One extra reduction is needed to guarantee that values can be extracted from proofs of existential sentences in this axiomatization. This relates to how inequality of lists is defined in terms of bisimulation.

7.8 Conclusion

Programs are naturally represented using recursive definitions of functions (McCarthy 1963a). These sets of equations can be given a least fixed point semantics (Scott and Strachey 1971). Least fixed point semantics suggest that a function should be undefined as often as possible.

Another possible semantics is the optimal fixed point (Manna and Shamir 1977). This suggests that if a function could only have one value, then we should choose that value, rather than choose undefined. The optimal fixed point is always as defined as the least fixed point, and thus can extract more content from our programs.

The optimal fixed point therefore seems like a much better choice of semantics. There remains but one obstacle, sometimes the optimal fixed point is not computable.

We avoid this pitfall by looking at the problem in another way, using the Curry-Howard-De Bruijn isomorphism. This relates programs in a lambda calculus with proof normalization in a constructive logic. The computation of the least fixed point, expressed in lambda calculus, gives a proof that the function is defined. We show that a proof that the function is uniquely defined corresponds to a computation of the optimal fixed point.

The constructive logic that we use is the positive normal form of constructible duality, and the lambda calculus is a variant of that used for normalizing intuitionistic proofs. We show that the calculus is confluent and strongly normalizable, and that it correctly computes the optimal fixed points.

Our approach gives a way of computing the optimal fixed point of recursive equations. This allows a computational interpretation of optimal fixed points, improving over the commonly used least fixed point.

Optimal fixed points are especially useful for co-recursively defined data-structures, but can differ from the least fixed points even on recursively defined data-structures like the natural numbers.

Chapter 8

Technical Summary

We presented an investigation of duality in the traditional logical manner. We extended Nelson's symmetrization of intuitionistic logic, constructible falsity, to a self-dual logic — constructible duality. We developed a self-dual model theory by considering an interval of worlds in an intuitionistic Kripke model. The duality arose through how we judged truth and falsity. Truth was judged forward in the Kripke model, as in intuitionistic logic, while falsity was judged backwards.

We developed a self-dual algebra such that every point in the algebra was representable by some formula in the logic. This algebra arises as an instantiation of a Heyting algebra into several categorical constructions. In particular, we showed that this algebra is an instantiation of the **Chu** construction applied to a Heyting algebra, the second Dialectica construction applied to a Heyting algebra, and as a posetal case of a suggestion for the study of recursion and corecursion. Thus the algebra provided a common base for these constructions, and suggested itself as a necessary part of any logical treatment of constructive duality.

Implicit programming was suggested as a new paradigm for computing with constructible duality as its formal system. We showed that proofs of definedness have terms associated with those proofs which compute the least fixed point and proofs of uniqueness have terms associated with those proofs which computes the optimal fixed point. When the proof method succeeds in calculating an output for every input, it is really the greatest fixed point being calculated because the optimal fixed point is the greatest fixed point when the greatest fixed point exists.

8.1 Constructive Duality

To understand constructive duality we needed to examine a notion of an implication which forms a congruence on equivalence classes of formulas from constructible falsity. This new implication, denoted

$\multimap$ is $\varphi \multimap \psi \equiv \varphi \supset \psi \ \wedge \ \neg\psi \supset \neg\varphi$. However, one important property of the intuitionistic implication, $\supset$, is how it interacts with intuitionistic conjunction, $\wedge$. Namely, $\varphi \supset (\psi \supset \chi) \equiv (\varphi \wedge \psi) \supset \chi$ i.e. $\supset$ is the residual of $\wedge$.

We defined a conjunction $\otimes$ which is the residual of this substitutable implication $\multimap$ for which we have deduction theorem i.e. $A \otimes B \multimap C = A \multimap (B \multimap C)$.

The unit for $\otimes$ is $\mathbf{I}$ which is as we stated before the *overdecided* truth value. This is the point where every propositional atom has been decided as both "true" and "false". We also added Thomason's $\perp$ — the *undecided* truth value where no propositional atom has been assigned either "true" or "false".

These units $\perp$ and I are related to the units of truth $\mathbf{1}$ and the unit for falsity $\mathbf{0}$ and each other via the following diagram ordered for information.

$$
\begin{tikzcd}
 & \mathbf{I} & \\
\mathbf{1} \arrow[ur] & & \mathbf{O} \arrow[ul] \\
 & \perp \arrow[ul] \arrow[ur] &
\end{tikzcd}
$$

8.2 The Model Theory

We gave a Kripke model having only two truth values, 0 and 1, for which we showed that constructible duality is sound and complete. Truth was defined on pairs (w_t, w_f) in the Kripke structure where w_f and w_t do not necessarily stand in any relation to each other. Truth is judged from one end w_t, falsity from the other w_f. We interpret strong negation as dualizing the Kripke structure with respect to both truth and accessibility.

The atoms that Thomason assigns the third value $\perp$ are those propositional atoms which are assigned 0 at w_t and 1 at w_f. The atoms that in the logic should be assigned $\mathbf{I}$ are those propositional atoms which are assigned 1 at w_t and 0 at w_f.

This model theory can be constrained to intervals to give a sound and complete model theory for constructible falsity.

8.3 The Exponentials

We suggested that exponentials be regarded as logic transformers. Classical or intuitionistic logic can be recaptured from the paraconsistent logics. We considered exponentials that mapped paraconsistent logics onto consistent ones.

Consistent logics appear as certain syntactic subsets of paraconsistent logics. We considered this embedding algebraically and see that an exponential, which maps a paraconsistent logic to a consistent logic, is both a *projection* and an *interior* operation.

Another exponential ¡ is definable in constructible duality which has no linear logic counterpart. This is also a projection and an interior operator. Closure and interior operations can be thought of as approximations. Thus consistent logics approximate paraconsistent ones.

8.4 The Algebra

The algebra for constructible duality has an onto representation theorem for $H \times H^{op}$; hence it is sound and complete for $H \times H^{op}$. $H \times H^{op}$ is also constructed via the Lindenbaum construction for the logic of constructible duality. Furthermore the free Heyting algebra generated by P, denoted H_f, every point in $H_f \times H_f^{op}$ is representable as some formula in constructible duality and $H_f \times H_f^{op}$ is the free algebra for constructible duality.

We showed that the algebra of constructible duality is a model for linear logic and that the exponentials in constructible duality satisfy both the logical and the algebraic definitions of exponentials in linear logic.

This algebra theory can be constrained to disjoint pairs to give a sound and complete model theory for constructible falsity.

8.5 Linear Logic

We have studied a logic *constructible duality* which is a conservative extension of intuitionistic logic and showed it has operators which we may interpret as multiplicatives while the operators from intuitionistic logic can be interpreted as the additives. Therefore, it is natural to compare this logic to linear logic.

We showed that the linear fragment of multiplicative constructible duality agrees with multiplicative linear logic.

8.6 Computational Interpretation

The Curry-Howard isomorphism between intuitionistic formulas and typed lambda-calculus terms carries over to an isomorphism between cut-elimination (or proof normalization) and lambda-calculus reductions. This means that proof normalization is isomorphic to typed-lambda calculus reductions. This is the "proofs as computation" interpretation.

We presented a natural deduction system for constructible duality. A natural deduction system is a system of proof rules which allows for a notion of proof equivalence. These proof systems play an important role in the Curry-Howard interpretation because β equivalence of λ-Calculus terms correspond to equivalence classes of proofs.

We presented a Curry-Howard isomorphism between formulas in constructible duality in positive normal form and the λ_{HAZ}-calculus.

8.7 Implicit Programming

If a program has a unique solution then programming semantics should evaluate to that solution. Optimal fixed points represent this unique solution; that is, if the only choice is between undefined and a specific defined value then we choose the defined value. However, optimal fixed points can be uncomputable. For this reason they have been relatively underutilized.

Here we provided a proof method for computing the optimal fixed point. If a proof of constructive uniqueness can be found, then the term associated with that proof computes the optimal fixed point. To do this we used proofs of uniqueness in constructible duality and terms in the λ_{HAZ} calculus.

We showed that reductions in the λ_{HAZ} calculus were proof normalizations in *CD* giving us an analogue of the Curry-Howard-De Bruijn isomorphism. Using this correspondence, we extracted programs from these proofs by using the terms of the λ_{HAZ} calculus which computes the optimal fixed point of a recursively defined function. When this uniqueness proof exists, the optimal fixed point is computable.

We gave several examples of where this fixed point gives intuitive answers. This new semantics is especially natural for functions on co-recursively defined data-types, like streams or circular lists.

Bibliography

Abramsky, S. 1993. Computational interpretations of linear logic. *Theoretical Computer Science* 111:3–57.

Abramsky, S., and R. Jagadeesan. 1994. New foundations for the geometry of interaction. *Information and Computation* 111(1):53–119.

Aczel, P. 1988. *Non-Well-Founded Sets*. Vol. 14 of *Lecture Notes*. CSLI.

Akama, S. 1988. On the proof method for constructive falsity. *Z. Math. Logik Grundlag. Math.* 34.

Arai, T. 1996. From the attic. Presented Notes at a Logic Lunch – Stanford, May.

Barr, M. 1979. $*$-*Autonomous categories, LNM 752*. Springer-Verlag.

Barwise, J., and L. Moss. 1996. *Vicious Circles*. CSLI.

Beth, E. 1952. *The Foundations of Mathematics*. North-Holland.

Birkhoff, G. 1933. On the combination of subalgebras. *Proc. Cambridge Phil. Soc* 29:441–464.

Birkhoff, G. 1948. *Lattice Theory*. Vol. 25. A.M.S. Colloq. Publications.

Birkhoff, G. 1967. *Lattice Theory*. Vol. 25. A.M.S. Colloq. Publications.

Blute, R. 1991. Proof nets and coherence theorems. In S. Abramsky, P.-L. Curien, A. M. Pitt, D. H. Pitts, A. Poigné, and D. E. Rydeheard (Eds.), *Category Theory and Computer Science*, Springer-Verlag LNCS 530, 121–137, Paris, France, September.

Blute, R. 1992. *-autonomous hyperdoctrines and predicate linear logic. Manuscript.

Blute, R. 1993. Linear logic, coherence and dinaturality. *Theoretical Computer Science* 115:3–41.

Brouwer, L. 1928. Intuitionistische Betrachtungen über den Formalismus. *Koninklijke Akademie van wetenschappen te Amsterdam, Proceedings of the section of sciences* 31:374–379. In *From Frege to Gödel*, Jean van Heijenoort (editor), Harvard, 1967, pages 490–492.

Carroll, L. 1872. *Through the Looking Glass*. Grosset & Dunlap.

Chen, T. 1996. On the structural properties of the set of fixpoints for nondeterministic recursive definitions. *J. of Computer and System Sciences* 52:80–86.

Cubric, D. 1994. Interpolation property for bicartesian closed categories. *Archive for Mathematical Logic* 33:291–319.

Cutland, N. 1980. *Computability: An introduction to recursive function theory.* Cambridge University Press.

Czelakowski, J. 1975. Logics bases on partial Bolean σ-agebras. *Studia Logica* XXXIII(4):371–396.

Davis, M., H. Putnam, and J. Robinson. 1961. The decision problem for exponential diophantine equations. *Annals of Math.* 74(2):425–436.

de Bruijn, N. 1970. The mathematical language AUTOMATH, its usage, and some of its extensions. In M. Laudet, D. Lacombe, D. Nolin, and M. Schützenberger (Eds.), *Symposium on Automatic Demonstration*, 29–61. Springer-Berlin.

de Paiva, V. 1989a. The dialectica categories. In *Categories in Computer Science and Logic*, Vol. 92 of *Contemporary Mathematics*, 47–62, held June 1987, Boulder, Colorado.

de Paiva, V. 1989b. A dialectica-like model of linear logic. In *Proc. Conf. on Category Theory and Computer Science, LNCS 389*, 341–356, Manchester, September. Springer-Verlag.

de Paiva, V., and M. Hyland. 1993. Full intuitionistic linear logic (extended abstract). *Annals of Pure and Applied Logic* 64(3):273–291.

Dold, A. 1992. Ginsberg's Bilattices and Nonmonotonic Consequence Operations. Technical report, Master Thesis, Karlsruhe University (in German).

Doyle, A. C. 1902. *The Hound of the Baskervilles.* Strand Magazine.

Dummett, M. 1977. *Elements of intuitionism.* Oxford: Clarendon Press.

Dunn, J. 1986. Relevant logic and entailment. In D. Gabbay and F. Guenthner (Eds.), *Handbook of Philosophical Logic*, Vol. III, 117–224. Dordrecht: Reidel.

Enderton, H. B. 1972. *A Mathematical Introduction to Logic.* New York: Academic Press.

Euclid. 300 B.C. Translation 1908. *Elements.* Vol. I-XIII. Cambridge University Press. translated by Heath.

Feferman, S. (Ed.). 1986. *Kurt Gödel Collected Works.* Oxford, UK: Oxford University Press.

Fidel, M. 1979. An algebraic study of logic with constructible negation. In *Third Brazilian Conference on Mathematical Logic*, 119–130.

Fiore, M. 1996. A coinduction principle for recursive data types based on bisimulation. *Information and Computation* 186–198.

Fitch, F. B. 1954. A definition of negation in extended basic logic. *Journal of Symbolic Logic* 19:29–36.

Fitting, M. C. 1969. *Intuitionistic logic, model theory and forcing*. North-Holland.

Fitting, M. 1988. Logic Programming on a Topological Bilattice. *Fundamenta Informaticae* XI(2):209–218.

Fitting, M. 1991. Bilattices and the semantics of logic programming. *Journal of Logic Programming* 11(2):91–116.

Freyd, P. 1990. Recursive types reduced to inductive types. *IEEE Symposium on Logic in Computer Science* 498–508.

Friedman, H. 1975. Provable equality in primitive recursive arithemtic with and without induction. *Pacific Journal of Mathematics* 57:379–392.

Gallier, J. 1980. On the existence of optimal fixpoints. *Math. Systems Theory* 13:209–217.

Gentzen, G. 1969. *The collected papers of Gerhard Gentzen*. North-Holland.

Ginsberg, M. 1988. Multivalued Logics. *Computational Intelligence* 4(3).

Girard, J.-Y. 1995. Linear logic: Its syntax and semantics. In J.-Y. Girard, Y. Lafont, and L. Regnier (Eds.), *Advances in Linear Logic*, 1–42. Cambridge University Press. Proceedings of the Workshop on Linear Logic, Ithaca, New York, June 1993.

Girard, J.-Y., A. Scedrov, and P. J. Scott. 1992. Normal forms and cut-free proofs as natural transformations. In Y. Moschovakis (Ed.), *Logic from Computer Science*, Vol. 21, 217–241. Springer-Verlag, MSRI publications edition.

Girard, J. 1987. Linear logic. *Theoretical Computer Science* 50:1–102.

Glabbeek, R. v., and G. Plotkin. 1995. Configuration structures (extended abstract). In Proceedings 10^{th} Annual IEEE Symposium on *Logic in Computer Science* Philadelphia, USA. IEEE Computer Society, June.

Goranko, V. 1985. The Craig interpolation theorem for propositional logics with strong negation. *Studia Logica* 44.

Gordon, A. 1994. A tutorial on co-induction and functional programming. In *GLA*, 78–95. Springer Workshops in Computing.

Gordon, A. 1995. Bisimilarity as a theory of functional programming. *Electronic Notes in Theoretical Computer Science* 1.

Gupta, V. 1994. *Chu Spaces: A Model of Concurrency.* PhD thesis, Stanford University, September. Tech. Report, available as ftp://boole.stanford.edu/pub/gupthes.ps.Z.

Gurevich, Y. 1977. Intuitionistic logic with strong negation. *Studia Logica* 36(1-2):49–59.

Hautamäki, A. 1982. The logic of viewpoints. *Studia Logica* XLI(2-3):187–196.

Heyting, A. 1966. *Intuitionism – An Introduction.* North-Holland.

Johnstone, P. 1982. *Stone Spaces.* Cambridge University Press.

Kleene, S. C. 1952. *Introduction to metamathematics.* North-Holland, Amsterdam.

Kleene, S. C. 1959. Recursive functionals and quantifiers of finite types i. *Trans. Am. Math. Soc.* 91:1–52.

Kleene, S. 1967a. *Introduction to Logic.* North-Holland, Amsterdam?

Kleene, S. 1967b. *Mathematical Logic.* John Wiley and Sons Inc., New York.

Klunder, B. 1992. Topos based semantics for constructive logic with strong negation. *Z. Math. Logik Grunlagen Math* 38:508–519.

Kolmogorov, A. 1932a. On the principle of the excluded middle: Russian. *Mat. Sb.* 32:646–667. Translated in *From Frege to Gödel*, Jean van Heijenoort (editor), Harvard, 1967, pages 414–437.

Kolmogorov, A. 1932b. Zur Deutung der intuitionistischen Logik. *Zeitschrift* 35:58–65.

Kozen, D. 1983. Results on the propositional μ-calculus. *Theoretical Computer Science* 27(3):333–354.

Kripke, S. A. 1965. Semantical analysis of intuitionistic logic I. In *Formal Systems and Recursive Functions*, 92–130. Amsterdam: (Proc. 8th Logic Colloq. Oxford 1963) North-Holland.

Lassez, J., and M. Maher. 1985. Optimal fixedpoints of logic programs. *Theoretical Computer Science* 39:15–25.

López-Escobar, E. G. K. 1972. Constructions and negationless logic. *Studia Logica* XXX:7–22.

Manna, Z., and A. Shamir. 1976. The theoretical aspects of the optimal fixedpoint. semantics and correctness of programs. *SIAM Journal of Computing* 5(3):414–426.

Manna, Z., and A. Shamir. 1977. The optimal approach to recursive programs. *Communications of the ACM* 11:824–831.

Maslov, S. Y. 1987. *Theory of deductive systems and its application.* MIT Press. Translated from the Russian by Michael Gelfond and Vladimir Lifschitz.

McCarthy, J. 1963a. A basis for a mathematical theory of computation. In P. Braffort and D. Herschberg (Eds.), *Computer programming and formal systems*, 33–70. North-Holland.

McCarthy, J. 1963b. A basis for a mathematical theory of computation. In P. Braffort and D. Hirschberg (Eds.), *Computer Programming and Formal Systems*, 33–70. Amsterdam: North Holland.

McKinsey, J., and A. Tarski. 1944. The algebra of topology. *Annals of Mathematics* 45:141–191.

McKinsey, J., and A. Tarski. 1946. On closed elements in closure algebras. *Annals of Mathematics* 47:122–162.

Milner, R. 1989. *Communication and Concurrency.* Prentice-Hall.

Milner, R., and M. Tofte. 1991. Co-induction in relational semantics. *Theoretical Computer Science* 87:209–220.

Mints, G. 1985. Resolution calculi for nonclassical logics. *Semiotics and information science* 25:120–135. Akad. Nauk SSSR, Vsesoyuz. Inst. Nauchn. i Tekhn. Inform., Moscow.

Mints, G. 1990. Gentzen-type systems and resolution rules I. Propositional logic. *Lecture Notes in Computer Science* 417:198–231.

Moschovakis, Y. N. 1974. *Elementary Induction on Abstract Structures.* North-Holland.

Moschovakis, Y. N. 1977. On the basic notions in the theory of induction. *Logic, Foundations of Mathematics, and Computability Theory* 207–236.

Moss, L., and N. Danner. 1996. On the Foundations of Corecursion. Technical Report 444, Indiana University, Bloomington.

Nelson, D. 1949. Constructible falsity. *Journal of Symbolic Logic* 14:14–26.

Nelson, D., and A. Almukdad. 1984. Constructible falsity and inexact predicates. *Journal of Symbolic Logic* 49(1):231–233.

Nielsen, M., G. Plotkin, and G. Winskel. 1981. Petri nets, event structures, and domains, Part I. *Theoretical Computer Science* 13(1):85–108.

Park, D. 1983. The fairness problem and nondeterministic computing networks. In *Foundations of Computer Science IV, Distributed Systems*, Vol. 159 of *Mathematisch Centrum Tracts*, 33–162. Mathematisch Centrum, Amsterdam.

Patterson, A. 1996a. Bisimulation and propositional intuitionistic logic. *Proceedings of the 8th International Conference on Concurrency Theory, Warsaw Poland.*

Patterson, A. 1996b. A symmetric model theory for constructible falsity. *Bulletin of the Association of Symbolic Logic. Summer Meeting of the ASL.*

Patterson, A. 1997. A Logical Treatment of Constructive Duality. *http://www-formal.stanford.edu/annap/.*

Patterson, A., and T. Costello. 1997a. Exponentials as projections from paraconsistent logics. *First World Congress on Paraconsistency, Ghent Belgium.*

Patterson, A., and T. Costello. 1997b. Guilt-free exponentials. *Bulletin of the Association of Symbolic Logic. Winter Meeting of the ASL.*

Patterson, A., and T. Costello. 1997c. Implicit programming and Computable Optimal Fixed Points. available from http://www-formal.Stanford.EDU/annap.

Patterson, A., and T. Costello. 1997d. Linear Logic through Constructible Falsity. 6th Annual CSLI Workshop on Logic, Language and Information.

Pearce, D., and H. Wansing (Eds.). 1992. *Nonclassical Logics and Information Processing*, Vol. 619 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag.

Pitts, A. 1994. A co-induction principle for recursively defined domains. *Theoretical Computer Science* 124:195–219.

Pratt, V. R. 1991. Event spaces and their linear logic. In *Proceedings of the Workshop on Algebraic Methodology and Software Technology*, 1–23, Iowa City, Iowa. Springer-Verlag.

Pratt, V. R. 1992. The duality of time and information. In W. Cleaveland (Ed.), *Proceedings of the Third International Conference on Concurrent Theory*, 237–253, Stonybrook, New York, August. Springer-Verlag LNCS 630.

Pratt, V. R. 1996. Broadening the Denotational Semantics of Linear Logic. In *Proceedings of Tokyo Conference on Linear Logic*, Vol. 3 of *Electronic Notes in Computer Science*, Tokyo, Japan. Elsevier. To appear.

Pratt, V. 1980. Dynamic algebras and the nature of induction. In *12th ACM Symposium on Theory of Computation*, Los Angeles, April.

Pratt, V. 1995. The Stone Gamut: A Coordinatization of Mathematics. In *Logic in Computer Science.* IEEE Computer Society, June.

Pratt, V. 1997. Towards full-completness for linear logic. In *Mathematical Foundations of Programming Semantics.*

Rasiowa, H. 1974. *An algebraic approach to nonclassical logics.* North-Holland, Amsterdam.

Rauszer, C. 1975. A formalization of the propositional calclus of the H-B logic. *Studia Logica* XXXIII(1):23–34.

Rauszer, C. 1977a. Application of Kripke models to Heyting-Brouwer logic. *Studia Logica* XXXVI(1-2):62–71.

Rauszer, C. 1977b. Model theory for an extension of intuitionistic logic. *Studia Logica* XXXVI(1-2):73–87.

Reddy, U. 1994. Passivity and independence. In *IEEE Symposium on Logic in Computer Science,* 342–352.

Restall, G. 1995. Four-valued semantics for relevant logics and some of their rivals. *Journal of Philosophical Logic* 24:139–160.

Routley, R. 1975. Semantical analyses of propositional systems of Fitch and Nelson. *Studia Logica* XXXIII(3):283–298.

Scott, D., and C. Strachey. 1971. Towards a mathematical semantics for computer languages. Technical Report Technical Monograph PRG-6, Oxford University Computing Laboratory.

Seely, R. 1989. Linear logic, $*$-autonomous categories and cofree algebras. In *Categories in Computer Science and Logic*, Vol. 92 of *Contemporary Mathematics*, 371–382, held June 1987, Boulder, Colorado.

Sendlewski, A. 1984. Some investigations of varieties of N-lattices. *Studia Logica* XLIII(3):257–280.

Sendlewski, A. 1989. Nelson algebras through Heyting ones: I. *Studia Logica* XLIX(1):105–126.

Sendlewski, A. 1995. Axiomatic extensions of the constructive logic with strong negation and the disjunction property. *Studia Logica* LV:377–388.

Smith, R. 1996. Aristotle on the uses of dialectic. *Synthese* 3:335–358.

Stone, M. 1936. The theory of representations for Boolean algebras. *Trans. Amer. Math. Soc.* 40:37–111.

Stone, M. 1937. Topological representations of distributive lattices and brouwerian logics. *Časopis Pěst. Math.* 67:1–25.

Tanaka, K. 1988. A note: On the proof method for constructive falsity. *Z. Math. Logik Grundlag. Math.* 34.

Thomason, R. 1969. A semantical study of constructible falsity. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik* 15:247–257.

Troelstra, A. S. 1973. *Metamathematical Investigation of Intuitionistic Arithmetic and Analysis*. Vol. 344 of *Lecture Notes in Mathematics*. Springer-Verlag. chapter I-IV, 1–323.

Troelstra, A. S., and H. Schwichtenberg. 1996. *Basic Proof Theory*. Vol. 43 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge Univeristy Press.

Troelstra, A. S., and D. van Dalen. 1988. *Constructivism in Mathematics; An Introduction Vol. 1.* Vol. 121 of *Studies in Logic and the Foundations of Mathematics*. Amsterdam: North Holland.

Troelstra, A. S. 1992. *Lectures on Linear Logic*. Stanford, California: CSLI Lecture Notes 29, Center for the Study of Language and Information.

Vakarelov, D. 1977. Notes on N-lattices and constructive logic with strong negation. *Studia Logica* XXXVI(1-2):109–126.

Vorobév, N. N. 1952. A constructive calculus of propositions with strong negation. *Doklady Akad. Nauk SSSR (N.S.)* 85:465–468.

Wagner, G. 1991. *Vivid Logic*. Vol. 764 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag.

Wansing, H. 1993. *The Logic of Information Structures*. Vol. 681 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag.

Yoneda, N. 1961. On universality I. *Sugaku* 13:109–112.

Yoneda, N. 1962. On universality II. *Sugaku* 14:39–43.

Zaslavskiĭ, I. 1973. On predicate and arithmetic calculi of symmetric constructive logic. *Soviet Math. Doklady* 14:751–754.

Vita

Anna L. Patterson

Computer Science Department
Stanford University
Stanford CA 94305
(650) 723-3334
E-mail: annap@cs.stanford.edu

Education

- **Advisor: Vaughan Pratt: Stanford University**
- **University of Illinois Champaign– Urbana** De jure advisor: Gul Agha
  January 1998 – Doctor of Philosophy, Computer Science
  Thesis Title – *Implicit Programming and the Logic of Constructible Duality*
- **Washington University**
  May 1987 – Bachelor of Science, Computer Science
  May 1987 – Bachelor of Science, Electrical Engineering

Experience

- **Visiting Research Associate – Formal Reasoning Group – Stanford University** May 1997 - present.
  Worked on developing several features of the Elephant 2000 programming language, including development of accomplishment-level specifications, and inter-process communication mechanisms involving speech-acts — specifically the dynamic creation obligations or *promises*.
- **Research Assistant – Concurrency Group – Stanford University** June 1994 - May 1997.
  Investigated and developed an extension to programming languages to allow corecursive functions. The main advantage of this approach is that corecursive functions terminate sometimes when recursive functions do not. The study of this extension necessitated extending domain theory and the lambda calculus. This can be understood through an extension of intuitionistic logic via the addition of constructively negative information and the free interaction of both positive and negative information. In addition to these changes two new operators, similar to linear logic exponentials, were added to the logic. These exponentials are related to greatest fixed points and least fixed points.

- **Visiting Fellow SRI** September 1993 - June 1994.
Investigating and developing general frameworks to describe the equivalence of systems of open heterogeneous parallel processes; thereby allowing formal reasoning about massively parallel systems. This system was formalized on PVS, Prototype Verification System, a theorem prover developed at SRI.

- **Andersen Research - Distributed Platform Group** May 1992 -September 1993.
Evaluated a variety of commercially available object-oriented open distribution architectures. These products needed to be evaluated on their capability of interoperating with dynamically created clients worldwide. Products included pre-released versions by Expersoft, Sun DOE, HP and HyperDesk.
Taught a course on multi-threaded programming. Taught a course on object-oriented programming.

- **Teaching Assistant – University of Illinois** January 1993 –May 1993.
Performed duties as teaching assistant and guest lecturer for the Data Structures and Algorithms course. This was the first time that such a class was taught with an object-oriented language at the University of Illinois. Languages included Modula-3, C++ and Smalltalk.

- **Research Assistant – University of Illinois** December 1990 – May 1992.
Worked on the semantics of the Actor model – a parallel object-oriented model whose language is very similar to Smalltalk, but compiled such that each method invocation maps to an asynchronous message send. Developed proof techniques to show the invariance of programs under various dependability protocols, messaging implementations, and synchronization constraint implementations. The project uses sound engineering methodology to widen the scope of the run-time platform and provide automated checkpoints to ensure that the additional capabilities still satisfy the original system specification.

- **Researcher/Developer – Northrop** May 1987 – August 1989.
Designed a process to statically check the consistency of a real-time system rule base. This work involved transforming the system rules into an equivalent set such that the new set satisfied unique readability, compactness and effectiveness theorems.
Worked on the design, implementation and maintenance of jamming techniques for the F-15E. The run-time system was a real-time embedded parallel machine using a combination of message passing and shared dataspace. Optimized theoretical models, performed extensive software simulations, tested these models on an embedded parallel machine, and provided support for flight testing of the system.

Programming/Software Package Experience

C, C++, HTML, Java, Modula-3, HAL, LISP, Prolog, Rosette, Parlog, Scheme, Ada, Solaris, IBM AIX, ULTRIX, UNIX, IRIX, UNIX System V, DOS, VMS, OS/2, Microsoft Word, FrameMaker, PageMaker, LaTex, Lotus 1,2,3, Excel, Epoch, Emacs

Honors and Awards

Visiting Scholar to the Concurrency Theory Group at Stanford University, Visiting Fellow to the Formal Methods group at SRI International, Langsdorf and Woodward Fellowship, John and Katie M. Bruner Memorial Scholarship, Eta Kappa Nu, Voted Tau Beta Pi's Outstanding Senior, President of the Engineering Council, Women's Leadership Award

Reviews

FOOL96, LICS96, Concur96, Concur95, Concur94, ECCOP96, Coordination96, Information and Computation, Lisp and Symbolic Computation, Parallel and Distributed Technology

Program Committees

Deputy Program Chair for the 6th International World Wide Web Conference Program Committee

Papers

A. Patterson *Bisimulation and Propositional Intuitionistic Logic*, Proceedings of the 8th International Conference on Concurrency Theory, Warsaw Poland 1997.

A. Patterson, T. Costello *Exponentials as Projections from Paraconsistent Logics*, Proceedings of the Workshop for Paraconsistent and Minimal Logics, Ghent Belgium, 1997.

A. Patterson, T. Costello *Guilt-free Exponentials*, 1997 Bulletin of the Association of Symbolic Logic.

A. Patterson, *A symmetric model theory for constructible falsity*, 1996 Bulletin of the Association of Symbolic Logic.

A. Patterson, *Bisimulation and Propositional Intuitionistic Logic.* Proceedings of the Workshop for Observational and Logical Equivalence 1996, Prague, Czech Republic.

U. Reddy, A. Patterson: *Towards a Typed Theory for Actors* Types 94.

G. Agha, W. Kim, S. Frolund, R. Panwar, A. Patterson, D. Sturman: *Abstraction and Modularity Mechanisms for Concurrent Computing*, Parallel and Distributed Technology, May Issue 1993.

R.K. Shyamasundar, A. Patterson and G. Agha: *An ACTOR Framework for Open Concurrent Systems*, Proceedings of the Chalmers Workshop on Concurrency, Baastad, Sweden, 1991.

Books

Chapter in **Research Directions in Concurrent Object-Oriented Programming** Editors: G. Agha, P. Wegner and A. Yonezawa Published by MIT Press, 1993.

Journals Edited

Editors: M. Genesereth and A. Patterson, Special Issue of *Computer Networks and ISDN Systems*, Elsevier containing 6th International World Wide Web Conference papers.

Invited Submissions to Special Issues of Journals/Books

A. Patterson, R. van Glabbeek *Intuitionistic Structures*, Fundamenta Informaticae.

A. Patterson, T. Costello *Exponentials as Projections from Paraconsistent Logics*, Acta Informaticae.

A. Patterson *Constructible Duality* Chapter in CSLI Lecture Notes, for Logic, Language and Computation.

J. McCarthy, A. Patterson, C. Talcott *I/O + World Model => Accomplishment* Chapter in Second Edition of "The Correctness Problem in Computer Science".