NORTHWESTERN UNIVERSITY

Intensionality and Abstraction: A Theory of Logical Form

SUBMITTED TO THE GRADUATE SCHOOL
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

DOCTOR OF PHILOSOPHY

Field of Philosophy

By
Ezra Joseph Cook

Evanston, Illinois

December 2018
Abstract

The analytical paradigm in philosophy has as a pillar an analysis of the sentence as a basic entity. New sentences may be built from old sentences recursively through the application of logical constants, recently including intensional operators. Models are built on how these logical constants interact with each other, and these models are applied to problems in philosophy. I argue that while this paradigm has its place, there exists a theory based on the empirical discipline of generative syntax. It is the interface of this syntax with a compositional, or sentence-internal, interpretation that generates the model of logical form I adopt. I argue that the Curry-Howard Isomorphism bridges this model and the intensional models of formal logic. Logical form is studied through the connection between intensionality and abstraction, specifically when considering the interpretation of modals and conditionals in natural language.
Acknowledgements

I would like to thank Northwestern for the opportunity to grow as a student within an interdisciplinary environment. The hearts and minds of many individuals went into the development of this piece. I would like to thank above all my dissertation chair Michael Glanzberg and his wife Ching-I, as well as Stefan and Magdalena Kaufmann. Without their weekly meetings I would never have made it through the process, and they have guided my hand in every aspect of this project. I would like to thank Fabrizio Cariani. We started at Northwestern in the same year, and his early support became the bedrock upon which this dissertation was built. And, finally, I would like to thank Peter Ludlow, whose approach to semantics differed in many ways from my own, and whose conversations I will never forget.

I would like to thank my family for their unwavering support. Without them this dissertation would not have been possible.
Contents

1 Introduction 8

2 Generative Syntax and Semantics 34

3 Logic and the $\lambda$-Calculus 62

4 Intensional Environments and Imperative Definitions 153

5 Syntax-Semantics Interface 200

6 Modals and Conditionals 216

A Derivation Comparison 274
List of Tables

3.1 Curry-Howard Isomorphism ............................................. 90
3.2 Lambda Cube ............................................................. 147
# List of Figures

2.1 Generative Semantics ................................................. 36
2.2 The Y-Model ............................................................. 42
2.3 ES Assumption ........................................................... 57
2.4 Curry-Howard Isomorphism ........................................... 60
3.1 4-Valued Lattice .......................................................... 106
3.2 Continuation Semantics ............................................... 144
3.3 Curry’s Paradox ......................................................... 151
4.1 Intensional Assumption ................................................. 154
4.2 The Sentence ............................................................. 170
5.1 Little vP ................................................................. 205
5.2 Product Constructors .................................................. 211
5.3 Entity Products ......................................................... 212
6.1 Movement of CondP ................................. 241
Chapter 1

Introduction

I investigate the justificatory connection between linguistic data and metaphysical theories with logical foundations. The claim made in this paper is (1) that there is such a connection, and (2) that this connection is indirect. These claims will be contrasted with a class of views that purport to provide a more direct connection.

Along the way, I defend a *schmentencite* strategy for interpreting language. I develop a theory of types with an algebraic foundation sufficient for interpreting schmentencite sentences of natural language, and I provide interpretations of modals and conditionals stable enough to handle contemporary data.

Where formal methods were first brought to bear on the study of proof theory in mathematics, extending these methods to the evaluation of scientific theories is not an objectionable enterprise. This enterprise differs, however, from the arguments presented for choosing one
logical theory over another. The philosophers of this school are often quite cavalier about what might be otherwise considered the linguistic data upon which they ground various aspects of their theories. In theoretical linguistics, on the other hand, a very robust model of the universal structural properties that underlie and connect the various human languages is provided. These theories are generally couched within an internalist framework. The linguist is satisfied to simply provide a model of certain cognitive properties that underly language production and understanding, without any real attempt to extend this model’s capabilities beyond that of the cognitive system at issue. If the model eschews this connection, certain logical properties such as truth and satisfaction must be reinterpreted epistemically.

Deduction, formally studied, codifies an explicit mental procedure not essential to the language faculty. The procedures that govern proper sentence production and the procedure that governs proper inferential patterns, while related, could possibly operate independently. The notions of truth and logical consequence, however, could provide a bridge that connects one to the other.

This methodological position can be traced to at least the 1928 lecture of Tarski’s “On Some Fundamental Concepts of Metamathematics”, where three methodologically interrelated ideas are formulated:

1. Explicit concern with metatheory signaled through the use of the metametalanguage,

Cf. (Stalnaker 2014, 30-31) for a criticism of comments made by (Williamson 2009, 136) on this point. Tarski (1985, xvii).
2. Explicit formulation of syntactic laws, through the use of recursive (generative) grammars, and

3. Explicit formulation and use of semantic and model theoretic ideas such as truth, satisfaction, and consequence.

I follow these Tarskian themes throughout. I offer an alternative model to those who think sentential operators exist within a linguistic framework. Instead, I provide what David Lewis calls a *schmentencite* strategy, where properties are consumed by other properties. I believe this framework more accurately maps the computational system of the human language faculty, and, for principled reasons, this investigation should lead us to a conceptual framework that is more closely related to the structure of reality than others in its general neighborhood.

In chapter 1 I focus on some historical debates where this type of analysis matters. These concerns directly impinge upon debates about universals, the structure of the human conceptual framework and it’s connection to the world, and how views in the philosophy of logic should be connected to contemporary research in linguistics. I highlight the descriptive metaphysics of Strawson in contrast to a Quinean quietism regarding the internal structure of the sentence. I believe that this quietism has lead to a romantic attachment by philosophers to sentence-level analysis. I hope to show that this attachment is misguided.

In chapter 2 I connect this attachment to the sentence to a view in linguistics called *generative semantics*. Generative semantics differs from generative syntax, which I adopt.
The generative semanticist hoped to generate semantic equivalence classes of sentences that corresponded to syntactic configurations that differed in surface structure but seemed to share the same meaning. This was a direct attempt to save the sentence as the basic level of logical analysis. Unfortunately, it did not survive contemporary syntactic analysis, largely due to arguments by Chomsky and Jackendoff. This leads us to the lexicalist hypothesis and the highlighting of the displacement properties that distinguish natural languages from purely formal languages. This brings us to a contemporary debate where I believe the philosophical hang-up on sentences being the atomic particles over which we provide an analysis is misguided. The debate itself has been termed the operators argument, but I believe the sentence-first methodology upon which it relies is far more pervasive than the parameters of the argument itself. At this point I propose my alternative account, where the empirical study of generative syntax provides a logical form for a sentence which is interpreted compositionally by a λ framework.

Chapter 3 develops the λ-calculus to a sufficient degree to model the system I prefer. The main issue to consider when designing a λ-system suitable for compositionally treating generative syntax trees is it’s normalization properties. Whether the system is strongly, or just weakly normalizing control whether the systems corresponding reductions have infinite branches or halting conditions. The composition rules used when interpreting syntax trees may be evaluated along this dimension. I develop an operational semantics with an algebraic interpretation capable of modeling continuations, side-effects, and sequencing, which then
allows for the transparently compositional treatment of some data provided by the sentence-
first theorists.

In chapter 4 I treat the arguments of the sentence-first theorists head-on. The data
they provide is given an alternative interpretation in line with the framework developed
herein. I provide both semantic and syntactic arguments against the sentence-first position.
Semantically, I argue that modeling an assignment function as an object to be interpreted by
the calculus is essentially a less-general version of a continuation semantics. Re-assignments
of values to variables that are required to provide a logical analysis of sentential operators
can be handled by the sequencing of side-effects, a mechanism already built into advanced
type theory. I then argue, syntactically, that there are no first-order quantifiers in natural
language and that the prejacent of an intensional operator does not correspond to a definite
syntactic category. The conclusion I draw from this is that traditional intensional logic is
not fine grained enough to capture the logical form of natural language, whose meanings can
be interpreted by an applicative type-theoretic model.

After providing negative arguments, I continue in Chapter 5 to lay the groundwork of
my positive theory. Here I develop a theory of logical form where generalized quantifiers are
mapped directly to generative syntactic trees. It is here that I provide an argument that
events may be decomposed into entities already present within the system. A theory of the
verb and its argument structure is outlined. This basic outline must be in place in order to
see how the intensional properties of modality manipulate the lower verb and its arguments.
In the final chapter, I investigate the logical form of modals and conditionals within my system. I develop the syntax according to my system and extend a semantics based on a Kratzerian system, taking insights from (Frank 1996). I then consider Frank-Zvolensky conditionals and provide a solution based on AGM style belief revision rules. Finally, I compare compositional frameworks: one based on an intensional function application rule and the other based on a combinatorial framework capable of representing lambda functions.

The Package: Historical Antecedents

This is the dilemma: one seems to be faced with a choice between either a robust metaphysical model grounded in cavalier linguistic theory or a robust linguistic theory lacking any real connection to the world. During the course of this inquiry I propose that these models be connected, and, in fact, establish such a connection.

I seek a structural connection, by proving that structure is preserved between two models on either side of the divide. These models are representative of the field. To dissolve the dilemma in this way, I must first abolish a pervasive assumption that largely passes by without argument in the literature. This assumption, sententiality or ES, that sentences embed under operators to form new sentences, does great violence to the particularly intricate nature of the model proposed here. Specifically, what is meant by ‘proposition’, in reference to a complicated syntactic-semantic structure, is at issue. I argue that the theorist that adopts ES is taking on board a type of generative semantics, which has had many arguments mar-
I argue for a separate proposal that is capable of generating a logical form I deem acceptable for bridging the divide between formal logical models and generative grammar. This model happens to be *schmentencite*. It also eschews syncategorematic function application rules in a novel way. I believe this captures the spirit of the *direct compositionalist* movement in linguistics, without relying on their specific choice of grammar. Instead I adopt a movement based approach, which, admittedly is at odds with the direct compositionalist, but I believe better captures the displacement properties Chomsky highlights as distinctive to natural language.

In what follows, I develop a natural picture of logical form based on contemporary work in generative syntax. This picture is to be contrasted with a formal logical picture that has become central to the philosophy. I argue that while the formal theorists might be approaching these problems in the wrong way, there still exists work for the formal picture to perform. I introduce the central test case: the operators argument of (Kaplan 1989b), (Kaplan 1989a) and (Lewis 1980). This argument relies on sententiality (*ES*). Discussion of this assumption leads to an explicit discussion of logical form, and my initial positive proposal. Here I suggest that the proper level of connection between the formal models is a matter of the choice of type-theory adopted. I propose a natural model that takes advantage of the Curry-Howard isomorphism to bridge the syntax-semantics divide. I develop close alternatives of (King 2003), (King 2007) and (Schaffer 2012). Then I develop the
arguments of (Weber 2012) and (Rabern 2012), which provide contemporary examples of the group founded by the work of Kaplan and Lewis. To properly illustrate the issues involved in this final section, I must develop some additional syntactic-semantic machinery, which then serves to deepen the development of the positive proposal while simultaneously providing the resources to better explain how the negative arguments from (Weber 2012) and (Rabern 2012) fail. Finally, I develop a broadly Kratzerian model of modals and conditionals that deepens the discussion of context brought about by this investigation.

The dominant view in theoretical linguistics is a Chomsky-inspired internalism. The view is that language is a biological process, an independently developed computational system of the mind. As such, each individual develops his own language, an $I$-Language, by setting certain parameters of the computational system based on the primary learning data received as a child. Unfortunately, Chomsky and his followers generally take this view to lend itself to an interpretation that models a conceptual system alone. That is to say, there is no immediate reason to consider the model as one that reaches into the external world; rather, it is enough to simply model the individuals conceptual representation of the world.

This last assumption is auxiliary: it is not directly motivated by positing a language faculty, nor by the empirical evidence that supports this posit. I aim to take the model provided by the Chomskian program of generative syntax on board without falling directly into his favored internalism. Clearing the conceptual space for a view of this type accommodates the advances in theoretical linguistics without the loss of properties that philosophers tend to
favor. For example, without truth conditions, one would require some more complex explanation of the information uptake and agreement/disagreement properties that accompany successful communication. This view is similar to that found in (Partee 2004).

I follow a Strawsonian approach, assuming the investigation into the deep structure of language reveals the human conceptual framework:

**Descriptive Metaphysics** Investigation into the deep structure of language reveals the human conceptual framework (because it yields the same framework).

Unlike Chomsky, Strawson believes that descriptive metaphysics describes “the actual structure of our thought about the world.”³ It is thought about the world that is the primary concern here, the idea being that the conceptual framework so revealed correctly describes the nature of reality. Strawson thinks that this structure is to a certain extent hidden, it “does not readily display itself on the surface of language, but lies submerged.”⁴ Further, a study of language which reveals this hidden structure could potentially uncover a “massive central core of human thinking [...] categories and concepts which, in their most fundamental character, change not at all.”⁵ These quotes are telling in the way that I intend to approach this topic. It is only through a study of the deep structure of language that the structure of thought, and therefore of the world, will be revealed.

³(Strawson 1959, 59).
⁴(Strawson 1959, 10).
directly considered. As I am only considering the connection between the empirical logical form of languages and the computational system of biology that generates and checks these logical and syntactical forms. There are too many prejudices to unravel to approach this project by attacking theories of propositions, and instead hope to undergird such a theory with theoretical work on logical form. However logical form is chopped up into units that are shared between language, thought, and the world, is not a position I take a determinate stance on. I only suggest certain themes that would be consistent with my view. I prefer to think that properties, not propositions are the conceptual glue that holds language and thought together, but I only provide arguments that connect properties to logical form here.

The focus of this paper is the study of linguistic deep structure. Specifically, the relation between sentential structure and logical structure. In chomskian generative syntax, the syntactic structure of a sentence is not a direct reflection of its surface structure. This mirrors the position of (Russell 1905), which called into question the relation between a sentences surface structure and its logical structure when investigating the logical properties of definite determiners. Since neither the syntactic structure nor the logical structure is

6I am not comitting myself to the Sapir-Whorf hypothesis here. That the structure of particular human languages influences the speaker’s conceptual structure is not an assumtion I need make. This hypothesis gets the direction of explanation incorrect. The human conceptual framework is stable in its potential, and it is only through an in-depth cross-linguistic inquiry can one hope to uncover any revealing generalizations about its nature. Cf. (Pelletier 2011) for in-depth discussion of the Sapir-Whorf hypothesis in connection with these issues.
a direct reflection of a sentence’s surface structure, the connection between the syntactic structure and semantic structure of the sentence must be spelled out theoretically.

The package I put together starts with some modest assumptions, which I think should be accepted regardless of one’s position on the spectrum of views described below.

Call the total package *Integrative Metaphysics*. This is a descriptive metaphysics revival. For at least the grammar of human languages, there exists an underlying, finely articulated structure:

**Structure Dependence** Grammatical operations are structure dependent.

Words in sentences are grouped into *constituents*, like a noun phrase (NP) or verb phrase (VP). Notice the relatedness of the following type of sentence pairs:⁷

(1.1) (a) Susan must leave.

(b) Must Susan leave?

(1.2) (a) The man has eaten the cake.

(b) Has the man eaten the cake?

(1.3) (a) The woman who is singing is happy.

(b) Is the woman who is singing happy?

⁷This assumption and the data that supports it is called the *poverty of stimulus* argument. For empirical support see (Crain and Nakayama 1987).
The postulation of an underlying grammatical structure, not necessarily directly related to the surface structure, of the sentences witnessed is required to provide a felicitous explanation of this data. This underlying structure might be considered the coarsest structure that makes a particular language a candidate natural language. The study of this structure is therefore driven by inquiry into the natural properties of these languages.

The second assumption ties these structures together cross-linguistically. There must be some linguistic universals that relate all human languages:

**Linguistic Universals** All natural languages are related at some explanatory level.

The contemporary manifestation of this idea is the project of discovering the principles and parameters of a general theory that distinguishes the worlds languages.\(^8\) That is to say, the combination of certain abstract principles or rules governing possible grammars and the setting of particular parameters that explains the variation witnessed between language systems. In fact, (Kayne 1994) takes an even stronger position with his linear correspondence axiom, suggesting that all of the worlds languages have only one strict, antisymmetric, structure.\(^9\) The observed cross-linguistic differences in word order are then derived by complex movement operations.\(^10\) These positions both take the constituent structure outlined above to be the proper explanatory level to spell out the content of LINGUISTIC UNIVERSALS.

\(^8\)Cf. (Chomsky 1957), (Chomsky 1965), (Chomsky 1980), (Chomsky 1995).

\(^9\)In fact, in (Kayne 1994) he shows how to derive the major properties of standard X-bar theory from this assumption.

\(^10\)Kayne’s research program has provided the foundation for the cartographic movement in generative syntax, which has found great empirical success in, for example, the work of (Rizzi 1997), (Rizzi 2004),
(Pelletier 2011) takes a view called *Modest Natural Language Metaphysics* to be reliant on the assumptions STRUCTURE DEPENDENCE and LINGUISTIC UNIVERSALS above. This view originates in the work of Emmon Bach.\textsuperscript{11} Bach is concerned with two lines of research:

\textbf{I.} Chomsky’s Thesis: Natural languages can be described as a unified formal system.

\textbf{II.} Montague’s Thesis: Natural languages can be described as an interpreted formal system.

The first of these theses is a particular instantiation of STRUCTURE DEPENDENCE and LINGUISTIC UNIVERSALS, while DESCRIPTIVE METAPHYSICS is itself concerned with an interpretation of the interpretation associated with the second thesis.

The interpretation associates a model structure populated by non-linguistic objects with the linguistic objects defined by Chomsky’s formal system, but just what this model structure represents has not yet been spelled out. The two most plausible candidates are spelled out most vividly by (Bach 1986b, 574):

\begin{quote}
[O]ne tradition, probably the most prominent one in the philosophical tradition, has it that they are real objects and relationships in the world (as well as, perhaps, their analogues in other possible worlds); the other, which seems most prominent in the tradition of generative theory, says that they are mental objects: concepts, representations, or the like.
\end{quote}

(Cinque 1999), and (Cinque 2006).

\textsuperscript{11} Cf. (Bach 1980), (Bach 1981), (Bach 1986b), and (Bach 1986a).
Up to this point, the assumptions made only get us as far as mental objects, what has been called modest natural language metaphysics. What is additionally required is a bridge, and there are two candidates:

**Realism** The conceptual framework of speakers correctly describe reality.

Realism takes the distinctions reflected in the overt and covert categories of natural language to correspond to the structure of the world. What has not been stated is at what explanatory level this correspondence occurs. There is a stronger version of this thesis that might require the correspondence to hold at the most fundamental explanatory level, carving at the natural joints of reality. I do not defend this stronger thesis directly. The topic under investigation here is the contrast between the type of integrative metaphysics outlined by the four assumptions **STRUCTURE DEPENDENCE, LINGUISTIC UNIVERSALS, DESCRIPTIVE METAPHYSICS, and REALISM** and a Quinean *regimented metaphysics* which takes itself to be continuous with scientific inquiry such as physics.

The difference between integrative metaphysics as I have described it and regimented metaphysics is that while integrative metaphysics seeks a deeper understanding of language, 

---

12(Bach 1986b), (Pelletier 2011).

13This investigation is not a thesis against anti-realism. The anti-realist position is perfectly consistent with Chomsky’s Thesis, and if one is concerned with integrating it with Montague’s Thesis, the technical points made below may be adopted. Here we are concerned with the additional baggage associated with the realist position, and whether it could be considered a motivating factor in the theory selection performed below.
regimented metaphysics seeks to clean up a tool considered too imprecise for a particular type of inquiry:  

Ordinary language is totally unsuited for expressing what physics really asserts, since the words of everyday life are not sufficiently abstract. Only mathematics and mathematical logic can say as little as the physicist means to say.

Indeed, this type of view is reflected more recently by Williamson, in the preface to *Modal Logic as Metaphysics*, when he states that “[O]ne role for logic is to supply a central structural core to scientific theories, including metaphysical theories.”  

The program of *generative semantics* hoped to find a deep logical structure that would function to sort linguistic phrases into synonomy classes. This could be seen as an extension of the regimented approach. As will be seen in what follows, however, there seems to be a mismatch between the formal structure of natural language and the formal structure of first-order logic. (Lewis 1970) and (Montague 1970) treat noun phrases uniformly as generalized quantifiers, which is a direct descendant from the analysis in (Russell 1905).

Disagreement begins on the formal structure of the deepest or most abstract level of representation. The point of contact between the two theories is the mathematical method of representation. The mathematical branch of *model theory* seeks to understand the semantic basis of logical representation. This mathematical study makes direct comparisons between

---

14 See (Russell 1931) and (Russell 1905).
15 (Williamson 2013, x).
the two views available. It provides the common ground between the theories.

Once this general project is taken on board, Quine’s dictum that *to be is to be the value of a variable*\(^{16}\) is considered dogma, for the entities constituting the core of the theory at issue are just those the existence of which the theory postulates. One way of thinking about this type of enterprise is a desire for a regimented language that is distinctly Quinean:\(^{17}\)

Simplification of theory is a central motivation likewise of the sweeping artificiality in notation of modern logic. Clearly it would be folly to burden a logical theory with quirks of usage that we can straighten.

(Stalnaker 2012, 91) suggests that, in the process of regimentation, Quine’s goal was “to find neutral linguistic frameworks in which controversial substantive issues about what there is, and about what can truly be said about what there is, could be framed.” So, rather than letting an antecedent requirement to capture patterns of entailment in natural language—the relations between our concepts—be the guide, the enterprise has floated free from the data in favor of a high minded and moralistic enterprise set out to enforce antecedent philosophical commitments directly.

This Quinean hangover leads directly to constructing models based purely in ideology. A.N. Prior suggests that “there is too much unconscious theological fantasy in modern

\(^{16}\)To be more precise: “to be assumed as an entity is, purely and simply, to be reckoned as the value of a variable.” Cf. Quine(1953, 13).

\(^{17}\)(Quine 1960b) and (Quine 1960a, 158).
logic.” To combat this “theological fantasy” he, when considering the logical structure of temporal relations, suggests that “[tense-forming operations] must be expressions that form sentences from sentences, and so must come out of the same box as the ‘not’ or ‘It is not the case that’ of ordinary propositional logic, and the ‘Necessarily’ or ‘It is necessary that’ of ordinary modal logic.” Prior claims that “The formation-rules of the calculus of tenses are not only a prelude to deduction but a stop to metaphysical superstition.” In other words, Prior would avoid reifying times in his ontology, which would be a byproduct of an analysis of temporality in a first-order manner, in conjunction with the Quinean assumption. (Sider 2011) mirrors Prior in his attitude toward regimentation when he suggests that “whether reality contains causal, or ontological, or modal structure is a matter of whether causal predicates, quantifiers (or names), and modal operators carve at the joints [of nature].”

Regimented metaphysics, based in Quinean regimentation, should be contrasted with the type of project that I will continue to fill out as we go along: the descriptive metaphysics revival. That is to say, I take seriously both that Chomskian linguistics has a model of the structure of language and that models in logic are perfectly suited to capture the inference patterns witnessed in communication and reasoning. Since both of these projects are sources of information about the world, I hope to bring them together in a way that (1) does

---

18 (Prior 1957, 50).
19 (Prior 1967, 15).
20 (Prior 1967, 19).
21 (Sider 2011).
violence to neither side and (2) does not lose the weak realism assumption above. Integrative
metaphysics is then a type of Strawsonian descriptive metaphysics informed by Chomskian
linguistics. To accomplish this, I develop a case against the assumption sententiality, which
I take to be shared among the regimented metaphysicians. Through the diagnosis of this
shared problem, I hope to develop a more productive alternative.

Sentence-First Methodology

The guiding idea behind the proposals in this paper is that there exists natural logical struc-
ture that emerges from the semantic interpretation of generative grammar. This structure
shares some similarities with the logics familiar to the philosopher, but nonetheless harbors
some striking differences. To draw out these differences I take a pervasive assumption in
the literature as a test case. I contend that this assumption is the crucial assumption of
the regimented metaphysics family, the main alternative to the positive proposal. While
an assumption like this might be perfectly acceptable when considered only for its ability
to capture certain patterns of inference within a formally constructed system, it diverges
from the natural logical structure that has emerged in the past thirty years of research in
theoretical linguistics. I follow (Cappelen and Hawthorne 2009) in calling the assumption
\( ES \):

\( ES \) In natural language, some embedding environments (\( E \)) take sentential arguments (\( S \)).
An embedding environment can be considered, given a formal language, something like the scope of an operator or quantifier: \( \Box(...) \), \( \exists x(...) \). In the first case, the necessity operator takes a sentence like \( P \) and makes a new sentence \( \Box P \). In the second case, the existential quantifier takes an open formula like \( P(x) \) and makes a sentence \( \exists x P(x) \). \( ES \) states that, effectively, embedding environments operate like \( \Box \) rather than like \( \exists \). The principle \( ES \) is just the kind of principle that would be required for any bridge to be developed that connects a regimented model to the world. Therefore, this is a rather clear case where rejecting this principle is grounds for rejecting the preceding types of views, on methodological grounds. I take issue with arguments in the literature that assume \( ES \), but in reality this assumption is much more widespread. For example, based solely on considerations of theoretical unity and fecundity, one might simply assume that there is a core class of logical vocabulary, such as the categories of epistemic, deontic, and metaphysical modality. Lexical items such as necessary, possibly, must, might, ought and should would be natural candidates for such a category. These terms might then be provided an operator analysis, much like what is described below. In fact, the persuasiveness of an argument that purports to find operators in unlikely places surely bases some of it’s support in the assumption that an operator analysis of terms such as those above is uncontroversially accepted. But by rejecting \( ES \), one rejects precisely this analysis of the above terms.

In the remainder of this paper, I consider three arguments that presuppose the \( ES \) assumption: first, the most famous of the three, the operators argument of (Kaplan 1989b),
(Kaplan 1989a) and (Lewis 1980). Then, two recent arguments by (Rabern 2012) and (Weber 2012). Along the way I outline a dodge of the operators argument made by (King 2003), (King 2007) and (Schaffer 2012) that is compatible with the rejection of ES, but does not go quite as far as I would like. These arguments serve to outline my positive theory, which I discuss concurrently.

To discuss the operators argument I must first introduce the technical notion of a proposition. Propositions are the abstract entities to which philosophers appeal in order to explain a variety of phenomena: they are the objects of assertion; they capture patterns of inference; they are the things over which we agree and disagree; they are the primary bearers of truth and falsity. (King 2003) aptly summarizes the package.22

A primary purpose of a semantics for a natural language is to compositionally assign to sentences semantic values that determine whether the sentences are true or false. Since natural languages contain contextually sensitive expressions, semantic values must be assigned to sentences relative to contexts. These semantic values are propositions. Sentence types may also be associated with higher level semantic values that are or determine functions from contexts to propositions (something like what David Kaplan calls “character”). Propositions are the primary bearers of truth and falsity. Propositions are also the objects of our attitudes: they are things we doubt, believe, and think.

22(King 2003, 1).
One commonly held view of propositions, borrowed from intensional logic, is just that they are sets of truth supporting circumstances, points at which truth is defined in a model. Call these points indices, abbreviated $i$, and let a proposition $p$ be the set \{\(i : p \text{ is true at } i\}\).

Philosophers of language such as Richard Montague, David Kaplan, and David Lewis have, historically, been concerned with sentence operators, developed in a finitely state-able artificial language. This “intensional assumption” is a consequence of philosophers following in the tradition of Rudolf Carnap and A. N. Prior. This could be due to an emphasis on, for example, the modal properties of sentences,\(^23\) rather than explaining the productivity facts exemplified in speaker competence that so interested Chomsky and the generative grammarians.

This type of emphasis is pointed out by (Yalcin 2014), at least with respect to the work of David Lewis:\(^24\)

Lewis expressed no interest in accounting for productivity. On the contrary, he explicitly expressed skepticism that semantics could be supplied with determinate foundations at the level of subsentential expressions. He offered only to ground the notion of a population’s using a language, where ‘language’ in his sense refers just to a function which pairs sentences with meanings, \([\ldots]\). This limited set of facts of “semantic value” were to be grounded in certain conventions—on Lewis’s analysis, in certain regularities in belief and action prevailing

\(^{23}\)These properties have been considered important since (Kripke 1980).

\(^{24}\)(Yalcin 2014, 25).
in a given population, owing to some common interest in communication. This story about the ground of sentential semantic value facts did not generally settle the choice between two compositional semantic theories agreeing on the semantic values of sentences, but differing at the subsentential level.

Quine believed that relations of synonymy were inscrutable at any level smaller than that of the sentence. Quine’s dismissal of this type of inquiry, of course, did not consider anything related to the methodology of contemporary linguistics, except by asking the reader to consider the circularity of “asking the natives” directly.

Further, and to the point, this line of reasoning fosters (Kaplan 1989b), (Kaplan 1989a) and (Lewis 1980)’s operators argument. The operators argument is characterized by (Kaplan 1989b), (Kaplan 1989a) when he states:

[I]ntensional operators must, if they are not to be vacuous, operate on contents which are neutral with respect to features of circumstance the operator is interested in.

(Lewis 1980) states the same point in the following way:

Often the truth (-in-English) of a sentence in a context depends on the truth of some related sentence when some feature of the original context is shifted. ‘There

---

25 Cf. (Quine 1953), (Quine 1970)

26 (Kaplan 1989b, 503,n 28).

27 (Lewis 1980, 39).
have been dogs.’ is true now iff ‘There are dogs.’ is true at some time before
now. ‘Somewhere the sun is shining.’ is true here iff ‘The sun is shining.’ is true
somewhere. ‘Aunts must be women.’ is true at our world iff ‘Aunts are women.’
is true at all worlds.

At this point, the account we are developing stipulates that propositions are set-theoretic
entities: sets of sequences of parameters. They are the \( n \)-tuples that determine the truth of a
sentence in a context. Much ink has been spilt over the exact sequence of parameters defined
by these \( n \)-tuples: whether it is just a world, or a world, time pair, or whether it requires
the addition of a judge, a location, a standard of precision, etc., and it is the operators
argument that is supposed to test for the existence of the particular elements in the \( n \)-tuples
that define this index. Because propositions, or “contents” in Kaplan’s terminology, are sets
of these sequences, if the language at issue happens to contain logical operators that shift
these parameters, then Lewis’s point is satisfied. But these operators that we speak of are
members of an intensional logic, a formal, artificial language. This language is not English.
An assumption that seems to have been riding in the background, without comment, is that
English itself, indeed any natural language, contains these operators. For this to be true,
some connection must be made between the structure of a sentence in natural language and
its logical structure: the logical form of a sentence.

(Cappelen and Hawthorne 2009) set out what they consider the assumptions of the
operators argument. Aside from \( ES \), which they also call Sententiality, they identify the
following:\(^{28}\)

**Parameter Dependence**  \(S\) is evaluable for truth only relative to some parameter \(M\).

**Uniformity**  \(S\) is of the same semantic type when it occurs alone or combined with \(E\).

**Vacuity**  If \(M\) is already supplied for \(S\), combining \(S\) with \(E\) does not affect truth value.

Of these three assumptions, I take no issue with **vacuity**, as it concerns itself specifically with the structure of the operators in question. Neither am I particularly concerned with **parameter dependence**, for, regardless of the label of the constituent in question, whether it be ‘\(S\)’ or otherwise, I do agree that the semantic component is primarily concerned with saturating properties through various compositional means. What I am concerned with is **uniformity**, for I think there are classic cases where an \(ES\) theorist would posit an operator, and yet **uniformity** seems to fail.

Telegraphing the type of arguments you will find in the final sections of the paper, consider the following pair:

(1.4) (a) John ought to be home.

(b) John is home.

If one were to apply the operators test here, one would consider (1.4 b) the \(S\) embedded under **ought** in (1.4 a). But uniformity patently fails here. The property expressed by the constituent embedded under the modal auxiliary verb in (1.4 a) is the property \(\lambda x. [x \text{ is home}]\).

\(^{28}\) (Cappelen and Hawthorne 2009, 71).
The content of (1.4b), on the other hand, is the result of applying John to this property: \( \lambda x. [x \text{ is home}](\text{John}) \). Their inflectional—and, therefore, functional—structure differs, as well. That is, \textit{to be home} is an infinitive, with \( \emptyset \) tense, whereas (1.4b) is non-past indicative. So it is my contention that \textsc{uniformity} fails, but the root of this issues lies not with \textsc{uniformity}, but with \textsc{ES}. For \textsc{uniformity} is only a plausible assumption if \textsc{ES} has already been assumed. Further, while Cappelen and Hawthorne claim that “[t]here is no single flaw common to all possible applications of the Operator Argument,” I disagree. The flaw common to all possible applications of the operators argument is that there is no stable syntactic category that embeds under a matrix clause which could be associated with ‘S’.

To wrap up the preceding discussion, we have seen that there is a strong pull toward a view that connects the objects of assertion to the semantic content of a sentence: the proposition. Some would like to go farther than this, finding the objects of assertion not only at the sentence level but at the sub-sentential level as well. The operators argument is provided as a way to test for a type of sub-sentential environment that would require a proposition as its object. This test relies on a methodological assumption: \textsc{ES}. Unfortunately, as will be shown below, the logical form of a sentence that this assumption generates differs from a number of different theories about how the logical structure of a sentence is represented syntactically in natural language. This makes the assumption an empirical issue worth investigating, but undermines any \textit{a priori} support. In the past, prior to the development

\(^{29}\text{(Cappelen and Hawthorne 2009, 73).}\)
of the generative program, this type of approximation could have been considered a helpful type of regimentation, but given the state of the art, it should now be considered at best an historical artifact. Thus, given the discussion in the introduction, while relying on $ES$ as an assumption might have initially seemed small or insignificant, as it turns out, this assumption is much more empirically significant than it initially appeared.
Chapter 2

Generative Syntax and Semantics

The operators argument suggests a methodology that is suspect in a number of ways. First, the syntax of natural language must contain a particular formal object, an operator, which shifts the point of evaluation of a closed formula. This closed formula is often termed a sentence of the language, hence the term sentential operator. This type of sentence is a formal construct. It has been generated by an artificial syntax. The logical form of an English sentence is thus represented. This type of logical form is different, however, from the logical form, termed LF, found in (May 1985), which has come to be considered a legitimate syntactic level in (Chomsky 1995) and elsewhere in the literature of generative syntax. This point is made clearly as early as (Cooper 1977) and (Cooper 1979), in regards to Montague's
system $PTQ$:$^{12}$

As it is often somewhat complicated to spell out in English what the denotation of a given phrase is, it is useful to represent it by an expression of Montague’s logic. It should be made clear that the translation of natural language expressions into the logic is simply a matter of convenience, and is not intended to represent a claim that such a translation should be included in a grammar of natural language.

If one’s intent is not to develop a compositional system that explains the productivity facts that are the central explananda of theoretical linguistics, however, then the theory developed should again be used, in Cooper’s words, as “simply a matter of convenience.” The theory itself should be independent of the grammar-interpretation mapping at issue. If these models do not connect in the right way to the prevailing models in theoretical linguistics, any claims made about natural language based on some property of the model should be considered with due caution.

How, then, would one relate the operators language to English in a more significant way? One might think that there is some semantically significant deepest level of representation (DS). For any given syntactically well-formed expression at this level, reduction of a large

$^1$(Cooper 1979, 62).

$^2$This point, of course generalizes straightforwardly to the more contemporary type-driven interpretations of (Klein and Sag 1985), (Bittner 1994), and (Heim and Kratzer 1998).
class of synonymous sentences—at surface structure (SS)—could be possible. This type of view might take the following form: Consider the passive transformation as an example:

(2.1) (a) Brutus killed Caesar.

(b) Caesar was killed by Brutus.

The linear view would suggest that (2.1a)/(2.1b) are derived from the same deep structure representation. The difference between (2.1a)/(2.1b) is syntactic, not semantic. In line with generative assumptions, a sentence has a syntactic deep structure that differs from the apparent surface structure. The seductive view suggests that once this deep structure is uncovered, it is all that is required for the semantics, and, further, this deepest level of representation could take the form of something like a quantified modal logic.

Unfortunately for those who have been seduced by this line of thinking, it has already
been tried, and the general consensus is that it failed. The research program to which I refer is *generative semantics*, pioneered by (Katz and Postal 1964). Their guiding idea was that syntactic transformations never add any additional semantic information, and thus, in the words of (Jackendoff 1972) “all semantic information is represented in underlying structure.” The development of generative semantics led to an attempt to dispense with deep syntactic structure, instead proceeding directly to surface structures from semantic representations by way of transformational rules.\(^3\) Unfortunately, this move requires incredibly powerful and unconstrained transformations which in turn require a multiplication of meanings. This undermines the explanatory value of the theory itself, as will be seen.

The program of generative semantics made a substantial empirical assumption: that form and meaning track each other on the basis of the logical structure. (Chomsky 1972), (Chomsky 1970b), (Chomsky 1970a), (Chomsky 1970c), and (Jackendoff 1972) criticized generative semantics for lacking any explanatory adequacy. This was due to the lack of constraints on transformations necessary to satisfy the empirical assumption. Specifically, (Chomsky 1970b) countered generative semantics with what has been termed the *Lexicalist Hypothesis*: that the lexicon underlies the relation between form and meaning, rather than the formal logical structure of a sentence.\(^4\) In other words, determining the logical structure

\(^3\)Cf. (McCawley 1968b), (McCawley 1968a), (McCawley 1970), (Lakoff 1970), and (Postal 1970) for further discussion.

\(^4\)The principal point that should be drawn from this discussion is that it is important to reign things in, in some way. The lack of constraints present in generative semantics undermines its explanatory force.
of a sentence is just another way that word meaning affects syntax. This assumption allows certain constraints on transformations, succinctly summarized by Jackendoff:

> The only changes that transformations can make to lexical items is to add inflectional affixes such as number, gender, case, person, and tense. Transformations will thus be restricted to movement rules and insertion and deletion of constants and closed sets of items.

(Chomsky 1970b) argues for this thesis by considering the relationship between (6.1 a)-(6.1 c)

(2.2) (a) The enemy destroyed the city.

(b) The enemy’s destruction of the city

(c) The enemy’s destroying the city

Chomsky suggests that while the sentence (6.1 a) and the gerundive nominal (6.1 c) are related transformationally, with the same deep structure, the derived nominal (6.1 b) is best analyzed as only morphologically related, with a separate listing in the lexicon. This provides an explanatory benefit for a number of reasons.

First, there are sentences for which gerundive nominals but no derived nominals exist. This suggests a morphological analysis; syntactic transformations should be regular and productive. Second, the meanings of derived nominals do not correspond to their verbal

Adopting the lexicalist hypothesis, while a very popular option, is by no means the only option. I adopt it here for concreteness.

5 (Jackendoff 1972, 13).
forms in the way that the gerundive nominals do. Finally, while derived nominals share many properties with nouns—the ability to appear with articles, be modified by adjectives, and take plural morphology—, gerundive nominals share none of these properties. These considerations strongly suggest that transformational derivation of (6.1 b) from (6.1 a) is completely unsatisfactory. According to (Chomsky 1970c), the differences noted between derived and gerundive nominals would simply be a “remarkable accident” from the point of view of the generative semanticist, and this would be a startling failure of explanatory adequacy. If we are to take natural language seriously, attempting to associate equivalence classes of sentences with some deep semantic representation is not an avenue which will be met with much empirical success.

(Jackendoff 1972) adds insult to injury here. He points out that if one adopts the Katz-Postal hypothesis, then focus effects must be read as a semantic ambiguity. Consider the passive transformation above, here with focus marked according to what is considered normal or default sentence stress:

(2.3) (a) Brutus killed CAESAR.

(b) Caesar was killed by BRUTUS.

Now consider the following responses:

(2.4) (a) No, Brutus killed Pompeii.

(b) No, Koba killed Caesar.
The response (2.4a) is acceptable as a response to (2.3a) but not to (2.3b). Conversely, the response (2.4b) is acceptable as a response to (2.3b) but not to (2.3a). The proponent of Katz-Postal would have to posit a difference at deep structure between (2.3a) and (2.3b) to accommodate this contrast. But this ambiguity contradicts what was posited earlier, that the passive transformation of (2.1a)/(2.1b) was a solely syntactic difference. If there are differences at deep structure, then a semantic difference is also predicted. As it turns out, what are semantically equivalent, on the assumptions of this theory, are the following two pairs:

(2.5) (a) BRUTUS killed Caesar.
   (b) Caesar was killed by BRUTUS.

(2.6) (a) Brutus killed CAESAR.
   (b) CAESAR was killed by Brutus.

Here the (a) and (b) cases are semantically equivalent, but (2.5a) and (2.6b) are not. But (2.5a) and (2.6b) are the standard focus readings, so it is a bit of a stretch to say that it was not an assumption above that (2.1a) and (2.1b) should be read with their default focus assignments. Here is a case where sentence meaning is affected by syntactic structure, namely the assignment of focus, in a way that is independent of the meanings of the words alone, providing a direct counterexample to the Katz-Postal hypothesis.
Chomskian Displacement

The system that replaced this model follows Chomsky in adopting the lexicalist hypothesis and, further, takes seriously that meaning can be affected by syntactic structure. This model contributes to the meaning of a sentence by generating an interpretable structure, LF, or logical form, governed by the rules of the human language system. Following (Chomsky 1995), a human language system, or Computational System of Human Language ($C_{HL}$), is the processing system that interfaces with the human sensorimotor system, the PF interface (Phonetic Form), and the human sound/gesture and interpretation system, the C-I interface (Conceptual-Interpretation).\textsuperscript{6} The PF interface and C-I interface rely on processes that are not dedicated to language. That is to say, neither the processes required to generate sounds and gestures nor the processes required to interpret sounds and gestures are necessarily dedicated to the human language system. There are, however, certain formal aspects of the lexicon and functional elements dedicated to the language system. $C_{HL}$ is a computational system connecting the form and interpretation systems. Thus, $C_{HL}$ must provide expressions that are well-formed according to both the PF and C-I systems. This provides a level of phonetic form and logical form, and the current working hypothesis is that $C_{HL}$ is the optimal

\textsuperscript{6}I say Conceptual-Interpretation here rather than Conceptual-Intensional because I would like to remain uncommitted to the function of the semantic system that interfaces with $C_{HL}$. As a type-driven framework is perfectly compatible with the output LF, I do not wish to beg any questions about the interpretation of this system on the whole.
system that allows for communication between the PF and C-I systems.

Given that the PF and C-I systems are independent, the dominant view is that at some point, what is called *spell-out*, the derivation must split, generating PF-interpretable and C-I interpretable structures: The level of logical form (LF) described here has its roots in (May 1985), and should be considered the contribution of syntax to meaning. This is distinct from the “logical form” of the artificial languages described above insofar as it is a legitimate linguistic level, in the sense of (Chomsky 1957). In fact, Chomsky goes so far as to suggest that these formal languages are only “languages” by metaphorical extension, as they do not exemplify properties of displacement common to all natural languages:
These “displacement” properties are one central syntactic respect in which natural languages differ from the symbolic systems devised for one or another purpose, sometimes called “languages” by metaphoric extension (formal languages, programming languages);⁷

As an example of displacement, take question formation:

(2.7) What did you say Jim brought?

In (2.7) what has two functions: it serves both to signal a question and is the direct object of brought. These two features can be seen in (2.8):

(2.8) What did you say Jim brought ⟨what⟩

Contemporary syntactic theory considers the leftmost what a copy of the rightmost what, and it is supposed that an effect of this copy is that all but the head of the chain ⟨what, what⟩ generated has had phonological features deleted, rendering the elements unpronounced. This differs from traditional movement, where the element moves from it’s source or argument position (A position) to its target position (A’ position), leaving an unpronounced trace element, t, behind.⁸

⁷(Chomsky 1995, 222).

⁸In line with current practice, let movement refer to this procedure, with unpronounced copies represented as traces.
Once this kind of displacement is witnessed, its prevalence and utility is astounding. Consider, for example, sentences involving quantification in object position, such as the following:

(2.9) Every student likes some professor.

There are two readings of (2.9), which correspond to the two possible scopes of the quantifiers: the $\forall \exists$ reading and the $\exists \forall$ reading. It is commonly assumed that a process of *quantifier raising* operates to generate the alternate scope reading. This is a straightforward displacement of *some professor*, where it moves above *every student*, leaving behind a trace that acts as a bound variable. The syntactic level at which this type of movement occurs is $\text{LF}$, which is considered a legitimate syntactic level due to the variety of syntactic constraints that have been witnessed to occur at this level. The contrasting picture, that of generative semantics, would seek to apply semantic interpretation to the deep structure of the sentence. The syntactic level that is generally considered prior to displacement of the type considered above. This is not to say that generative semantics would not be able to countenance an explanation like quantifier raising. On the contrary, this type of movement would be necessary in many cases. The difference between the views lies in the constraints placed on these transformations. Generative semantics is unable to constrain the transformations in the ways that generative syntax can with the syntactic level $\text{LF}$.

So the regimented metaphysician might be pulled toward a view that posits a deep structure similar to the well-formed formulas of the preferred logic of the day. What has been
witnessed, however, is a picture where syntactic structure, rather than being arbitrarily posited for semantic reasons, has a number of very natural, independent, constraints. Further, something very similar to what logicians might consider “logical form” appears in this model, which is generated from the class of logical expressions in the lexicon, expressions such as ‘or’, ‘and’, ‘must’, ‘might’, ‘every’, ‘some’, and so on. Interestingly, what might be considered the “logical” class of expressions in the lexicon cross-cuts the biologically occurring classes such as noun and verb. Thus, the syntax-semantics interface is not governed by an independent deep structure dictated by a particular choice of logical form; logical form is itself one of the outputs of the syntactic component, governed by rules and dictated by biology.

A basic sentence is traditionally composed of a subject and a verb. Contemporary syntax complicates this picture by adding what is known as the left periphery, a hierarchy of functional projections. Functional projections contains tense information about the subject and verb, as well as the complementizer that begins the sentence, and a landing site for word order inversions found when processing questions. This breakdown is simplistic, but it will suit the needs of the current project.

Lets start with the verb phrase, the terminating node of the left periphery.\(^9\)

\(^9\)Example trees taken from (Adger 2003).
The verb phrase $vP$ shows movement similar to the left periphery. This movement is required to handle ditransitive verb phrases. The ditransitive construction seems to require a ternary branching structure, which is avoided by positing additional movement. Ditransitive constructions involve predicates with three arguments:

(2.10) Pat gave the cloak to Mikey.

(2.11) Mikey received the cloak from Pat.

In most standard conceptions of the verb phrase, an event is being considered with, in this case, three participants: Pat, Mikey, and the cloak. The NPs and PPs are assigned Θ-roles by the predicates *give* and *receive*. To incorporate these constructions within a binary branching account, VP recursion must be posited. This is similar to recursion seen at the level of adjunct phrases, but adjuncts do not normally take Θ-role assignments. Note the following asymmetry due to the reflexive ‘himself’:
(2.12) *Emily showed himself Ben in the mirror.

(2.13) Emily showed Ben himself in the mirror.

By assuming that ditransitives have a hidden structure related to causation, known as a *causative* construction, these asymmetries can be resolved within a binary branching framework. The causative construction assumes a *light verb* $v$:

(2.14) Emily showed Ben himself in the mirror.
Within the hierarchy of projections, the light verb $vP$ always has $VP$ as its complement; it is an extension of the projection of $VP$.\textsuperscript{10} The object that satisfies the predicate then moves from $\bar{V}$ to $\bar{v}$\textsuperscript{11}. As it turns out, within the functional categories of a syntactic tree, this recursive structure appears frequently.

For example, consider the nominal phrase, which has been argued has a determiner head.

\textsuperscript{10}(Adger 2003, 131-136)
\textsuperscript{11}Use \langle . . . \rangle to denote traces.
Within the DP, there is a vP-like structure used to introduce the agent, theme, and goal. This is similar to the introduction of the causative structure within the vP. Little v and n are both obligatory projections. Even if there is no agent, the noun will raise from NP to n.\textsuperscript{12} Some linguists consider T to be the topmost of a series of functional categories within the clause. Unembedded sentences then start with a TP root:

\begin{center}
\begin{tikzpicture}


\end{tikzpicture}
\end{center}

\textsuperscript{12}(Adger 2003, 268-269)
DP

D    nP
  /
the  n   NP
  /
  cat  n   (cat)

(Internal diagram of noun phrase structure.)
The object that saturates the verbal predicate\textsuperscript{13} moves from $\bar{v}$ to the complement of $T$. The verbal predicate is then modified by a temporal predicate, which constrains the subject by a temporal relation: some relation to an interval of time. However, in certain cases, when the temporal relation is further embedded by a predicate of attitudes or some other verbal predicative structure, a complementizer phrase is necessary:\textsuperscript{14}

Here ‘does’ is a complementizer that is traditionally a pronominal element that has moved from the object of the verbal predication structure into $C$. Broadly speaking, in this case

\textsuperscript{13}“Subject” is no longer considered the best labeling system.

\textsuperscript{14}(Adger 2003, 289-340)
the pronominal element *does* serves to question the common ground on a specific topic. The information contained in the common ground of the conversation will entail an answer, which the conversational participant best positioned to answer will provide. If a clausal complement carries a temporal predication structure, it is considered *finite*.

Once this structure is in place, an interesting distinction can be made. The distinction surfaces when verbs are considered that can take either a finite complement or a non-finite complement. Consider ‘believe’:

(2.15) London believed \([_TP\text{Sherwood would go}]\).

(2.16) London believed \([_CP\text{that }_TP\text{Sherwood would go to the cinema}]\).

The distinction being made is between a non-finite temporal predication structure and a finite complementizer headed structure. It is always possible to add a null CP in (2.15) to make the structures in (2.15)/(2.16) identical. But the interesting counterpart is to distinguish (2.15) from (2.16). This viewpoint is more efficient, it allows *v* to carry accusative case, which explains exceptional case marking (ECM) by distinguishing it from control structures and *for*-clauses:/footnoteNote that I follow Boeckx et. al. (2006) in treating control as movement in my own theory, eliminating the use of PRO. This does not, however, collapse the two conceptions, as there are many empirical differences. I provide this alternative, which is in fact more commonplace in linguistics, as an example of an empirical stumbling block to an *a priori* theory.
What London arranged was [CP for [DP Sherwood to accept the call]]. (for-)

What Sherwood attempted was [CP PRO [TP to sell the car]]. (control)

*What Sherwood believed was [CP London [TP to be a buyer]]. (ECM)

Tarah believed [TP the report]. (ECM)

In each of these cases, the verbal structure take a CP complement, except in ECM constructions, where v carries an accusative θ-role. Distinguish these cases from the case of raising.

In a raising construction, certain single argument verbs such as *seem* and *appear* allow both non-finite and finite clausal constructions:

(2.21) It seems [CP that [TP Jose left]].

(2.22) Jose seems [TP to have left].

The same θ-roles are assigned in (2.21) as in (2.22). Most contemporary views would want to assert that a proposition would be assigned θ-roles in both cases, even though the complement clause differs between (2.21) and (2.22). In (2.21) there is a traditional complementizer clause. On the contrary, in (2.22) subject-to-subject raising eliminates the necessity of a CP layer. In fact, ECM constructions are explained by this difference. This lends theoretical weight to the claim that the syntactic structures differ in spite of their θ-role assignments. This complicates the definition of proposition, however, because the syntactic structures now differ, without a seeming difference in proposition. Consider the difference between (2.21)/(2.22):
(2.23) It seems that Jose left.

(2.24) Jose seems to have left.
This is a semantic-first methodology that opposes the syntactic nature of ES. There is a
difference of opinion in linguistics whether a biclausal construction model, where a CP is always present, is the correct model of human language. This relegates the theory of propositions to a theory of argument structure. The arguments that follow suggest that argument structure is informed in certain ways by the syntax. In the following chapters the semantic side, the LF, will take form.

With these considerations in mind, remember now that our target is the assumption sententiality, restated here:

\textbf{ES.} There exist embedding environments \((E)\) in natural language that take sentential arguments \((S)\).

(Cappelen and Hawthorne 2009) state this as follows:\textsuperscript{15}

As applied to a given expression \(E\), the key syntactic assumption made about \(E\) has to do with \textit{how} it syntactically combines with other expressions to generate yet larger expressions. The assumption of Sententiality is that \(E\) combines with one or more \textit{sentences} to yield larger sentences.

What I have been attributing to the proponents of \(ES\) up to this point is the adherence to a view that looks something like the following diagram: Here, on the right-hand-side, I have placed the main players we have discussed so far: Montague, Prior, Kaplan, and Lewis. Each of this group, in their own way, constructs an intensional logic in order to describe certain

\textsuperscript{15}(Cappelen and Hawthorne 2009, 70).
Figure 2.3: ES Assumption
phenomena in natural language. Indeed, this is the theme, and anyone who considers an intensional language an appropriate model for some natural language phenomena should be considered members in this category. On the left-hand-side, I have placed Chomsky’s program of generative grammar. This model posits a computational module $C_{HL}$ that produces well-formed expressions at the level of PF and LF. At LF, the level of interest here, (Klein and Sag 1985), (Bittner 1994), and (Heim and Kratzer 1998), among others, have shown how a direct type-driven interpretation can be applied to LF, providing a well-founded semantics. What I have claimed is that the group on the right provide a revisionary reading of LF, enforcing their assumption $ES$. Further, if by commitment to STRUCTURE DEPENDENCE and LINGUISTIC UNIVERSALS or some other commitment to realism, in either its weak or strong form, you claim to describe reality but fail to respect the best empirical models, you have a weakness in your conceptual system. I have never seen a convincing a priori argument that empirical data should be ignored.

It is worth noting that while Montague might be considered an additional candidate on the right-hand-side, his contribution is somewhat of an outlier in the above description of the situation. Specifically, (Montague 1970), (Montague 1973) provides the foundation for the approaches taken by (Klein and Sag 1985), (Bittner 1994), (Partee 2004), and (Heim and Kratzer 1998). Application of the typed lambda calculus was indispensable in the development of a rich and descriptive semantic program. However, Montague himself constructed an artificial syntax for his language, defining a set of meaningful expressions $ME_{\alpha}$ isomorphic
to the recursively defined hierarchy of types. It was not until later that the typed lambda calculus would be brought more closely together with the generative program in syntax. At the time, Montague considered himself in agreement with the Chomskian program:¹⁶

There is in my opinion no important theoretical difference between natural languages and the artificial languages of logicians; indeed, I consider it possible to comprehend the syntax and semantics of both kinds of languages within a single natural and mathematically precise theory. On this point I differ from a number of philosophers, but agree, I believe, with Chomsky and his associates.

There is a kernel of truth to Montague’s statement here: his use of the lambda calculus is integral to the contemporary approach. However, his syntax is artificial, not empirical. This provides a transparent syntax-semantics interface, albeit a stipulated one. While this might be considered a formal virtue, it is not explanatory in the way we require. The two camps on the left-hand-side and right-hand-side of the diagram above can then be characterized by branching from two different elements of Montague’s approach. Those on the left-hand side began applying type-driven interpretations directly to trees derived by the Chomskian model, while those on the right-hand-side took up the idea that a sufficiently rich intensional system can be used well-enough to describe natural language. I take the left-handed approach.

My opponents might attempt a retreat: perhaps the left-hand-side of the diagram is not “real” or “serious”, in some way philosophically insignificant. Rather than reading ES as

¹⁶(Montague 1970, 1).
an assumption that bridges from the right to the left, there need not be any bridge at all. There could be posited such a thing as “philosophical semantics”, where formal models are developed in a sufficiently rich intensional logic to describe the joints of reality.

If one lends credence to the connection between linguistic and conceptual structure and the structure of reality argued for so far, then it is certainly unclear how a level of logical form posited by the best empirical model of humankind’s linguistic capacity could fail to be philosophically interesting in this way. Because of this, I redraw the diagram above according to what I consider a less revisionary method of bridging the left- and right-hand side of the diagram.

Here the Curry-Howard Correspondence provides a computational bridge between the Logical Form of Generative Grammar and Constructive Logic. The computational basis provided by the untyped λ-calculus I claim suffices to connect our linguistic faculties to
the computational logic processing unit that is modeled in the ideal by artificial logical systems. In fact, where typing is concerned, I develop a system $\lambda P$, also known as $\lambda LF$, which is capable of representing logical form within the generative program. I then go on to show that within this system, a number of the arguments made by the supporters of the operators argument may be explained compositionally. Finally, I explore a modified version of Kratzer’s system of modality within the extensional framework developed. This version relies on (Frank 1996) and generalizes to multiple ordering sources, providing a unified account of deontic and epistemic modality.
Chapter 3

Logic and the \(\lambda\)-Calculus

According to (Pierce 2002),

A type system is a tractable syntactic method for proving the absence of certain program behaviors by classifying phrases according to the kinds of values they compute.

We will be building a type system on top of the \(\lambda\)-Calculus, a small Turing-complete language. To model natural language, the type system will be classifying phrases according to how they compose with each other. Computation will terminate with a \texttt{BOOL} value. The type system itself tracks the nature of the higher order properties which are matched with generative syntax trees. I will be following (Michaelson 2011), (Pierce 2002), (Hindley and Seldin 2008), and (Barendregt 2012).
λ Calculus (λβ)

Following (Hindley and Seldin 2008) and (Pierce 2002), assume there is given a countable sequence of variables \(v_1, v_2, v_3, \ldots \in \mathcal{V}\). The set of \(\lambda\)-terms is the inductively defined set \(\mathcal{T}\):

(i.) \(\mathcal{V} \subseteq \mathcal{T}\) (atoms).

(ii.) if \(M, N \in \mathcal{T}\), then \((MN) \in \mathcal{T}\) (application).

(iii.) if \(M \in \mathcal{T}\) and \(x \in \mathcal{V}\), then \((\lambda x.M) \in \mathcal{T}\).

Scope For a particular occurrence of \(\lambda x.M\) in a term \(A\), the term \(M\) of the occurrence \(\lambda x\) on the left is called the scope.

Free and Bound Variables An occurrence of a variable \(x\) in a term \(A\) is called

(i.) bound if it is in the scope of a \(\lambda x\) in \(A\),

(ii.) bound and binding if it is the \(x\) in \(\lambda x\),

(iii.) free otherwise.

A term \(x\) is a bound variable of \(A\) if \(x\) has at least one binding occurrence in \(A\). A term \(x\) is a free variable of \(A\) if \(x\) has at least one free occurrence in \(A\). The set of all free variables in \(A\) is \(FV(A)\). A closed term does not have any free variables.

Substitution For any \(x, M, N\) define \([N/x]M\) or \([N \rightarrow x]M\) to be the result of substituting \(N\) for every free occurrence of \(x\) in \(M\).
**α-conversion** Let a term $A$ contain an occurrence of $\lambda x. M$, and let $y \notin FV(M)$. Replacing $\lambda x. M$ for $\lambda y. [y/x]M$ is called an α-conversion in $A$. $A$ α-converts to $B$, $A \equiv_\alpha B$, iff $A$ can be changed to $B$ by a finite series of changes in bound variables.

**Lemma** If $A \equiv_\alpha B$, then $FV(A) = FV(B)$. The relation $\equiv_\alpha$ is an equivalence relation.

**β-reduction** Any term $(\lambda x. M)N$ is called a β-redex and the term $[N/x]M$ is its contractum. If a term $A$ contains an occurrence of $(\lambda x. M)N$, the result $A'$ of replacing that occurrence with $[N/x]M$ is a contraction of the redex occurrence in $A$. We note this with $A \triangleright_1 \beta A'$. If a term $A$ can be changed to a term $B$ by a finite series of $\triangleright_1 \beta$ contractions and α-conversions, we say $A$ β-reduces to $B$: $A \triangleright_\beta B$.

**Lemma** $A \triangleright_\beta B \Rightarrow FV(A) \supseteq FV(B)$.

**β-equality** $A$ is β-convertible to $B$, $A =_\beta B$, iff $A$ can be obtained from $B$ by a finite or infinite series of β-contractions, reversed β-contractions, and α-conversions. A term is β-equal to at most one β-normal form, modulo changes in bound variables.

**λI-terms** λI-terms are the λ-terms defined in definition 1.1, except that $\lambda x. M$ is allowed to be a λI-term only when $x$ occurs free in $M$. This is the system of only closed terms. The full system is called the λK-system and its terms are called λK-terms.

**Omega** $(\Omega) (\lambda x.xx)(\lambda x.xx)$
The function $\Omega$ is an infinite redex: $\Omega \triangleright_{\beta} \Omega \triangleright_{\beta} \Omega \triangleright_{\beta} \ldots$. This term has a stable form but infinite reduction path, which differs from a term such as:

$$(\lambda x.xxx)(\lambda x.xxx) \triangleright_{\beta}$$

$$(\lambda x.xxx)(\lambda x.xxx)(\lambda x.xxx) \triangleright_{\beta}$$

$$(\lambda x.xxx)(\lambda x.xxx)(\lambda x.xxx)(\lambda x.xxx) \triangleright_{\beta}$$

\ldots

This term, also allowable in $\lambda K$, enlarges itself through $\beta$-reduction. The $\triangleright_{\beta}$ path models the growth path of the successor function. These properties are all properties of recursive functions. A modification to $\Omega$ is useful to define a fixed point term:\(^1\)

**Fixed Point (Z)** $\lambda f.((\lambda x.f(\lambda y.xxx))(\lambda x.f(\lambda y.xxx)))$

Fixed points can be used to recursively define mathematical functions. This function is formulated as a *call-by-value* function. This evaluation strategy requires that a functions argument be evaluated before it is passed to the function. By contrast, in a *call-by-name* setting, the function argument is substituted directly into the function body, modulo $\alpha$-conversion. In a call-by-value setting, free variables should be thought of as ranging over

\(^1\)This construction is (Plotkin 1975)'s Z combinator. It is often called the *call by value* fixed point combinator. Cf. (Pierce 2002, 65).
values, not terms.\textsuperscript{2} Crucially, in this setting, the term $\Omega$ is considered a value. However, if an extensionality axiom is added, $\Omega$ can no longer be considered a value.\textsuperscript{3}

In a call-by-name setting, on the other hand, the simpler fixed-point term may be used:

**Fixed Point (Y)** $\lambda f. (\lambda x. (f(xx)) \lambda x. (f(xx)))$

This is due to the fact that the infinite cycle may be passed as a name without first evaluating it. This combinator has the property of evaluating into a new version of itself at each iteration: $Y \beta Y' \beta Y'' \beta \ldots$.\textsuperscript{4}

(Church 1936) defined the $\lambda$-definable functions over the positive integers and showed their equivalence to the general recursive functions. (Turing 1937) proved that $\lambda$-definability and Turing computability coincide, but was unaware of Church’s work on recursive functions. (Rosser 1939) pointed out that the turing computable functions are equivalent to the general recursive/$\lambda$-definable functions. (Post 1936) considered this a natural law. (Kleene 1952, 320) calls this equivalence a fundamental class.\textsuperscript{5} This principle is the Church-Turing Thesis:

**Church-Turing Thesis** a function is Turing computable iff it is $\lambda$-definable iff it is general recursive.

The fixed point combinator implements recursion. Lacking the fixed point combinator,

\textsuperscript{2}Cf. (Plotkin 1975, 135)
\textsuperscript{3}Cf. (Plotkin 1975, 136). To be specific, Plotkin’s Theorem 1 provides a counter-example to the Church-Rosser property.
\textsuperscript{4}Cf. (Sørensen and Urzyczyn 2006, 11)
\textsuperscript{5}Cf. (Lobina 2017)
the system is not turing complete. An operational semantics capable of defining the \( \lambda \)-computable functions will be provided below.

According to (Barendregt 2012, 187.9.2) there exists a system of numerals upon which the partial recursive functions may be defined in \( \lambda I \), so at this point the choice between the two systems seems arbitrary. There may, however, be linguistic reasons to value one system over the other. For example, (Jacobson 1999), (Jacobson 2002), and (Jacobson 2007) forwards a semantics where pronouns are represented by the term \( \lambda x.x \). This is a closed term representation, rather than an open variable representation, which I will pursue here. The \( \lambda I \) system is a restriction of the \( \lambda K \) system to only closed terms, which suggests that Jacobson’s may be after a restriction to \( \lambda I \). However, due to the properties of the \( \lambda \)-binder and its correspondence to the \( K \) combinator, I find this avenue of approach an unlikely contender.

The system \( \lambda I \) recommends itself on the basis of the conservation theorem.

**Conservation Theorem (\( \lambda I \))** Let \( WN_\beta \) be the set of terms s.t. \( M \in WN_\beta \) iff there exists a normal form \( N \) s.t. \( M \rhd_\beta \ldots \rhd_\beta N \). The set \( WN_\beta \) is the set of weakly normalizing terms. Let \( SN_\beta \) be the set of terms s.t. \( M \in SN_\beta \) iff all reduction sequences starting with \( M \) are finite. If \( M \not\in SN_\beta \), we write \( M \in \infty_\beta \).

The difference between \( WN_\beta \) and \( SN_\beta \) is that weakly normalizing terms can have infinite reduction paths. With these terms, however, there exists at least one reduction path that is finite.
Finite Reduction Paths  For any weakly normalizing term $\alpha$, there exists an algorithmic process to find a finite reduction path for $\alpha$.

Given that there are algorithms for finding both finite and infinite reduction paths for $WN_3$ terms, I do not see any issue with adopting an approach that is weakly normalizing. However, when higher-order type theories are introduced, the theories that we will be considering are all strongly normalizing. It is worth noting that by adopting strongly normalizing theories, recursion is rejected. I place the locus of recursion in the grammar, when considering the biological mechanism we are studying.

From the overall perspective of human cognition, we have recursive faculties of thought, I just don’t believe they are a model requirement necessary for the semantics of natural language. Recursive syntactic structures may still be modeled academically. However, the semantics of the terms involved do not require the evaluation of fixed points.

We have just developed two distinctions that cross-cut one another: the distinction between call-by-name and call-by-value and the distinction between weak normalization and strong normalization. At this point the introduction of type systems provide a set of stricter requirements. I stay conservative in my arguments, preferring the weight of the full system to lend justification to the particular parts I use in my interpretation functions.

6See (Sørensen and Urzyczyn 2006, §1.5)
Combinatory Logic (CLw)

(Pierce 2002, 55) calls all closed terms *combinators*. (Hindley and Seldin 2008) studies the connection between the $\lambda$-calculus and *combinatory logic*. I will be using a system that is capable of handling traces, which are normally considered open variables, without resorting to the combinator $I$, as Jacobson does. However, I will be developing a $\lambda$-abstraction neutral theory, building in the operator $K$ into the semantics of composition. This will lead us to require a reconfiguration of some movement principles, and could, in principle be extended with $I,S$ to make a fully functional combinatorial logic. I leave $S$ as an understood part of our system, it is represented by the connecting edges of our binary tree. I go on now to introduce the combinatory system and its connection to the $\lambda$-calculus.

**Definition 2.1** Assume that the set of variables $V$ above is the set of constants $C$, where the set $T$ is the same as above. Now add three basic combinators $I, K, S$. Define the inductive set of *CL-terms* $T$:

(i.) $C \subseteq T$

(ii.) $I, K, S \in T$

(iii.) if $X, Y \in T$, then $(XY) \in T$.

The set $C$ is the set of constants or *atomic terms*. A *closed term* is a term containing no variables. A *combinator* is a term whose only atoms are basic combinators.

(Hindley and Seldin 2008) defines a type of *weak reduction* $\triangleright_w$ equivalent to $\lambda\beta$'s $\triangleright_\beta$. 
**Weak Reduction** Any term IX, KXY or SXYZ is called a weak redex. Contracting an occurrence of a weak redex to a term U means replacing one occurrence of IX by X, KXY by X. or SXYZ by XZ(YZ). If this changes U to U’, we say that U (weakly) contracts to U’: U ⊳₁wu’. If V is obtained from U by a finite series of weak contractions, we say that U (weakly) reduces to V: U ⊳ₚw V.

Both ⊳ₚ and ⊳ₚw have the Church-Rosser property⁷ that reductions are order independent or confluent. Further, CLw can define an abstraction operation:

**Abstraction** For every CL-term P and every variable x, a CL-term λ∗x.P is defined by induction on P.⁸

(i.) λ∗x.x ≡ I,

(ii.) λ∗x.P ≡ KP if x ∉ FV(P),

(iii.) λ∗x.PQ ≡ S(λ∗x.P)(λ∗x.Q).

One issue with the λ-calculus is the role played by bound variables in connection to substitution and variable capture specifically. The notion of free and bound variables is replaced here by a set of constants and a set of axioms that corresponds to λ-introduction.

⁷(Church and Rosser 1936)

⁸(Barendregt 2012, 152) Definition 7.2.1. Notation taken from (Hindley and Seldin 2008).
Church Booleans and Numerals

With this basic framework in place, logical operations can be defined following (Michaelson 2011). We start with two basic boolean functions and a currying function, which makes a pair out of two arguments and then feeds them into a higher order function:

**TRU** \( \lambda x. \lambda y. x \)

**FLS** \( \lambda x. \lambda y. y \)

**PAIR** \( \lambda x. \lambda y. \lambda f.((fx)y) \)

Feeding a sequence \( \langle a, b, \text{TRU} \rangle \) returns ‘a’:

\[
\lambda x. \lambda y. \lambda f.((fx)y)(a)(b)(\lambda x. \lambda y. x) \triangleright_{\beta}
\]

\[
\lambda y. \lambda f.((fa)y)(b)(\lambda x. \lambda y. x) \triangleright_{\beta}
\]

\[
\lambda f.((fa)b)(\lambda x. \lambda y. x) \triangleright_{\beta}
\]

\[
\lambda x. \lambda y.((xa)b) \triangleright_{\beta}
\]

\[
\lambda y.(ab) \triangleright_{\beta}
\]

\[
a
\]

Whereas the sequence \( \langle a, b, \text{FLS} \rangle \) returns ‘b’:

\[
\lambda x. \lambda y. \lambda f.((fx)y)(a)(b)(\lambda x. \lambda y. y) \triangleright_{\beta}
\]
\[
\lambda y.\lambda f.((fa)y)(b)(\lambda x.\lambda y.y) \triangleright_\beta
\]

\[
\lambda f.((fa)b)(\lambda x.\lambda y.y) \triangleright_\beta
\]

\[
\lambda x.\lambda y.((ya)b) \triangleright_\beta
\]

\[
\lambda y.(yb) \triangleright_\beta
\]

\[
b
\]

The pairing function provides the necessary combinatorics to generate boolean values from the functions TRU and FLS, which simply choose the first or second element in a pair.

We can then define the boolean operations:

\textbf{NOT} \quad \lambda x.(((\text{PAIR FLS})\text{TRU})x)

\textbf{AND} \quad \lambda x.\lambda y.(((\text{PAIR } y)\text{FLS})x)

\textbf{OR} \quad \lambda x.\lambda y.(((\text{PAIR TRU})y)x)

As would be predicted, NOT TRU evaluates to FLS and NOT FLS evaluates to TRU. AND FLS evaluates to FLS and so does AND TRUE FLS. OR evaluates to TRUE in the expected cases.

The natural numbers may also be generated:

\textbf{ZERO} \quad \lambda x.x

\textbf{SUC} \quad \lambda n.\lambda s.((s \text{ FLS})n)

\textbf{ISZERO} \quad \lambda n.(n \text{ TRU})

\textbf{PRE} \quad \lambda n.(((\text{ISZERO } n)(n \text{ FLS}))
Operational Semantics $\lambda_{\text{NB}}$

Terms are the result of syntactic formation rules. Values are the potential results of evaluation: terms of a particularly simple form. I follow the implementation of operational semantics found in (Pierce 2002) which is derived from (Plotkin 1981). Terms are defined in BNF form:

**TERMS** $t ::= x | \lambda x.t | t \ t | \top | \bot | \text{if t then t else t} | 0 | \text{succ t} | \text{pred t} | \text{iszero t} | \text{nv}$

**VALUES** $v ::= \lambda x.t | \top | \bot | \text{nv}$

**NUMERIC VALUES** $\text{nv} ::= 0 | \text{succ nv}$

**EVALUATION RULES** $t \rightarrow t'$

\[
\frac{t_1 \rightarrow t'_1}{t_1 \ t_2 \rightarrow t'_1 \ t_2} \quad \text{(E-App1)}
\]

\[
\frac{t_2 \rightarrow t'_2}{v_1 \ t_2 \rightarrow v_1 \ t'_2} \quad \text{(E-App2)}
\]

\[
(\lambda x.t_{12})v_2 \rightarrow [v_2/x]t_{12} \quad \text{(E-AppAbs)}
\]

\[
\text{if } \top \text{ then } t_2 \text{ else } t_3 \rightarrow t_2 \quad \text{(E-IfTrue)}
\]

\[
\text{if } \bot \text{ then } t_2 \text{ else } t_3 \rightarrow t_3 \quad \text{(E-IfFalse)}
\]

\[
\frac{t_1 \rightarrow t'_1}{\text{if } t_1 \text{ then } t_2 \text{ else } t_3 \rightarrow \text{if } t'_1 \text{ then } t_2 \text{ else } t_3} \quad \text{(E-If)}
\]
\[
\begin{align*}
    t_1 \rightarrow t'_1 & \quad \text{(E-Succ)} \\
    \text{succ } t_1 \rightarrow \text{succ } t'_1 & \\
    \text{pred } 0 \rightarrow 0 & \quad \text{(E-PredZero)} \\
    \text{pred } (\text{succ } n v_1) \rightarrow n v_1 & \quad \text{(E-PredSucc)} \\
    t_1 \rightarrow t'_1 & \quad \text{(E-Pred)} \\
    \text{pred } t_1 \rightarrow \text{pred } t'_1 & \\
    \text{iszero } 0 \rightarrow \top & \quad \text{(E-IsZeroZero)} \\
    \text{iszero } (\text{succ } n v_1) \rightarrow \bot & \quad \text{(E-IsZeroSucc)} \\
    t_1 \rightarrow t'_1 & \quad \text{(E-IsZero)} \\
    \text{iszero } t_1 \rightarrow \text{iszero } t'_1 &
\end{align*}
\]

These operational rules define boolean operations and the natural number line. The evaluation or reduction relation \( \rightarrow \) is rigorously studied in (Pierce 2002). I provide the rules of computation that this relation expresses.\(^9\) The rules come in two forms: congruence rules and computation rules. Congruence rules (E-App1), (E-App2), (E-If), (E-Succ), (E-Pred), (E-IsZero) determine a particular evaluation strategy for conditional statements. Computation rules (E-AppAbs), (E-IfTrue), (E-IfFalse), (E-PredZero), (E-PredSucc), (E-IsZeroZero), (E-IsZeroSucc) tell us what to do when we reach a conditional statement whose guard is already false.

\(^9\)I do not formalize stuck states in these definitions, although a simple extension covered in (Pierce 2002) 3.5.16 could be developed. What could be formalized as run-time errors I consider meaningless states (e.g., \text{succ true}).
fully evaluated. Let the arabic numerals refer to the sequence \(0, \text{succ } 0, \text{succ succ } 0, \ldots\)
defined recursively by \textsc{numeric values} above.

These operational definitions have completely defined the structure of the church booleans and numerals above with primitive booleans and numbers. Specifically, one could use church booleans and numerals within the \(\lambda\beta\) system and accomplish everything that can be accomplished within the \(\lambda\mathbb{N}\mathbb{B}\) system. In this sense the \(\lambda\mathbb{N}\mathbb{B}\) system reduces to the \(\lambda\beta\) system, but I find the operational definitions and proof method easier to work with.

\textit{Operational Semantics}, pursued here, defines an abstract machine which has a state, with several components, and a set of primitive instructions. The machine is specified by showing how the components of the state are changed by the primitive instructions. The semantics of the programming language is defined in terms of the machine. This machine abstracts from questions of efficiency and instead focus on eliminating ambiguity of order code. The semantic description of the programming language specifies a translation into this code. The code in this case is defined by the \(\lambda\)-calculus. The abstract machine is a representation of Chomsky’s \(C_{HL}\). The operational semantics specifies a translation between a Chomsky Grammar and \(\lambda\mathbb{N}\mathbb{B}\).

To draw the connection I want between my system and natural language, I draw some examples from early Programming Language Theory. When programming languages were first being developed, the mathematical question asked was how to describe the semantics of all possible programs of a particular programming language. The most direct way to do this
is with a complete implementation. Languages were “as defined by” a particular compiler.\textsuperscript{10} This provided an implementation specific definition. What was sought was a ‘standard implementation’. A standard implementation is more general than a compiler specific implementation, as the choice of type system and verification method is compiler specific. The method of generalizing from specific implementations to a generic implementation involves translating program text into a parse tree and then translating the parse tree into a generic program for a standardized abstract machine. This abstract machine, via translation rules, provides a machine state transformation (MST) which is identical to that of the programs execution by a computer.

As applied to natural language, the enterprise pursued here could be considered the mathematical development of an abstract machine. This machine represents $C_{HL}$ in that $C_{HL}$ contains an standard implementation for a linguistic compiler, where sentences in a language are considered programs to be executed when spoken by those processes of $C_{HL}$ that allow for processing the meaning of terms in a shared language. Each compiler is language specific, and these language specific compilers assign types to meaningful values to ensure their well-formedness for linguistic execution. The problem here is that the compiler itself, as well as the actual implementation of $C_{HL}$, is unknown to us. All that is left is the mathematical model and an empirical success scale based on coverage of data in accordance to the best speaking practices within a language. If the MST itself is unknown to us, as the actual computer is

\textsuperscript{10}Cf. (Stoy 1977, 14)
an organic masterpiece, all that we are left with is the mathematical simulation. For this reason operational semantics, which provides a state-based transition machine is sufficient to provide a high-level language usable by linguists to compare their programs. One of the main benefits of operational semantics is that it suggests techniques for implementing the language to the compiler writers. Since the broad question of concern here is which particular abstract implementation best represents the organic processes of $C_{HL}$, all formal linguists should consider themselves compiler writers. At this point the entire enterprise consists of the study of syntactic symbol manipulation. A symbolic string of natural language is parsed in a formal Chomsky grammar and then theoretically “compiled” by an operational semantics, which ensures its derivational correctness. “Meanings” in Programming Language Theory were simply the MST of the program. *Denotational semantics*, on the other hand, is concerned with providing a canonical function that represents the abstract meanings of a program on the basis of the meanings of its sub-components. The techniques available here lend themselves to language designers, who are concerned with the mathematical correctness of their programs.

Denotational semantics specifies an abstract mathematical value in some suitable value space that represents the meaning of an program parse tree. This lends itself to linguistic analysis in that the MST can be completely avoided, as it may not be relevant to $C_{HL}$. This allows for the identity of parse trees based on identity of their abstract value. Providing such an abstract structure to compare programs allows for the discussion of *correctness*
of implementation—that equal expressions remain equal under the implementation. It also allows various standards of effectiveness to be defined.\footnote{Cf. (Stoy 1977, 16)} Thus, denotational semantics provides a normative environment from which one may validate the operational tools used to solve specific problems within formal linguistic semantics. A specific tool provided by the denotational approach which will be considered in what follows is the ability to simulate imperative commands within a referentially transparent environment.

In a referentially transparent environment, (Stoy 1977, 190) points out that open variables are well behaved:

**Well-Behaved Variables** A variable $v$ is *well-behaved* just in case one can show by structural induction on terms that all occurrences of $v$ denote the same value.

A call-by-value setting compromises the well-behavedness of variables even if we are not in a state-based system.

The term *referential transparency* was first used by (Whitehead and Russell 1910-1913) to contrast the logic of the following arguments:

1. All men are mortal.
2. Socrates is a man.

\[\therefore\] Socrates is mortal.

1. Everything Xenophon said about Socrates is true.
2. Xenophon said: “Socrates is mortal”.

jadi. Socrates is mortal.

Whitehead and Russell contrasted the content of an argument from the circumstances of its assertion, the context. All of the statements in the first argument are referentially transparent. However, in the second argument, premise (2) requires that both of the facts that Xenophon said $p$ and that $p$ is ‘about Socrates’ should be added to the argument for it to be valid.

In (Quine 1960b), ‘referential transparency’ is used to refer to the substitutivity of identities. Contrast the following two sentences:

1. Tully was a Roman.

2. William Rufus was so-called because of the colour of his hair.

In (1), ‘Tully’ may be replaced by ‘Cicero’ in a truth-functionally equivalent manner. In (2), however, replacing ‘William Rufus’ by ‘King William II’, the sentence becomes untrue. Another name for the second cases above is an opaque environment. I follow (Stoy 1977) in using the term Referential Transparency to mean a fact of mathematics:12

The only thing that matters about an expression is its value, and any subexpression can be replaced by any other equal in value. Moreover, the value of an expression is, within certain limits [of scope], the same whenever it occurs.

12(Stoy 1977, 5)
This mathematical fact seems to be violated in the context of opaque environments.\textsuperscript{13} Contemporary programming languages have developed imperative operations which violate this principle directly:\textsuperscript{14}

**Imperative Definition** if $x > y$ then $x := x - 1$

In this example the value of $x$ depends on the value of $y$ within the scope of the declaration of $x$. These imperative command definitions, where the identity sign is prefixed with a colon, can be represented in a language where referential transparency holds. This is one main reason for choosing the $\lambda$-calculus to provide a semantic description of our language.

In computer science, functional programming languages, of which the $\lambda$-calculus is the simplest, differ from the more traditional *imperative* programming languages in ways that parallel the differences we have been tracking.

Traditional imperative programming languages revolve around changeable associations between names and values: *variables*. Imperative languages consist of sequences of commands, each command an *assignment* which changes a variables value. This involves working out the value of an expression and associating it with a name:\textsuperscript{15}

\[
\langle \text{name} \rangle := \langle \text{expression} \rangle
\]

\textsuperscript{13} *Opacity* is a term coined by Quine to refer to those environments in natural language that violate referential transparency.

\textsuperscript{14} Cf. (Stoy 1977, 6)

\textsuperscript{15} (Michaelson 2011, 1-14)
The commands link together by referring to multiple variables whose values have been changed by preceding commands. Values are passed through this chain of command.

The common paradigm in linguistics is a boolean valued state based system. The effects of executing a command is a change of state. The appropriate value of a command, then is a state transformation function. If \( S \) is a set of states, then \( \gamma := S \rightarrow S \) is the value of a command operation. Expressions have results (which depend on the state), and these results are confined to the \texttt{BOOL} domain of truth conditions. So in general, an expression value is a function \( \delta := S \rightarrow \texttt{BOOL} \) that maps states to results. Generalized quantifiers are a good example of this type of function. A generalized quantifier is of type \( \langle \varphi, \texttt{BOOL} \rangle \). Predicate modification is in general a command operation, the command being a restriction on a predicate.

Functional languages, based on a hierarchy of nested function calls, receive and pass values as functions compose with each other. Names are only introduced as the formal parameters of functions. Once a name has been given an actual value there is no way for that name to be associated with a new value. This generates an execution order sensitivity to imperative languages that functional languages lack. It is much easier to parallel process functional languages due to this execution order insensitivity.

This directly mirrors intensional logic, where logical operators use assignment functions to shift the value of an evaluation sequence. This operation is not allowed in a strict functional language. Thus the confusion. (Michaelson 2011, 3) states this point forcefully:
Once a formal parameter is associated with an actual parameter value there is no way for it to be associated with a new value. There is no concept of a command which changes the value of a name through assignment. Thus, there is no concept of a command sequence or command repetition to enable successive changes to values associated with names.

As an example, consider the following function call:

\[ F(A(D), B(D), C(D)) \]

The parameter \( D \) is shared as inputs to the functions \( A, B, C \), but the execution orders of these functions does not matter. The value of parameter \( D \) cannot be changed by the functions \( A, B, C \). Functional languages are referentially transparent. While imperative languages can associate new values with the same name throughout the processing of new commands, functional languages rely on recursive function calls instead. The values of names cannot be shifted. If a new value is introduced, it must correspond to a new name.

One perceived downside to functional programming languages are their handling of data structures. The substructure of data structures cannot be changed \textit{a la carte} in functional languages because they lack an assignment function. Instead, data structures like \textit{lists} are used, where operations on the whole structure are described by recursive substructural operations. Since there is no assignment, if a substructural change needs to be made to a structure already in place, the entire data structure must be passed explicitly as an actual parameter of a function.
(Landin 1964),(Landin 1966) proposed the use of the λ-calculus to model imperative languages. Landin’s Stack, Environment, Control, Dump (SECD) machine was one theoretical attempt. It involved an operational description of the λ-calculus defined in terms of an abstract interpreter, the SECD machine. In (Strachey 1966) and (Scott and Strachey 1971) approached the problem by constructing a denotational semantics where every imperative language construct would have an equivalent function denotation. Some current approaches in linguistics follow this vein. For example, (Charlow 2017), (Kennedy 2014), (Kobele 2010), and (Sternefeld 1997) all model imperative assignment functions within functional languages.

Continuations and jumps developed by (Strachey and Wadsworth 1974) and implemented in linguistics by (Barker and Shah 2014), among others, are the result of the early work attempting to model imperative languages in a functional calculus. Their success has not yet been fully appreciated by the community, and I hope to add to the intelligibility of these operations and their uses in philosophy and linguistics. I start by introducing a typing system, which then leads to the introduction of the unit type and a store, which is then capable of modeling an assignment function within the system. This then clarifies the response I am making to the operators argument and classifies it as a case of referential transparency failure, not compositionality failure. As the system is based on the typed lambda calculus, a functional language, it can be said to be fully compositional.
Typing

There are two different approaches to typing systems which could be adopted here. First, the evaluation relation ‘→’ could be defined directly on the syntax of a simply typed calculus. In this Church-style system (Church 1940), typing is prior to semantics. Terms which are not well-typed do not exist in the system. The converse, Curry-style system (Curry and Feys 1958) first defines terms and provides them with a semantics showing how they behave. Then, a type system is developed which rejects the behaviors of some terms we don’t like.

The most basic property of any desirable type system is soundness. Well-typed terms are guaranteed to be meaningful. Computation on well-typed terms do not generate stuck states. This property is formalized by two theorems, progress and preservation:

PROGRESS A well-typed term is not stuck. Either it is a value or it can take a step according to the evaluation rules.

PRESERVATION If a well-typed term takes a step of evaluation, then the resulting term is also well typed.

As is usual in the development of typing systems, I start with what looks like a simple Church-style system, and then aim to develop into the more theoretically complex Curry-style system. Seperating the well-typability of a term from it’s value seperates the two conceptually.

\(^{16}\)See (Pierce 2002, 95) for a detailed soundness proof.
The converse theorem, for the purposes of studying natural language mathematically, might be called *linguistic completeness*:

**Linguistic Completeness** Every string generally acceptable by a group of language speakers to be a member of a natural language \(L\) has a well-typed derivation. Generally, for all natural languages \(L\), the class of terms \(\tau \in \bigcup L\) are well-typable.

This is an assumption I make and the end goal of this linguistic inquiry. Formatting this thesis within a Curry-style system further distinguishes the values of terms from their types.

**Operational Semantics** \(\lambda NB \rightarrow^\times\)

I provide this extension as an explicitly typed, Curry-style system. I will speak both of terms with types and those without, and I show in the following section why it makes sense to do so. Terms are defined in BNF form:

\[
\text{TERMS } t ::= x \mid \lambda x:T.t \mid t \cdot t \mid \top \mid \bot \mid \text{if } t \text{ then } t \text{ else } t \mid 0 \mid \text{succ } t \mid \text{pred } t \mid \text{iszero } t \mid \text{nv } \mid \{t,t\} \mid t.1 \mid t.2
\]

\[
\text{VALUES } v ::= \lambda x:T.t \mid \top \mid \bot \mid \text{nv } \mid \{v,v\}
\]

\[
\text{NUMERIC VALUES } \text{nv } ::= 0 \mid \text{succ } \text{nv}
\]

\[
\text{TYPES } T ::= (T,T) \mid \text{bool } \mid \text{nat } \mid T_1 \times T_2
\]

\[
\text{ENVIRONMENTS } \Gamma ::= \emptyset \mid \Gamma, x : T
\]
EVALUATION RULES $t \rightarrow t'$

\[
\frac{t_1 \rightarrow t_1'}{t_1 \ t_2 \rightarrow t_1' \ t_2} \quad (E-\text{App1})
\]

\[
\frac{t_2 \rightarrow t_2'}{v_1 \ t_2 \rightarrow v_1' \ t_2'} \quad (E-\text{App2})
\]

\[
(\lambda x : T_{11} . t_{12}) v_2 \rightarrow [v_2/x] t_{12} \quad (E-\text{AppAbs})
\]

\[
\frac{\text{if } \top \text{ then } t_2 \text{ else } t_3 \rightarrow t_2}{ \quad (E-\text{IfTrue})}
\]

\[
\frac{\text{if } \bot \text{ then } t_2 \text{ else } t_3 \rightarrow t_3}{ \quad (E-\text{IfFalse})}
\]

\[
\frac{t_1 \rightarrow t_1'}{\text{if } t_1 \text{ then } t_2 \text{ else } t_3 \rightarrow \text{if } t_1' \text{ then } t_2 \text{ else } t_3} \quad (E-\text{If})
\]

\[
\frac{t_1 \rightarrow t_1'}{\text{succ } t_1 \rightarrow \text{succ } t_1'} \quad (E-\text{Succ})
\]

\[
\frac{\text{pred } 0 \rightarrow 0}{ \quad (E-\text{PredZero})}
\]

\[
\frac{\text{pred } (\text{succ } n v_1 ) \rightarrow n v_1}{ \quad (E-\text{PredSucc})}
\]

\[
\frac{t_1 \rightarrow t_1'}{\text{pred } t_1 \rightarrow \text{pred } t_1'} \quad (E-\text{Pred})
\]

\[
\frac{\text{iszero } 0 \rightarrow \top}{ \quad (E-\text{IsZeroZero})}
\]

\[
\frac{\text{iszero } (\text{succ } n v_1 ) \rightarrow \bot}{ \quad (E-\text{IsZeroSucc})}
\]
\[ \frac{t_1 \rightarrow t'_1}{\text{iszero } t_1 \rightarrow \text{iszero } t'_1} \quad (\text{E-IsZero}) \]

\[ \frac{\{v_1, v_2\}.1 \rightarrow v_1}{(\text{E-PairBeta1})} \]

\[ \frac{\{v_1, v_2\}.2 \rightarrow v_2}{(\text{E-PairBeta2})} \]

\[ \frac{t_1 \rightarrow t'_1}{t_1.1 \rightarrow t'_1.1} \quad (\text{E-PairBeta1}) \]

\[ \frac{t_1 \rightarrow t'_1}{t_1.2 \rightarrow t'_1.2} \quad (\text{E-PairBeta2}) \]

\[ \frac{t_1 \rightarrow t'_1}{\{t_1, t_2\} \rightarrow \{t'_1, t_2\}} \quad (\text{E-PairBeta2}) \]

\[ \frac{t_2 \rightarrow t'_2}{\{t_1, t_2\} \rightarrow \{t_1, t'_2\}} \quad (\text{E-PairBeta2}) \]

\[ \frac{t_1 \rightarrow t'_1}{t_1.2 \rightarrow t'_1.2} \quad (\text{E-PairBeta2}) \]

**Typing Rules** \( \Gamma \vdash t : T \)

\[ \vdash \top : \text{BOOL} \quad (\text{T-True}) \]

\[ \vdash \bot : \text{BOOL} \quad (\text{T-True}) \]

\[ \frac{\Gamma \vdash t_1 : \text{BOOL} \quad \Gamma \vdash t_2 : T \quad \Gamma \vdash t_3 : T}{\Gamma \vdash \text{if } t_1 \text{ then } t_2 \text{ else } t_3 : T} \quad (\text{T-App}) \]

\[ \vdash 0 : \text{NAT} \quad (\text{T-Zero}) \]
\[
\frac{\Gamma \vdash t_1 : \text{NAT}}{\Gamma \vdash \text{succ } t_1 : \text{NAT}} \quad \text{(T-Succ)}
\]

\[
\frac{\Gamma \vdash t_1 : \text{NAT}}{\Gamma \vdash \text{pred } t_1 : \text{NAT}} \quad \text{(T-Pred)}
\]

\[
\frac{\Gamma \vdash t_1 : \text{NAT}}{\Gamma \vdash \text{iszero } t_1 : \text{BOOL}} \quad \text{(T-IsZero)}
\]

\[
\frac{x : T \in \Gamma}{\Gamma \vdash x : T} \quad \text{(T-Var)}
\]

\[
\frac{\Gamma, x : T_1 \vdash t_2 : T_2}{\Gamma \vdash \lambda x : T_1. t_2 : \langle T_1, T_2 \rangle} \quad \text{(T-Abs)}
\]

\[
\frac{\Gamma \vdash t_1 : \langle T_{11}, T_{12} \rangle \quad \Gamma \vdash t_2 : T_{11}}{\Gamma \vdash t_1 t_2 : T_{12}} \quad \text{(T-App)}
\]

\[
\frac{\Gamma \vdash t_1 : T_1 \quad \Gamma \vdash t_2 : T_2}{\Gamma \vdash \{t_1, t_2\} : T_1 \times T_2} \quad \text{(T-Pair)}
\]

\[
\frac{\Gamma \vdash t_1 : T_{11} \times T_{12}}{\text{Gamma} \vdash t_{1.1} : T_{11}} \quad \text{(T-Proj1)}
\]

\[
\frac{\Gamma \vdash t_1 : T_{11} \times T_{12}}{\text{Gamma} \vdash t_{1.2} : T_{12}} \quad \text{(T-Proj2)}
\]

Notice that terms are not assigned types in the syntax. Types are assigned by the typing rules, which characterize a typing relation independent of the evaluation relation. This is a Curry-style typing system. According to (Pierce 2002, 109), “most compilers for full-scale programming languages actually avoid carrying [type] annotations at run time.” Define the following erasure function mapping terms in $\lambda\mathbb{N}\mathbb{B}^{\rightarrow \times}$ to corresponding terms in $\lambda\mathbb{N}\mathbb{B}$:
ERASE

$$erase(x : T) = x$$

$$erase(\lambda x : T_1.t_2) = \lambda x.erase(t_2)$$

$$erase(t_1t_2) = erase(t_1)erase(t_2)$$

A term $m$ in the untyped $\lambda$-calculus is said to be typable if there exists some term $t$, type $T$, and context $\Gamma$ s.t. $erase(t : T) = m$ and $\Gamma \vdash t : T$. If $t \rightarrow_{\text{ANB} \rightarrow \times} t'$, then $erase(t) \rightarrow_{\text{ANB}} erase(t')$. Further, if $erase(t) \rightarrow_{\text{ANB}} m'$, then there is a term $t'$ s.t. $t \rightarrow_{\text{ANB} \rightarrow \times} t'$ and $erase(t') = m'$. Variables are not introduced with a basic type, they are provided a basic type by a formal environment. This procedure provides a sort on the variable domain. When a variable is first introduced, it must be provided a type. The classification of variable types, or the environment relation $R(\Gamma, x, T)$ provides the basic denotational structure.

If one were comparing Curry style systems to Church style systems, the erasure function is exactly what is needed:

**Church$\Rightarrow$Curry** If $M \triangleright_{\beta} N$, then $erase(M) \triangleright_{\beta} erase(N)$, and if $\Gamma \vdash_{\text{church}} M : \sigma$, then

$$\Gamma \vdash_{\text{curry}} erase(M) : \sigma.$$  

**Curry$\Rightarrow$Church** If $M \triangleright_{\beta} N$ and $M = erase(M')$, then $M' \triangleright_{\beta} N'$ for some $erase(N') = N$, and if $\Gamma \vdash_{\text{curry}} M : \sigma$, then there is a typed term $M'$ with $erase(M') = M$ and

$$\Gamma \vdash_{\text{church}} M' : \sigma.$$
Here we see that it is possible to translate properties between Church and Curry style systems. \textbf{Church}⇒\textbf{Curry} shows that it is possible to lower typed terms to their untyped counterparts, while \textbf{Curry}⇒\textbf{Church} shows how to lift Curry derivations into Church derivations.

\textbf{Curry-Howard Isomorphism}

There are two kinds of typing rules. Type \textit{introduction} rules such as (T-Abs) describe how elements of a type are created. Type \textit{elimination} rules such as (T-App) explain how elements of a type can be used. When introduction and elimination forms come together, a \textit{redex} is formed, which represents a computational state. Introduction and elimination rules are used as in the natural deduction systems of formal logic. Type systems correspond very generally to constructive logics:\textsuperscript{17}

\begin{table}[h]
\centering
\begin{tabular}{l l}
\hline
Propositions & Types \\
\hline
Proposition \( P \supset Q \) & Type \( \langle P, Q \rangle \) \\
Proposition \( P \land Q \) & Type \( P \times Q \) \\
Proof of \( P \) & Term \( t \) of type \( P \) \\
\hline
\end{tabular}
\caption{Curry-Howard Isomorphism}
\end{table}

\textsuperscript{17}(Pierce 2002, 109)
The logics represented often omit the law of excluded middle, requiring a constructive proof of either $Q$ or $\neg Q$. The $\triangledown_{\beta}$ relation corresponds to the cut elimination rule:

$$\frac{\Gamma \vdash A \quad \Pi, A \vdash B}{\Gamma, \Pi \vdash B} \quad \text{(Cut Elimination)}$$

Here $\Gamma$ represents a term of type $A$, while $\Pi$ represents a term of type $\langle A, B \rangle$. Thus, putting $\Gamma$ and $\Pi$ together provides a term of type $B$. If one were to consider the Curry-Howard Isomorphism from the perspective of combinatory logic, $\lambda$-introduction corresponds to the implication introduction rule. Those who prefer combinatory systems eschew this rule for a set of axioms:

**A1** $\varphi \rightarrow \psi \rightarrow \varphi$

**A2** $(\varphi \rightarrow \psi \rightarrow \sigma) \rightarrow (\varphi \rightarrow \psi) \rightarrow \varphi \rightarrow \sigma$

The result is a Hilbert-style proof system. These axioms correspond to the typed combinators $K$ and $S$:

**C1** $\Gamma \vdash K : \varphi \rightarrow \psi \rightarrow \varphi$

**C2** $\Gamma \vdash S : (\varphi \rightarrow \psi \rightarrow \sigma) \rightarrow (\varphi \rightarrow \psi) \rightarrow \varphi \rightarrow \sigma$

In general, any Hilbert style axiom may be added to a combinatory system as a constant. Since these axioms are schemata, the constants are assigned polymorphic types. Further, combinatory logic provides the restrictions necessary to study the implicational fragment

$^{18}$(Sørensen and Urzyczyn 2006, 103-126).
substructural logics such as relevance logic, which corresponds to the $\lambda I$ restriction. Relevance logic requires that each assumption in a derivation be used at least once. Specifically, $\text{A1}/\text{C1}$ should be ruled out. This axiom defines a constant function that allows one to ignore premises of the argument, which is against the relevant logicians aims. The axioms of the relevant propositional logic $\text{R}_{\rightarrow}$ are as follows:

\[
\text{A}_S \quad \Gamma \vdash S : (\phi \to \psi \to \gamma) \to (\varphi \to \psi) \to \varphi \to \gamma
\]

\[
\text{A}_B \quad \Gamma \vdash B : (\varphi \to \gamma) \to (\varphi \to \psi) \to \varphi \to \gamma
\]

\[
\text{A}_C \quad \Gamma \vdash C : (\varphi \to \psi \to \gamma) \to \psi \to \phi \to \gamma
\]

\[
\text{A}_I \quad \Gamma \vdash I : \varphi \to \varphi
\]

Two other logics may be defined: $\text{BCK}$ and $\text{BCI}$, which correspond to the implicational fragment of affine logic and linear logic. Affine logic consists of the axioms $\text{A}_B$, $\text{A}_C$ and $\text{C}_1$. Hence the name $\text{BCK}$. Affine terms are defined as follows:

**Affine Terms** Affine terms (also known as $\text{BCK}$-terms) are defined as follows:

1. Every variable is an affine term;

2. An application $MN$ is an affine term iff both $M$ and $N$ are affine terms, and $\text{FV}(M) \cap \text{FV}(N) = \emptyset$;

3. An abstraction $\lambda xM$ is an affine term iff $M$ is an affine term.
These terms use each variable only once, representing the idea that assumptions are disposable. *Linear* terms maintain control over assumptions by allowing them to be used exactly once. The axioms of the implicational fragment of linear logic are \( A_B, A_C, \) and \( A_I. \) The combinators \( B \) and \( C \) are definable by \( S \) and \( K:\)

**SK Reduction** \( B = S(KS)K \)

**SBK Reduction** \( C = S(BBS)(KK) \)

With \( B \) and \( C, \) we can extend our abstraction operation:

**Abstraction** For every CL-term \( P \) and every variable \( x, \) a CL-term \( \lambda^x.P \) is defined by induction on \( P: \)

(i.) \( \lambda^x.x \equiv I, \)

(ii.) \( \lambda^x.P \equiv KP \) if \( x \not\in FV(P), \)

(iii.) \( \lambda^x.PQ \equiv S(\lambda^x.P)(\lambda^x.Q) \) if \( x \in FV(P) \cap FV(Q), \)

(iv.) \( \lambda^x.PQ \equiv C(\lambda^x.P)Q \) if \( x \in FV(P) \) and \( x \not\in FV(Q) \)

(v.) \( \lambda^x.PQ \equiv BP(\lambda^x.Q) \) if \( x \not\in FV(P) \) and \( x \in FV(Q) \)

To those advocating a minimal logical fragment such as (Jacobson 1999), (Jacobson 2002), (Jacobson 2007), the combinator methodology just outlined provides an avenue of research. In what follows I use (ii) of the abstraction rules to demonstrate the effects of this type

\(^{19}(S\o\r ensen and Urzyczyn 2006)\) Lemma 5.6.3.
of system in a movement driven framework. This partial decomposition of $\lambda$ is going to ignore that the applicative structure can in fact be mapped by $S$, as it would needlessly complicate the derivations. I now strengthen the calculus we are constructing, allowing control operations that simulate the classical propositional calculus.

The picture developed here differs from the picture sketched in the first part of this work:

Here the Curry-Howard Correspondence provides a computational bridge between the Logical Form of Generative Grammar and Constructive Logic. The computational basis provided by the untyped $\lambda$-calculus I claim suffices to connect our linguistic faculties to the computational logic processing unit that is modeled in the ideal by artificial logical systems. Having a type system that is erasable at runtime, or is even Turing-complete itself, would be capable of providing a sound mathematical and logical basis, provided organic inputs that were capable of writing their own complex code. This function is still mysterious at best, but best approximations are being refined by the artificial intelligence community.

This is a good starting point, but there is a stronger connection between formal logic
and the logical form of sentences in a natural language—as represented by $\lambda$-terms. The Curry-Howard isomorphism shows that type-checking and reduction is a logical process. When considering the content of a sentence $\varphi$, the Curry-Howard correspondence ensures that the subcomponents of the sentence compose logically. There is a further question here, however, which seeks to understand the logical properties of these sub-sentential chunks of information. To contrast these two projects I first develop an algebraic semantics capable of comparing the logics in question.

For any partially ordered set $\langle A, \leq \rangle$, if every two element subset of $a, b \in A$ has a least upper bound $\inf(a, b) := a \land b$ and greatest lower bound $\sup(a, b) := a \lor b$, then $A$ is a lattice. This mathematical structure investigates the partial ordering relation, which is defined by the following properties:

**Reflexivity** $a \leq a$

**Antisymmetry** $a \leq b$ and $b \leq a$ imply that $a = b$

**Transitivity** $a \leq b$ and $b \leq c$ imply that $a \leq c$

These conditions generate an interesting property known as the *duality principle*

If a statement $\varphi$ is true in all posets, then its dual is also true in all posets. If $\varphi$ holds for $\langle A, \leq \rangle$, its dual holds for $\langle A, \geq \rangle$.

Lattices are a poset with the additional restriction that every two element set $a, b \in A$ has a least upper bound and greatest lower bound:
**Lattice** A poset $\langle A, \leq \rangle$ is a lattice $\langle A, \wedge, \vee \rangle$ just in case $\wedge, \vee$ are idempotent, commutative, and associative:

1. $a \vee a = a$, $a \wedge a = a$
2. $a \vee b = b \vee a$, $a \wedge b = b \wedge a$
3. $(a \vee b) \vee c = a \vee (b \vee c)$, $(a \wedge b) \wedge c = a \wedge (b \wedge c)$

The operations $\wedge, \vee$ are the greatest lower bound (infimum) and least upper bound (supremum) respectively. These operations are connected to partial orders by the *absorption identities*:

1. $a \leq b$ iff $a \wedge b = a$, $a \leq b$ iff $a \vee b = b$
2. $a \wedge (a \vee b) = a$, $a \vee (a \wedge b) = b$
3. $a \wedge (b \wedge a) = a \wedge b$,

The connection in (1) generates the identities in (2). Once we see these identities, it becomes obvious that lattices can be defined algebraically without a generating partial order:

**Algebraic Identities** Let $\langle A, \circ \rangle$ be an algebra with one binary operation $\circ$. The algebra $\langle A, \circ \rangle$ is a semilattice if $\circ$ is idempotent, commutative, and associative.

1. Let the algebra $\mathfrak{A} = \langle A, \circ \rangle$ be a meet semilattice. Let $a \leq b$ iff $a \circ b = b$. Then $\mathfrak{A}^p = \langle A, \leq \rangle$ is a poset, and the poset $\mathfrak{A}^p$ is a meet semilattice.
2. Let the poset $\mathfrak{A} = \langle A, \leq \rangle$ be a meet semilattice. Let $a \land b$ be the infimum of \{a, b\}. Then the algebra $\mathfrak{A}^a = \langle A, \land \rangle$ is a meet semilattice.

3. Let the poset $\mathfrak{A} = \langle A, \leq \rangle$ be a meet semilattice. $(\mathfrak{A}^a)^p = \mathfrak{A}$

4. Let the algebra $\mathfrak{A} = \langle A, \circ \rangle$ be a meet semilattice. Then $(\mathfrak{A}^p)^a = \mathfrak{A}$

5. By the duality principle, (1)-(4) hold for join semilattices and therefore lattices in general.

**Complete Lattice** We say that a lattice $\langle A, \land, \lor \rangle$ is complete if for every subset $X \subseteq A$ there exists an infimum $\land X$ and supremum $\lor X$.

**Compactness** Let $\mathfrak{L}$ be a complete lattice. An element $a \in L$ is compact iff for any $X \subseteq L$, if $a \leq \lor \{x : x \in X\}$, then there exists a $Y \subseteq X$ that is finite and $a \leq \lor \{y : Y\}$

**Algebraic Lattice** A lattice $\mathfrak{L}$ is algebraic iff

1. $\mathfrak{L}$ is complete, and

2. For every $a \in L$ there exists a subset $X \subseteq L$ s.t. $a = \lor \{x : x \in X\}$ and every $x \in X$ is compact.

**Isomorphism** Two lattices $\mathfrak{A}_1 = \langle A_1, \circ \rangle$, $\mathfrak{A}_2 = \langle A_2, \circ \rangle$ are isomorphic, and the map $\Phi : A_1 \to A_2$ is an isomorphism if $\Phi$ is one-to-one and onto, and $a \circ b \in \mathfrak{A}_1$ iff $\Phi(a) \circ \Phi(b) \in \mathfrak{A}_2$ for all operators $\circ$. 
Monotone Map The map $\Phi : \mathcal{A}_1 \rightarrow \mathcal{A}_2$ is a monotone or isotone map of the poset $\mathcal{A}_1$ into
the poset $\mathcal{A}_2$, if $a \leq b \in \mathcal{A}_1$ implies $\Phi(a) \leq \Phi(b) \in \mathcal{A}_2$.

Homomorphism A homomorphism of a semilattice $\mathcal{A}_1 = \langle A_1, \circ \rangle$ into $\mathcal{A}_2 = \langle A_2, \circ \rangle$ is a
map $\Phi : \mathcal{A}_1 \rightarrow \mathcal{A}_2$ satisfying $\Phi(a \circ b) = \Phi(a) \circ \Phi(b)$. A one-to-one homomorphism is
called an embedding.

Reminding ourselves, a function $f$ is one-to-one just in case it preserves distinctness of elements. A function $f$ is onto just in case, for every element $y$ in the co-domain of $f$, there exists an element $x$ in the domain s.t. $f(x) = y$. If a function is both one-to-one and onto it is a correspondence or bijection: every element of the codomain is mapped to by a unique element in the domain. By our algebraic identities, meet-homomorphisms, join-homomorphisms, and lattice-homomorphisms are all isotone maps.

Subalgebra A sublattice $\mathcal{L}$ of a lattice $\mathcal{A}$, $\mathcal{L} \subseteq \mathcal{A}$, is defined on a nonvoid subset $L \subseteq A$
closed under subprenumm and infimum ($\wedge, \vee$). Generally, a subalgebra $\mathcal{X} \subseteq \mathcal{A}$ just
in case the non-empty subset $X \subseteq A$ is closed under algebraic operations $\circ \in \mathcal{A}$ and
induced operations.

Convex Lattice The sublattice $\mathcal{L} \subseteq \mathcal{A}$ is convex just in case for $a, b \in L, c \in A$ and
$a \leq c \leq b$ imply that $c \in L$.

For $a, b \in L$, the closed interval $[a, b] = \{x : a \leq x \leq b\}$ is an example of a convex sublattice.

Two important types of convex sublattices are ideals and filters.
Ideal A poset $\mathcal{I} \subseteq \mathfrak{A}$ is an ideal iff

1. for every $x \in I, y \in A$, $y \leq x$ implies $y \in I$ ($I$ is a lower set), and

2. for every $x, y \in I$, there is some $z \in I$ s.t. $x \leq z$ and $y \leq z$ ($I$ is a directed set).

Filter A poset $\mathcal{G} \subseteq \mathfrak{A}$ is a filter iff

1. for every $x \in G, y \in A$, $y \geq x$ implies $y \in I$, and

2. for every $x, y \in I$, there is some $z \in I$ s.t. $x \geq z$ and $y \geq z$.

Remark Equivalently, a sublattice $\mathcal{I} \subseteq \mathfrak{A}$ is an ideal iff it is a lower set that is closed under suprema. The same sublattice is a filter iff it is an upper set closed under infimum.

Proper Ideals and Filters An ideal or filter $\mathfrak{X} \subseteq \mathfrak{A}$ is proper just in case $X \neq \varnothing(A)$.

Remark Let $X$ be any subset of a Heyting algebra $\mathfrak{A}$. The set $F = \{ a \in \mathfrak{A} : a \geq a_1 \wedge \ldots \wedge a_n \}$ for some $a_1, \ldots, a_n \in X$ is the least filter containing $X$. The filter $F$ is proper iff $a_1 \wedge \ldots \wedge a_n \neq \emptyset$ for all finite sets $\{a_1, \ldots, a_n\} \subseteq X$. Let $F$ be a proper filter in $\mathfrak{A}$ and let $a \notin F$. There exists a prime filter $G$ s.t. $F \subseteq G$ and $a \notin G$.

Prime Ideals and Filters If the set-theoretic complement of an ideal is a filter, then the ideal is considered prime. Let $I \subseteq (A, \leq)$ be a prime ideal iff

1. $I$ is a proper ideal of $A$, and

2. for all $x, y \in A$, $x \wedge y \in I$ implies $x \in I$ or $y \in I$.
Principal Ideals and Filters A principal ideal or filter of $L$ is the closure under join or meet of a unit member of $\{L\}$.

A principal ideal is not prime. A principal ideal generated by $\{x\}$ is the open interval $(x]$. Dually, a principal filter generated by $\{x\}$ is the open interval $[x)$.

Finite Intersection Property (FIP) Let $W$ be a nonempty set and let $E \subseteq \wp(W)$. By the filter generated by $E$ we mean the intersection $F$ of the collection of all filters over $W$ that include $E$:

$$F = \bigcap \{G : E \subseteq G \text{ and } G \text{ is a filter over } W\}$$

$E$ has the finite intersection property if the intersection of any finite number of elements of $E$ is non-empty.

I begin by introducing a standard boolean algebra:

Formula Algebra Let $\mathcal{P}$ be a set of proposition letters. It is possible to provide an algebra representing the dynamics of formula construction. The propositional formula algebra over $\mathcal{P}$ is the algebra

$$\mathfrak{F}orm(\mathcal{P}) = \langle \text{Form}(\mathcal{P}), +, - \rangle$$

where the operators ‘+’ and ‘−’ are defined $-\varphi := \neg \varphi$ and $\varphi + \psi := \varphi \lor \psi$ respectively. The carrier of this algebra is the collection of propositional formulas over the set of proposition letters, where the operators ‘+’ and ‘−’ provide a mathematical picture of the construction dynamics.
The algebra of truth values is \( \mathbb{2} = \langle \{0, 1\}, +, -, 0 \rangle \), where ‘−’ is defined by \( -a = 1 - a \) and ‘+’ is defined by \( a + b = \max(a, b) \).

**Formula Homomorphism** Let \( \mathcal{P} \) be a set of proposition letters. Given any assignment \( v : \mathcal{P} \to \mathbb{2} \), the function \( \tilde{v} : \text{Form}(\mathcal{P}) \to \mathbb{2} \) assigning to each formula its meaning under the valuation \( v \) is a homomorphism from \( \text{Form}(\mathcal{P}) \) to \( \mathbb{2} \).

Algebraic semantics is essentially *equational*. Algebraic logic then provides a means for determining when two terms are equal. When considering classical validity, we have the following correspondence:

\[
\Vdash_{\text{cpl}} \varphi \iff \mathbb{2} \models \varphi \approx \top \\
\mathbb{2} \models \varphi \approx \psi \iff \Vdash_{\text{cpl}} \varphi \leftrightarrow \psi \\
\Vdash_{\text{cpl}} \varphi \leftrightarrow (\varphi \leftrightarrow \top)
\]

An equation \( \varphi \approx \psi \) is valid in an algebra \( \mathfrak{A} \) if for every assignment of variables occurring in the terms, \( \varphi \) and \( \psi \) have the same meaning in \( \mathfrak{A} \). To properly algebraize a logic, one must establish

1. the membership of a formula \( \varphi \) in the logic can be rendered algebraically as the validity of some equation \( \varphi \approx \) in some class of algebras,

2. there is a translation of equations to formulas s.t. the equation holds in the class of algebras iff its translation belongs to the logic, and
3. translating a formula $\varphi$ into an equation $\varphi^\approx$ and then translating the equation back into a formula $\varphi'$ must leave one with a formula equivalent to $\varphi$.

The algebra 2 is intuitively connected to *power set algebras*:

**Power Set Algebra** Let $A$ be an arbitrary set with powerset $\wp(A)$. The power-set algebra

$$\mathfrak{B}(A) = \langle \wp(A), \cup, -, \emptyset \rangle.$$ Where ‘$\emptyset$’ denotes the empty set, ‘$-$’ denotes set complementation, and ‘$\cup$’ denotes set union.

**Remark** Every power set algebra is isomorphic to a power of 2, and conversely.  

This classical development is sufficient to provide an algebraic interpretation of semantic validity. If, however, one wanted to add a natural deduction system of some kind, we would need a syntactic analog of our current system:

**Boolean Algebra** Let $\mathfrak{A} = \langle A, +, -, 0 \rangle$ be an algebra of the boolean similarity type. Here the construction $x \cdot y$ is an abbreviation for $-(x + y)$, and, similarly, 1 abbreviates $-0$. We call $\mathfrak{A}$ a *boolean algebra* iff it satisfies the following identities:

1. $x + y = y + x, x \cdot y = y \cdot x$
2. $x + (y + z) = (x + y) + z, x \cdot (y \cdot z) = (x \cdot y) \cdot z$
3. $x + 0 = x, x \cdot 1 = x$
4. $x + (-x) = 1, x \cdot (-x) = 0$

---

$^{20}$(Patrick Blackburn and Venema 2001, 268) Proposition 5.8
5. \( x + (y \cdot z) = (x + y) \cdot (x + z), \ x \cdot (y + z) = (x \cdot y) + (x \cdot z) \)

Here top and bottom elements are represented by 1 and 0 respectively. The operations + and \( \cdot \) are join and meet, the least upper bound (subpremum) and greatest lower bound (infimum). Elements of a boolean algebra are ordered by defining \( a \leq b \) if \( a + b = b \). Because negation is defined as set complementation, a boolean lattice requires a lower bound, but does not require an upper bound. Define a **bounded lattice**:

**Bounded Lattice** A lattice \( \mathcal{L} = \langle L, \wedge, \vee, \bot, \top \rangle \) is bounded just in case it contains two nullary operations \((\bot, \top)\) s.t. \( a \wedge \bot = a \) and \( a \vee \top = a \).

**Distributive Lattice** A lattice \( \mathcal{L} = \langle L, \vee, \wedge \rangle \) is a distributive lattice iff for every \( a, b, c \in A \):

1. \( a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c) \), and
2. \( a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c) \)

Axioms (1)-(5) characterize lattices in general, with (6) and (7) satisfying the requirement of distributivity. Note that axioms (1) and (5) are the only things seperating distributive lattices from boolean algebras. These additional axioms allow us to provide an algebraic interpretation of intuitionistic logic. Out of a distributive lattice we build a **Heyting algebra**:

**Heyting Algebra** A bounded distributive lattice is said to be a Heyting algebra if for every \( a, b \in A \) there exists an element \( a \to b \) s.t. for every \( c \in A \) we have:

\[
  c \leq a \to b \text{ iff } a \wedge c \leq b
\]
In every Heyting algebra $\mathfrak{A}$ we have that

$$a \rightarrow b = \bigvee \{c \in A : a \land c \leq b\}$$

A distributive lattice $\mathfrak{A} = \langle (A, \leq), \land, \lor, \bot, \top \rangle$ is a Heyting algebra iff there is a binary operation $\rightarrow$ on $A$ s.t. for every $a, b, c \in A$:

1. $a \rightarrow a = \top$
2. $a \land (a \rightarrow b) = a \land b$
3. $b \land (a \rightarrow b) = b$
4. $a \rightarrow (b \land c) = (a \rightarrow b) \land (a \rightarrow c)$

A further condition on lattices will be useful in what follows:

**Infinite Distributivity Law** A complete distributive lattice $\langle A, \land, \lor, \bot, \top \rangle$ is a Heyting algebra iff it satisfies the *infinite distributivity law*

$$a \land \bigvee_{i \in I} b_i = \bigvee_{i \in I} a \land b_i$$

**Valuation** Let $\mathfrak{A}$ be a Heyting algebra with signature $\langle A, \land, \lor, \bot, \top, \rightarrow \rangle$.

Let $\mathcal{P}$ be a set of atomic variables and $\text{Form}(\mathcal{P})$ the set of all formulas, inductively defined. A function $v : \mathcal{P} \rightarrow A$ is a valuation into Heyting algebra $\mathfrak{A}$. Extend the valuation from $\mathcal{P}$ to $\text{Form}(\mathcal{P})$ in the following way:

$^{21}$From this point forward we refer to the poset $(A, \leq)$ as $A$, suppressing the underlying order.
1. \( \tilde{v}(\varphi \land \psi) = \tilde{v}(\varphi) \land \tilde{v}(\psi) \)

2. \( \tilde{v}(\varphi \lor \psi) = \tilde{v}(\varphi) \lor \tilde{v}(\psi) \)

3. \( \tilde{v}(\varphi \rightarrow \psi) = \tilde{v}(\varphi) \rightarrow \tilde{v}(\psi) \)

4. \( \tilde{v}(\bot) = \bot \)

**Validity** A formula \( \varphi \) is true in \( A \) under \( v \) if \( v(\varphi) = \top \). \( \varphi \) is valid in \( A \) if \( \varphi \) is true for every valuation in \( A \).

Intuitionistic propositional calculus (IPC) requires an infinitary system. There are, however, finitely bounded restrictions for which we will find some use. The difference between the classical propositional calculus (CPC) and the intuitionistic calculus is the definition of negation. In IPC \( \neg \varphi := \varphi \rightarrow \bot \), where \( \bot \) stands for contradiction. This condition is weaker than the condition of CPC:

**Boolean Algebra** Let \( A = \langle A, \land, \lor, \bot, \top, \rightarrow \rangle \) be a Heyting algebra. The following are equivalent:

1. \( A \) is a *Boolean* algebra.

2. For \( a \in A \), \( a \lor \neg a = \top \).

3. For \( a \in A \), \( \neg a = a \).

So far we have been setting up our structure using what is called the *coalesced sum*. For the sum of two complete lattices, if the \( \bot, \top \) in the sum domain is the image of the \( \bot, \top \) in each
of the component domains, the resulting lattice is a coalesced sum. Alternatively, one could provide the *seperated sum*, where there is a $\bot, \top$ standing for weaker, stronger respectively, seperated from all of the elements of the component lattice—including, here, true and false:

In this case let $\leq$ be an informational ordering. Consider the following approximations to real numbers: The set of closed intervals $[x, \bar{x}]$ where $x, \bar{x} \in \mathbb{R}$, and $x \leq \bar{x}$. Here the perfectly accurate value is somewhere between $x$ and $\bar{x}$. Now consider $x = [x, \bar{x}]$ and $y = [y, \bar{y}]$. If $y \leq \bar{x}$ and $x \leq \bar{y}$, then $x \leq y$. That is to say, $x$ is consistent with $y$ but more accurate. In this case, $\bot, \top$ represent the absence of information, such as $[-\infty, \infty]$, and too much information, such as the least upper bound $[0, 1] \vee [2, 3]$, which does not exist. In the case of the truth-value lattice above, the additional values $\bot, \top$ represent the undefined state and the overdefined state, respectively.

There are two correspondences between the intuitionistic and classical calculus to note. First, an application of intuitionistic completeness is *Glivenko’s theorem*:

$$\vdash_{cpc} \varphi \iff \vdash_{ipc} \neg\neg\varphi$$
Glivenko’s theorem does not extend to predicate logic. Here, however, we satisfy ourselves with an extension of the propositional calculus to modal propositional logic. As (Gödel 1986) shows, IPC has an S4 interpretation. Define a Kripke model as follows:

**Kripke Model** A Kripke model $\mathcal{M}$ consists of a frame $\mathcal{F} := \langle W, R \rangle$ where $W$ is a set of points, usually interpreted as worlds, and $R$ is a partial order on $W$. Let $\mathcal{M} := \langle \mathcal{F}, V \rangle$ where $V$ is a *persistent* valuation:

**Persistence** A valuation $V$ is persistent iff

1. $wRw'$ and $w \in V(p)$, then $w' \in V(p)$,
2. $w \models \varphi \land \psi$, then $w \models \varphi$ and $w \models \psi$,
3. $w \models \varphi \lor \psi$, then $w \models \varphi$ or $w \models \psi$, and
4. $w \models \varphi \rightarrow \psi$, then for $w' \in W$, if $wRw'$ and $w' \models \varphi$, then $w' \models \psi$.

In IPC, by our definition of $\neg \varphi$, we have that:

1. $w \models \neg \varphi$ iff for all $w' \in W$, if $wRw'$, then not $w' \models \varphi$
2. $w \models \neg \neg \varphi$ iff for $w' \in W$ if $wRw'$, then there exists a $w'' \in W$ s.t. $w'Rw''$ and $w'' \models \varphi$

To compare Kripke frames and models with Heyting algebras, we need some additional terminology:

**Characterization** A modal logic $\mathcal{L}$ *characterizes* a class of frames $\mathcal{F}$ iff $\mathcal{F} \models \mathcal{L}$.
Frame Validity A model $\mathcal{M} \models \varphi$ iff for $w \in W$, $w \models \varphi$, a frame $\mathcal{F} \models \varphi$ iff for all $\mathcal{M}$ definable on $\mathcal{F}, \mathcal{M} \models \varphi$.

Generated Subframe Let $R(w) := \{w' \in W : wRw'\}$ the generated subframe $\mathcal{F}_w := \langle W, R' \rangle$ of $\mathcal{F} = \langle W, R \rangle$ where $R'$ is the restriction of $R$ to $R(w)$.

Disjoint Union For frames $\mathcal{F}_1$ and $\mathcal{F}_2$, the disjoint union $\mathcal{F}' := \mathcal{F}_1 \uplus \mathcal{F}_2$ is the set $W' = W_1 \uplus W_2$ and the relation $R' = R_1 \cup R_2$.

Bounded Morphism If $\mathcal{M}$ and $\mathcal{M}'$ are models, then $f : W \to W'$ is a bounded morphism from $\mathcal{M}$ to $\mathcal{M}'$ iff

1. For $w, w' \in W$, $wRw'$ iff $f(w)R'f(w')$

2. For $w \in W$ and $w' \in W'$, if $f(w)R'w'$, then there exists a $w'' \in W$ s.t. $wRw''$ and $f(w'') = w'$.

3. For $w \in W$, $w \in V(p)$ iff $f(w) \in V'(p)$.

Now that we have some background machinery, I illustrate the connection between Kripke frames and Heyting algebras.

Kripke-Heyting Bridge Let $\mathcal{F}$ be an intuitionistic Kripke frame. For every $w \in W$ and $U \subseteq W$ let

1. $R(w) = \{w' \in W : wRw'\}$

22A generated submodel is a generated subframe with a valuation $V$ restricted to it.
2. $R^{-1}(w) = \{ w' \in W : w'Rw \}$

3. $R(U) = \bigcup_{w \in U} R(w)$

4. $R^{-1}(U) = \bigcup_{w \in U} R^{-1}(w)$

A subset $U \subseteq W$ is an up-set if $w \in U$ and $wRv$ implies $v \in U$. Let $Up(\mathcal{F})$ be the set of all up-sets of $\mathcal{F}$. For $U, V \in Up(\mathcal{F})$ let

$$U \rightarrow V = \{ w \in W : \text{for every } v \in W \text{ with } wRv \text{ if } v \in U \text{ then } v \in V \}$$

$$= W/R^{-1}(U/V)$$

**Fact** $\langle Up(\mathcal{F}), \land, \lor, \top, \bot, \rightarrow \rangle$ is a Heyting algebra.

Now that we have basic contact between Kripke frames and Heyting algebras, we would like to investigate the details and generality of this connection.

Given a Heyting algebra $\mathfrak{A} = \langle A, \land, \lor, \top, \bot, \rightarrow \rangle$ a set $F \subseteq A$ is a filter if $a, b \in F$ implies $a \land b \in F$, and if $a \in F$ and $a \leq b$ then $b \in F$. A filter is prime just in case $a \lor b \in F$ implies $a \in F$ or $b \in F$. If $\mathfrak{A}$ were a boolean algebra, every prime filter of $A$ would be an ultrafilter. In a Heyting algebra, this is not the case.

**Heyting-Kripke Bridge** Let $v$ be a valuation in a Heyting algebra $\mathfrak{A}$ where $\top \neq \bot$. There is a Kripke model $\mathfrak{M} = \langle W, \leq, \Vdash \rangle$ s.t. the conditions $\mathfrak{A}, v \Vdash \varphi$ and $\mathfrak{M} \Vdash \varphi$ are equivalent for all $\varphi$.

(Sørensen and Urzyczyn 2006) Lemma 2.5.8 and Theorem 2.5.9 prove this point in generality.

I provide an abbreviated proof of this fact.
Proof Take $W$ to be the set of all prime filters in $\mathfrak{A}$. The relation $\leq$ is inclusion. Define

$$F \models p \text{ iff } v(p) \in F.$$ 

For all formulas $\varphi$ I prove

$$F \models \varphi \text{ iff } [\varphi]_v \in F$$

by induction w.r.t. $\varphi$. I prove the case where $\varphi = \phi \to \psi$. Suppose that $F \models \phi \to \psi$ but $[\phi \to \psi]_v = [\phi]_v \to [\psi]_v \notin F$. Take the least filter $G'$ containing $F \cup \{[\phi]_v\}$. By our definitions above we have that $G' = \{b : b \leq f \land [\phi]_v \text{ for some } f \in F\}$ and $[\psi]_v \notin G'$. $G'$ is proper. Extend $G'$ to the prime filter $G$ not containing $[\psi]_v$. By the induction hypothesis $G \models \phi$ because $[\phi]_v \in G$. But since $F \models \phi \to \psi$ it follows that $G \models \psi$ or $[\psi]_v \in G$. Contradiction. For the converse assume $[\phi \to \psi]_v \in F$ and $F \subseteq G \models \phi$. By the induction hypothesis we have that $[\phi]_v \in G$ and since $F \subseteq G$ we have that $[\phi]_v \to [\psi]_v \in G$. Thus $[\psi]_v \leq [\phi]_v \land ([\phi]_v \to [\psi]_v) \in G$. Because of this we have that $[\psi]_v \in G$. Apply the induction hypothesis to show that $G \models \psi$.

This is a good connection, but it is not complete. To show its deficiency, we must first cover some basic algebraic operations.

Homomorphism Let $\mathfrak{A} = \langle A, \land, \lor, \top, \bot, \to \rangle$ and $\mathfrak{A}' = \langle A', \land', \lor', \top', \bot', \to' \rangle$ be Heyting algebras. A map $h : A \to A'$ is called a Heyting homomorphism if

1. $h(A \land B) = h(A) \land' h(B)$
2. $h(A \lor B) = h(A) \lor' h(b)$
3. $h(A \to B) = h(A) \to' h(B)$
4. \( h(\bot) = \bot' \)

**Subalgebra** \( \mathfrak{A}' \) is a subalgebra of \( \mathfrak{A} \) if \( A' \subseteq A \) and \( \forall a, b \in A', a \land b, a \lor b, a \rightarrow b, \bot \in A' \).

**Product** A product \( \mathfrak{A} \times \mathfrak{A}' \) is the algebra \( \langle A \times A', \land, \lor, \bot, \top, \rightarrow \rangle \) where

1. \( \langle a, a' \rangle \land \langle b, b' \rangle := \langle a \land b, a' \land b' \rangle \)
2. \( \langle a, a' \rangle \lor \langle b, b' \rangle := \langle a \lor b, a' \lor b' \rangle \)
3. \( \langle a, a' \rangle \rightarrow \langle b, b' \rangle := \langle a \rightarrow b, a' \rightarrow b' \rangle \)
4. \( \bot := \langle \bot, \bot' \rangle \)

Now I define two category theoretic entities, the categories **Heyt** and **Krip**. Let **Heyt** denote the category of Heyting algebras whose morphisms are Heyting homomorphisms. Let **Krip** denote the category of intuitionistic Kripke frames whose morphisms are bounded morphisms. Now we define two functors:

**Left** \( \Phi : \text{Heyt} \rightarrow \text{Krip} := \mathfrak{A} \mapsto \Phi(\mathfrak{A}) = \langle W, R \rangle \). For a homomorphism \( h : \mathfrak{A} \rightarrow \mathfrak{A}' \) let

\[ \Phi(h) : \Phi(\mathfrak{A}') \rightarrow \Phi(\mathfrak{A}) \] such that for every element \( F \in \Phi(\mathfrak{A}) \) we have that \( \Phi(h)(F) := h^{-1}(F) \).

**Right** \( \Psi : \text{Krip} \rightarrow \text{Heyt} := \) for every Kripke frame \( \mathfrak{F} \) let \( \Psi(\mathfrak{F}) = \langle Up(\mathfrak{F}), \land, \lor, \top, \bot, \rightarrow \rangle \).

If \( f : \mathfrak{F} \rightarrow \mathfrak{F}' \) is a bounded morphism, then \( \Psi(f) := \varphi(\mathfrak{F}') \rightarrow \Phi(\mathfrak{F}) \) is s.t. for every element of \( U \in \Psi(\mathfrak{F}') \) we have that \( \Psi(f)(U) = f^{-1}(U) \).

(de Jongh n.d.) provides the following duality:

**Theorem** Let \( \mathfrak{A} \) and \( \mathfrak{B} \) be Heyting algebras and \( \mathfrak{F} \) and \( \mathfrak{G} \) be Kripke frames.
1. If $\mathcal{A}$ is a homomorphic image of $\mathcal{B}$, then $\Psi(\mathcal{A})$ is isomorphic to a generated subframe of $\Psi(\mathcal{B})$.

2. If $\mathcal{A}$ is a subalgebra of $\mathcal{B}$, then $\Phi(\mathcal{A})$ is a bounded morphic image of $\Phi(\mathcal{B})$.

3. If $\mathcal{A} \times \mathcal{B}$ is a product of $\mathcal{A}$ and $\mathcal{B}$, then $\Phi(\mathcal{A} \times \mathcal{B})$ is isomorphic to the disjoint union $\Phi(\mathcal{A}) \uplus \Phi(\mathcal{B})$.

4. If $\mathcal{F}$ is a generated subframe of $\mathcal{G}$, then $\Psi(\mathcal{F})$ is isomorphic to a homomorphic image of $\Psi(\mathcal{G})$.

5. If $\mathcal{F}$ is a bounded morphic image of $\mathcal{G}$, then $\Psi(\mathcal{F})$ is a subalgebra of $\Psi(\mathcal{G})$.

6. If $\mathcal{F} \uplus \mathcal{G}$ is the disjoint union of $\mathcal{F}$ and $\mathcal{G}$, then $\Psi(\mathcal{F} \uplus \mathcal{G})$ is isomorphic to the product $\Psi(\mathcal{F}) \times \Psi(\mathcal{G})$.

Remark $\Psi(\mathcal{F}) = \langle Up(\mathcal{F}), \land, \lor, \top, \bot, \rightarrow \rangle$ is a complete lattice. Not every Heyting algebra is complete. Restrictions of $\Phi$ and $\Psi$ to categories of finite Heyting algebras and finite Kripke frames are dually equivalent.

Finitary Restriction For every finite Heyting algebra $\mathcal{A}$ there exists a Kripke frame $\mathcal{F}$ s.t. $\mathcal{A}$ is isomorphic to $Up(\mathcal{F})$.

Extensions For every Heyting algebra $\mathcal{A}$ the algebra $\Psi\Phi(\mathcal{A})$ is a cannonical extension of $\mathcal{A}$. For every Kripke frame $\mathcal{F}$, the frame $\Phi\Psi(\mathcal{F})$ is called a prime filter extension of $\mathcal{F}$. These extensions have the following properties:

1. $\mathcal{A}$ is a subalgebra of $\Psi\Phi(\mathcal{A})$. 
2. $\mathfrak{A}$ is not isomorphic to a homomorphic image of $\Psi\Phi(\mathfrak{A})$.

3. $\mathfrak{F}$ is a bounded morphic image of $\Phi\Psi(\mathfrak{F})$.

4. $\mathfrak{F}$ is not isomorphic to a generated subframe of $\Phi\Psi(\mathfrak{F})$.

We previously referred to a valuation being connected to a Heyting algebra. I assume familiarity with how valuations are added to Kripke frames, but we have not yet seen how a valuation is applied to an algebra.

**Tarski** Let $\mathbb{K}$ be a class of algebras of the same signature. We say that $\mathbb{K}$ is a *variety* if $\mathbb{K}$ is closed under the operations of taking homomorphic images ($H$), subalgebras ($S$), and products ($P$). $\mathbb{K}$ is a variety iff $\mathbb{K} = HSB(\mathbb{K})$.

**Birkhoff** A class of algebras forms a variety iff it is equationally defined. As we have seen, any boolean algebra is isomorphic to a field of sets: a sub-algebra of a power of $2$. Therefore, the variety of boolean algebras is generated by the algebra $2$.

The class $\textbf{Heyt}$ is a variety. Completeness requires the construction of a Lindenbaum-Tarski algebra.

**Lindenbaum-Tarski Algebra** Define an equivalence relation $\equiv$ on $Form(\mathcal{P})$ by putting $\varphi \equiv \psi$ iff $\vdash_{IPC} \varphi \leftrightarrow \psi$. Let $[\varphi]$ denote the equivalence class defined by $\varphi$. $Form(\mathcal{P})/\equiv := \{[\varphi] : \varphi \in Form(\mathcal{P})\}$. Define the operations on $Form(\mathcal{P})/\equiv$ as follows:

1. $[\varphi] \land [\psi] = [\varphi \land \psi]$
2. \([\varphi] \lor [\psi] = [\varphi \lor \psi]\)

3. \([\varphi] \rightarrow [\psi] = [\varphi \rightarrow \psi]\)

Denote by \(F(\omega)\) the algebra \((\text{Form}(\mathcal{P})/ \equiv, \land, \lor, \top, \bot, \rightarrow)\). \(F(\omega)\) is called either the Lindenbaum-Tarski algebra of IPC or the \(\omega\)-generated free Heyting algebra. This algebra provides the following restriction:

1. \(F(\alpha)\), for \(\alpha \leq \omega\) is a Heyting algebra.

2. \(\vdash_{\text{IPC}} \varphi\) iff \(\varphi\) is valid in \(F(\omega)\).

3. \(\vdash_{\text{IPC}} \varphi\) iff \(\varphi\) is valid in \(F(n)\) for any formula of \(n\) variables.

This duality shows that our type system carries a logic that captures a model system definable in a Kripke frame. This is good as far as it goes: it shows that there is a logic driven computational system that underlies the logic of composition. What has not been shown is the connection of this logic to what is traditionally considered the logical form of a sentence. The fact that our type system carries an intuitionistic logic does not in itself constrain the possible logics that might govern the meaning of natural language sentences. The logic of composition and the logic that connects semantic entailments within a grammatical system are not the same. What we have just done is provide a mathematical foundation upon which an argument over the logic of content might be posed.

To review, we have developed a Lindenbaum algebra \(\mathfrak{A}\) and connected it to an S4 Kripke frame \(\mathfrak{F}\) which simulates constructive proof. However, when considering how to best cap-
ture the logical content of a natural language sentence, we should generalize to arbitrary modalities.

**τ-frame** Let \( \tau \) be a modal similarity type. A **τ-frame** is a tuple \( \mathfrak{F} \) including

1. a non-empty set \( W \), and
2. for each \( n \geq 0 \) and each \( n \)-ary modal operator \( \Delta \) in the similarity type \( \tau \), an \( n + 1 \)-ary relation \( R_\Delta \).

I write \( \mathfrak{F} = \langle W, R_\Delta \rangle \Delta \in \tau \) to represent the collection of similarity relations \( \Delta \in \tau \). The arity of a similarity relation \( \Delta \) is defined by \( \rho(\Delta) \). In a model \( \mathfrak{M} = \langle \mathfrak{F}, V \rangle \), when \( \rho(\Delta) > 0 \) we define

\[
\mathfrak{M}, w \models \Delta(\varphi_1, \ldots, \varphi_n) \text{ iff }
\]

for some \( v_1, \ldots, v_n \in W \) with \( wR_\Delta v_1, \ldots, v_n \) we have

\[
\text{for each } i, \mathfrak{M}, v_1 \models \phi_1
\]

The modal type \( \Delta \) is a generalization of \( \Diamond \). The generalization of \( \Box \) is represented as \( \nabla \).

Now let us define the corresponding algebraic similarity type:

**Formula Algebra** Let \( \tau \) be a modal similarity type and \( \varphi \) a set of proposition letters. The

*formula algebra* of \( \tau \) over \( \varphi \) is the algebra \( \mathfrak{F}_{\text{form}}(\tau, \varphi) = \langle \text{Form}(\tau, \varphi), \wedge, \vee, \top, \bot, \rightarrow \rangle \)
For each modal operator $\Delta$, the operation $f_\Delta(t_1, \ldots, t_n) = \Delta(t_1, \ldots, t_n)$. Where $\Delta$ is a static part of a logical formula, the function $f_\Delta$ provides an interpretation of $\Delta$ as an operator on terms.

**Algebraic Similarity Type** Let $\tau$ be a modal similarity type. The corresponding algebraic similarity type $F_\tau$ contains as function symbols all modal operators together with the boolean symbols. For a set $\phi$ of variables let $\text{Ter}_\tau(\phi)$ denotes the collection of $F_\tau$ terms over $\phi$. The algebraic similarity type $F_\tau$ is the union of the modal similarity type $\tau$ and the collection of logical operators (with your choice of interpretation). By these definitions $\text{Form}(\tau, \phi) = \text{Ter}_\tau(\phi)$.

**Term Algebra** Let $\mathcal{A}$ be an algebraic similarity type and $X$ a set of variables. The term algebra of $\mathcal{A}$ over $X$ is the algebra $\text{Ter}_\mathcal{A}(X) = \langle \text{Ter}_\mathcal{A}(X), I \rangle$ where every function symbol $f$ is interpreted as the operation $I(f)$ on $\text{Ter}_\mathcal{A}(X)$ given by

$$I(f)(t_1, \ldots, t_n) = f(t_1, \ldots, t_n)$$

For term algebras generally the carrier of the algebra is the set of inductively generated terms over the atomic variables:

**Algebraic Carrier** The interpretation $I(f)$ maps an $n$-tuple of terms $t_1, \ldots, t_n$ to the term $f(t_1, \ldots, t_n)$. Substitution of terms for variables is an endomorphism—a homomorphism from an algebra to itself.
Substitution

Let $A$ be a similarity type and $X$ a set of variables. A substitution is a map $\sigma : X \to \text{Ter}_A(X)$ mapping variables to terms. Extending the map to obtain $\tilde{\sigma} : \text{Ter}_A(X) \to \text{Ter}_A(X)$ is defined inductively

$$\tilde{\sigma}(x) := \sigma(x)$$

$$\tilde{\sigma}(f(t_1, \ldots, t_n)) := f(\tilde{\sigma}(t_1), \ldots, \tilde{\sigma}(t_n))$$

Let $\sigma : \text{Ter}_A(X) \to \text{Ter}_A(X)$ be a substitution. Then $\sigma : \overline{\text{Ter}}_A \to \overline{\text{Ter}}_A$ is a homomorphism. In fact, given an algebra $\mathfrak{A}$ and an assignment $v$ to variables $X$, the extension $\tilde{v}$ is now a homomorphism from $\overline{\text{Ter}}_A \to \mathfrak{A}$.

Complex Algebra

Let $\tau$ be a modal similarity type and $\mathfrak{F} = \langle W, R_\Delta \rangle_{\Delta \in \tau}$ a $\tau$-frame. The full complex algebra of $\mathfrak{F}$, notation $\mathfrak{F}^+$, is the expansion of the power-set algebra $\mathfrak{B}(W)$ with operations $m_{R_\Delta}$ for every operator $\Delta \in \tau$. A complex algebra is a sub-algebra of a full complex algebra. If $K$ is a class of frames, the class of full complex algebras of frames in $K$ is denoted $\mathbf{CmK}$. Given a particular model, with $\tilde{v}(\varphi)$ the set of states where $\varphi$ is true, then $\tilde{v}(\Delta(\varphi_1, \ldots, \varphi_n)) = m_{R_\Delta}(\tilde{v}(\varphi_1), \ldots, \tilde{v}(\varphi_n))$.

Heyting Algebra with Operators

Let $\tau = \langle O, \rho \rangle$ be a modal similarity type where $\rho$ returns the arity of operators $R_\Delta \in O$. A Heyting algebra with $\tau$-operators is an algebra

$$\mathfrak{A} = \langle A, \wedge, \vee, T, \bot, \to, f_\Delta \rangle_{\Delta \in \tau}$$
Here $\langle A, \land, \lor, \top, \bot, \rightarrow \rangle$ is a Heyting algebra and every $f_{\Delta}$ is an operator of arity $\rho(\Delta)$. To be such an operator $f_{\Delta}$ must satisfy

**Normality** $f_{\Delta}(a_1, \ldots, a_{\rho(\Delta)}) = \bot$ whenever $a_i = \bot$ for some $i(0 < i \leq \rho(\Delta))$.

**Additivity** For all $i$ s.t. $(0 < i \leq \rho(\Delta))$, $f_{\Delta}(a_1, \ldots, a_i \lor a'_i, \ldots, a_{\rho(\Delta)}) =$

$$f_{\Delta}(a_1, \ldots, a_i, \ldots, a_{\rho(\Delta)}) \lor f_{\Delta}(a_1, \ldots, a'_i, \ldots, a_{\rho(\Delta)})$$

The two operations normality and additivity above correspond to the following modal formulas:

$$\Diamond \bot \leftrightarrow \bot$$

$$\Diamond (p \lor q) \leftrightarrow \Diamond p \lor \Diamond q$$

These formulas can be used to axiomatize the normal modal logic $K$, which is the basis of our current framework. Further, our modal operators are all monotonic. An operation $g$ on a Heyting algebra is monotonic if $a \leq b$ implies $ga \leq gb$. Specifically, if $a \leq b$, then $a \lor b = b$ so $ga \lor gb = g(a \lor b) = gb$ and therefore $ga \leq gb$. Monotonicity here corresponds to the rule of proof

$$\text{if } \vdash p \rightarrow q \text{ then } \vdash \Diamond p \rightarrow \Diamond q$$

**General Frame** Where $\mathcal{F}$ is a Kripke frame and $\mathfrak{A}$ is the carrier of a complex algebra over $\mathcal{F}$, $\mathfrak{g} = \langle \mathcal{F}, \mathfrak{A} \rangle$ is a general frame. To say that $\mathfrak{A}$ carries an algebra is just to say that $\mathfrak{A}$ is a non-empty collection of subsets of $W$ which is closed under the boolean operations and under the modal operation $m_R$. 
Definition  Given an \((n + 1)\)-ary relation \(R\) on a set \(W\), we define the following \(n\)-ary operation \(m_R\) on the powerset \(\wp(W)\) of \(W\):

\[
m_R(X_1, \ldots, X_n) = \{w \in W : wRw_1, \ldots, w_n \text{ for some } w_1 \in X_1, \ldots, w_n \in X_n\}
\]

General Frame Validity  Let \(g\) be a general frame. A formula \(\varphi\) is valid on \(g\) if \(\varphi\) holds in every state of \(g\) and under every admissible valuation \(V\). A similar definition holds for sets of formulas and classes of general frames. For a set of formulas and a normal modal logic \(\Lambda\), a general frame is called a \(\Lambda\)-frame if \(\Lambda\) is valid on that frame. Let \(K\) be a class of general frames, \(\Sigma\) a set of formulas, and \(\varphi\) a formula. Let \(\varphi\) be a semantic consequence of \(\Sigma\) over \(K\) if for every general frame \(g \in K\), every admissible valuation \(V\) on \(g\) and every state \(s \in g\), we have that \((g, V), s \models \Sigma\) implies that \((g, V), s \models \varphi\).

This construction has an interesting feature: a formula may be valid on a general frame while it is invalid on its underlying Kripke frame. The following is (Patrick Blackburn and Venema 2001, 305)’s example 5.63

Proof  Take a Kripke frame \(\mathfrak{F} = \langle W, R \rangle\), where \(W = \{e, v, w, x\} \cup \{v_n, w_n : n \in \omega\}\) and \(R = \{\langle u, v \rangle, \langle u, w \rangle, \langle v, v_n \rangle, \langle w, w_n \rangle, \langle v_n, x \rangle, \langle w_n, x \rangle, \langle x, x \rangle\}\) for all \(n\). Let \(g = \langle \mathfrak{F}, A \rangle\) a general frame where \(A\) is the collection of all finite and co-finite subsets of \(W\). It is claimed that \(g \models \lozenge \Box p \rightarrow \Box \lozenge p\) while \(\mathfrak{F} \not\models \lozenge \Box p \rightarrow \Box \lozenge p\). Consider the latter. The valuation \(V\) given by \(V(p) = \{v_n : n \in \mathbb{N}\}\). From \((\mathfrak{F}, V), v \models \Box p\) derive \((\mathfrak{F}, V), u \models \lozenge \Box p\), but \((\mathfrak{F}, V), u \not\models \Box \lozenge p\) since \((\mathfrak{F}, V), w \not\models \lozenge p\). To prove that \(g \models \lozenge \Box p \rightarrow \Box \lozenge p\), suppose that
for some admissible valuation $V$, $\Diamond \Box p$ holds at $u$. We may assume that $(g, V)$, $v \models \Box p$, so $p$ holds at all $v_i$. But $V(p)$ is not finite. Since $V(p)$ must be admissible it must be co-finite. Therefore we have co-finitely many $w_i$ with $(g, V), w_i \models p$. But then $\Diamond p$ holds at $w$ and thus $\Box \Diamond p$ at $u$.

This could be seen as a problem, and it is provided by Blackburn and Venema without comment. I am currently exploring the extension of our algebraic semantics to operators of arbitrary arity, as would be required to model the generalized quantifiers of natural language upon which they are based. This proof shows that a weaker Kripke frame may be the foundation for a stronger model. Not all convergent Kripke models generate convergent general frames. We restrict our attention in what follows to descriptive general frames. These frames provide a general completeness result: general frames share with boolean algebras with operators the property of providing an adequate semantics for all normal modal logics.\textsuperscript{23}

We must first restrict our attention to descriptive general frames. A general frame is descriptive just in case it is differentiated, tight, and compact:

**Differentiated** For all $s, t \in W$, $s = t$ iff $\forall a \in A (s \in a$ iff $t \in a)$.

**Tight** If for all $\Delta \in \tau$, assuming $\rho(\Delta) = n$ and for all $s, s_1, \ldots, s_n \in W$, $s R_{\Delta} s_1, \ldots, s_n$ iff

$$\forall a_1, \ldots, a_n \in A (\bigwedge_i s_i \in a_i \text{ then } s \in m_R(a_1, \ldots, a_n))$$

**Compact** For all $X \subseteq A$ with the finite intersection property, $\bigcap X \neq \emptyset$.

\textsuperscript{23}Cf. (Patrick Blackburn and Venema 2001)
A descriptive general frame is capable of representing IPC according to the definitions above.

We now provide some properties of general frames:

Descriptive General Frames Let \( \mathfrak{g} = \langle \mathfrak{F}, \mathfrak{A} \rangle \). A subset \( c \) of the universe is closed if it is the intersection of a collection of admissible sets:

\[
c = \bigcap \{ a \in \mathfrak{A} : c \subseteq a \}
\]

Let \( \mathfrak{g} \) be a descriptive general frame. We have that

1. Every singleton is closed,

2. The collection of closed sets is closed under finite unions and arbitrary intersections,

3. If \( c \) is a closed set, then so is the set \( R_\Delta[c] \),

4. For every state \( s \) and sequence of diamonds \( \beta \), the set \( R_\beta[s] \) is closed,

5. Let \( C \) be a family of closed sets with the finite intersection property. \( C \) has a non-empty intersection.

All normal modal logics are representable in a descriptive general frame.\(^{24}\) To classify the admissability of a set, we turn to topology.\(^{25}\)

Topology A pair \( \mathcal{H} = \langle H, \pi \rangle \) is a topological space when \( H \neq \emptyset \) and \( \pi \) is a set of subsets of \( H \) s.t.

\(^{24}\)(Patrick Blackburn and Venema 2001, 308) proposition 5.69.

\(^{25}\)(de Jongh n.d.)
1. \( H, \emptyset \in \pi \),

2. \( U, V \in \pi \) then \( U \cap V \in \pi \), and

3. If, for \( i \in I, U_i \in \pi \), then \( \bigcup_{i \in I} U_i \in \pi \).

**Interior** For \( Y \subseteq X \) the *interior* of \( X \) is \( \text{Int}(Y) = \bigcup \{ U \in \pi : U \subseteq Y \} \)

**Heyting Topos** For every \( U, V \in \pi \) let \( U \rightarrow V = \text{Int}((X/U) \cup V) \). The signature \( \langle \pi, \cup, \cap, \rightarrow, \emptyset \rangle \) defines a Heyting algebra.

We can extend this topography to cover modal algebras as follows:\(^{26}\) Let \( \mathcal{M} \) be an *topological model* \( \langle \mathcal{M}, \pi, V \rangle \) with points \( \mathcal{M} \) and \( \pi \) satisfies the usual conditions, and in addition a valuation \( V \) of modal models:

**Modal Topos Semantics** \( \square \varphi \) is true at a point \( s \) in \( \mathcal{M} \), \( \mathcal{M}, s \models \square \varphi \) if \( s \in \text{Int}(\llbracket \varphi \rrbracket^\mathcal{M}) \):

\[
\exists X \in \pi : x \in X \land \forall t \in X : \mathcal{M}, t \models \varphi
\]

This simple extension is unfortunately restrictive, as the modal axioms of S4 express basic topological properties:

**Inclusion** \( \square \varphi \rightarrow \varphi \)

**Idempotence** \( \square \square \varphi \rightarrow \varphi \)

**Cl(\cap) Open** \( \square(\varphi \land \psi) \rightarrow \square \varphi \land \square \psi \)

\(^{26}\)(van Benthem 2010, 19.2, 220)
Here $\text{Cl}(\cap)$ is the closure under arbitrary intersection of open sets. This leads directly to the consideration of *Alexandroff topologies*:

**Alex. Topos** Define a topology $\mathcal{M} = \langle M, \pi, V \rangle$ s.t. $\pi$ is equal to the *open bases* $s^\leq = \{ t \in W : s \leq t \}$ for a given preorder $\leq$. Each $s^\leq$ defines an open neighborhood as an *upward cone*. Here truth throughout an open neighborhood of a world $s$ is equivalent to truth in all of $s$’s relational successors.

**Remark** Here arbitrary intersections of open sets are open. This *Alexandroff property* can be defined as an infinite distribution law:

\[ \Box \bigwedge_{i \in I} \varphi_i \leftrightarrow \bigwedge_{i \in I} \Box \varphi_i \]

Further topological notions can be defined, such as *continuous maps*, as desired by (Stoy 1977) and the Scott-Strachey approach to programming language theory. Here the notion of a *continuation* plays a primary role. We define the relationship between modal valuations and continuous maps as follows:

**Continuous Map** Consider a continuous map $f$ from $\mathcal{M}$ onto $\mathcal{N}$, and let $V$ be a propositional valuation on $\mathcal{N}$. Taking inverse images $f^{-1}[V(p)]$ induces a corresponding valuation on $\mathcal{M}$.

Here we have an S4 relation modelled, which characterizes the Gödel translation of intuitionistic logic.
Requiring continuity in effect provides a stance on how to approximate infinitary objects. Infinitary objects are the least upper bound, or limit, of some set of finite approximations. To see this define a directed set:

**Directed Set** A nonempty set $X$ is directed iff every pair of elements has an upper bound in $X$.

Now a continuous function may be defined:

**Continuous Function** A function $f : D \to D'$ is continuous if $f(\bigcup X) = \bigcup \{f(x) : x \in X\}$ for all directed $X \subseteq D$.

**Monotonicity and Continuity** A function $f : D \to D'$ is continuous iff it is monotonic and $f(\bigcup X) = \bigcup \{f(\bigcup X_f) : X_f \subseteq X \text{ and } X_f \text{ finite}\}$ for all $X \subseteq D$.

**Remark** All monotonic functions on finite lattices are continuous.\(^{27}\)

Continuity is required as a necessary condition of the following condition:

**Strachey Condition** For all $X \subseteq E$ there exists a $Y' \subseteq E'$ s.t. $\bigcup Y' = f(\bigcup X)$, and for all $y' \in Y', y' \subseteq f(\bigcup X_f)$ for some finite $X_f \subseteq X$.

A function is continuous if it preserves limits. Where computation is modeled as a join of a directed set, a map is considered continuous if it is compatible with the formation of directed joins. A model is continuous if the subsets of all directed sets are directed. Continuous maps

\(^{27}\)Cf. (Stoy 1977, 97-105)
need not preserve bottoms, as the empty set is not directed. Every continuous map is order preserving, however, continuity is a strictly stronger condition than order preservation. Since Kratzerian functions, which we will be discussing in the course of the work, use pre-orders to represent better approximations to the ideal set of scenarios, it is natural to require continuity as a property of the order.

This topological description of $S4$ can be weakened to a *neighborhood* semantics, which is weaker than $K$. This type of frame does not validate modal distribution generally. Define a neighborhood $\langle W, N \rangle$ with $W$ a generating set and $N : W \to \wp(\wp(W))$. This function assigns to each point $w \in W$ a set of subsets of $W$: a *neighborhood*. If $\mathcal{M}$ is a model on the neighborhood frame $\langle W, N \rangle$, then

$$\mathcal{M}, w \models \Box \varphi \text{ iff } [\varphi]^\mathcal{M} \in N(w)$$

Without further restriction on our neighborhoods, distribution over conjunction and disjunction fail. Upward monotonicity remains:

$$\Box \varphi \to \Box (\varphi \lor \psi)$$

Neighborhoods are defined on topological spaces as follows:\textsuperscript{28}

**Neighborhood Spaces** Let $\langle X, \pi \rangle$ be a topological space. A subset $N$ of $X$ is a *neighborhood* of a point $a \in X$ if $N$ contains an open set that contains $a$. A subset $O$ of $X$ is *open* iff $O$ is a neighborhood of each of its points. The following are properties of neighborhoods:

\textsuperscript{28}I follow (Mendelson 1990)'s definitions.
N1 For each point \( x \in X \) there exists a neighborhood \( N \) of \( x \).

N2 For each point \( x \in X \), if \( N \) is a neighborhood of \( x \), then \( x \in N \).

N3 For each point \( x \in X \), if \( N \) is a neighborhood of \( x \) and \( N' \supset N \), then \( N' \) is a neighborhood of \( x \).

N4 For each point \( x \in X \) and each pair of neighborhoods \( N, M \) of \( x \), \( N \cap M \) is a neighborhood of \( x \).

N5 For each point \( x \in X \) and neighborhood \( N \) of \( x \), there exists a neighborhood \( O \) of \( x \) s.t. \( O \subset N \) and \( O \) is a neighborhood of each of its points.

**Closed Set** Given a topological space \( \langle X, \pi \rangle \), a subset \( F \) of \( X \) is called closed set if the complement \( C(F) \) is open. Let \( A \) be a subset of a topological space \( \langle X, \pi \rangle \). A point \( x \) is said to be in the closure of \( A \) if, for each neighborhood \( N \) of \( x \), \( N \cap A \neq \emptyset \). The closure of \( A \) is denoted \( \overline{A} \). \( A \) is closed iff \( A = \overline{A} \).

**Generalization** For each point \( x \) in a topological space \( \langle X, \pi \rangle \), the collection of neighborhoods \( \mathfrak{N}_x \) of all neighborhoods of \( x \) is called a complete system of neighborhoods at the point \( x \). This generalization allows for (N1)-(N5) to be appropriately paraphrased:\(^{29}\)

N1 For each \( x \in X \), \( \mathfrak{N}_x \neq \emptyset \).

N2 For each \( x \in X \) and \( N \in \mathfrak{N}_x \), \( x \in \mathfrak{N} \).

N3 For each \( x \in X \) and \( N \in \mathfrak{N}_x \), if \( N' \supset N \) then \( N' \in \mathfrak{N}_x \).

\(^{29}\)See (Mendelson 1990, 75-76)
N4 For each $x \in X$ and $N, M \in \mathfrak{N}_x$, $N \cap M \in \mathfrak{N}_x$.

N5 For each $x \in X$ and $N \in \mathfrak{N}_x$, there exists an $O \in \mathfrak{N}_x$ s.t. $O \subset N$ and $O \in \mathfrak{N}_y$ for each $y \in O$.

The collection of all topological spaces and collection of all neighborhood spaces are in one-one correspondence.\textsuperscript{30} Neighborhood semantics, however, falls outside of our general completeness results, as it is non-normal.

We are now in a position to give a topological characterization of a general frame $g = \langle W, R, \mathfrak{A} \rangle$ for the basic similarity type. According to (Patrick Blackburn and Venema 2001, 315), one can consider $\mathfrak{A}$ as a base for a topology $\mathcal{T}_\mathfrak{A}$.

**g-closure** Let $g = \langle W, R, \mathfrak{A} \rangle$ be a descriptive general frame. The general frame $g$ is descriptive iff $\langle W, \mathcal{T}_\mathfrak{A} \rangle$ is a boolean space with $\mathfrak{A}$ the set of clopens, and $R$ is a point-closed relation. Here the definition of admissible sets is the same as the definition of descriptive general frames above:

1. Every singleton is closed.

2. The collection of closed sets is closed under finite unions and arbitrary intersection.

3. If $c$ is a closed set, then so is the set $R_\Diamond[c] := \{t : \exists s \in cR_\Diamond st\}$ for every $\Diamond$.

4. For every state $s$ and sequence $\sigma$ of diamonds, the set $R_\sigma[s]$ is closed.

\textsuperscript{30}(Mendelson 1990, 79-80) Theorem 3.8.
5. Let $C$ be a family of closed sets with the finite intersection property. Then $C$ has a non-empty intersection.

**Clopen** A set $X$ is *clopen* just in case $\overline{X}$ is open. The closure of a set $X$ may preserve the property of being a neighborhood at each of its points.

Within this topography, the fragment IPC that represents CPC according to *Glivenko’s theorem* is representable within the typed $\lambda$-calculus. This corresponds directly to the relation between the a descriptive general frame $g$ and the Gödel coding of IPC into S4. It has been shown by (Griffin 1990) that the classical fragment of Glivenko’s itself corresponds to a type of control operator or *continuation passing style* of representation: the $\lambda\mu$-calculus.

Let us define the propositional calculus $\text{NK}(\land, \lor, \rightarrow, \bot)$ as the following set of rules:31

$$\frac{}{\Gamma, \varphi \vdash \varphi} \quad \text{(Ax)}$$

$$\frac{\Gamma, \varphi \vdash \psi}{\Gamma \vdash \varphi \rightarrow \psi} \quad \text{($\rightarrow$I)}$$

$$\frac{\Gamma \vdash \varphi \rightarrow \psi}{\Gamma \vdash \varphi} \quad \frac{\Gamma \vdash \varphi}{\Gamma \vdash \psi} \quad \frac{\Gamma \vdash \varphi \rightarrow \psi}{\Gamma \vdash \varphi} \quad \text{($\rightarrow$E)}$$

$$\frac{\Gamma \vdash \varphi \quad \Gamma \vdash \psi}{\Gamma \vdash \varphi \land \psi} \quad \text{($\land$I)}$$

$$\frac{\Gamma \vdash \varphi \land \psi}{\Gamma \vdash \varphi} \quad \frac{\Gamma \vdash \varphi \land \psi}{\Gamma \vdash \psi} \quad \text{($\land$E)}$$

$$\frac{\Gamma \vdash \varphi \lor \psi}{\Gamma \vdash \varphi} \quad \frac{\Gamma \vdash \varphi \lor \psi}{\Gamma \vdash \psi} \quad \text{($\lor$I)}$$

31 (Sørensen and Urzyczyn 2006)
The Hilbert-style system for classical logic is obtained by adding the axiom:

\[ \Gamma \vdash \varphi \lor \psi \quad \Gamma, \varphi \vdash \vartheta \quad \Gamma, \psi \vdash \vartheta \quad (\lor E) \]

\[ \Gamma, \neg \varphi \vdash \bot \quad (\neg E) \]

This is a form of Pierce’s law. The deduction theorem follows:

\[ \Gamma, \varphi \vdash \psi \text{ iff } \Gamma \vdash \varphi \rightarrow \psi \]

Axiom \( A_\bot \) \((\varphi \rightarrow \bot) \rightarrow \bot \) \rightarrow \varphi.\]

Axiom \( A_\bot \) corresponds to \( \neg E \), given that \( \neg \alpha := \alpha \rightarrow \bot \). This structure may be provided the Kolmogorov translation:

\[ k(\alpha) = \neg \neg \alpha \]

\[ k(\varphi \circ \psi) = \neg \neg (k(\varphi) \circ k(\psi)) \text{ for } \circ \in \{\land, \lor, \rightarrow\} \]

Over which the following theorem holds:\[^{32}\]

**K-Translation Theorem.**

1. \( \vdash \varphi \rightarrow k(\varphi) \text{ and } \vdash k(\varphi) \rightarrow \varphi \) in \( \text{CPC}(\land, \lor, \rightarrow, \bot) \).

2. \( \vdash \varphi \) in \( \text{CPC}(\land, \lor, \rightarrow, \bot) \) iff \( \vdash k(\varphi) \) in \( \text{IPC}(\land, \lor, \rightarrow, \bot) \).

Significantly, the term \( k \) has stood in various places for both *intensions* and *continuations*.

Where \( k \) is a type product over worlds, temporal intervals, individuals, etc., \( k \) is injected

\[^{32}(\text{Sørensen and Urzyczyn 2006, 154}) \text{ Theorem 6.4.3/6.7.4} \]
into a restricted category of types. Where $k$ represents a continuation, it is interpolated over every type. That is, for every type $\alpha$ there exists a continuation type $\langle k, \alpha \rangle$. Before defining the $\lambda \mu$-calculus in terms of the $\lambda$-calculus, I first set out the rules $\lambda \mu$:

\[
\frac{\Gamma, x : \tau \vdash x : \tau}{(Ax)}
\]

\[
\frac{\Gamma, x : \sigma \vdash M : \tau}{\Gamma \vdash (\lambda x : \sigma M) : \tau} \quad (A1)
\]

\[
\frac{\Gamma \vdash M : \langle \sigma, \tau \rangle \quad \Gamma \vdash N : \sigma}{\Gamma \vdash (MN) : \tau} \quad (A2)
\]

\[
\frac{\Gamma, a : \neg \sigma \vdash M : \bot}{\Gamma \vdash (\mu a : \neg \sigma M) : \sigma} \quad (A3)
\]

\[
\frac{\Gamma, a : \neg \sigma \vdash M : \sigma}{\Gamma, a : \neg \sigma \vdash ([a]M) : \bot} \quad (A4)
\]

Terms of the form $[a]M$ and $\mu a : \neg \sigma.M$ are called address application and abstraction. The logical system does not distinguish between names and addresses, as we have here. Thus, there is no reflex of the address application rule (A4). Note that (Ax) does not have a corresponding version for addresses, as they are not terms. Instead, an instance of (A4) is

\[
\Gamma, a : \neg \sigma \vdash (\lambda x : \sigma.[a]x) : \neg \sigma
\]

Where $\lambda x : \sigma.x$ is the identity function of arguments type $\sigma$, the address $[a]$ is incompatible with these arguments. The term $\lambda x : \sigma.[a]x$ returns $\bot$ when given an argument for $x$, and
is thus of type $\neg\sigma$, interpreted as $\langle\sigma, \bot\rangle$. Thus, $\sigma$ type arguments are incompatible with an address that is typed $\neg\sigma$. If this address is present, however, an application of (A3) provides

$$\Gamma \vdash (\mu a : \neg\sigma. \lambda x : \sigma. [a] x : \sigma)$$

This returns us to the original value of $M$. More generally, when an occupied address, $[a]$ rather than $[]$, is applied to an argument, it defaults, generating the contradictory state $\bot$. This mechanism re-assigns the type of the term. In the language of computation, what is happening is a variable is being re-assigned a meaning locally. When a term $M$ encounters a box $[]$, if the box is empty, the term fills the box and generates a new empty location. This is one way of reading $M = [][M]$. Conversely, if a term $M$ generates an address $[M]$, it does not provide a meaning until it finds empty memory space $[]$.

(Sørensen and Urzyczyn 2006) calls $\mu$-abstraction and address application a channel device. In the term $\mu a : \neg\sigma \ldots [a] M \ldots$, the address $a$ is a channel that values can be transmitted along. When you see $[a] M$, it should be read as $M$ is the value transmitted to address $a$, within the context of $\mu a : \neg\sigma \ldots$. The value of $\mu a : \neg\sigma \ldots [a] M \ldots$, then, is something like $M$. This value has just been moved addresses. We can see this by inspecting the syntax of assignment contexts, where $[][M] = M$.

Reduction on $\lambda\mu$-expressions captures the control operation we have just introduced. To define reduction, we must first define $\lambda\mu$-contexts:

$\lambda\mu$-context The set of $\lambda\mu$-contexts is defined:

$$C ::= []|CM|[a]C$$
\textbf{λµ-term} The \(\lambda\mu\)-term \(C[M]\) is defined:

1. \([[]]M = M\)

2. \((CN)[M] = C[M]N\)

3. \(([a]C)[M] = [a]C[M]\)

\textbf{Free Variables} The set \(FV(C)\) is defined:

1. \(FV([]) = \emptyset\)

2. \(FV(CM) = FV(C) \cup FV(M)\)

3. \(FV([a]C) = FV(C) \cup \{a\}\)

\textbf{Address Assignment} For a \(\lambda\mu\)-context \(C\) and an address \(a\), define \(M[a := C]\), where \(y, b \notin FV(C)\) and \(b \neq a\):

1. \(x[a := C] = x\)

2. \((\lambda y : \sigma.M)[a := C] = \lambda y : \sigma.M[a := C]\)

3. \((M_1M_2)[a := C] = M_1[a := C]M_2[a := C]\)

4. \((\mu b : \neg\sigma.M)[a := C] = \mu b : \neg\sigma.M[a := C]\)

5. \(([a]M)[a := C] = C[M[a := C]]\)

6. \(([b]M)[a := C] = [b]M[a := C]\)

To define a concept of reduction, it is easiest to work in a two different definitions of exten- sionality: \(\eta, \zeta\). The first, \(\eta\)-reduction, is required over \(\lambda\) and \(\mu\):
\[
\begin{align*}
(\eta) & \quad \frac{\lambda x. M x = M}{x \not\in \text{FV}(M)} \\
(\zeta) & \quad \frac{M x = N x}{M = N} \text{ if } x \not\in \text{FV}(MN)
\end{align*}
\]

(\beta) \quad (\lambda x :: \varphi M) \xrightarrow{\beta} M[x := N]

(\eta_\mu) \quad \mu a :: \neg \varphi. [a] M \xrightarrow{\eta_\mu} M \text{ if } a \not\in \text{FV}(M)

(\beta_\mu) \quad [b](\mu a :: \neg \varphi. M) \xrightarrow{\beta_\mu} M[a := [b][]]

(\zeta) \quad (\mu a :: \neg(\langle \varphi, \psi \rangle) M N) \xrightarrow{\zeta} \mu \beta :: \neg \psi. M[a := [b][[N]]] \text{ if } a \neq b \not\in \text{FV}(MN)

The relation \(\xrightarrow{\mu}\) is the union \(\xrightarrow{\beta} \cup \xrightarrow{\eta_\mu} \cup \xrightarrow{\beta_\mu} \cup \xrightarrow{\zeta}\).

In this system we write \(\varepsilon_\varphi(M) = \mu a :: \neg \varphi M\) when \(a \not\in \text{FV}(M)\). This relation, required for NK(\(\to, \land, \lor\)), the substructural fragment with sums and products, captures the sense of a *miracle*:

\[
\frac{\Gamma \vdash M : \bot}{\Gamma \vdash \varepsilon_\varphi(M) : \varphi} \quad (\varepsilon)
\]

This miracle construction requires that \(M\) be typed as \(\bot\) already, and then sets as the result of the *reductio* represented by \(M\) the preferred inference \(\varphi\). The reduction relation \(\xrightarrow{\varepsilon}\) is the smallest relation satisfying

(\(\varepsilon_1\)) \quad \varepsilon_{\langle \varphi, \psi \rangle}(M) N \xrightarrow{\varepsilon} \varepsilon_{\psi}(M)

(\(\varepsilon_2\)) \quad \varepsilon_{\varphi}(\varepsilon_{\bot}(M)) \xrightarrow{\varepsilon} \varepsilon_{\varphi}(M)
In the case of \( \varepsilon_1 \) we can see that \( \varepsilon \) is capable of absorbing an argument \( N \) of arbitrary type. \( \varphi \) here can be seen as a polymorphic type variable. This exemplifies the explosiveness of classical contradiction. The second case \( \varepsilon_2 \) shows that contradictions absorb each other. Assuming that a contradiction follows from a contradiction provides us with the same resources as the initial contradiction.

The rule \( \eta \mu \) connects transmittal and receipt of a result across a channel. The rule \( \beta \mu \) is a form of jump optimization. For a term \([b](\mu a.\neg \sigma.M)\), suppose that inside \( M \) a term \( \tau \) is transmitted to address \( a \): \( M = N([a]\tau) \). The rule \( \zeta \) moves an application down to a transmitted term.

In \( \lambda \mu \), terms are separated into expressions with \( \lambda \)-abstractions termed procedures, and commands, also called routines, here represented by \( \mu \)-bindings and their corresponding addresses. (Milne and Strachey 1976, 185) points out that while expressions yield results, commands do not. They instead alter the machine state without leaving any “significant result.” However, since executing expressions can have side effects this blurs the distinction between expressions and commands. Milne calls commands “sequencers” and pointed out that unlike expressions, label settings or addresses cannot be separated from the commands in which they occur.

In all of these cases the label settings or addresses, as we have been calling them, have functioned as a store in the sense of Cooper Storage. We have introduced identifiers and environments. Prior to introducing identifiers, the environments of expressions did not affect the
computation of a terms denotation. Once identifiers of expressions are explicitly introduced, computation of a terms denotation depends directly on the expressions environment. We can see this dependency by introducing let bindings:\(^{33}\)

**Terms (Let Binding)** \( t ::= \ldots | \text{let } x = t \text{ in } t \)

**Evaluation Rules** \( t \rightarrow t' \)

\[
\frac{\text{let } x = v_1 \text{ in } t_2 \rightarrow [x \mapsto v_1]t_2}{(E-\text{LetV})}
\]

\[
\frac{t_1 \rightarrow t'_1}{\text{let } x = t_1 \text{ in } t_2 \rightarrow \text{let } x = t'_1 \text{ in } t_2} \quad (E-\text{Let})
\]

**Typing Rules** \( \Gamma \vdash t : T \)

\[
\frac{\Gamma \vdash t_1 : T_1 \quad \Gamma x : T_1 \vdash t_2 : T_2}{\text{let } x = t_1 \text{ in } t_2 \rightarrow \text{let } x = t'_1 \text{ in } t_2} \quad (T-\text{Let})
\]

This extension witnesses the environmental dependence of let binding as a transformation on typing derivations that maps a let derivation to a derivation involving abstraction and application:

\[
\frac{\Gamma \vdash t_1 : T_1 \quad \Gamma x : T_1 \vdash t_2 : T_2}{\text{let } x = t_1 \text{ in } t_2 \rightarrow \text{let } x = t'_1 \text{ in } t_2} \quad (T-\text{Let})
\]

\( \Rightarrow \)

\(^{33}\)Here I am extending the operational definitions above. See (Pierce 2002, 174).
\[
\Gamma, x : T_1 \vdash t_2 : T_2 \\
\Gamma \vdash \lambda x : T_1 . t_2 : \langle T_1, T_2 \rangle \quad \text{(T-Abs)} \\
\Gamma \vdash t_1 : T_1 \quad \text{(T-App)}
\]

A more general version of \textit{continuation} must be introduced. As Milne puts it, “Some programming languages attempt to unify commands and expressions by providing every command with some conventional “result”, which is then ignored by the equations specifying the meaning of the commands.”\textsuperscript{34}

In what follows we will see a number of ways of doing this. It has been claimed that control operators generate non-compositional contexts. I argue that this is the flaw in (Weber 2012) and (Rabern 2012)’s arguments. I now show how these apparently non-compositional “side effects” may be compositionally incorporated.

In higher-order polymorphic systems, let binding is used to create truly polymorphic definitions, which are generalized typings that cannot be emulated using normal λ-abstraction and application. This new control operation generates a dependence on the environment of combinators. This type of dependence was formalized by (Milne 1974), and can be described simply by semantically modelling the ‘:=’ of the assignment statement \( M[x := y] \) found in model-theoretic semantics.\textsuperscript{35} Let binding changes the state of the machine in the same way as the assignment of values of variables. By explicitly referencing the domain of storable values and seperating it from the names of it’s states, denoted by a set of locations, we obtain

\textsuperscript{34}(Milne and Strachey 1976, 190)
\textsuperscript{35}I follow the exposition of (Stoy 1977, 282-320).
an explanation of the following examples:

\[ x := y \]

\[ y := (\text{if } a > b \text{ then } a \text{ else } b) \]

\[ (\text{if } a > b \text{ then } a \text{ else } b) := y \]

\[ n := n + 1 \]

We call the left-hand side of the equations the \( L \)-values, something like an address or location in memory, and the right-hand side the \( R \)-values the contents of these addresses. Let a flat lattice \( L \) denote the locations and a domain of storable values \( V \) different from the total set of denotations, since it is not necessarily true that everything that can be denoted can be stored. The domain of expression results is often the total set of denotations \( D + V \), where \( L \subseteq D \). To deal with side-effects, of which continuations are a primary example, a two-stage mapping between a name and its \( R \)-value is necessary. First a mapping between a name and it’s address must be part of the environment. Second a mapping between the address and it’s content in \( V \) must be part of the operational state. At this point, further inspection of an arbitrary operational state \( \sigma \) generates the following functions:

**Location Map** \( Map(\sigma) \in L \rightarrow V \)

**State Area** \( Area(\sigma) \in L \rightarrow \{true, false\} \)

Where the location map generates pairs of locations and their values, and the state area returns true on addresses currently in use. Here \( \sigma \subseteq \sigma’ \) if \( Map(\sigma) \subseteq Map(\sigma’) \) and \( Area(\sigma) \subseteq \)
Area(σ'). Ignoring input-output completely, a possible model for states would be

\[ S = ((\mathcal{L} \to V) \times (\mathcal{L} \to \{true, false\})) \]

With Map(σ) = σ.1 and Area(σ) = σ.2, two primitive store functions can be defined:

**Contents** \((\mathcal{L} \to (S \to V))\)

\[
\text{Conts}(\alpha)(\sigma) = \text{Area}(\sigma) \longrightarrow \text{Map}(\sigma)(\alpha), \text{error}
\]

The contents of a location \(\alpha\) in a state \(\sigma\) is an error result if \(\alpha\) is not in the area of \(\sigma\). The \(\longrightarrow\) is the conditional defined by our operational semantics. Assuming completely strict updates, an update function may be defined:

**Strict Update** The function \(Update : ((\mathcal{L} \times V) \to (S \to S))\) specifies a state transformation where \(Update(\alpha, \beta)\sigma\) produces a new state \(\sigma'\) s.t. the contents of \(\alpha\) in \(\sigma'\) is \(\beta\) and everything else is as it was. Define the following axioms:

**Map** \(\text{Map}(Update(\alpha, \beta)\sigma) = \lambda\alpha'.(\alpha = \alpha') \longrightarrow \beta, \text{Map}(\sigma)\alpha'\)

**Area** \(\text{Area}(Update(\alpha, \beta)\sigma) = \text{Area}(\sigma)\)

Further functions can be defined, such as a function for finding new and unused locations in \(\mathcal{L}\):

\(^{36}\)See (Stoy 1977, 286-287).
New Loc  $\text{New} : S \to \mathcal{L}$ defines $\text{New}(\sigma)$, a location not in $\text{Area}(\sigma)$.

$$\text{Area}(\sigma)(\text{New}(\sigma)) = \text{false}$$

$\text{New}(\sigma)$ is independent of $\text{Map}(\sigma)$, depending only on $\text{Area}(\sigma)$. While if $\sigma_1 = \sigma_2$, then $\text{New}(\sigma_1) = \text{New}(\sigma_2)$, it is not necessarily the case that if $\text{Area}(\sigma_1) = \text{Area}(\sigma_2)$ then $\text{New}(\sigma_1) = \text{New}(\sigma_2)$. Since $\mathcal{L}$ is a flat lattice, $\text{New}(\sigma_1) = \text{New}(\sigma_1 \cup \sigma_2) = \text{New}(\sigma_2)$. By monotonicity $\text{New}(\sigma_1) \supseteq \text{New}(\sigma_1 \cup \sigma_2)$. Once a new location is identified, further functions may be defined to manually adjoin a new location to the area in use and remove it once it has exhausted it’s extent.$^{37}$

I have added an error exit to the formalism without introducing it’s definition. In this case we can use the $\text{Unit}$ type to provide an adequate error sequence. In languages with side-effects, such as this, a sequencing notation $t_1; t_2$ may be defined. In languages with side-effects, often it is the side-effect, not the result, that is more important to the reader:

**Terms** $t ::= \ldots | \text{unit}$

**Values** $v ::= \ldots | \text{unit}$

**Types** $T ::= \ldots | \text{Unit}$

**Typing Rules** $\Gamma \vdash t : T$

$^{37}$See (Stoy 1977, 288). Also see (Fritz Henlein and Niss 2005) for an overview of the state of the art.

There are effects-based languages which model the flow of control.
\[ \Gamma \vdash \text{unit} : \text{Unit} \quad \text{(T-Unit)} \]

\[ \Gamma \vdash t_1 : \text{Unit} \\
\Gamma \vdash t_2 : T_2 \\
\Gamma \vdash t_1 ; t_2 : T_2 \quad \text{(T-Seq)} \]

**Evaluation Rules** \( t \rightarrow t' \)

\[ \frac{t_1 \rightarrow t'_1}{t_1 ; t_2 \rightarrow t'_1 ; t_2} \quad \text{(E-Seq)} \]

\[ \frac{\text{unit} ; t_2 \rightarrow t_2}{\text{(E-SeqNext)}} \]

**Derived Form** \( t_1 ; t_2 =_{df} (\lambda x : \text{Unit}. t_2) t_1 \), where \( x \not\in \text{FV}(t_2) \)

The derived form can take the place of the second typing rule and both evaluation rules. This formalization can be extended to explicitly deal with a referencing mechanism and a store, as the \( \lambda \mu \)-calculus above, with the following typing rules:

**Typing Rules** \( \Gamma \vdash t : T \)

\[ \frac{\Gamma \vdash t_1 : T_1}{\Gamma \vdash \text{ref } t_1 : \text{Ref } T_1} \quad \text{(T-Ref)} \]

\[ \frac{\Gamma \vdash t_1 : \text{Ref } T_1}{\Gamma \vdash ! t_1 : T_1} \quad \text{(T-DeRef)} \]

\[ \frac{\Gamma \vdash t_1 : \text{Ref } T_1}{\Gamma \vdash t_1 := t_2 : \text{Unit}} \quad \text{(T-Assign)} \]

As can be seen above, (T-Assign) shows that it is the side-effect that is more important than the result. The evaluation of a term may cause side-effects on the store that affect the evaluation of future terms. To handle this encumbrance, as we have already seen in the \( \lambda \mu \) system, we must relativize the evaluation rules to both a term and a store:
**Old Evaluation** \( t \rightarrow t' \)

**New Evaluation** \( t|\mu \rightarrow t'|\mu' \)

Here \( \mu \) is the store at a beginning state and \( \mu' \) is it’s ending state. This extension requires addition to the set of terms. One must include store locations \( (l) \) which are separate from the end results of computations. These intermediate computations can be formalized into an *intermediate language*. The typing language of course generalizes as well. The previous three place relation between contexts, terms, and types to a four place relation including locations, as well. To do this I introduce a *store typing* \( (\Sigma) \), a function from locations in the store into types:

\[
\Sigma(l) = T_1 \\
\frac{}{\Gamma|\Sigma \vdash l : \text{Ref} T_1} \quad (\text{T-Loc})
\]

I re-introduce the full set of rules \( \lambda \rightarrow \text{Unit Ref} \) before explaining the significant details:

**Terms** \( t ::= \text{unit (constant)} \mid x \ (\text{variable}) \mid \lambda x : T.t \ (\text{abstraction}) \mid tt \ (\text{application}) \mid \text{ref } t \) (reference creation) \mid \text{!t (dereference)} \mid t := t' \ (\text{assignment}) \mid l \ (\text{store location})

**Values** \( v ::= \text{unit} \mid \lambda x : T.t \mid l \)

**Types** \( T ::= \text{Unit} \mid \langle T, T \rangle \mid \text{Ref } T \)

**Stores** \( \mu ::= \emptyset \mid \mu, l = v \ (\text{location binding}) \)

**Contexts** \( \Gamma ::= \emptyset \mid \Gamma, x : T \ (\text{term variable binding}) \)
Store Typings $\Sigma ::= \emptyset | \Sigma, l : T$ (location typing)

Evaluation $t|\mu \rightarrow t'|\mu'$

$$
\frac{t_1|\mu \rightarrow t'_1|\mu'}{t_1 t_2|\mu \rightarrow t'_1 t'_2|\mu'} \quad \text{(E-App1)}
$$

$$
\frac{t_2|\mu \rightarrow t'_2|\mu'}{v_1 t_2|\mu \rightarrow v_1 t'_2|\mu'} \quad \text{(E-App2)}
$$

$$(\lambda x : T_{11}.t_{12})v_2|\mu \rightarrow [x \mapsto v_2]t_{12}|\mu$ \quad \text{(E-AppAbs)}$$

$$
\frac{l \notin \text{Dom}(\mu)}{\text{ref } v_1|\mu \rightarrow l|(\mu, l \mapsto v_1)} \quad \text{(E-RefV)}
$$

$$
\frac{t_1|\mu \rightarrow t'_1|\mu'}{\text{ref } t_1|\mu \rightarrow \text{ref } t'_1|\mu'} \quad \text{(E-Ref)}
$$

$$
\frac{\mu(l) = v}{!l|\mu \rightarrow v|\mu} \quad \text{(E-DerefLoc)}
$$

$$
\frac{t_1|\mu \rightarrow t'_1|\mu'}{!t_1|\mu \rightarrow !t'_1|\mu'} \quad \text{(E-DerefLoc)}
$$

$$
l := v_2|\mu \rightarrow \text{unit } |[l \mapsto v_2]|\mu \quad \text{(E-Assign)}
$$

$$
\frac{t_1|\mu \rightarrow t'_1|\mu'}{t_1 := t_2|\mu \rightarrow t'_1 := t'_2|\mu'} \quad \text{(E-Assign1)}
$$

$$
\frac{t_2|\mu \rightarrow t'_2|\mu'}{v_1 := t_2|\mu \rightarrow v_1 := t'_2|\mu'} \quad \text{(E-Assign2)}
$$

Typing $\Gamma|\Sigma \vdash t : T$
In (van Eijck and Unger 2010), the *continuation passing style* splits composition into a series of *computations on continuations*. Computations are functions representing the context around a continuation. To generate this style of functions and their arguments, instead of composing directly, jointly compose with a higher order combinator which encodes the next step in the computation (compositional process).

The basic example of continuations in linguistics is Montague’s lifting of the type NP from $e$ to $\langle \langle e, t \rangle t \rangle$. This lift generates a *computation*, which needs it’s *continuation*: a function of type $\langle e, t \rangle$. Here the continuation would represent the verbal shell.
Alice helped Dorothy.

\[
\lambda P : \langle e, t \rangle. [P(Alice)] \\
\lambda x : e. [helped(x)(Dorothy)]
\]

Computation: \(\langle\langle e, t \rangle t\rangle\)  
Continuation: \(\langle e, t \rangle\)

Figure 3.2: Continuation Semantics

Generally, continuations from values to results \(\langle a, r \rangle\) correspond to computations of type \(\langle\langle a, r \rangle, r \rangle\).\(^{38}\) The continuations for intransitive verbs, common nouns, and transitive verbs are:

<table>
<thead>
<tr>
<th>CN/IV</th>
<th>Initial Value</th>
<th>Continuation</th>
<th>Computation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(\langle e, t \rangle)</td>
<td>(\langle\langle e, t \rangle, t\rangle)</td>
<td>(\langle\langle\langle e, t \rangle, t\rangle, t\rangle)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TV</th>
<th>Initial Value</th>
<th>Continuation</th>
<th>Computation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(\langle ee, t \rangle)</td>
<td>(\langle\langle ee, t \rangle, t\rangle)</td>
<td>(\langle\langle\langle ee, t \rangle, t\rangle, t\rangle)</td>
</tr>
</tbody>
</table>

As can be seen by contrasting (CN/IV) and (TV), in principle it does not matter how many values the initial value \(a\) contain, not only could it contain, for example, an additional agent, time, and world \(\langle s, i, e \rangle\), it could in principle contain it’s own complex type. The pattern remains.

\(^{38}\)(van Eijck and Unger 2010, 308)
An extension of the (K-Translation Theorem) above provides a translation of store-based continuations into the continuation-passing style.\footnote{I take the translation from (Sørensen and Urzyczyn 2006, 140-144)}

**CPS Translation 1** If $\Gamma \vdash M : \varphi$ in $\lambda \mu$-calculus, then $\Gamma \vdash M : k(\varphi)$ in $\lambda \rightarrow$.

**CPS Translation 2** If $M =_\mu N$ in $\lambda \mu$, then $M =_\beta N$ in $\lambda \rightarrow$ for any restricted $M, N$.

The terms $M$ and $N$ are restricted in the following way:

$$M ::= x | MM | \lambda x : \sigma.M | \mu a : \neg \sigma. [b] M$$

This recursive definition restricts the $\lambda \mu$ terms in two ways. First, the storage location $[b]$ must always be accompanied by an abstraction $\mu a$, where $a, b$ may be identical. Either we are transmitting between address $\mu a$ and $[a]$, or because $\mu a$ is a type coercion allowing the application $[b] M$ to be placed in a context. Adopting a fixed address as $\neg \bot$, unrestricted $\lambda \mu$ may be translated to restricted $\lambda \mu$ of the same type. The restricted $\lambda \mu$-calculus is closed under reduction using the following auxiliary properties:\footnote{See (Sørensen and Urzyczyn 2006)}

$$\begin{align*}
\lambda k : \varphi.Pk & =_\beta P \\
P[x := Q] & =_\beta P[x := Q] \\
P[a := [b]] & =_\beta P[a := b] \\
P[a := [b]([]Q)] & =_\beta P[a := \lambda m^{k(\varphi) \rightarrow k(\psi)}.mQb]
\end{align*}$$

A full translation of $\lambda \mu$ terms into $\lambda$ terms is as follows:
\[ x^\varphi = \lambda h : \neg \varphi.x^{k(\varphi)}h \]

\[ \lambda x : \varphi.M^\psi = \lambda h : \neg(k(\varphi) \rightarrow k(\psi)).h(\lambda x : k(\varphi).M^{k(\psi)}) \]

\[ \overline{MN} = \lambda h : \neg \varphi.M^{\neg(\neg(k(\varphi) \rightarrow k(\psi)))}(\lambda m : k(\psi) \rightarrow k(\varphi).mN^{k(\psi)})h \]

\[ \mu a : \neg \varphi.\overline{M^\perp} = \lambda a : \neg \varphi.\overline{M^{\neg \perp}}(\lambda v : \perp.v) \]

\[ [a^{\neg \varphi}].M^\varphi = \lambda d : \neg \perp.\overline{M^{k(\varphi)}}a^{\neg \varphi} \]

In each of these cases I represent complex types with arrows \(a \rightarrow b\), rather than angles \(\langle a, b \rangle\). These representations differ only in notation, and the arrows better represent the connection to the Curry-Howard Isomorphism.

Because we are not using a storage system to represent scope, the system developed here is normalizing. If we restrict our type system to one that is strongly normalizing, we have access to the system \(\lambda P\) and its morphisms.

**Syntax \(\lambda P\)**

**Terms**

\[ t ::= s \text{ (sort)} | x \text{ (variable)} | \lambda x : t.t \text{ (abstraction)} | t.t \text{ (application)} | \]

\[ \Pi x : t.t \text{ (dependent product type)} \]

**Sorts**

\[ s ::= * \text{ (sort of proper types)} | \square \text{ (sort of kinds)} \]

**Contexts**

\[ \Gamma ::= \emptyset \text{ (empty)} | \Gamma, x : T \text{ (variable binding)} \]

**Typing**

\[ \Gamma \vdash t : T \]

\[ \Gamma \vdash * : \square \text{ (T-Star)} \]
\[ x : T \in \Gamma \quad (T-\text{Var}) \]

\[ \Gamma \vdash x : T \quad (T-\text{Var}) \]

\[ \Gamma \vdash S : * \quad \Gamma, x : S \vdash t : T \quad (T-\text{Abs}) \]

\[ \Gamma \vdash \lambda x : S.t : \Pi x : S.T \quad (T-\text{Abs}) \]

\[ \Gamma \vdash t_1 : \Pi x : S.T \quad \Gamma \vdash t_s : S \quad \Gamma \vdash t_1 \cdot t_2 : [x := t_2]T \quad (T-\text{App}) \]

\[ \Gamma \vdash S : s_i \quad \Gamma, x : S \vdash T : s_j \quad (T-\text{Pi}) \]

\[ \Gamma \vdash S : \Pi x : S.T : s_j \quad (T-\text{Pi}) \]

\[ \Gamma \vdash t : T \quad T \equiv T' \quad \Gamma \vdash T' : s_i \quad (T-\text{Conv}) \]

Here \( \langle s_i, s_j \rangle \in \{ \langle *, * \rangle, \langle *, \square \rangle \} \). The sorts used here, characterizing \( \lambda P \), comprise of \( * \), the kind of all proper types, and \( \square \), which is the sort that classifies well-formed kinds. There are two places where this system can be modified. First, the set over which \( \langle s_i, s_j \rangle \) ranges can be modified:

<table>
<thead>
<tr>
<th>Table 3.2: Lambda Cube</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \lambda \rightarrow )</td>
</tr>
<tr>
<td>( \lambda \mathcal{P} )</td>
</tr>
<tr>
<td>( \mathcal{F} )</td>
</tr>
<tr>
<td>( \mathcal{F}^\omega )</td>
</tr>
<tr>
<td>( \mathcal{C} )</td>
</tr>
</tbody>
</table>

Second, the axiom (T-Star) may be modified or copied to obtain type systems without corresponding logics. These systems may be implemented as programming languages where
programmer logic takes a primary role. For our purposes, however, we want to restrict our attention to systems that further characterize the Curry-Howard Isomorphism. The systems defined above require that the underlying term language be strongly normalizing:

**Strong Normalization** Every term \( \tau \), every sequence of \( \beta \)-reductions ends in a normal form.

Typechecking dependent types requires deciding equality of terms as a subtask. If the underlying term language is not strongly normalizing, the dependent system will be undecidable.

From the perspective of the Curry-Howard Isomorphism, if propositions are types, then proofs are terms. Similar to modal logic, a type constructor \( Prf \) may be introduced, which maps a type formula \( A \) into the type of its proofs. In this language, a proof of \( B \) from \( A \) is a \( \lambda \)-term \( \langle Prf A, Prf B \rangle \). The type constructor \( Prf \) may be omitted. When omitted, propositions are identified with the type of their proofs. In this way a proof of \( B \) from \( A \) is just \( \langle A, B \rangle \).

The generalization of the Curry-Howard Isomorphism to first-order predicate logic is representable by a dependent type system. A predicate \( B \) over a type \( A \) is a type-valued function on \( A \): a proof of \( \forall x : A. B(x) \). Constructively, such a proof is a procedure that, given an arbitrary element \( x \) of type \( A \) produces a proof of \( B(x) \). This construction is represented in our system as dependent product: \( \Pi x : A. B(x) \). Proof of the universal quantification is a member of this term. Martin-Löf (1984) was motivated by this extension of the Curry-Howard Isomorphism.
Dependent types may also classify lower-level type systems. Suppose we were to classify \( \lambda \rightarrow \) with the following declarations:\(^{41}\)

\[
\begin{align*}
Ty & :: * \\
Tm & :: \langle Ty, * \rangle \\
base & : Ty \\
arow & : \langle Ty \langle Ty, Ty \rangle \rangle \\
app & \Pi A : Ty. \Pi B : Ty. \langle Tm(arrowAB), \langle Tm A, Tm B \rangle \rangle \\
lam & \Pi A : Ty. \Pi B : Ty. \langle \langle Tm A, Tm B \rangle, Tm(arrowAB) \rangle
\end{align*}
\]

Here \( Ty \) represents the type of simple type expressions. For \( A : Ty \) the type \( Tm : A \) represents the type of lambda terms of type \( A \). There is a constant \( base : Ty \) representing the base type and function \( arrow \) representing type combination.\(^{42}\) There are two functions \( app \) and \( lam \). The first, \( app \) accepts two types \( A \) and \( B \), then a term of type \( (arrowAB) \) and a term of type \( A \) and yeilds a term of type \( B \), the combination of the two. Lambda abstraction is represented by \( lam \). This function takes a function mapping terms of type \( A \) to terms of type \( B \), and returns a term of type \( (arrowAB) \). This use of dependent type

\(^{41}\)This construction belongs to (Pierce 2005)

\(^{42}\)The naming here reflects the commonality that types are represented either in arrow notation \( A \rightarrow B \) or sequential notation: \( \langle A, B \rangle \).
systems, using functions at one level to represent dependencies at another level, is useful in studding syntax with binders. The technique is known as Higher-Order Abstract Syntax: 43

\[ \text{idA} = \text{lam} ~ A ~ A (~ \lambda x : Tm A. x) \]

This term represents the identity function on \( A : Ty \), as an example.

The final row in our system of dependent types is the Calculus of Constructions first introduced by (Coquand and Huet 1988). This system can be formulated with an additional basic type \( \text{Prop} \) and type family \( \text{Prf} \). The type family \( \text{Prf} \) assigns to each proposition \( p : \text{Prop} \) the type \( \text{Prf} p \) of its proofs. This completes an abbreviated overview of a group of type systems of varying strength. Using dependent types requires the factoring out of recursion into the Chomsky Grammar, as the underlying term language must be strongly normalizing. In what follows I will investigate a few different ways of developing these systems with syntactic-semantic issues in mind.

One needs to be careful here to outline how close we have come to what (Barendregt 2012) Appendix B, calls Illative Combinatory Logics: 44

**Definition** An ICL or illative combinatory logic is an extension of the alphabet of CL with a set of logical constants \( \text{I} \) s.t.:

1. The terms of ICL are defined inductively:

   (a) Every variable or constant is a term,

43 Chapter 2 of (Pierce 2005).
44 (Barendregt 2012, 573)
(b) If $M, N$ are terms, $(MN)$ is a term,

2. Formulas of $\text{ICL}$ are defined as follows:

(a) If $M, N$ are terms of $\text{ICL}$, then $M = N$ is a formula,

(b) If $M$ is a term of $\text{ICL}$, then $M$ is a formula.

Such a logic falls victim to Curry’s Paradox:

**Proof** Let $M$ be an arbitrary term. By a fixed point theorem, construct $X = X \rightarrow M$. The following is a derivation of $M$:

Essentially what is happening here is that when fixed point theorems are introduced into the language, the conditional becomes closer to a material conditional than we want and the paradox becomes an issue.

The logics we are considering do not have fixed-point constructions, as they are strongly normalizing. In its strongest form, CC allows one to define $\text{Prop}$ and $\text{Proof}$, which can also be used to escape the paradox. In this case, since the term language allows for recursion, it will be at best weakly normalizing, allowing for infinite reduction paths. Curry’s Paradox
relies on one of these infinite reduction paths in his definition of a fixed point. The Prop and Proof type and type family track the finite reduction paths in this weakly normalizing language, essentially disallowing recursion from occurring within their fragment. However, most practical programming languages provide general recursion with possible non-termination. These languages rely heavily on programmer-logic to prevent programming errors, where in linguistics it should be a property of the system that errors are ruled out. Hence, for linguistic purposes strong normalization should be preferred. Simply adding dependent types to a Turing-complete term language invariably leads to undecidable type-checking.
Chapter 4

Intensional Environments and Imperative Definitions

Intensionalism and Imperative Languages

At this point, I reintroduce the main problem of the work, that sentences don’t embed. There is no clear bridge between the formal syntax of intensional logic and generative syntax. In addition, the most straightforward semantic reduction–generative semantics–has had major arguments leveled against its adoption. Taking away this escape route, we now look at the contemporary views which hinge on the assumption, here called ES, which attempts to bridge natural language and intensional logic.

In this section we target the assumption ES:
**ES.** There exist embedding environments ($E$) in natural language that take sentential arguments ($S$).

This assumption draws the following picture: The reason this position looks initially feasible is that the type theorists on the left-hand side are what (Barker 2007) would call *globalists*: those that interpret displacement properties from a global perspective, interpreting moved objects semantically after their movement has taken effect.

According to the framework developed here, sentences are of type $\text{BOOL}$. The only embedding environments that take type $\text{BOOL}$ as an argument are the logical operations. All other operations generate computable redexes out of argument pairs $\langle \tau, \langle \rho, \text{BOOL} \rangle \rangle$ for
arbitrary types $\tau, \rho \in \mathfrak{S}$. But this instance is not strong enough to provide support for
the existential claim ES above. I claim that the existential must be read as a stronger
context-restricted existential for the arguments to find support:

**ES** There exist [non-truth-functional] embedding environments (E) in natural language that
take sentential arguments (S).

If you examine the arguments, the embedding environments relied upon are not truth-
functional environments, they are instead intensional environments. I claim these envi-
ronments are characterized by the violation of Referential Transparency:

> The only thing that matters about an expression is its value, and any subex-
pression can be replaced by any other equal in value. Moreover, the value of an
expression is, within certain limits [of scope], the same whenever it occurs.

Remembering the arguments from the previous chapter, imperative definitions violate this
principle. It is also violated by opaque environments, those environments that are claimed
to be sentences in $ES$ above. I claim that opaque environments are generated by imperative
definitions within the semantics of the operator that generates the environment.

The examples of (Weber 2012) and (Rabern 2012) are repeated here. First Weber suggests
that (4.1a)/(4.1b) have the same content:

(4.1) (a) It is raining in Canberra.

\(^1\) (Stoy 1977, 5)
(b) It is raining in Canberra on the 22nd of August 2010 at 2:36pm.

In the framework under development, the semantic type of (4.1a)/(4.1b) is shared. They are both type BOOL. There is a further claim that $\Gamma \vdash a : \text{BOOL} = b : \text{BOOL}$, but this is not the case for (4.2a)/(4.2b):

(4.2) (a) It is always the case that it is raining in Canberra.

(b) It is always the case that it is raining in Canberra on the 22nd of August 2010 at 2:36pm.

In this pair the first example (4.2a) ignores some constant values that have been explicitly filled in (4.2b), generating a divergence in truth value. At the level of the sentence, however, both (4.2a)/(4.2b) are of type BOOL, but the semantic content of the subsentential environment diverges. Under the scope of ‘it is always the case that’, the content which has the same surface form as (4.1a) diverges in (4.2a).

(Rabern 2012) makes the same fundamental error when suggesting that the semantic content of the below (a) cases should be the same as the semantic content found in the subsentential environments that share their surface forms in the (b) and (c) cases:

(4.3) (a) Dave might be in Oxford.

(b) It is consistent with what I know that Dave is in Oxford.

(c) Leon said that Dave might be in Oxford.
(4.4) (a) Licorice is tasty.
(b) Licorice is tasty to me.
(c) According to Jonathan Licorice is tasty.

(4.5) (a) He is mortal. [Socrates identified as salient male.]
(b) Socrates is mortal.
(c) Every man is such that he is mortal.

The content of each of (4.3 a-4.5 c) is of type BOOL, but the assumption ES seeks to connect these examples on a different level. Specifically, it is that the embedding environments themselves are, in the ideology of the opponents, scoped over by an intensional operator.

All such intensional operators are in fact control operations, which are imperative by their very nature, and it is this characterization which catches the natural class of such problems. Once you recognize how these control structures work, it becomes easy to see why they violate referential transparency. This violation is not identical with a violation of compositionality.

To show this first let us return to a basic construction with a unit type capable of defining functions with side-effects. In languages with side-effects, such as this, a sequencing notation \( t_1; t_2 \) may be defined. In languages with side-effects, often it is the side-effect, not the result, that is more important to the reader in the sequencing notation, it is not uncommon for both \( t_1 \) and \( t_2 \) to return undefined. It is what else that has happened within the function
body that is important:

**Terms** $t ::= \ldots \mid \text{unit}$

**Values** $v ::= \ldots \mid \text{unit}$

**Types** $T ::= \ldots \mid \text{Unit}$

**Typing Rules** $\Gamma \vdash t : T$

\[
\Gamma \vdash \text{unit} : \text{Unit} \quad (\text{T-Unit})
\]

\[
\frac{\Gamma \vdash t_1 : \text{Unit} \quad \Gamma \vdash t_2 : T_2}{\Gamma \vdash t_1 ; t_2 : T_2} \quad (\text{T-Seq})
\]

**Evaluation Rules** $t \to t'$

\[
\frac{t_1 \to t'_1}{t_1 ; t_2 \to t'_1 ; t_2} \quad (\text{E-Seq})
\]

\[
\text{unit} ; t_2 \to t_2 \quad (\text{E-SeqNext})
\]

**Derived Form** $t_1 ; t_2 =_{df} (\lambda x : \text{Unit}.t_2)t_1$, where $x \not\in \text{FV}(t_2)$

The derived form can take the place of the second typing rule and both evaluation rules.

This formalization can be extended to explicitly deal with a referencing mechanism and a store, as the $\lambda\mu$-calculus above, with the following typing rules:

**Typing Rules** $\Gamma \vdash t : T$
\[ \Gamma \vdash t_1 : T_1 \quad \Gamma \vdash \text{ref } t_1 : \text{Ref } T_1 \qquad (\text{T-Ref}) \]

\[ \Gamma \vdash t_1 : \text{Ref } T_1 \quad \Gamma \vdash !t_1 : T_1 \qquad (\text{T-DeRef}) \]

\[ \Gamma \vdash t_1 : \text{Ref } T_1 \quad \Gamma \vdash t_2 : T_1 \quad \Gamma \vdash t_1 := t_2 : \text{Unit} \qquad (\text{T-Assign}) \]

As can be seen above, the assessment of (T-Assign) shows that it is the side-effect that is more important than the result. The evaluation of a term may cause side-effects on the store that affect the evaluation of future terms. To handle this encumbrance, as we have already seen in the \(\lambda\mu\) system, we must relativize the evaluation rules to both a term and a store. There are various ways of doing this. We have seen two, a third is presented in (Barker and Shah 2014), where a new "towers" notation is developed to help parse issues of scope. Since here the issues are not scope related, I will not be going into the details of these orthogonal accounts.

What must be noticed here is that the extension of the typing system \(\lambda_{\rightarrow \text{Unit Ref}}\), there is a definition of the assignment function in the form of \(t := t'\) (assignment). This is a valid term in the syntax. Once you have a unit type, which can represent an undefined state of a function which carries an operation that has not been returned, as well as a sequencing operation such as ‘;’ that processes these functions, then you may use functions with unit types to re-assign variables. The re-assignment happens within the store. Without a store there would be no temporary location to hold the new value of the variable. In the context of
our previous discussions, we need only reflect on the definitions of our First-Order quantifiers or the Intensional Operators that mimic them:

**Existential 1** For a model \( M \) and variable assignment \( \mu \), \( M, \mu \models \exists x \phi(x) \) is true according to \( M \) and \( \mu \) if there exists an evaluation \( \mu' \) of the variables that only differs from \( \mu \) regarding the evaluation of \( x \) and such that \( \phi \) is true according to the interpretation \( M \) and the variable assignment \( \mu' \).

This formal definition captures the idea that \( \exists x \phi(x) \) is true if and only if there is a way to choose a value for \( x \) such that \( \phi(x) \) is satisfied.

**Universal 1** For a model \( M \) and assignment \( \mu \), \( M, \mu \models \forall x \phi(x) \) is true according to \( M \) and \( \mu \) if \( \phi(x) \) is true for every pair composed by the interpretation \( M \) and some variable assignment \( \mu' \) that differs from \( \mu \) only on the value of \( x \). This captures the idea that \( \forall x \phi(x) \) is true if every possible choice of a value for \( x \) causes \( \phi(x) \) to be true.

In each of these cases the store \( \mu \) must be re-evaluated point-wise at an index. In more complex cases, this could be expanded to evaluation at multiple indices, or relations over indices. What matters is that \( \mu \) shifts, or changes. This is a control operation. It can be handled by a function that returns the type \( \text{Unit} \). This does not look like a compositional operation, but by the sequencing operation ‘;’, it combines with any function of appropriate type.

There is a second version of the definitions that attempts to avoid this problem:
**Existential 2** For a model $M$, $M \models \exists x \phi(x)$ if there is some $d$ in the domain of discourse such that for every free occurrence of $x \in \phi$, $\phi[x := c_d]$ is true according to $M$.

**Universal 2** For a model $M$, $M \models \forall x \phi(x)$ if, for every $d$ in the domain of discourse such that for every free occurrence of $x \in \phi$, $\phi[x := c_d]$ is true according to $M$.

But here the store $\mu$ is simply replaced by a suppressed reference to a contextual domain: one of the focuses of this dissertation. But the linearization of a domain of discourse is a merely notational difference. Computationally, each storage location $\mu$ requires space in memory, which constitutes it’s domain. It becomes a function in this setting when the locations it stores are mapped. Pointing out that the domain of alternation for a particular open space in a formula can be generated by context, rather than by a function, is not a difference at all.

As has been argued at length, modal operators can be extensionalized in the following way:

**Necessity** For a model $M$ and world $w$, $M, w \models \Box \phi$ iff $M, w \vdash \forall w' \phi(w')$.

**Possibility** For a model $M$ and a world $w$, $M, w \models \Diamond \phi$ iff $M, w \models \exists w' \phi(w')$.

As you can see, I have written these definitions in quantifier form, highlighting the similarity. Extending these operations to sequences of arbitrary length is academic.

The curry-howard isomorphism generates natural constraints on logical form, this differs from the artificial constraints enforced by an assumption like $ES$. The logical form of the
sentence is constrained by the type theory which follows logical rules. The optimal result of this system is that there should always exist a proof of a well-formed sentence. Every term given to you by LF should be typable. Thus, intensionality is a part of the model, not a connection between two different kinds of syntax.

These examples which require the assumption ES are used to argue that compositionality of the language fails. They claim that the only type of compositionality is that generated by an applicative model. I have developed an applicative model, and it in fact captures Rabern’s argument that the ordering source must be part of the compositional structure. Following (Glanzberg and King m.s.), in the tradition of (May 1985), take the following example:

\[(4.6) \left[_{CP}Eros_{VP}loves_{DP}every\ \text{woman}\right].\]

\[(4.7) \left[_{CP}_{DP}every\ \text{woman}\right]\left[\left[1\left[_{CP}Eros_{VP}loves\ t_{1}\right]\right]\right].\]

To generate an interpretation from the quantifier raising a rule that interprets ‘1’ as a lambda is used:

**PM** Let \(\alpha\) be a branching node and \(\beta, \gamma\) be \(\alpha\)’s daughters, where \(\beta\) dominates only a numerical index \(i\). Then, for any variable assignment \(g\), \([\alpha]^g = \lambda x.\left[\gamma\right]^{g[i:=x]}\).

This rule is in the similar spirit of the earlier VIFA:\(^2\)

**VIFA** If \(\alpha\) is a branching node and \(\{\beta, \gamma\}\) the set of its daughters, \(\beta\) is of type \(\langle\langle a, b\rangle c\rangle\) and \(\gamma\) is of type \(b\) and contains a free variable indexed \(i\) of type \(a\), and \((\lambda x_{a}.\left[\gamma\right]^{g[i:=x]}\) is in

\(^2\)See (Glanzberg and King m.s.) for a detailed comparison of the two rules.
the domain of $[\beta]^g$, then $[\alpha]^g = [\beta]^g(\lambda x_a. [\gamma]^g[i:=x])$.

The difference being VIFA does not require the ‘1’ node to exist in (4.7), only the trace. (Rabern 2013) claims that properly treating these operations in a way that does not require syncategorematic rules such as (PM) and (VIFA) is monstrous in Kaplan’s sense of context shifting operations. In what follows I develop a view where context is factored out of the variable assignment used to assign meaningful objects to indices and is not ever shifted. This view uses continuations, as (Rabern 2013) does in his interpretation of (4.7), repeated here, following (Glanzberg and King m.s.):

(4.8) ‘Eros loves $t_1$’ is of type $\langle \gamma, t \rangle$ (instead of $t$).

(4.9) ‘1’ is of type $\langle \langle \gamma, t \rangle, \langle \gamma, e \rangle, \langle \gamma, t \rangle \rangle$ (no type before).

(4.10) ‘$[1_{CP\text{Eros}[VP\text{loves } t_1]}]$’ is of type $\langle \langle \gamma, e \rangle, \langle \gamma, t \rangle \rangle$ (instead of type $\langle e, t \rangle$).

(4.11) ‘Every woman’ is of type $\langle \langle \langle \gamma, e \rangle, \langle \gamma, t \rangle \rangle, \langle \gamma, t \rangle \rangle$ (instead of type $\langle \langle e, t \rangle, t \rangle \rangle$).

(4.12) ‘loves’ is of type $\langle \langle \gamma, e \rangle, \langle \langle \gamma, e \rangle, \langle \gamma, t \rangle \rangle \rangle$ (instead of type $\langle e, e, t \rangle \rangle$).

(4.13) ‘Eros’ and ‘$t_1$’ are of type $\langle \gamma, e \rangle$ (instead of type $e$).

(4.14) ‘woman’ is of type $\langle \langle \gamma, e \rangle, \langle \gamma, t \rangle \rangle$ (instead of type $\langle e, t \rangle$).

(4.15) (4.7) is of type $\langle \gamma, t \rangle$ (instead of type $t$).

These definitions are in continuation-passing style and could be rewritten with a store. The claim being made is that an applicative structure is the only compositional structure, but
(Glanzberg and King m.s.) disagree. They claim that compositionality in natural language does not require the strength of an applicative model. They define composition within the following structure:\(^3\)

Think of the syntax of a language as consisting of a set \(E\) of (simple and complex) expressions and a set of “syntactic rules” \(F\) such that each rule in \(F\) is an \(n\)-ary partial function \(f\) (for some \(n\)) taking \(n\) members of \(E\) as arguments and yielding a member of \(E\) as value if \(f\) is defined for the arguments in question. So we can identify the syntax of a language with a *partial algebra* \(\langle E, F \rangle\), with \(E\) and \(F\) understood as above. The members of \(F\) in such partial algebras are often referred to as *operations on* \(E\).

The example given is that ‘most dogs’ will be interpreted as \(f_{DP}(most, dogs) = [DP[D_{most}] [NP_dogs]]\) for an \(f_{DP} \in F\). This assumes complex members of \(E\) have constituent structure, which is innocent. Now consider a meaning assignment \(\kappa\) that is a function from \(E\) to the set \(M\) of meanings for expressions in \(E\). Consider \(f \in F\) an \(n\)-place operation on \(E\). The meaning function \(\kappa\) is a function \(\kappa(\alpha, j)\) from \(E\) and the relevant set of parameters \(j\) for expressions in \(E\). Supposing a type adjusting rule \(R\) built from families of meanings generated by manipulating parameters in \(j\), \(R\) will provide potentially distinct type adjusting functions for each argument place \(i\), which we can write \(T_R^i(\kappa, j, e_i)\). (Glanzberg and King m.s., 35) form a definition of compositionality similar to (Pagin and Westerståhl 2010):

\(^3\)(Glanzberg and King m.s., 20)
Strong Type Adjusting Compositionality A function $\kappa$ is \textit{strongly type adjusting compositional with respect to} $f$ iff there is an $n$-place meaning operation $r_f$ (a function from $M^n$ to $M$) s.t. given expressions $e_1, \ldots, e_n \in E$ for which $f$ is defined, for every (value of parameters in) $j$, $\kappa(f(e_1 \ldots e_n), j) = r_f(T_R^n(\kappa, j, e_1), \ldots, T_R^n(\kappa, j, e_n))$.

$\kappa$ is \textit{strongly type adjusting compositional} iff it is strongly type adjusting compositional with respect to every $f \in F$.

This rule is weaker than the simple applicative compositionality we have been considering, but arguably it may be all natural language requires, which would make the rule (VIFA) the simplest way of handling intensional constructions. However, Pagin and Westerståhl’s geberal compositionality has shifted meanings computed compositionally from other shifted meanings, while type adjusting compositionality shifts meanings locally, in the process of composition. This may be considered a benefit, but it does mean that type adjusted meanings are not technically meaning assignments, although there are special cases where they align. The variable IFA account allows for, in the terminology of Peter Pagin, a “generalized switch” able to treat a range of quantifiers and operators spanning the standardly assumed classes.\(^4\)

I go on to show how an applicative model might be used to handle intensional constructions, using the K combinator to eliminate argument heads on the basis of syntactic feature sets. This model allows for a specific integration of the syntax and semantics that eliminates any claim that the system could be monstrous, while simultaneously allowing constructions\(^4\) (Glanzberg and King m.s., 34-35).
that, in English, would seem monstrous, such as interpreting ‘John said I am hungry’ as ‘John said John is hungry’ when John is not the speaker. These constructions are found in languages other than English, and if the feature set carried by the verb said, in these other languages, allows for this construction, so be it.

**Syntax and Sententiality**

While the approaches of King and Schaffer correctly reject ES and oppose the operators argument, their positive proposals are couched in artificial logical languages. I argue that the envelope should be pushed. Rather than couching the debate in a first-order language, I develop the tools of generative syntax necessary to describe the data at issue.

In opposing the operators argument, both (King 2003), (King 2007) and (Schaffer 2012) note the expressive equivalence between sentential operators and object-language quantifiers.\(^5\) King argues that the formal landscape need not be exhausted by an operators treatment of the phenomena considered by Kaplan and Lewis. I will be extending this conversation to the comparison between operators and generalized quantifiers of various types. Considering tense, King puts the point in the following way:\(^6\)

\[\text{[I]f the proper way to treat tenses is not as index shifting sentence operators,}\]

\[\text{then there is no need for temporal coordinates in indices of evaluation. This, in}\]

\(^5\) Cf. (Vlach 1973), (van Benthem 1977), (Quine 1960a), and (Cresswell 1990)
\(^6\) (King 2003, 223).
turn, means that we are no longer forced to hold that [...] semantic values are, or determine, functions from worlds, locations, standards of precision and times to truth values, as Lewis claimed.

Speaking just in terms of tense and times for the moment, if tense is not to be translated as an index-shifting sentence operator, then not only could indices be removed from the index of evaluation, as King notes, but the functional phrase governing tense would be free to take any type of complement. This differs from the operators approach, which would require the tense phrase to take a sentential complement, for no other reason than to force a certain logical analysis of the “deep structure”.

Given the close parallel between tense and pronouns pointed out by (Partee 1973), following (Kusumoto 1999), King treats tense through the use of covert pronominals and relations over them in the object language. To see an example of this analysis, let \( t^* \) designate the time of the context; let yesterday denote the temporal interval covering the day before the day of utterance; and let ‘\( \leq \)’ denote a part of relation over times.\(^7\)

\[
\begin{align*}
(4.16) & \quad (a) \text{ Maggy is happy.} \\
& \quad (b) \exists t (t = t^* \land \text{happy}(Maggy)(t)).
\end{align*}
\]

\[
\begin{align*}
(4.17) & \quad (a) \text{ Maggy was happy.} \\
& \quad (b) \exists t (t < t^* \land \text{happy}(Maggy)(t)).
\end{align*}
\]

\(^7\)Cf. (King 2003, 221-222).
(4.18) (a) Maggy was happy yesterday.

(b) \( \exists t (t < t^* \land t \leq \text{yesterday} \land \text{happy}(Maggy)(t)) \).

Here we have a non-past utterance in (4.16a)/(4.16b), a past tense utterance in
(4.17a)/(4.17b), and a temporal frame adverbial serving as a predicate of times in
(4.18a)/(4.18b).

This kind of argument has been further extended by (Schaffer 2012). Schaffer suggests,
along with (Stone 1997), (Percus 2001), (von Stechow 2003), (Schlenker 2006), and (Szabolcsi
2011), that this extensionalization strategy should extend to mood and modality, as well.

(4.19) (a) Maggy is happy.

(b) \( \exists t (t = t^* \land \text{happy}(Maggy)(w^*)(t)) \).

(4.20) (a) Maggy might be happy.

(b) \( \exists t (t = t^* \land \text{might}(w^*)(\lambda x.\text{happy}(Maggy)(x)(t)))). \)

Here we have a covert pronominal element representing the world of the context, ‘\( w^* \)’, and
what were intransitive verbs are now relativized to both worlds and times.

Up to this point, we have still been operating within the framework of an artificial
language. The arguments made here are all related to expressive equivalence between two
formal languages. While this approach is consistent with the rejection of ES, as it does not
rely on formal operators, it still relies on, for example, quantifiers of a first-order logic. In
what follows, I develop, in more detail, the particulars of a grammar in the generative style, and show that these details provide good reason to abandon this approach in favor of that sketched by the second diagram above.

Philosophers used to the simple syntax of artificial languages might consider a sentence to be simply the scope of an operator $\Box(...)$, or a closed formula such as $\exists x (Fx \lor \neg Fx)$. In fact, while in its early development, generative syntax attempted to mirror this structure by labeling a node ‘S’. Unfortunately, due to the research on the left periphery, this simple assumption has since been discarded. Formulated in an X-bar ($\bar{X}$) framework, contemporary generative grammar provides the following recipe for a sentential structure:

With this in mind, the displacement of the direct object in the question formation above can be modeled syntactically in the following way:

(4.21) What did you say Jim brought?

8I have not split the IP into additional AgrP functional elements along the lines of (Pollock 1989), (Chomsky 1989), (Chomsky 1995), as this additional complication is not necessary for the arguments that follow.
Figure 4.2: The Sentence
This kind of displacement is central to, for example, (May 1985)’s explanation of quan-
tificational determiners found in object position, where a level of logical form is realized as an output of \( C'_{HL} \). The phrase structure trees are interpreted directly in a type-driven framework.

The observations made previously regarding quantifier scope can then be seen to operate by the same mechanism.\(^9\) If one is attempting to get operators into the system, however, focusing on the property of surface compositionality is a more respectable approach than simply assuming \( ES \).

(4.22) Every student likes some professor.

\(^9\)Note that quantifier raising can be eliminated, along with the variables, through the use of a categorical logic (See (Jacobson 1999)).
In the above example, the two readings, $\forall \exists$ and $\exists \forall$, are distinguished based on movement out of their argument positions into binding position. According to this system, the denotation of quantificational determiners are just higher order relations over sets. For example, some denotes the relation $R = \{\langle X, Y \rangle : X \cap Y \neq \emptyset\}$. The two sets at issue are termed the restriction and the scope. Consider:

(4.23) Some dog is barking.

In this case $\{x : x \text{ is a dog}\}$ is the restriction and $\{y : y \text{ is barking}\}$ the scope. For the determiner some, what is being determined is whether the intersection of the restriction and
scope is nonempty. If so, there exists a barking dog, and (5.3) is true.\textsuperscript{10} In (5.2), either the DP \textit{every student} is taking as an argument the property of \textit{liking some professor} or \textit{some professor} is taking as an argument the property of \textit{being liked by every student}.

At this point, it is worth noting that both King and Schaffer have been operating in an intermediate translation language, which is fine for illustrative purposes, but, as noted above, is independent of a direct semantic interpretation of LF at the syntax-semantics interface. In fact, on the direct approach there are no first-order quantifiers of the type mentioned in (4.16a)-(4.20b) above. Instead, there are simply higher-order relations over sets, according to the system of generalized quantification just outlined.\textsuperscript{11} Aside from the explanatory power and theoretical fecundity of the theory of generalized quantification, one might nonetheless wonder why first-order quantifiers are not adequate in the realm of generative syntax. (Cooper 1977) explains this eloquently. He points out that the following is not a syntactically motivated structure of English:

\begin{equation}
\text{(4.24) Some dog is barking.}
\end{equation}

\textsuperscript{10} Cf. (May 1985) and the references therein for a detailed explanation of the set theoretic operations that correspond to the various determiners.

\textsuperscript{11} Cf. (Barwise and Cooper 1981), (Heim and Kratzer 1998).
Specifically, there is evidence that some dog forms a constituent at some point in the derivation, as would be predicted standardly:

(4.25) Some dog is barking.

The structure in (4.24) is what would be required syntactically were quantificational determiners treated as the standard quantifiers of first-order logic. Cooper notes that “[s]aying
that our semantic theory requires [(4.24)], then, is rather like saying that the semantics cannot treat English itself, but a language that is like English except that it has structures like [(4.24)]." On the other hand, (4.25) is what is predicted when generalized quantifiers are introduced.

This type of interpretation generalizes straightforwardly to a treatment of tense and mood, as the functional projections in the left periphery of the clause that are required to account for inflectional morphology would provide adequate landing sites for these pronominal elements. It is worth noting, however, that this is exactly what King and Schaffer aim for, as generalized quantifiers combine compositionally with NP and VP denotations rather than taking a sentence as an argument and parametrically shifting its point of evaluation.

Sentence Schmentence

(Lewis 1980) describes the strategy taken so far:13

We can perfectly well build a compositional grammar in which it never happens that sentences are constituents of other sentences, or of anything else. [...] In this grammar sentences are the output, but never an intermediate step, of the compositional process.

I would first like to amend this picture slightly. The schmentence need not refuse the view

12(Cooper 1977, 898).
13(Lewis 1999, 32).
that there are functions of type \(< s, s >\). One less generalized version of negation than I
prefer is of this type. The better version has this type as an instance, or it is an example of
a formula schema, you might say:

\[\text{Negation} ~ \neg : \langle\langle \alpha, \beta \rangle, \langle \alpha, \beta \rangle \rangle := \lambda P_{\langle \alpha, \beta \rangle}. [D_\alpha \times D_\beta / P]\]

I am not the only person who has embraced schmentencism to a degree. (Schroeder 2011)
considers at least one reading of a deontic modal auxiliary verb scoping over its prejacent as
a syntactic control construction, where the object that the operator requires is not itself a
sentence but instead a property, as I advocate.

This picture I am fighting against might be illustrated with an analogy: a closed formula
is one where open variables have been bound by operators such as \(\exists\) and \(\forall\). Taking this
analogy further, if operators embed sentences, then in the closed formula \(\exists x (F x \lor \neg F x)\), the
sentence embedded under \(\exists\) would be \((F x \lor \neg F x)\). But there is a problem! According to the
rules of our language, this is an open formula, rather than a sentence. The schmentencite
strategy suggests that the complements embedded (for example in modals and conditionals)
act like the open formulae above.

(Enderton 2001), for example, provides a precise concept of definability on a structure:
Consider a structure \(\mathcal{M}\) and a formula \(\varphi\) whose free variables are among \(x_1, \ldots, x_n\). Con-
struct an \(n\)-ary relation on \(|\mathcal{M}|\)

\[\{ \langle a_1, \ldots, a_n \rangle | \models_{\mathcal{M}} \varphi[a_1, \ldots, a_n] \}\]
Call this the \( n \)-ary relation \( \varphi \) defines in \( \mathcal{M} \). An \( n \)-ary relation on \( |\mathcal{M}| \) is said to be definable in \( \mathcal{M} \) iff there is a formula (whose free variables are among \( x_1, \ldots, x_n \)) that defines it there. So the idea captured by Lewis above is one where, each individual part of the syntactic structures above come together to form a sentence. When looking at the individual parts, however, we find instead open formula, which define certain properties on \( |\mathcal{M}| \). (Enderton 2001, 90) provides the following example:

\[
\text{Take the directed graph}
\]

\[
\mathcal{U} = \langle \{a, b, c\}, \{\langle a, b\rangle, \langle a, c\rangle\} \rangle
\]

where the language has parameters \( \forall \) and \( \exists \). Then in \( \mathcal{U} \), the set \( b, c \) (the range of the relation \( E^U \)) is defined by the formula \( \exists v_2(Ev_2v_1) \) In contrast, the set \( b \) is not definable in \( \mathcal{U} \). This is because there is no definable property in this structure that would separate \( b \) and \( c \).

Here we have an example of another connection between formal logical models and linguistic LF that is only uncovered once ES is rejected.

If this example is not helpful, let us contrast the interpretation of a first-order language with the lambda calculus. There is an intuitive connection to the typed-driven interpretation of the syntactic trees we have been discussing. The received standard view in linguistics is already schmentencite. Lambda terms provide a way of stating a closed formula with the content of its open counterpart in a formal system. Let \( P \) define some property, for
concreteness, the property *is red*. Now compare the open formulae of First-Order logic (4.26a) with the λ-term (4.26b):

(4.26) (a) $Px$

(b) $\lambda x.[Px]$

First, we have (4.26a), an open formula. It is well-formed, but as ‘$x$’ is not bound, it is not a sentence. On the contrary, (4.26 b) is closed by the $\lambda$. However, the denotation of (4.26a) and (4.26 b) is identical. According to the definition above, the denotation of $Px$ is the set $\{y|\models_M Px[y]\}$. The denotation of $\lambda x.[Px]$ on the other hand is $\{y|\lambda x.[Px](y) = 1\}$. By simply chasing definitions, both turn out to be the set $\{x|x$ is red$\} \subseteq D_M$, which is just the set of red things in the domain of the model.

Now consider the following pair, with some minimal syntactic and semantic structure revealed:

(4.27) John hopes to be home.
(4.28) John is home.

In the above pair, we have a candidate for an operators analysis. But the term in the complement of *hopes* in (4.27) is not identical to the term in (4.28). One is a variable which
is then bound by a $\lambda$ through the course of derivation. The other is a constant. Further, the argument taken by *ought* is of type $\langle e, t \rangle$, which is generated by an intensional function application rule such as (VIFA):\(^{14}\)

**VIFA** If $\alpha$ is a branching node and $\{\beta, \gamma\}$ the set of its daughters, $\beta$ is of type $\langle \langle a, b \rangle \rangle$ and $\gamma$ is of type $b$ and contains a free variable indexed $i$ of type $a$, and $(\lambda x_a. [\gamma] g[i := x])$ is in the domain of $[\beta]^g$, then $[\alpha]^g = [\beta]^g(\lambda x_a. [\gamma] g[i := x]).$

This rule only applies to arguments in the functional projection of the matrix clause.

While the attitude verb might take the same type of clause, a TP, as it’s complement, there are two properties of this clause that differ from (4.28). First, as an infinitival, the TP embedded under *hopes* in (4.27) has $\emptyset$ tense, as can be seen by the lack of inflection in the complement. Second, *hopes* seems to demonstrate subject control. I follow (Boeckx, Hornstein and Nunes 2010) in modeling control as movement. But some would traditionally leave a PRO in place of a trace argument. This results in the denotation of the entire clause being $t$ temporarily in the derivational process, where the trace acts as an open, but indirectly indexed, variable. But notice that, even though the TP share a type, their denotation could easily diverge. The initial values of $g$, when not set by context, are completely arbitrary. There is no guarantee that the initial value of $\langle John \rangle = John$. To generate this reading, the variable must be $\lambda$-bound when provided as an argument of *hopes* to generate a *de se* reading. Thus, while the final type of (4.28) is $t$, the argument of the verb here considered

\(^{14}\)See (Glanzberg and King m.s.) for a defense of this principle.
is of type \( (e, t) \)\(^{15}\)

So far, the intensional component has been ignored, but following the schmentencite strategy is straightforward. First, the arity of lexical verbs must be increased to accommodate a world and time, as these elements must move into the left periphery according to how the agreement relations for \([±indictive]\) and \([±past]\) are spelled out, in much the same way that the subject moves out of VP. This allows for these pronominals to serve as the arguments of modal auxiliaries, as they climb their way up the syntactic tree. Given a simple referential analysis of these pronominals, all that would change in the above argument is that the contrast between the two TP in (4.27) and (4.28) would be thus:

\[
\begin{align*}
(4.29) \quad (a) & \quad home(x)(t)(w) \\
(b) & \quad home(John)(w^*)(t^*)
\end{align*}
\]

Where the \(-^*\) diacritic forces the variables interpretation to be that dictated by the context.\(^{16}\) The difference here, aside from the difference between a variable and a constant seen with \textit{John} and \(x\), is that the variables in (4.29a) do not have diacritics restricting their interpretation. Thus, the denotations of (4.29a) and (4.29b) could potentially diverge in three different ways.

Another key point is that the class of auxiliaries and adverbs have many potential landing sites in the left periphery, and moving higher or lower along the inflectional spine can change

\(^{15}\)I have suppressed quite a few details of the semantic derivation here. For a more detailed derivation in these kinds of cases, see (Cook 2013).

\(^{16}\)Cf. (Schlenker 2003), (Schlenker 2006), (von Stechow 2003)
the semantic type of the complement required. For example, the linguistic data might suggest that the epistemic modals *might, must* are ambiguous between a *de dicto* reading and a *de se* reading, such as the following:

(4.30) (a) John must be late.

(b) I might be Lingens.

In this case, *might* in (4.30b) would require a property of centered worlds, abstracting out the subject of the complement, : \( \lambda w_i. \lambda t_r. \lambda x_e. [is(Lingens)(x)(t)(w)] \). The example (4.30b) then has an analysis similar to (4.27) above. This differs from the analysis of (4.30a), which requires the subject as an argument of the VP prior to combining with *must*: \( \lambda w_i. \lambda t_r. [late(John)(t)(w)] \). Thus, it would seem that there is no single semantic category to be associated with the complement of modal auxiliaries. Further, the complements under consideration here all act as if they were open properties, although \( \lambda \)-bound. Thus, we have encapsulated the schmentencite strategy. The complement of embedding environments, according to this story, more accurately resemble open formulae than they do sentences.

It is not clear why (Lewis 1980) goes on to consider this type of move “both cheap and pointless”, for, the discerning reader would note, this view is precisely parallel to a view espoused by (Lewis 1979). In “Attitudes *De Dicto* and *De Se*”, Lewis argues that the objects of the propositional attitudes are properties, rather than propositions. Properties much like \( \lambda w_i. \lambda t_r. \lambda x_e. [home(x)(t)(w)] \), the embedded complement in (4.27). Indeed, Lewis recognizes

\(^{17}\) (Lewis 1999, 33).
this position in his (1979) when he advocates:¹⁸

Some philosophers would favor sentential objects, drawn either from natural language or from some hypothetical language of thought. Others would favor sentence meanings, entities enough like sentences to have syntactic structure and indexicality. If you are of one of these persuasions, my advice to you is by no means new: do not limit yourself to complete, closed, nonindexical sentences or meanings. Be prepared to use predicates, open sentences, indexical sentences, or meanings thereof—something that can be taken to express properties rather than propositions.

The position provided for by (Lewis 1979) is then exactly what is observed when syntactic structure is taken seriously. Unfortunately, if anything is shown by the widespread reliance upon ES, it is that the view of (Lewis 1980) has become dominant.

At this point, one might object: operators were only required to take sentences in propositional modal logic. In quantified modal logic (QML), operators can perfectly well take open formula, for consider:

$$\exists x \Box P x$$

In this case the open formula $\Box P x$ is perfectly well defined. I agree with this point, and, further, believe it actually supports my position. Consider how $\Box$ affects the relation $P x$

¹⁸(Lewis 1983, 150).
defines on $|\mathcal{M}_{QML}|$:

$$\{y | \models_{\mathcal{M}_{QML}} P_x[y] \} \subseteq \{y | \models_{\mathcal{M}_{QML}} \Box P_x[y] \}$$

To wit, the necessarily red things are a subset of the red things. But there is already a class of lexical items with this effect on the denotation of a predicate: the non-intersective adjectives. This lexical class already has a well defined denotation: $\langle \langle e, t \rangle \langle e, t \rangle \rangle$. Consider:

(4.31) (a) The very red ball

(b) The necessarily red ball

In just the same way that the set of very red things is a subset of the set of red things, the set of necessarily red things is a (possibly empty) subset of the red things: $[[\text{necessarily red}]] \subseteq [[\text{red}]]$. Notice that the converse is true of the modal dual:

(4.34) (a) The almost hero

(b) The possible hero

$^{19}$Notice that while necessary might be considered intersective, necessarily is at best subsective:

(4.32) (a) The CIA is a necessary evil.

(b) The CIA is an agency.

(c) $\Rightarrow$ The CIA is a necessary agency.

(4.33) (a) The witness is necessarily silent.

(b) $\not\Rightarrow$ The witness is necessary.
The adjective in (4.34a) is *privative*: \[\text{almost hero} \cap \text{hero} = \emptyset\]. There are no heros in the set of almost heros. The adjective *possible* could be analyzed in the same way: a possible moon is not a moon. So it would seem that a bridge might be built between quantified modal logic and the class of modal non-intersective adjectives. But this is just the type of bridge I have been advocating. In fact, assuming this kind of connection points to an interesting case: in any quantified modal logic stronger than \(T\), \(P \rightarrow \Diamond P\) is validated. If this were the natural logic of English, however, then the following questionably contradictory sentence should be considered acceptable:

(4.35) ?? The moon is a possible moon.

If (4.35) is deemed acceptable, then *possible* should have a *hypernym* denotation, rather than a privative one: \([\text{possible moon}] \supseteq [\text{moon}]\). In this case everything that is a moon is a possible moon, and there could even be more things that are possible moons, besides. Regardless of the particular denotation, each of these cases provides a denotation structure that, once intensionalized, is of type \(\langle\langle s\langle et\rangle \rangle\rangle\).\(^{20}\) They are mappings between sets (predicate/noun denotations) that follow the particular constraints of their lexical class, as outlined above.

\(^{20}\) See Appendix 1 for further intensionalization options. The specific method of generating intensional constructions does not affect the details of the proposal above.
The Critics

I now consider the arguments of (Weber 2012) and (Rabern 2012), whose reliance on faulty premises renders their arguments vacuous, if valid. (Weber 2012) argues that the extensionalization strategy of King and Schaffer is not a successful dodge. Considering tense, Weber introduces eternalization pairs:

Eternalization Members of an eternalization pair have the same semantic value – they express the same eternal proposition.

Consider the following minimal pair:

(4.36) (a) It is raining in Canberra.

(b) It is raining in Canberra on the 22nd of August 2010 at 2:36pm.

Weber suggests that while (4.36a)/(4.36b) have the same content, the following embedding generates a divergence in truth value:

(4.37) (a) It is always the case that it is raining in Canberra.

(b) It is always the case that it is raining in Canberra on the 22nd of August 2010 at 2:36pm.

This divergence supposedly generates a problem for King’s strategy along the same lines as the operators argument: there is a substitution failure for (4.36a)/(4.36b). But this
substitution failure is only a problem if one assumes that a sentence and the complement
of an embedding environment are the same type of entity, and can therefore be substituted
salva veritate. If there were no guarantee that any given embedding environment would
require the same type of entity as any other embedding environment, then there would be
no reason to think that providing a sentence as an argument in these cases would even result
in a well-formed structure. As has been seen above, when something that could be considered
a “sentence” enters into an embedding environment, the binding configurations that the verb
subcategorizes for must be considered. These differences can potentially change the semantic
type of the complement. Thus, when (4.36a)/(4.36b) are considered as the complements of
the matrix verbs in (4.37a)/(4.37b) above, they should no longer be considered eternalization
pairs.

Rabern (2012) also relies on ES when he suggests that if one assumes an identity thesis
of assertoric content and semantic content, the intuition that the (a) and (b) cases below
“say the same thing” poses a problem for the compositional semantics of the (c) cases:

(4.38) (a) Dave might be in Oxford.

(b) It is consistent with what I know that Dave is in Oxford.

(c) Leon said that Dave might be in Oxford.

(4.39) (a) Licorice is tasty.

(b) Licorice is tasty to me.
(c) According to Jonathan *Licorice is tasty.*

(4.40) (a) He is mortal. [Socrates identified as *salient* male.]
(b) Socrates is mortal.
(c) Every man is such that *he is mortal.*

In each of these cases, if one assumes that the (a) and (b) cases have the same semantic content,\(^{21}\) then it is supposedly a problem that the complement of the matrix verb in each of the (c) cases differs in semantic value. Rabern acknowledges his reliance on *ES*, calling it “embedment”, and points out that its denial is tantamount to the schmentencite strategy. This move is an odd development in the history of the philosophy of language. The disparagement of a theory that has been designed specifically with compositional issues in mind, and then the forced application of a framework that was initially only concerned with the properties of sentences extended to compositional issues in this way. What reason would one have to assume that any verb would subcategorize for something with the properties of a matrix clause? In fact, Glanzberg (2011) provides an elaboration of this reaction to *ES*. He points out that a matrix clause requires something more than a simple predication structure such as the *VP*, where a predicate meets an argument. At minimum inflectional elements such as tense and mood, as well allowance for *wh*- movement, complementization, and other types of dislocation are needed.

\(^{21}\)This assumption is objectionable in its own right, but outside the scope of the current inquiry.
Cappelen and Hawthorne have pointed out additional syntactic problems with constructions traditionally considered analyzable in terms of ES, such as ‘S in L’ or ‘S at T’:

Consider ‘in Boston’. One very natural—and utterly standard—account if its syntactic life is that it is an adverb that combines with a verb phrase to compose a verb phrase. On this picture, the syntactic home of ‘in Boston’ is much better revealed by ‘He spends long hours in Boston’ than by ‘In Boston, he spends long hours’. Indeed, the acceptability of the latter is explained by a special rule of fronting that allows us, in certain circumstances, to move adverbs from their home to the front of a sentence.\(^{22}\)

Thus, the environment \(E\) does not combine with a sentence, but rather with a \(VP\). Surface structure can, at times, be misleading. Topicalization by fronting generates the following structure:

\[
\text{(4.41) In Boston, John works.}
\]

\(^{22}\)(Cappelen and Hawthorne 2009, 74).
At this point one additional component has been added: the features $[\pm past]$ and $[\pm ind]$. These syntactic markers are used to distinguish indicative mood from interrogative or subjunctive—in languages with an active subjunctive—and the past tense from English non-past, which can be considered present in the null case. These features can then probe for a
It is not at all clear what reason one would have for thinking that the moved PP should be analyzed semantically as taking a sentential argument. In fact, languages, such as English, with *exceptional case marking (ECM)* provide evidence that a CP is itself not always sentence-like, as it is not always acceptable as a matrix clause:

(4.42) I need \([CP[T_{TP} \text{ him} [T', \emptyset [V_{TP} \text{ to build a wall }]]]\)
In this case the CP has the wrong inflection to be considered a sentence by native speakers.
Embedding environments generate inflectional changes that do not map directly to judgments of sentencehood when taken outside of their environment. One might argue that their logical forms are the same and thus there is an equivalence class which includes sentences that can be defined over these objects, but this reduces to a type of generative semantics.

Since both tense and modal auxiliary verbs live inside $I'$, there is no reason to think that tense or modal phenomena are properly treated as sentential operators at all. One might retreat to the position that the relevant embedded clause is a $CP$ headed by the complementizer *that*, which can be witnessed in many of the examples like (4.37a)/(4.37b), but here restrictive relative clauses provide a stumbling block:

(4.43) John wants $[CP/IP]_{DP}$ the $[NP$ thing]$]_1[CP$ that $[TP you like t_1]]]]$
In (4.43) neither the embedded $DP$ nor its embedded $CP$ can function alone as a matrix clause; both are plainly unassertable. At this point one might remark that sentence-hood is not a syntactic category with any real content. What is important is the matrix clause,
which, by definition, cannot be embedded. Thus, the operators argument requires a vacuous
syntactic premise, and it never gets off the ground in any of its forms.

The recent *cartographic* movement in generative syntax provides some further evidence in
support of these arguments. This movement starts with (Kayne 1994), *The Anti-Symmetry
of Syntax*. In this work, Kayne proposes that phrase structure completely determines linear
order, so, for example, English and Japanese phrases consisting of a verb and its complement
should no longer be thought of as symmetric to one another in hierarchical structure. Instead,
one is derived by movement from the other, more basic, structure. Kayne formulates this
theory in the form of a *linear correspondence axiom*, from which he derives the essentials
of X-bar theory. The details of this proposal are not essential to the current project, but
an extension of this proposal most certainly is. Specifically, (Cinque 1999), (Cinque 2006)
argues that “the functional portion of the clause, in all languages, is constituted by the same,
richly articulated and rigidly ordered, hierarchy of functional projections.” What does this
mean? For our purposes, it means that in between the CP and the VP, there is a rigid
hierarchy, a portion of which is displayed here:

\[
(4.44) \text{MoodP}_{\text{speech act}} > \text{MoodP}_{\text{evaluative}} > \text{MoodP}_{\text{evidential}} > \text{ModP}_{\text{epistemic}} > \text{TP}(\text{Past}) > \\
\text{TP}(\text{Future}) > \text{MoodP}_{\text{irrealis}} > \text{ModP}_{\text{alethic}} > \text{AspP}_{\text{habitual}} > \ldots > \text{ModP}_{\text{volitional}} > \\
\text{AspP}_{\text{celerative}(I)} > \text{TP}(\text{Anterior}) > \text{AspP}_{\text{terminative}} > \ldots > \text{ModP}_{\text{obligation}} > \\
\text{ModP}_{\text{permission/ability}} > \ldots
\]

According to Cinque, all functional verbs are inserted directly into the head of their respective
functional projection, forming a monoclausal structure, rather than analyzing functional verbs according to a biclausal structure in which the functional verb takes a CP complement:

(4.45) Biclausal: $[\text{CP} \ldots [\text{FP} \ldots [\text{FP} \ldots [\text{FP} \ldots \text{VP} V \text{restr} \text{CP} \ldots [\text{FP} \ldots [\text{FP} \ldots \text{VP} V ]]]]]]]$

(4.46) Monoclausal: $[\text{CP} \ldots [\text{FP} \ldots [\text{FP} \ldots \text{VP} V \text{restr} \text{FP} \ldots [\text{FP} \ldots \text{VP} V ]]]]]$

Cinque’s work, based on careful empirical cross-linguistic study, suggests that modal auxiliary verbs such as *must* and *might* are inserted directly into the functional hierarchy above as the head of $\text{ModP}_{\text{epistemic}}$. If this is the case, then it is direct evidence that the verbs which are the classic candidates of the operators analysis do not in fact embed a full CP. This fact removes the last plank under the sentence first position. Even if one were to attempt to identify the classic node label ‘S’ with CP, modal auxiliary verbs would not embed them. Further, according to this decomposition of the inflectional component, attempted identity with any other node label would be completely arbitrary.

In fact, Cinque’s hierarchy condition could be weakened and the argument would still succeed. All that is necessary is that for any given language, there exists a linearization schema that this language follows. These schemata may have general connections, but if linearization even in a language-specific form is a live option as a map of the structure of auxiliary verbs and their prejacent clauses—and the movement required to generate the proper semantic interpretation—then the eternalization assumption is on questionable grounds.
Conclusion

When introducing this project I set up a dilemma: one must choose between (1) a comprehensive formal model of natural language accompanied by a deficient metaphysics or (2) a comprehensive metaphysics accompanied by a cavalier approach to natural linguistic structure. It is my hope that the horns of this dilemma can be avoided, through the construction of semantic bridge principles related to algebraic models and the Curry-Howard Isomorphism. This investigation will reveal some overlooked properties of languages including intensional operators as they have been conceived up to this point.

In the preceding sections, I attempted to develop a picture of natural logical form in line with the generative program. This picture was contrasted with the artificial logical form of formal logic. In the natural picture I developed, logical structure is determined largely by the lexicon, whereas in the formal picture structure is enforced by the requirements of the logical constants. The empirical failures of the formal picture, through consideration of generative semantics and the Katz-Postal hypothesis, were then discussed. It was then argued that even given a package including this natural logical form, there still exists work for the formal picture, just not where the contemporary flow of research has placed it. By bridging the type-theoretic interpretation of the natural logical form, we found that a formal language is capable of capturing the logical properties of a class of logical verbs. By bridging the theories in this way, the study of the logical fragments of natural language can be studied from “the ground up”.
To motivate the view just described, I provided arguments that sought to undermine the following assumption:

\[ E S \] There exist embedding environments \( (E) \) in natural language that take sentential arguments \( (S) \).

Rejecting this assumption is tantamount to rejecting the sentence-first structure of intensional logic. The operators argument of Kaplan and Lewis was considered, where the assumption \textit{uniformity}, that \( S \) is of the same semantic type when it occurs alone or when it combines with \( E \), was shown to fail for versions of Chomsky Grammars. Two contemporary variations on these themes, by (Weber 2012) and (Rabern 2012), are guilty by association, and demonstrate the artificial approach we are rejecting.

A more successful endeavor might start with the type system favored as an interpretation of a generative grammar, and attempt to extract the logic it’s various fragments represent. This is not a position derived from an ideological standpoint related to any specific logical system. Instead I seek to uncover empirical facts about the logical structure of natural languages. Along the way I hope to deepen the general understanding of how the logical form of natural language connects to the mathematical structures studied by logicians.
Chapter 5

Syntax-Semantics Interface

The importance of correctly mapping the syntax-semantics interface locates the role of the model theory in relation to the type system that provides an interpretation of the syntax. I choose a movement theory of syntax as I assume that Chomsky is correct in identifying a key property of natural language: *displacement*. The semantic model should be capable of modeling displacement in a natural way. This displacement property is represented by movement of the direct object in the question formation:

(5.1) What did you say Jim brought?
This kind of displacement is central to, for example, (May 1985)'s explanation of quantifi-
cational determiners found in object position, where a level of logical form is realized as an output of the syntactic derivation. The phrase structure trees are interpreted directly in a type-driven framework.

The observations made previously regarding quantifier scope can then be seen to operate by the same mechanism:

(5.2) Every student likes some professor.

In the above example, the two readings, $\forall\exists$ and $\exists\forall$, are distinguished based on movement out of their argument positions into binding position. According to this system, the denotation of quantificational determiners are just higher order relations over sets. For example, some
denotes the relation \( R = \{\langle X, Y \rangle : X \cap Y \neq \emptyset\} \). The two sets at issue are termed the restriction and the scope. Consider:

(5.3) Some dog is barking.

In this case \( \{x : x \text{ is a dog}\} \) is the restriction and \( \{y : y \text{ is barking}\} \) the scope. For the determiner *some*, what is being determined is whether the intersection of the restriction and scope is nonempty. If so, there exists a barking dog, and (5.3) is true.¹ In (5.2), either the DP *every student* is taking as an argument the property of *liking some professor* or *some professor* is taking as an argument the property of *being liked by every student*.

The recent *cartographic* movement in generative syntax provides some further evidence in support of these arguments.² This movement starts with (Kayne 1994), *The Anti-Symmetry of Syntax*. In this work, Kayne proposes that phrase structure completely determines linear order, so, for example, English and Japanese phrases consisting of a verb and its complement should no longer be thought of as symmetric to one another in hierarchical structure. Instead, one is derived by movement from the other, more basic, structure. Kayne formulates this theory in the form of a *linear correspondence axiom*, from which he derives the essentials of X-bar theory. The details of this proposal are not essential to the current project, but an extension of this proposal most certainly is. Specifically Cinque’s hierarchy. (Cinque 2006) argues that “the functional portion of the clause, in all languages, is constituted by the same,

¹Cf. (May 1985) and the references therein for a detailed explanation of the set theoretic operations that correspond to the various determiners.

²Cf. (Cinque 1999).
richly articulated and rigidly ordered, hierarchy of functional projections.” What does this mean? For our purposes, it means that in between the CP and the VP, there is a rigid hierarchy, a portion of which is displayed here:

(5.4) Mood\textsubscript{speech act} > Mood\textsubscript{evaluative} > Mood\textsubscript{evidential} > MP\textsubscript{epistemic} > TP(Past) >

TP(Future) > Mood\textsubscript{irrealis} > MP\textsubscript{alethic} > Asp\textsubscript{habitual} > ... > MP\textsubscript{volitional} >

Asp\textsubscript{celerative(I)} > TP(Anterior) > Asp\textsubscript{terminative} > ... > MP\textsubscript{obligation} >

MP\textsubscript{permission/ability} > ...

According to Cinque, all functional verbs are inserted directly into the head of their respective functional projection, forming a monoclausal structure, rather than analyzing functional verbs according to a biclausal structure in which the functional verb takes a CP complement:

(5.5) Biclausal: $[CP \ldots |FP \ldots |FP \ldots |VPV\textsubscript{restr}|CP \ldots |FP \ldots |VPV \ldots |VPV$]

(5.6) Monoclausal: $[CP \ldots |FP \ldots |VPV\textsubscript{restr}|FP \ldots |VPV \ldots |VPV$]

Cinque’s work, based on careful empirical cross-linguistic study, suggests that modal auxiliary verbs such as must and might are inserted directly into the functional hierarchy above as the head of MP\textsubscript{epistemic}. This differs from the picture to be sketched below, which suggests that all verbs and their arguments could first be generated in VP before moving to their respective places in the hierarchy on the basis of their feature projections. I compare the semantics of these two views.
Semantics of the Light Verb

Lets start with the verb phrase, the terminating node of the left periphery. This phrase is responsible for both the main verb, its head, and the verbal arguments. These arguments take the form of the subject, the direct, and indirect object. It has been claimed by (Chomsky 1995) and (Kratzer 1996) that both the subject and the main verb move and merge into a light verb, which acts something like an unpronounced auxiliary verb, specifying the method of combination of these arguments:

In what follows I claim that for semantic reasons the copied phrase leaves behind not a semantic trace or copy, but instead a higher-order composition map, which tracks the
combinations that movement invokes. For our first example, consider the following sentence:

(5.7) John sent a letter to Mary.

\[
\begin{array}{c}
\text{vP} \\
\quad \\
\text{John} \quad v' \\
\quad \\
\text{v+send} \quad \text{VP} \\
\quad \\
\text{NP} \quad V' \\
\quad \\
\text{a letter} \quad V \quad \text{PP} \\
\quad \\
\langle \text{send} \rangle \quad \text{to mary}
\end{array}
\]
\[
\lambda x \cdot \lambda y \cdot \lambda z \cdot [send(x)(y)(z)]
\]

\[
\lambda y \cdot \lambda z \cdot [Q(mary)(y)(z)] 
\]

\[
\lambda z \cdot [send(mary)(letters)(z)]
\]

\[
\lambda z \cdot [send(mary)(letters)(John)]
\]

\[
\lambda x \cdot \lambda y \cdot \lambda Q \langle e,(e,e,t) \rangle \cdot \lambda z \cdot [Q(x)(y)(z)]
\]

\[
\lambda x \cdot \lambda y \cdot \lambda Q \langle e,(e,e,t) \rangle \cdot \lambda z \cdot [Q(x)(y)(z)]
\]

\[
\lambda Q \langle e,(e,e,t) \rangle \cdot \lambda z \cdot [Q(mary)(letter)(z)]
\]

\[
\lambda Q \langle e,(e,e,t) \rangle \cdot \lambda z \cdot [Q(mary)(letter)(z)]
\]

\[
\lambda Q \langle e,(e,e,t) \rangle \cdot \lambda z \cdot [Q(mary)(letter)(z)]
\]

\[
\lambda Q \langle e,(e,e,t) \rangle \cdot \lambda z \cdot [Q(mary)(letter)(z)]
\]

\[
\lambda Q \langle e,(e,e,t) \rangle \cdot \lambda z \cdot [Q(mary)(letter)(z)]
\]

\[
\lambda x \cdot \lambda y \cdot \lambda Q \langle e,(e,e,t) \rangle \cdot \lambda z \cdot [Q(x)(y)(z)]
\]

\[
\lambda x \cdot \lambda y \cdot \lambda Q \langle e,(e,e,t) \rangle \cdot \lambda z \cdot [Q(x)(y)(z)]
\]

\[
\lambda x \cdot \lambda y \cdot \lambda Q \langle e,(e,e,t) \rangle \cdot \lambda z \cdot [Q(x)(y)(z)]
\]

\[
\lambda x \cdot \lambda y \cdot \lambda Q \langle e,(e,e,t) \rangle \cdot \lambda z \cdot [Q(x)(y)(z)]
\]

\[
\lambda x \cdot \lambda y \cdot \lambda Q \langle e,(e,e,t) \rangle \cdot \lambda z \cdot [Q(x)(y)(z)]
\]

\[
\lambda x \cdot \lambda y \cdot \lambda Q \langle e,(e,e,t) \rangle \cdot \lambda z \cdot [Q(x)(y)(z)]
\]

\[
\lambda x \cdot \lambda y \cdot \lambda Q \langle e,(e,e,t) \rangle \cdot \lambda z \cdot [Q(x)(y)(z)]
\]

\[
\lambda x \cdot \lambda y \cdot \lambda Q \langle e,(e,e,t) \rangle \cdot \lambda z \cdot [Q(x)(y)(z)]
\]

\[
\lambda x \cdot \lambda y \cdot \lambda Q \langle e,(e,e,t) \rangle \cdot \lambda z \cdot [Q(x)(y)(z)]
\]

\[
\lambda x \cdot \lambda y \cdot \lambda Q \langle e,(e,e,t) \rangle \cdot \lambda z \cdot [Q(x)(y)(z)]
\]

\[
\lambda x \cdot \lambda y \cdot \lambda Q \langle e,(e,e,t) \rangle \cdot \lambda z \cdot [Q(x)(y)(z)]
\]

\[
\lambda x \cdot \lambda y \cdot \lambda Q \langle e,(e,e,t) \rangle \cdot \lambda z \cdot [Q(x)(y)(z)]
\]

\[
\lambda x \cdot \lambda y \cdot \lambda Q \langle e,(e,e,t) \rangle \cdot \lambda z \cdot [Q(x)(y)(z)]
\]

\[
\lambda x \cdot \lambda y \cdot \lambda Q \langle e,(e,e,t) \rangle \cdot \lambda z \cdot [Q(x)(y)(z)]
\]

\[
\lambda x \cdot \lambda y \cdot \lambda Q \langle e,(e,e,t) \rangle \cdot \lambda z \cdot [Q(x)(y)(z)]
\]

\[
\lambda x \cdot \lambda y \cdot \lambda Q \langle e,(e,e,t) \rangle \cdot \lambda z \cdot [Q(x)(y)(z)]
\]

\[
\lambda x \cdot \lambda y \cdot \lambda Q \langle e,(e,e,t) \rangle \cdot \lambda z \cdot [Q(x)(y)(z)]
\]

\[
\lambda x \cdot \lambda y \cdot \lambda Q \langle e,(e,e,t) \rangle \cdot \lambda z \cdot [Q(x)(y)(z)]
\]

\[
\lambda x \cdot \lambda y \cdot \lambda Q \langle e,(e,e,t) \rangle \cdot \lambda z \cdot [Q(x)(y)(z)]
\]

\[
\lambda x \cdot \lambda y \cdot \lambda Q \langle e,(e,e,t) \rangle \cdot \lambda z \cdot [Q(x)(y)(z)]
\]

\[
\lambda x \cdot \lambda y \cdot \lambda Q \langle e,(e,e,t) \rangle \cdot \lambda z \cdot [Q(x)(y)(z)]
\]

\[
\lambda x \cdot \lambda y \cdot \lambda Q \langle e,(e,e,t) \rangle \cdot \lambda z \cdot [Q(x)(y)(z)]
\]

\[
\lambda x \cdot \lambda y \cdot \lambda Q \langle e,(e,e,t) \rangle \cdot \lambda z \cdot [Q(x)(y)(z)]
\]

\[
\lambda x \cdot \lambda y \cdot \lambda Q \langle e,(e,e,t) \rangle \cdot \lambda z \cdot [Q(x)(y)(z)]
\]

\[
\lambda x \cdot \lambda y \cdot \lambda Q \langle e,(e,e,t) \rangle \cdot \lambda z \cdot [Q(x)(y)(z)]
\]

\[
\lambda x \cdot \lambda y \cdot \lambda Q \langle e,(e,e,t) \rangle \cdot \lambda z \cdot [Q(x)(y)(z)]
\]

\[
\lambda x \cdot \lambda y \cdot \lambda Q \langle e,(e,e,t) \rangle \cdot \lambda z \cdot [Q(x)(y)(z)]
\]

\[
\lambda x \cdot \lambda y \cdot \lambda Q \langle e,(e,e,t) \rangle \cdot \lambda z \cdot [Q(x)(y)(z)]
\]

\[
\lambda x \cdot \lambda y \cdot \lambda Q \langle e,(e,e,t) \rangle \cdot \lambda z \cdot [Q(x)(y)(z)]
\]

\[
\lambda x \cdot \lambda y \cdot \lambda Q \langle e,(e,e,t) \rangle \cdot \lambda z \cdot [Q(x)(y)(z)]
\]

\[
\lambda x \cdot \lambda y \cdot \lambda Q \langle e,(e,e,t) \rangle \cdot \lambda z \cdot [Q(x)(y)(z)]
\]

\[
\lambda x \cdot \lambda y \cdot \lambda Q \langle e,(e,e,t) \rangle \cdot \lambda z \cdot [Q(x)(y)(z)]
\]

\[
\lambda x \cdot \lambda y \cdot \lambda Q \langle e,(e,e,t) \rangle \cdot \lambda z \cdot [Q(x)(y)(z)]
\]

\[
\lambda x \cdot \lambda y \cdot \lambda Q \langle e,(e,e,t) \rangle \cdot \lambda z \cdot [Q(x)(y)(z)]
\]

\[
\lambda x \cdot \lambda y \cdot \lambda Q \langle e,(e,e,t) \rangle \cdot \lambda z \cdot [Q(x)(y)(z)]
\]

\[
\lambda x \cdot \lambda y \cdot \lambda Q \langle e,(e,e,t) \rangle \cdot \lambda z \cdot [Q(x)(y)(z)]
\]

\[
\lambda x \cdot \lambda y \cdot \lambda Q \langle e,(e,e,t) \rangle \cdot \lambda z \cdot [Q(x)(y)(z)]
\]

\[
\lambda x \cdot \lambda y \cdot \lambda Q \langle e,(e,e,t) \rangle \cdot \lambda z \cdot [Q(x)(y)(z)]
\]

\[
\lambda x \cdot \lambda y \cdot \lambda Q \langle e,(e,e,t) \rangle \cdot \lambda z \cdot [Q(x)(y)(z)]
\]

\[
\lambda x \cdot \lambda y \cdot \lambda Q \langle e,(e,e,t) \rangle \cdot \lambda z \cdot [Q(x)(y)(z)]
\]

\[
\lambda x \cdot \lambda y \cdot \lambda Q \langle e,(e,e,t) \rangle \cdot \lambda z \cdot [Q(x)(y)(z)]
\]

\[
\lambda x \cdot \lambda y \cdot \lambda Q \langle e,(e,e,t) \rangle \cdot \lambda z \cdot [Q(x)(y)(z)]
\]

\[
\lambda x \cdot \lambda y \cdot \lambda Q \langle e,(e,e,t) \rangle \cdot \lambda z \cdot [Q(x)(y)(z)]
\]

\[
\lambda x \cdot \lambda y \cdot \lambda Q \langle e,(e,e,t) \rangle \cdot \lambda z \cdot [Q(x)(y)(z)]
\]
vP

John_e[N] vP

t^e[T] vP

w^e[W] v'

v

v send_{e(e(sie,t))} VP

DP VP

⟨John[N]⟩ e PP VP

α ⟨t^e[T]⟩_i PP VP

α ⟨w^e[W]⟩_i DP V' a letter V DP

⟨send⟩_{e(e(sie,t))} to Mary
Note that the \( \Lambda \) representative function takes two different forms. First, for the moved verb, we have \( \lambda x.e.\lambda y.e.\lambda Q(\langle \text{eesie},t \rangle,\langle \text{iesie},t \rangle)\lambda z.e.\lambda z'.\lambda z''. [Q(x)(y)(z)(z')(z'')] \). This function raises the verb and effectively delays its composition. The second function, which represents the external semantic arguments \( \langle \text{John},w,t \rangle \) that form the semantic center of the sentence, is \( \lambda R(\langle \text{eesie},t \rangle)\lambda z.e.\lambda z'.\lambda z''. [R(\langle \text{eesie},t \rangle)(z')(z'')] \). This function works for all three basic types.

Suppose we start with merge. But before we merge the verb, we merge its full set of arguments. These arguments are met with a cartesian product constructor defined as follows:

The product type is a pair of types \( \tau_1 \times \tau_2 \). There exists functions \( \pi_n \) for \( n \in \mathbb{N} \) that take a product sequence and return its \( n \)-th member. The product types we are interested in is just that finite sequence of arbitrary length that defines the context under discussion. This
sequence includes an agent, the subject, a world, and a time, at the least. Various standards of precision could be included, with empirical testing determining the final outcome of the sequence.

One option opened up by this enriched system is the merging of argument stacks. I assume that types are assigned to terms during the merge process. So merged terms come with types variables assigned. Thus the operation \textit{merge} is the operation $\Gamma \cup \{x\}$ for a merged term $x$. Take the example sentence

(5.8) John sent a letter to Mary.

When constructing this sentence, I assume first \textit{mary, letter,} and \textit{send} combine. Assuming \textit{mary} and \textit{letter} are basic terms, send is an $n$-ary function. The arguments of ‘send’ have not yet been specified. We have a number of different options. First, the surface structure VP+DP+DP could be respected. In this case VP would take DP\times DP as a single argument. The DPs would combine with the $\times i$ constructor:
Figure 5.3: Entity Products

Since the function $\langle e \times e, t \rangle$ is identical to the function $\langle e(e, t) \rangle$ I will abbreviate the type of send as $\langle e e, t \rangle$. We can of course continue to move the verb, but this move gives us an alternative formulation of the external arguments:

$(5.9)$ John send a letter to Mary.
Here the DP merged into Comp, vP is complex. It contains additional prepositional phrases in argument position. Prepositions are thus polymorphic, they can play the role of either an argument or a verbal modifier. Additionally, with product types it is possible to posit phrasal movement of a complex DP.

(5.10)

(5.11) John send a letter to Mary.
\[
\frac{\lambda x_e \cdot \lambda y_e \cdot \lambda Q_{(e(eis,t))} \cdot \lambda z_e \cdot \lambda z'_e \cdot \lambda z''_e \cdot [Q(x)(y)(z)(z')(z'')] \quad \text{mary} (\rightarrow e) \quad \text{letter} (\rightarrow) \quad \text{send} (t,w) : i \times s}{\lambda y_e \cdot \lambda Q_{(e(eis,t))} \cdot \lambda z_e \cdot \lambda z'_e \cdot \lambda z''_e \cdot [Q(\text{mary})(y)(z)(z')(z'')] \quad \text{mary} (\rightarrow e) \quad \text{letter} (\rightarrow) \quad \text{send} (t,w) : i \times s}
\]
Chapter 6

Modals and Conditionals

Modality and Conditionals

Before I begin the section on modality and conditionals, I would like to set the stage for my discussion philosophically. At this point, I have been discussing the logical form of natural language sentences and comparing it to a variety of different logical systems. I believe this to be the primary concern of the interface between natural language semantics and philosophical logic. There are other viewpoints, and I have criticized some attempts to bridge the divide. I do, however, think there is room for a type of philosophical viewpoint which I will describe here, as long as it makes no claim to using *a posteriori* facts about natural language as evidence for itself. I use an example from deontic theory, where arguments are made about what people ought to do—a theory of agency. Whether and how the study of deontic logic is
connected to these theories is an open question.

‘Ought’ and ‘should’ are common sense normative words. Their meanings are various, and sometimes, philosophers dissect this variety and select a particular shade to focus on and conceptually analyze. This analysis is often regimented into some logical model. In this sense natural language provides the means by which the logical formula can be reasoned about in a common sense fashion, but the confounding factors and multiple interpretations of natural language make the consequences of our intuitions harder to parse when considering the *a priori*. Further, the claims of the generative semanticists, that there could be a single underlying logic to similarity classes of sentences in natural languages that share lexical building blocks, has been refuted.

This is the main point of the arguments canvassed earlier against generative semantics, the view that:

> [T]he only changes that transformations can make to lexical items is to add inflectional affixes such as number, gender, case, person, and tense. Transformations will thus be restricted to movement rules and insertion and deletion of constants and closed sets of items.\(^1\)

When (Horty 2001) creates an artificial ‘oughting’ noun that is related the the shade of meaning he is studying his “violence” to natural language reverses the order of explanation. Language is used to explain a particular concept, rather than studying language to uncover

\(^1\)(Jackendoff 1972, 13).
the natural concepts and their logical relations. If one wishes to non-circularly define a concept and analyze it, I do not have qualms with this approach. It should be noted, however, that linguistic data only provides weak support for this line of argumentation, as it is generally rife with counterexamples to the concept in question. As noted earlier, (Chomsky 1970b) counters this thesis by considering the relationship between (6.1a)-(6.1c)

(6.1) (a) The enemy destroyed the city.
    (b) The enemy’s destruction of the city
    (c) The enemy’s destroying the city

Chomsky suggests that while the sentence (6.1a) and the gerundive nominal (6.1c) are related transformationally, with the same deep structure, the derived nominal (6.1b) is best analyzed as only morphologically related, with a separate listing in the lexicon. This provides an explanatory benefit for a number of reasons.

First, there are sentences for which gerundive nominals but no derived nominals exist. This suggests a morphological analysis; syntactic transformations should be regular and productive. Second, the meanings of derived nominals do not correspond to their verbal forms in the way that the gerundive nominals do. Finally, while derived nominals share many properties with nouns—the ability to appear with articles, be modified by adjectives, and take plural morphology—, gerundive nominals share none of these properties. These considerations strongly suggest that transformational derivation of (6.1b) from (6.1a) is completely unsatisfactory. According to (Chomsky 1970c), the differences noted between
derived and gerundive nominals would simply be a “remarkable accident” from the point of view of the generative semanticist, and this would be a startling failure of explanatory adequacy. If we are to take natural language seriously, attempting to associate equivalence classes of sentences with some deep semantic representation is not an avenue which will be met with much empirical success.

I start my inquiry with a broadly Kratzerian account of modality. Kratzer’s account, construed loosely, defines a generalized polyadic quantifier $\text{NECESSITY}(\text{ORDER})(\text{BASE}) \subseteq P$. The sets ORDER and BASE are sets of propositions, or subsets of the powerset of worlds. They are generated by functions from the world of evaluation. These functions will be represented here as either $F/f$ or $G/g$. Both of these sets are further modified. The order itself generates a POSET which contains a set of "best" worlds or "ever better" worlds—where the are no changes between the worlds that effect "betterness". The modal base is intersected with itself in order to find a set of worlds—events$^2$—rather than propositions, to be restricted by the ordering source. If this restricted modal base is a subset of the proposition in question, the necessity modal is true. I have argued in (Cook 2013) that this framework needs to be relativized to at least agents. I now argue that if we extend this analysis to both times and worlds, in conformity with the system developed so far, the simplicity and fecundity of the extensions is worth investigating. First, I show that this relativization firmly distinguishes ought-to-be and ought-to-do modals, and, then, I show that by incorporating rules based on

$^2$I will not be considering event-based analyses here as I have previously argued that events may be reduced to the entities I work with
AGM belief revision theory, the Frank-Zvolensky conditional problem may be circumvented.

I close with some thoughts on further extensions of system.

I take the distinction between *ought to be* and *ought to do* modality as a legitimate distinction to be made in the logical form of natural language sentences. The first case is agent irrelevant. The second case directly involves an agent. Moral philosophers seek to connect the beliefs that underly these sentences with agent’s actions, but this is not an extension that will be made here.

It could be that you morally ought to do one thing, rationally ought to do another, and prudentially ought to do a third thing. I consider these different strands to be contained by a sequence of ordering sources $G_{\text{moral}}, G_{\text{prudential}}, G_{\text{rational}}$, and any others you may find in your linguistic investigation. This differs from Kratzer’s account, which only has one ordering source. The type and number of these different strands has a contextually defined precedence relation which determines how these sources override each other. I go into the details of this relation in what follows. Some might think there is an *all-things-considered* ought, and, if so, it might require the combination of these different ordering sources. These ordering sources could model what (Broome 2013, 26) might call a *source of requirements*. Not all of these sources would be considered ‘normative’ by Broome, but they must all be contended with when considering the shades of meaning found in the logic of modals and conditionals.

Another way of looking at this project is just this: I am concerned with the *prima facie*
logic of the language, what direct compositionalists value. If a moral theorist thinks this is not the best logic for their specific axiology, they are free to construct a logical system on their own. I only worry insofar as that system is justified by *a posteriori* facts about natural language.

Once a natural language semantics is fully generalized, it should be connected to the best theory of morality. This connection would happen wherever one found a neutral container of content that bridges language and thought, if such a thing were possible. It is this container that I would label a *proposition*, but I have not made heavy use of this terminology.

**Syntax**

Syntactically, I assume for simplicity that modals are base generated in the left periphery. I also assume, generally following (Brennan 1993), (Cinque 1999), (Cinque 2006), (Butler 2003), and (Nauze 2008) that syntactically, epistemic modals figure in the left periphery higher than deontic modals. Somewhat more controversially, I assume that these two categories are split by tense. This is to say that deontic modals are constrained by subject-to-subject raising. This assumption, in conjunction with the feature set to be developed here, which I attach to *extended projection principle* (EPP) features,\(^3\) will predict the distinction observed above.

This feature system, which will govern the covert entities that are labeled worlds and times by our type system, is an instance of what Pesetsky (2000) calls a complementizer that

---

\(^3\)The EPP account I follow is found in (Adger 2003).
requires multiple specifiers (*multispecifier complementizer*). Bulgarian shows this feature of the underlying universal grammar in multiple *wh* questions. Given the universality of the system, it stands to reason that the semantics could pick up this structure and co-opt it’s use. This is the direction we are headed. The number of specifiers can be reduced by product formation in the semantics, but they cannot be completely eliminated.

The modal term itself can be considered a polyadic quantifier of three arguments, taking two conversational backgrounds as restrictions and the complement clause as scope. The claim made is that the covert restrictions, the modal base $f$ and ordering source $g$, probe for their arguments. The features that determine this probe generate syntactic restrictions similar to those that govern personal pronouns and reflexives.

I develop this distinction in the following way: first, I posit the feature ($\pm \ast$). In its negative form ($-\ast$) probes for an argument to fill its position. The external arguments of the lexical verb, on their way to their final landing point in the inflectional layer, through successive-cyclic movement, leave traces behind where probed. In it’s positive form ($+\ast$) prevents this kind of movement, as it calls for a ($+\ast$) pronominal to be base generated in its argument position. Pronominals with ($+\ast$) diacritics generate a context-dependent reading, as they take their values from the contextually generated assignment function when not bound from outside of their local domain. I abbreviate ‘$+\ast$’ as ‘$\ast$’ and ‘$-\ast$’ as ‘$-$’, with [$-+\ast$], [$--\ast$], etc., superscripted to conversational backgrounds to provide a guide to their arguments. I am making several simplifying assumptions here regarding the verbal
shell vP (cf. (Larson 1988), (Chomsky 1995, 329-334)).

First, while independent heads for external arguments are assumed, they are not overtly represented here. I assume, contra (Kratzer 1996), that voice information is not semantically encoded in a davidsonian manner (Davidson 1967). Lexical verbs take their external arguments directly, as intensional information is already encoded within this system. That is to say, the property \( \lambda w_s. \lambda t_i. \lambda x_e. [shave(x)(t)(w)] \) already denotes an event of shaving—the set of \( \langle a, t, w \rangle \) s.t. a shaves at t in w. If it becomes necessary to distinguish situations that are more finely grained than sets of agent, time, world triples, event structure can be simulated by finite sequences including additional arguments required to satisfy the conditions of the event. I assume the verb selects for these arguments, but I make the simplifying assumption here that we are dealing only with three arguments.

Second, a relative tense must be introduced in vP to account for the fact that the time of shaving is constrained to some contextually determined future interval. Following (Reichenbach 1947), (Ogihara 1996), I treat tense quantificationally, and, following (Partee 1973), (Cresswell 1979), the lexical content of this tense could be found as the head of voiceP: \( \lambda t_i. \lambda P_{(i,et)}. \lambda x_e. [\exists t' \in I^*(t < t' \wedge P(t')(x) = 1)] \). Finally, this feature system must be complicated to accommodate PRO and infinitival environments generally. I leave detailed treatment of event structure and infinitival environments for future work, pointing only toward a development I would consider an extension of my own framework where necessary.
The two initial feature sets we see are \([- + +]\) and \([- - -]\), corresponding to the modal base and ordering source. This set explains the immunity of the modal base to binding within its local domain, and, conversely, the order being immune to capture from above (outside of the local domain).

The covert contextual material must undergo subject-to-subject raising (raising for short). The modal base, on the other hand, does not probe for these arguments. As per the \((+\ast)\) diacritic, pronominal elements are generated that take values from context already saturating their argument positions. Thus, the only probe the modal base requires is that of a world. This feature pattern predicts the local/non-local distinction witnessed in (??)/(??) above. Deontics can be bound from within their local domain via abstraction over the movement chains discussed, while epistemics cannot be so bound because the modal base does not probe for these arguments. Epistemics can be bound from above because certain classes of intensional verbs have the ability to delete \((+\ast)\) features, as has been developed by (von Stechow 2003).

In the case of world pronouns, there is a choice between both base and order generating their own contextually dependent world variable \((+\ast)\) or having the world pronoun generated by the lexical verb move through both arguments \((-\ast)\), fixing the reference. The first option would prevent modals from being embedded in a way contrary to the data. For example:

(6.2) John thinks that it might be raining.

Allowing worlds to be generated as \((+\ast)\) would allow independent variation of the world that
generates the modal base for 'might' under the scope of 'thinks'. What is wanted is the modal base of 'might' to be generated pointwise at each world compatible with John's thoughts. This is predicted on the (−∗) account, but must be stipulated on the (+∗) account. This further supports the package advocated here. Since mood is found above epistemic modals in the hierarchy above, the account can be sharpened: the modal base/ordering source probe for only those arguments that must move past their maximal projection during the derivation cycle. Any additional arguments must be base generated (+∗).

These diacritics are semantic. They are interpreted by semantic functions \( s^n \) that map indices to shifts in the value of the variable, as you can see in the following diagram:
(6.3) John should shave.

The movement of the DP is directly connected to the extended projection principle (EPP), it simply carries semantic material along for the ride. In the following section I first introduce a basic Kratzerian structure. I modify this structure with my own theory of context, which is at this point connected to my theory of compositionality. I provide a basic example derivation within this framework. I then generalize the account to conditionals. Here I add
broadly AGM style belief revision rules on top of the Kratzerian account to overcome the Frank-Zvolensky data. It is at this point that some interesting properties of my theory of compositionality surface. I compare the method of movement I adopt with the distinction between overt and covert operators approaches developed by (Kaufmann and Kaufmann forthcoming), and then extend my framework along the lines of (Frank 1996), contra Kratzer. I provide a framework of premise structures, initially developed by (Kaufmann 2013) and (Graham Katz and Rubinstein 2012), and then construct a method of generating a single set of worlds from this structural framework. This reduces the structure to the more familiar Kratzerian system in most standard cases. Following this, I extend the general framework to epistemic modals, showing that it is capable of unifying the two accounts. In the appendix I compare the framework developed here to one using the same system, only with the (VIFA) composition rule en force in lieu of the combinatorial approach.

**Semantics**

Let modals be defined as usual, factoring out a modal base and ordering source as functional phrases, we have the addition of a shifting function defined over \( \mathbb{N} \) which takes an argument and shifts it to the \( n \)th place for \( n \in \mathbb{N} \) of a contextually privileged sequence. Thus, this contextual selection function effectively shifts the referent of a pronoun to a contextually defined referent. This function is called by either the modal base or ordering source, when merged, according to it’s EPP feature content. The external argument product moves through the
modal base and ordering source on its way to be evaluated by tense.

**Restriction** Let \([f^i : \langle eis, \langle \langle eis, t \rangle, t \rangle \rangle = c(i) : \langle eis, \langle \langle eis, t \rangle, t \rangle \rangle]\).

Note at this point the restriction takes a variable \(n \in \mathbb{N}\), but the superscripts above are coded based on the required \(s^n\) functions defined below. This is a simple definitional issue, as the combinations of feature values for \(f\) and \(g\), once empirically verified, can be provided a simple numerical coding.

**Preorder** Let \(\leq\) define a preorder for a function \(A : \langle \langle eis, t \rangle, t \rangle \rangle\) s.t.

\[\forall uv : e \times i \times s \mid u \leq A v \iff \forall p : \langle eis, t \rangle \rangle \text{ s.t. } A(p) = 1 \rightarrow p(u) = 1\]

**Upper Bounds** Let \(S^U\) the upper bound function be defined

\[\lambda A : \langle \langle eis, t \rangle, t \rangle \rangle \lambda X : \langle eis, t \rangle \rangle [S^U(A)(X) = \lambda y : \langle eis, t \rangle \rangle \{(\forall x : \langle eis \rangle \rangle ((X(y) = 1 \land (\cap A)(x) = 1) \rightarrow (y \leq A x \rightarrow x \leq A y)\)]]\]

We may preliminarily define *should* as

\[\lambda f_{\langle eis, eis, t \rangle} \lambda g_{\langle eis, eis, t \rangle} \lambda P_{\langle eis, t \rangle} \lambda x : \langle eis, t \rangle \rangle [S^U(g(x, y, z))((\cap f(x, y, z)) \subseteq \lambda x' : \langle eis, t \rangle \rangle [P(x) = 1]]\]

We have a set of function shifting functions \(s^n\) for \(n \in \mathbb{N}\) that mirrors the deconstruction rule \((\times ne)\). The rule for these functions, here assigned type \(\langle \langle eis(eis, t) \rangle, eis(eis, t) \rangle \rangle\) can be defined by the following sequence

\[\lambda f \lambda x y z \ldots \lambda p. [s^1(f(x)(y)(z) \ldots (p)) = f(K(c(1))(x))(y)(z) \ldots (p)]\]
\[
\lambda f. \lambda xyz \ldots \lambda p. [s^2(f(x)(y)(z) \ldots (p)) = f(x)(\mathbf{K}(c(2))(y))(z) \ldots (p)] \\
\lambda f. \lambda xyz \ldots \lambda p. [s^3(f(x)(y)(z) \ldots (p)) = f(x)(y)(\mathbf{K}(c(3))(z)) \ldots (p)]
\]

for \( s^n \) where \( n \in \mathbb{N} \), the function \( c : \mathbb{N} \rightarrow \ldots \) takes the naturals to the privileged sequence defined by the context of conversation. I allow this sequence to be of arbitrary length. However, our current set of parameters is only \( \langle a, t, w \rangle \). \( \mathbf{K} \) is the combinator that returns a constant function. Thus, for each position \( n \) in the contextually defined sequence, we have a function \( s^n \) that takes a variable and returns the constant function returning the \( n^{th} \) position in the sequence. Note that up to this point, we have been working in the \( \lambda I \) subsystem, where the \( \mathbf{I} \) combinator is representable but the \( \mathbf{K} \) combinator is not. To represent the semantic content generated by composition in the inflectional phrase, the \( \mathbf{K} \) combinator is required, which increases the power of our system to \( \lambda K \).

During the movement phase the DP representing the external arguments \( \langle John, t, w \rangle \) requires the type of it’s sister when the copy-move cycle continues, since the semantic residue left behind is of the form \( \lambda x : \alpha. \lambda y : \beta. xy \) where \( \alpha \) is the type of \( \Lambda \)’s sister and \( \beta \) is the type of the copied object itself. See the following derivation:

(6.4) John should shave.
\[
\begin{array}{c}
\lambda F_{(eis,t)} \cdot \lambda x_e \cdot \lambda y_i \cdot \lambda z_s . [F(x, y, z)] \\
\lambda x_e \cdot \lambda y_i \cdot \lambda z_s . \lambda p_e \cdot [g(x, z, y)(p)]
\end{array}
\]

\[
\begin{array}{c}
\lambda g_{(eis,t)} \cdot \lambda P_{(eis,t)} \cdot \lambda x_e \cdot \lambda y_i \cdot \lambda z_s . [S^U (g(x, y, z))(\cap \lambda p_{(eis,t)} . f(K(c(1))(x))(K(c(2))(y))(z)(p)) \subseteq P] \\
\lambda x_e \cdot \lambda y_i \cdot \lambda z_s . \lambda p_e \cdot [g(x, y, z)(p)]
\end{array}
\]

\[
\begin{array}{c}
\lambda P_{(eis,t)} \cdot \lambda x_e \cdot \lambda y_i \cdot \lambda z_s . [P(x, y, z)] \\
\lambda x_e \cdot \lambda y_i \cdot \lambda z_s . [shave(x, y, z)]
\end{array}
\]

\[
\begin{array}{c}
\lambda x_e \cdot \lambda y_i \cdot \lambda z_s . [shave(x, y, z)] \\
\lambda x_e \cdot \lambda y_i \cdot \lambda z_s . [S^U (\lambda p_{(eis,t)} . [g(x, y, z)(p)])] \\
(\cap \lambda p_{(eis,t)} . f(K(c(1))(x))(K(c(2))(y))(z)(p)) \subseteq P]
\end{array}
\]

Here right node labels \((\rightarrow e)\) have been suppressed, and our final result is

\[
\lambda x_e \cdot \lambda y_i \cdot \lambda z_s . [S^U (\lambda p_{(eis,t)} . [g(x, y, z)(p)])] \cap \lambda p_{(eis,t)} . f(K(c(1))(x))(K(c(2))(y))(z)(p)) \subseteq
\lambda x'_e \cdot \lambda y'_i \cdot \lambda z'_s . [shave(x', y', z')]\]

which feeds seemlessly into the temporal phrase TP. Note that the combinator

\[
K(x)(y) := \lambda xy.x.
\]
Conditionals

One problem which immediately arises when considering this definition of modality is that it revolves around removing the inconsistencies from the ordering source. In common practice, context can not be guaranteed to generate a consistent set of propositions that determines a privileged set of possible worlds. When debating, what is considered best among the conversational participants is almost always inconsistent. Any political debate provides evidence for this claim.

Notation There are several type assignment patterns that can be reduced through the use of variables. Where $\nu$ is type $\langle eis \rangle$. This $\nu$ sequence represents the possible initial sequences of the contextual function $c(n)$. These would be the sequences that the main verb depends on as it’s subject. The propositional function $p$ is then of type $\langle \nu,t \rangle$. And our Kratzerian functions $f$ are of type $\langle p,t \rangle$. Looking ahead, ‘must’ is then a function of type $\langle fp,t \rangle$, ‘should’ is a function of type $\langle ffp,t \rangle$, and ‘if’ is a function of type $\langle pf(f,t),t \rangle$. For an arbitrary type, ususally types $\nu$, $p$, and $f$, I consider the set denoted by $\lambda x : \tau. M = 1$ to be the same as $M$, where $M$ is of type $\langle \tau,t \rangle$.

Addition $\lambda F_f, \lambda A_p, [+(F, A)] = F' : f$ s.t. for all $w : s,t : i,x : e,p : p, F'(w)(t)(x)(p) = F(w)(t)(x)(p)$ except $F'(w)(t)(x)(A) = 1$.

4I treat should as a generic modal with deontic flavour. ‘Ought’ would do just as well, but I find ‘should’ to be more relaxing. There may be empirical differences between these auxiliary verbs, but nothing that follows hinges on these differences.
Subtraction For $\Gamma, \Delta : f, -(\Gamma, \Delta)$ is a function of type $f :=$ the contextually determined maximal set $\gamma : f \subseteq \Gamma : f$ s.t. there is no $\varphi : f \subseteq \gamma : f$ s.t. $\exists \delta : p \in \Delta : f(\cap \varphi \subseteq \delta)$.

Strong Necessity $\lambda F_{(p,t)} \cdot \lambda P_p.[\cap F \subseteq P]$

Weak Necessity $\lambda F_{(p,t)} \cdot \lambda G_{(p,t)} \cdot \lambda P_p.[S^\mu(G)(\cap -(F, \{P^C, P}\})) \subseteq P]$

The analysis that follows is in the spirit of (Kratzer 2012), with some upgrades. First, I implement a restrictor analysis of ‘if’ within the current framework: conditionals impose a restriction on overt modal expressions in the consequent clause. This is a version of what (Kaufmann and Schwager 2006) call the Overt Conditional Operator (OCO) approach. In what follows, I first develop the OCO approach and then show how to extend the framework to its Covert Conditional Operator (CCO) cousin. Along the way, I show how to overcome one major hurdle of the OCO approach: the problematic theorems of (Frank 1996, 30), (Zvolenszky 2002), and (Zvolenszky 2006).

The characterization used here assumes implicitly that context will select the best of the maximal consistent subsets, if ties arise. How the best subset is determined by context will be covered later. I assume here that the difficulties involved will not translate into any intractable compositional difficulties. For a different implementation of this idea within a dynamic framework see the downdating operation of (Willer 2014, 14). One difference between these operations is that Willer’s requires an additional channel of contextual information to generate an objective ranking over worlds.
If $\varphi \models \psi$, sentences of the following schematic form are true trivially, yet in certain cases informative:

\[ \text{⌜if } \varphi, \square \psi \⌝ \]

This is a special problem for weak modals—those that require an ordering source. The additional restriction on the domain should be informative. Yet, in these cases, the naïve analysis effectively ignores it. At this point I have defined a weak modal that is insensitive to the truth value of its prejacent and a conditional which first nullifies the negation of it’s antecedent before addition. Consider the following examples

(6.5) If John speeds, then he should speed.

(6.6) If The Dalai Lama is mad, then he should be mad.

The analysis provided predicts that both (6.5) can be false and (6.6) can be true. The solution to this problem, I claim, lies not just in the analysis of the conditional restriction, but factors into the analysis of weak necessity modals generally. One strategy for eliminating this type of problem since at least (Frank 1996, 184-186) has been a neutralization of the modal base with respect to the scope of the modal. This kind of neutralization will be implemented here as part of the meaning of weak necessity modals, in the form of a subtraction operation (Df 6.2.2). Consider the following derivation:

**COORD1** If $\alpha$ is a branching node, $\{\beta, \gamma\}$ is the set of $\alpha$’s daughters, and $\beta : \langle x, x \rangle$, $\gamma : \langle xy, t \rangle$, then $[\alpha] = \lambda x. \lambda y.[\gamma(\beta(x))(y)]$
If $\alpha$ is a branching node, $\{\beta, \gamma\}$ is the set of $\alpha$’s daughters, and $\beta : \langle \langle x, t \rangle x, t \rangle, \gamma : \langle \langle x, t \rangle x, t \rangle$, then $[\alpha] = \lambda x : \langle x, t \rangle.[\beta(x) = \gamma(x) = 1]$.

$\text{DELAY } \alpha : \langle xy, t \rangle, \text{ then } \lambda x : x.\lambda y : y.[\alpha(x)(y)] = \lambda y : y.\lambda x : x.[\alpha(x)(y)]$. Thus, $\alpha : \langle xy, t \rangle = \alpha' : \langle yx, t \rangle$.

(6.7) If John$_1$ shaves, John$_1$ should$_2$ shave.
\[
\lambda F_{\text(eis}(t)} \cdot x \cdot \lambda y \cdot \lambda z \cdot \lambda p_{\text(eis}(t)} \cdot \\
\quad \frac{F(x, y, z)(p)}{g(x, y, z)(p)} \\
\lambda x \cdot \lambda y \cdot \lambda z \cdot \lambda p_{\text(eis}(t)} \cdot \\
\quad \frac{[g(x, y, z)(p)]}{[g(x, y, z)(p)]}
\]

\[
\lambda x \cdot \lambda y \cdot \lambda z \cdot \lambda p_{\text(eis}(t)} \\
\quad \frac{[\text{shave}(x, y, z)]}{[\text{shave}(x, y, z)]}
\]

\[
\lambda P_{\text(eis}(t)} \cdot \lambda x \cdot \lambda y \cdot \lambda z \\
\quad \frac{[P(x, y, z)]}{[\text{shave}(x, y, z)]} \\
\lambda x \cdot \lambda y \cdot \lambda z \\
\quad \frac{[\text{shave}(x, y, z)]}{[\text{shave}(x, y, z)]}
\]

\[
\frac{w : s \cdot t : i}{(w, t) : s \times i \times i} \quad \frac{\text{john} : e}{(w, t, \text{john}) : s \times i \times e}
\times i
\]

\[
\lambda x \cdot \lambda y \cdot \lambda z \cdot \lambda f_{\text(eis}(t)} \cdot [S^+(g(x, y, z))((\lambda p_{\text(eis}(t)}: [f(\text{K}(c(1)))(y)](z)(p) = f(x)(\text{K}(c(1)))(y)](z)(p) = f(x)(y)(z)(p) = 1) \subseteq P] \\
\lambda x \cdot \lambda y \cdot \lambda z \cdot \lambda f_{\text(eis}(t)} \\
\quad \frac{[f(\text{K}(c(1)))(y)](z)(p) = f(x)(\text{K}(c(1)))(y)](z)(p) = f(x)(y)(z)(p) = 1) \subseteq P}
\]

\[
\lambda x \cdot \lambda y \cdot \lambda z \cdot \lambda f_{\text(eis}(t)} \\
\quad \frac{[S^+(g(x, y, z))((\lambda p_{\text(eis}(t)}: [f(\text{K}(c(1)))(y)](z)(p) = f(x)(\text{K}(c(1)))(y)](z)(p) = f(x)(y)(z)(p) = 1) \subseteq P] \\
\lambda x \cdot \lambda y \cdot \lambda z \cdot \lambda f_{\text(eis}(t)} \\
\quad \frac{[f(\text{K}(c(1)))(y)](z)(p) = f(x)(\text{K}(c(1)))(y)](z)(p) = f(x)(y)(z)(p) = 1) \subseteq P}
\]

\[
\frac{\lambda x \cdot \lambda y \cdot \lambda z \cdot \lambda f_{\text(eis}(t)} \cdot [S^+(g(x, y, z))((\lambda p_{\text(eis}(t)}: [f(\text{K}(c(1)))(y)](z)(p) = f(x)(\text{K}(c(1)))(y)](z)(p) = f(x)(y)(z)(p) = 1) \subseteq P] \\
\lambda x \cdot \lambda y \cdot \lambda z \cdot \lambda f_{\text(eis}(t)} \\
\quad \frac{[f(\text{K}(c(1)))(y)](z)(p) = f(x)(\text{K}(c(1)))(y)](z)(p) = f(x)(y)(z)(p) = 1) \subseteq P}
\]
Notice how the addition of the restriction + to the modal base applies prior to the subtraction −, which is inert since in this case $F = \neg(F, A, A)$. More generally,

$$\neg(+(-(-F, \neg A), A), \{A, \neg A\})$$

is only identical to $F$ if $F = \neg(F, \{\neg A\})$. This is similar to Ramsey Revision, found in (Levi 1996). Ramsey revision removes both $A, \neg A$ from the information state before adding $A$. In this case, we have an order reversal: first the levi identity is used to expand the information state, and then both $A, \neg A$ are contracted. Since in general $+(+(-F, A), \{A\}), A \neq +F, A$, the Recovery Postulate does not hold on this account:

**Counterexample** $+(+(-F, A), \{A\}, A) \neq +F, A$.

Suppose for an arbitrary $w \in W, t \in T, a \in A$:
(a) \( F(w)(t)(a) = \{ A \land B, C \} \),

(b) \( +(F, A)(w)(t)(a) = \{ A, A \land B, C \} \),

(c) \( -(F, \{ A \})(w)(t)(a) = \{ C \} \),

(d) \( \{ A, A \land B, C \} \neq \{ A, C \} \)

Hence, AGM revision and Ramsey revision come apart.

In conditional syntax I follow (Bhatt and Pancheva 2006) and (Haegeman 2010) in assuming that, for syntactic reasons, conditional clauses are base generated in Spec, Mood\(P_{\text{irrealis}}\) and then undergo I-to-C movement to Spec, CP.\(^6\) However, I differ in the semantic analysis assigned to the conditional phrase. Specifically, I assume that conditional phrases come with a \(-f\) feature that probes for a c-commanded modal base. In the analysis of deontic conditionals, this would require head movement of the modal base in the embedded deontic. The modal base head \( f\) would then move to Spec, Cond\(P\).

In this case, remember that the function \( f\) is interpreted as a function \( F \in D_{\langle s i e \langle (s i e, t) \rangle \rangle} \), so \( \text{if} \) then takes as an argument this function \( F \) and a restriction of type \( \langle s i e, t \rangle \), returning a property of functions \( F \): \( \langle f(p(\langle f, t \rangle t)) \rangle \).

This interpretation evaluates the consequent clause relative to a modal base \( F \) restricted by the antecedent clause \( A\). The modal base \( f\) that has moved from the consequent clause must then taken as an argument, which provides the basis of the further restriction by the antecedent environment.

\(^6\)Cf. (Bhatt and Pancheva 2006) and (Haegeman 2010) for the relevant data.
Figure 6.1: Movement of CondP
But not just any restriction will do, for conditionals with counter-to-fact information in the antecedent will undoubtedly introduce contradictory information into an epistemic modal base. What must be defined is a consistent extension of the modal base by $A$, which is accomplished through the *Levi Identity*.\(^7\)

The solution applied here suggests a particular division of labor between the modal base and the ordering source. Specifically, deontic conditionals of the form $⌜$if $\varphi$, $\square \psi⌝$ where $\varphi \models \psi$ are only predicted true when the set of propositions in the ordering source jointly entail at least $\psi$.

The deviation of labor described here has connections to what has been called the *Overt Conditional Operator* (OCO) approach versus the *Covert Conditional Operator* (CCO) approach by (Kaufmann and Schwager 2006). The difference between these approaches turns on how they handle the restriction of overt modal operators found in the consequent of conditional constructions. There is need to posit a special mechanism that generates a covert epistemic modal in, e.g., indicative conditionals that do not contain any overt modal operators for the conditional to restrict. Once the existence of these operators is granted, however, how cases with overt operators are handled becomes an open question. The OCO approach claims that in cases of overt operators, the conditional restricts the overt operator. The CCO approach claims that the covert epistemic modal is both generated and restricted as the general case.

\(^7\)The *Levi Identity* is a simplification of Levi’s *Commensurability Thesis* (Levi 1991, 65). See also (Levi 1977), (Gärdenfors 1981), and (Alchourrón, Gärdenfors and Makinson 1985).
The approach sketched so far is a version of the OCO approach. No mechanism for generating covert operators has been specified, and while Frank-Zvolenszky conditionals, one main objection to the OCO approach, has been overcome, the empirical coverage could be extended with covert epistemic operators. Fortunately, this analysis already has the resources to provide a restrictive mechanism that generates these operators.

First, note that the syntactic movement postulated by (Bhatt and Pancheva 2006) and (Haegeman 2010) requires *if* to move across epistemic “space” on Cinque’s Heirarchy:
Suppose movement was constrained so that the epistemic space must be filled. This would require, in the absence of an overt epistemic modal, the generation of a covert modal. This would allow the benefits of the CCO approach, without the overgeneration of covert modals.
that the proposal in (Kratzer 2012) initially generates:\(^8\)

\begin{align*}
\text{CP} \\
\text{CondP} & \quad \text{CondP'} \\
\quad & \quad \cdots \\
\neg f & \quad \text{ModP}_{\text{epistemic}} \\
\quad & \quad \cdots \\
\text{ModP}_{\text{epistemic}} & \quad \cdots \\
\text{BaseP} & \quad \text{ModP}_{\text{epistemic}} & \quad \text{MoodP}_{\text{irrealis}} \\
\quad & \quad \cdots & \quad \cdots \\
\ldots F^* & \quad \forall & \quad \text{t} & \text{MoodP'}_{\text{irrealis}} \\
\quad & \quad \cdots \\
\end{align*}

\(^8\)See (Kaufmann and Kaufmann forthcoming) for discussion of this overgeneration and proof that, semantically, it is inert.
If this constraint existed, then deontic conditionals would generate a case of *modal concord*, where multiple modals co-occur within a single environment but modal force is expressed only once. Take a case like (A.2) above, with a deontic necessity modal in the consequent of a conditional. In this case, I propose, when the covert epistemic modal is created, it attracts the head of the lower modals BaseP, which then moves to Comp, CondP:
On this account, because the movement chain terminates in the conditional clause, no addi-
tional modification of the modal semantics is required.

As pointed out in (Kaufmann and Kaufmann forthcoming, 356), the CCO construal has problems with the following type of example:

(6.8) If Max buys a car, he will have to pay car taxes.

A solution to this problem is suggested by (Frank 1996, 54), where the modal base of the overt modals and the covert modal are connected anaphorically. The framework developed here establishes an identical connection compositionally—through syntactic movement operations corresponding to semantic binding operations. It is noted in (Kaufmann and Kaufmann forthcoming, 54) that connecting the proposals in this way re-introduces the problem of Frank-Zvolenszky conditionals. But a solution to this problem is already available, and it can be shown to be independently plausible in its ability to solve related problems, such as the problem of Tichý and Morgenbesser conditionals, as well as problems related to past-shifted counterfactuals and past-shifted deontic modal constructions made problematic by (Arregui 2010).

The strategy taken so far neutralizes the modal base of the antecedent before considering the consequent. This kind of operation returns content to the previously vacuously true sentences considered above. What comes next is an extension to multiple ordering sources, which has been implemented in (von Fintel and Iatridou 2008), (Graham Katz and Rubinstein 2012), and (Kaufmann 2013).

(von Fintel and Iatridou 2008) suggest a type of ordering source promotion based on the
following datum:

(6.9) After using the bathroom, everybody ought to wash their hands; employees have to.

This demonstrates that ‘ought’ is a weak necessity modal; one that picks out a particular alternative out of several that is the best, considering the ordering. ‘Have to’ or ‘must’ provide only a single alternative, as strong necessity modals should. Specifically, weak necessity modals pick out a ‘best’ alternative given a range of alternatives, while strong necessity modals show that a particular fact is true throughout all possible alternatives available. Weak necessity modals are cross-linguistically marked with counterfactuality morphosyntactic markings.9 If counterfactuality is considered a movement outside of the Stalnakerian context set, which is similar in concept to the epistemic modal base in (Cook 2013), then we have reason to believe that an ordering source is involved. One way the context set may be disrupted is, as we have seen above, through a modulation of the modal base via a subtraction operation.

What (von Fintel and Iatridou 2008) show is that, at times, strong necessity modals can show sensitivity to a secondary ordering source. This allows a contrast to be drawn between weak necessity in the actual world and strong necessity in some counterfactual worlds, which they require to handle the data. To do this, (von Fintel and Iatridou 2008) suggest a multiple ordering sources approach. They previously argued that strong necessity modals were dependent on an ordering source which categorizes conditions as being “normal” or

9Cf. (von Fintel and Iatridou 2008)
“stereotypical”. This normalcy condition is argued for in (Frank 1996) as a partial replacement for a Kratzerian graded modality system. Frank’s “multiple relative modality” differs from the Kratzerian system in a way similar to our approach, requiring the modal base to be anaphorically related to a complex object. This object is the combination of the modal base and the traditionally Kratzerian order, only in a way that differs from Kratzer’s graded modal constructions. Frank takes issue with the modeling of certain modalities using the tools developed to handle gradability, and proposes the following alternative: a relation of context reduction that is similar to Gärdenförs contraction, compatibility restricted union:

\[ \text{CRU } f(w) \cup !g(w) = \{ X \subseteq (f(w) \cup g(w)) : \text{consistent}(X) \land (\forall Y)X \subset Y : \text{inconsistent}(Y) \land f(w) \subseteq X \} \]

Here, rather than split the modal base from the potentially plural ordering source(s), the ordering source propositions are combined with the modal base in a way that eliminated contradictions. It is then the maximal consistent subsets of their combination that are entertained. The set \( f(w) \cup !g(w) \) is the set \( h(w) \) of all maximally consistent subsets \( h_i(w) \in h(w) \) of \( f(w) \cup g(w) \). The interpretation of basic modals w.r.t. (CRU) then follows. Let \( f, g \) be conversational backgrounds, \( w \) the world of evaluation, and \( p, q \) propositions. The compatibility restricted union \( h \), where \( h(w) = f(w) \cup !g(w) \) allows for the following basic definition of necessity and possibility:

**CRU-Necessity** \([\text{necessary } p]^h = \{ w \in W : [p] \text{ follows from } h_i(w) \in h(w) \}\)

**CRU-Possibility** \([\text{possible } p]^h = \{ w \in W : [p] \text{ is compatible with } h_i(w) \in h(w) \}\)
**CRU-least** \( [p \text{ is at least as good a possibility as } q]^{h_i} = \{ w \in W : P([p]^{h_i}) \geq P([q]^{h_i}) \} \)

**CRU-better** \( [p \text{ is a better possibility than } q]^{h_i} = \{ w \in W : P([p]^{h_i}) > P([q]^{h_i}) \} \)

**CRU-weak** \( [\text{weak necessity } p]^{h_i} = [p \text{ is a better possibility than } \neg p]^{h_i} \)

**CRU-slight** \( [\text{slightly possible } p]^{h_i} = [\text{possible } p]^{h_i} \cap [\text{weak necessity } \neg p]^{h_i} \)

Frank states that this compatibility restricted union strategy provides a better inferential basis than Kratzer’s gradable modality account when considering cases of practical inference, where, on the (CRU) account, one modal base must be considered in toto.\(^{10}\) In basic cases the two accounts do not diverge, but since Frank individuates outcomes according to consistent subsets \( h_i \), their accounts differ in outcome when handling contradictions. The more \( g(w) \) diverges from \( f(w) \) factually, the fewer of \( g(w) \) that may be jointly satisfied, given \( f(w) \).

Kratzer’s account quantifies over the outcomes \( h_i(w) \in h(w) \):

**CRU-Kratzer** \( [\text{necessary } p]^{h} = \{ w \in W : [p] \text{ follows from every } h_i(w) \in h(w) \} \)

**CRU-Kratzer** \( [\text{possible } p]^{h} = \{ w \in W : [p] \text{ is compatible with some } h_i(w) \in h(w) \} \)

Frank says, regarding the qualities that determine choice among the outcomes \( h_i(w) \in h(w) \), that rather than quantify, it is a matter of complex common sense reasoning, intentions, and preferences, that does not pertain to the domain of semantics and pragmatics proper, but rather to a theory of action and planning.\(^{11}\)

\(^{10}\)(Frank 1996, 45).

\(^{11}\)(Frank 1996, 44).
Frank considers (CRU) the result of two possibly conflicting conversational contexts, where the context with the “harder facts” takes precedent. Frank further explains that it could be that there is a ranking between contexts:\footnote{Frank 1996, 45.}

In general we assume the factual, circumstantial and also epistemic backgrounds to constitute “harder facts” than the non-factual bouletic, or deontic backgrounds. Is there any evidence for this assumption? Now everything that is determined to be factual in our world will always be so. […] So it is impossible—if we act rationally—to drop part of our factual context information and instead adopt a conflicting nonfactual assumption—at least for the interpretation of indicative conditionals.

I agree in this principle, that the premise sets which factor into modal interpretation take on a hierarchy. (Graham Katz and Rubinstein 2012) suggest an ordered merging of premise sets which we adopt.

Often, when describing a context in a common sense way, higher-order properties of the context that affect the interpretation of modal semantics are obscured. This is a key point that motivates the following theory, and the example provided in (Graham Katz and Rubinstein 2012) shows this explicitly. Taking an example from (Lassiter 2011):

Bill is extremely predictable. He almost always drives to and from work, arrives home by 6pm, and has macaroni for dinner.
(6.10) It is more likely that Bill will have something other than Macaroni for
dinner than it is that he will both fail to be home by 6pm and fail to drive
his car.

The set of expectations in this case is \{Bill drives., Bill arrives home by 6pm., Bill has
macaroni for dinner.\}. Katz, Portner, and Rubenstein argue that the appropriate truth
conditions may be derived by considering an order which encodes the notion that “other
things being equal, the more expectations that are satisfied, the better.”

Premises Adding Up For any set of propositions \(A\), \(\text{Premise}_{\text{add-up}}(A) = \text{def} \bigcup_{i=0}^{|A|} \{p_i\}\),
where \(p_i = \{w : \text{at least } i \text{ propositions in } A \text{ are true in } w\}\).

Ordering-Source Add-Up For any ordering source \(g\), \(\text{OS}_{\text{add-up}}(g) = g'\), where for any
world \(w\), \(g'(w) = \text{Premise}_{\text{add-up}}(g(w))\).

Higher-order explanations such as this one provide a key element to explaining the problem-
atic data in what follows. It also provides evidence for a multiple ordering-source theory.
This type of theory provides an explanation to the following scenario.

A doctor has a choice of two medicines, A or B, to administer to a patient. A has
a small chance of providing a total cure and a large chance of killing the patient.
B is sure to save the patient’s life, but will leave them slightly debilitated. Doing
nothing will surely lead to death:

\(^{13}\) (Graham Katz and Rubinstein 2012, 493).
(6.11) It is better to administer medicine B than medicine A.

Given a standard analysis, the wrongly predicted result is that medicine A should be preferred, as it has the most desirable outcome in the most ideal worlds. However, there seems to be an additional constraint in play. The likelihood of the worlds themselves relative to the actual world seem to change the interpretation, since the most desirable of the most likely worlds are worlds in which medicine B is administered.

This puzzle revolves around the prioritization of two separate sets of considerations. First, there is the desirability of the outcomes: it is more desirable that medicine A is administered and the patient miraculously survives than that medicine B is administered. However, the likelihood of the outcomes must also be considered. The most desirable of the most likely worlds are those in which medicine B is administered.

To resolve this issue, a binary merge operation over ordering sources is defined. We follow (Kaufmann 2013) in it’s development:\textsuperscript{14}

\textbf{Definition 6.2.6 (Sequence Structure)} Call a poset \( \langle \varphi, \subseteq \rangle \), where \( \varphi : \{f,t\} \) a \textit{sequence structure}.

\textbf{Premise Structure} Given a sequence structure \( \langle \varphi, \subseteq \rangle \), define the \textit{premise structure}:

\textsuperscript{14}To see the specifics of the solution to the medicine problem, see (Graham Katz and Rubinstein 2012). They extend this type of solution provisionally to the \textit{miner’s paradox}, which I do not cover in this manuscript. See also (Cariani, Kaufmann and Kaufmann 2013).
\( \text{Prem}(\varphi, \subseteq) \) is the poset \( \langle \varphi', \subseteq' \rangle \) s.t. (1) \( \varphi' \) is the set of sequences \( \varphi' \in \varphi \) s.t. \( \bigcap \varphi' \neq \emptyset \) and (2) \( \subseteq' \) is the restriction of \( \subseteq \) to \( \varphi' \).

**Lexicographic Order** \( \langle X, Y \rangle \leq_{\text{lib}} \langle X', Y' \rangle \) iff (1) \( X \subseteq X' \) and (2) if \( X = X' \) then \( Y \subseteq Y' \).

If \( \langle \varphi_1, \subseteq_1 \rangle \), \( \langle \varphi_2, \subseteq_2 \rangle \) are sequence structures, then so is \( \langle \varphi_1 \times \varphi_2, \subseteq_1 \times \subseteq_2 \rangle \). For two arbitrary orderings over sets of sequences \( \subseteq_1, \subseteq_2 \), the operation \( \subseteq_1 \times \subseteq_2 \) defines the lexicographic order on \( \varphi_1 \times \varphi_2 \) s.t. \( \forall \langle a, b \rangle, \langle a', b' \rangle \in \varphi_1 \times \varphi_2, \langle a, b \rangle \leq_{\text{lib}} \langle a', b' \rangle \) iff (1) \( a \subseteq_1 a' \) and (2) if \( a = a' \), then \( b \subseteq_2 b' \).

**Remark** This order allows for stacks of ordering sources of arbitrary length to be factored into the meaning of a modal. Provided a contextually defined stack of ordering sources \( G_1, \ldots, G^n \) and a contextually defined center \( \langle a, w, t \rangle \), we take the order \( \text{Prem}(G^1_\varphi(a)(w)(t), \subseteq) \ast \ldots \ast \text{Prem}(G^n_\varphi(a)(w)(t), \subseteq) \).

**TOP** Given \( \langle \varphi, \leq_{\text{lib}} \rangle \), \( \text{TOP}(\langle \varphi, \leq_{\text{lib}} \rangle) := \{ x \in \varphi : \forall y \in \varphi (x \leq_{\text{lib}} y \rightarrow y \leq_{\text{lib}} x) \} \).

Since the orders involved here are not total, there is no guarantee that \( \text{TOP}(\cdot) \) will provide a unique result. Take the following toy example: Suppose we are combining three ordering sources \( G^1(a)(w)(t) = \{pqr\} \), \( G^2(a)(w)(t) = \{abc\} \), \( G^3(a)(w)(t) = \{abc\} \), where \( \{abc\} \models \bot \).

Now consider \( \text{TOP}(\text{Prem}(G^1_\varphi(a)(w)(t)) \ast \text{Prem}(G^2_\varphi(a)(w)(t)) \ast \text{Prem}(G^3_\varphi(a)(w)(t))) \). Since premise structures are each sets of consistent sequence structures, our \( \text{TOP}(\cdot) \) operation generates the following set of sequences:

\( \{\{ab\}, \{ac\}, \{bc\}\} \)
None of these individual sequences can be unioned without generating the very same contradictory set we just attempted to avoid, but incorporating Frank’s compatibility restricted union has the exact effect we are looking for:

\[
\langle \{ab\}, \{bc\}, \{ac\} \rangle
\]
\[
\langle \{ac\}, \{ab\}, \{bc\} \rangle
\]
\[
\langle \{ac\}, \{bc\}, \{ac\} \rangle
\]
\[
\langle \{bc\}, \{ab\}, \{ac\} \rangle
\]
\[
\langle \{bc\}, \{ac\}, \{ab\} \rangle
\]

**TOP** Given \( \langle \varphi, \leq_{lib} \rangle \), \( \text{TOP}(\langle \varphi, \leq_{lib} \rangle) := \{ \bigcup ! (x \in \varphi) : \forall y \in \varphi (x \leq_{lib} y \rightarrow y \leq_{lib} x) \} \).

This additional maneuver reduces the sequences above, removing repetition, to the following set of set of premises:

\[
\{ \{ab\}, \{ac\}, \{bc\} \}
\]

The question now facing the interpretation is how to now deal with these three conflicting orders. I follow my previous line of argument and suggest that a contextually provided selection function is available \( S_{(x,t), \chi} \). This selection function is built into the lexical semantics of the modal, we assign it with the contextual assignment function \( c \), as above:

**Necessity Weak** \( \lambda F_1. \lambda G_1 \ldots. \lambda G_p. \lambda P_p. \)

\[
[S^\ell (c(n_s)(\text{TOP}(\text{Prem}(G_1^1, \subseteq) \ast \ldots \ast \text{Prem}(G_p^n, \subseteq)) \cap (F^1, \{ P^{C_1}, P \})) \subseteq P]
\]
Negation $\neg : \langle \langle \alpha, \beta \rangle, \langle \alpha, \beta \rangle \rangle := \lambda P_{\langle \alpha \beta \rangle}. [D_\alpha \times D_\beta / P]$ 

Notice that through this construction $\textsc{top}(\cdot)$ coincides with Kratzer’s account when the premise sets do not generate contradictions. However, when contradictions occur, $\textsc{top}(\cdot)$ can potentially generate multiple incomparable premise sets. This is exactly the model of (Frank 1996), which differs from Kratzer’s account in that, rather than quantifying over the incomparable tops, we instead allow for a metasemantic determination based on the best theory of action.

The method of selection in this particular case could be explained in a number of different ways. First, let us compare it with a move made by (von Fintel and Gillies 2011). Rather than define a contextual selection function to pick one out of the possible consistent sets, von Fintel and Gillies argue that on assertion, a cloud of propositions are asserted, one for each possible resolution. It is then up to the pragmatics to determine which proposition is updated on by the conversational participants. In the case of bare epistemic modals, this argument is made with respect to the resolution of the group of agents’ epistemic state to which the modal is sensitive. The group level reading involved is called aggregation:

**Aggregation** Suppose a context $c$ determines a modal base $B$ by determining a group $G$ and $c'$ determines $B'$ by $G'$: if $G \subseteq G'$, then $[B']^{c'i} \subseteq [B]^{ci}$.

**Distribution** Fix a $c$ relevant group $G_c$: $[B]^{ci} = \bigcap_{x \in G} f_x(i)$.

The authors go on to argue, however, that the distributed group reading is difficult for an agent to assert, given the usual norms of assertion. Specifically, because the distributed
reading can contain more information in it than contained by any individual member of the group, and there is no way to know—pre assertion—whether this information state does in fact contain additional information, no member of the group should be licensed to assert such a modal claim, as such an assertion risks asserting something false. The objective state of distributed belief among the group is not in general transparent to the members of the group subjectively. It is this state that is an aim or conversational goal, as I will argue. But first let us canvas the pragmatic rules von Fintel and Gillies believe govern selection from the cloud.

This cloud is generated by underdetermination of the context. The authors state:

It is important to realize that the proposal is not that some kind of objective context does provide a determinate resolution [...] and that the conversational parties are ignorant of or indifferent towards what the context is. There is no such thing as “the context”, only the contexts admissible or compatible with the facts as they are. The context of the conversation really does not provide a determinate resolution and we propose to model this by saying that there is a cloud of contexts at the given point of the conversation.

I think this attitude toward context is entirely too subjective. It is my belief that there exists a determinate context, contributed by the facts as they are. This context takes the form of a sequence of typed arguments, which are then assigned throughout the logical form of the sentence. That we are ignorant in general, as theorists, to the exact contents of this sequence,
is no difficulty. We can safely assume that it is finite in length, although possibly unbounded. It is the theory in place here, developed in (Cook 2013), that the epistemic modal base just is the distributed beliefs of the contextually specified group. The problematic data can be entirely resolved when considering the full range of game theoretic logical relations at play in a conversation. As I will demonstrate.

First, however, let me consider one example that von Fintel and Gillies provide which initially seems to threaten my view.

Suppose Alex and Billy are searching for their keys. If Alex and Billy are bickering already, one could encounter a dialogue like this one:

(6.12) A: The keys might be in the car.

(6.13) B: They’re not. I still had them when we came into the house. Why did you say that?

(6.14) A: Look, I didn’t say the *were* in the car. I said they *might be* there—*and they {??might / could} have been.*

Here I take the emphasis or stress on ‘were’ and ‘might be’ to be quotational uses, and I do not find the second might felicitous, unless it is read with the modal flavor of ‘could’, which is counterfactual. None of this suggests that Alex is retreating to a solipsistic epistemic state.

The conversion of might to a counterfactual reading, on this account, has a very natural story. There may be many cases where the distributed beliefs of the group is in fact the
empty set. Since factivity (reflexivity) has not been considered a constraint at this point, groups with conflicting beliefs will have an empty modal base. This, of course, is where the multiplicity of ordering sources, and their internal hierarchy, can be put to good use. In cases of counterfactual reasoning, a *totally realistic* order may be used to capture the facts about the actual world. I contend that this order factors in to even epistemic modality, albeit in a weak way. A weak completely factual order, in combination with an epistemic order that represents levels of justification or evidential support—potentially partly based on evidence sources, as work on evidentials would suggest—, could produce the same results as those contended by the advocates of epistemic modality representing a state of knowledge. When the epistemic modal base empties, it is the sequence of ordering sources, their relative order and therefore relative strength, all of which is determined by context, that governs the interpretation of the modal. I return to this topic when I reconsider the moral contradictions that opened this section.

First, however, I would like to expand on one pragmatic principle that von Fintel and Gillies suggest, *defeasible closure*:

**Defeasible Closure** If $H$ knows that $\varphi$ is compatible with what $x$ knows, for each $x \in G$, then it is reasonable for $H$ to defeasibly infer that $\varphi$ is compatible with what $G$ knows.

This is the beginning of a conventionalized game-theoretic account of the norms of assertion that govern modality, those with epistemic flavor specifically. It derives from a “mutual expectation in a conversation that partners in it are similarly situated.” There are many
cases where this inference is defeated, cases both with high practical stakes and those where
information asymmetry between group members is a commonly known.

I suggest that two game-theoretic principles of information sharing could expand on this
analysis, in line with my own account:

**Information Sharing** If a group $G$ distributively believes $\varphi$, then the group ought to at-
tempt, through conversational moves, to attain a state of common belief that $\varphi$.

**Information Assessment** If a group $G$ commonly believes $\varphi$, then the group ought to
attempt, through conversational moves, to attain a state of common knowledge that $\varphi$.

These conventionalized norms explain why in certain cases a modal is asserted when it’s
modal base—the distributed knowledge of the group—rules out the prejacent.

Consider the well known *muddy children* problem in the dynamic epistemic logic. In this
eexample, it is “public announcements of ignorance which drive the solution process towards
common knowledge of the true state of affairs.”\(^{15}\)

After playing outside, two of three children have mud on their foreheads. They
all see the others, but not themselves, so they do not know their own status.
Now their father comes and says: ‘At least one of you is dirty’. He then asks:
‘Does anyone know if he is dirty?’ The children answer truthfully. As questions
and answers repeat, what will happen?

\(^{15}\)(van Benthem 2004, 99).
In the first round no child knows the answer. However, for any \( n \) muddy children, all children will know in \( n \) rounds of the father repeating the question. Knowledge has been gained from the announcements of ignorance. Announcements of epistemic modals, specifically ‘might’, could be considered, in certain cases, such expressions of ignorance. But these announcements are driven by Information Sharing and Information Assessment. Without the ability to direct the inquiry in an epistemic fashion, via the use of the modal, one would not be able to query the game-theoretic group states in an efficient manner. Reactions to these queries, in the positive or negative, reflect asymmetries of information about the game-theoretic states of the group the participants are in.

Information Sharing should not be confused with attempting to attain a state of common belief that something is distributively believed by the group. For it is possible for it to be commonly believed that it is distributively believed that \( \varphi \) without it being commonly believed that \( \varphi \). This fact can be witnessed by the idea of a zero knolwedge proof in cryptographic theory. Schmers of Fitch’s Knowability Paradox will immediately recognize a resemblance:

John and Mary are exploring a tomb for hidden treasure. They are outside of a trapped room, which can be reached by one of two hallways, by one of two entrances, one on the left and one on the right. The doors are open until someone enters the room, at which point they shut. There is a password that opens the

---

16 See (Manuel Blum and Micali 1988).
17 See (Kvanvig 2008) and (Hand 2010).
right door and a password that opens the left door. But each time one of the
doors is open, the passwords change according to a specific algorithm. Mary
can prove to John that she knows the algorithm without revealing it or any of
the password to John. Mary simply walks into the trapped room and allows the
doors to be shut. John then requests ‘right’ or ‘left’. Mary says the magic word–
outside of the earshot of John–, and returns through the appropriate passage.
This process is repeated successfully as many times as John requires to be sure
that he knows that Mary knows the algorithm. John has not gained access to
the secret algorithm, or any of the passwords through which he could reverse
engineer the algorithm.

In this case, if Mary knows the algorithm then the group {John, Mary} distributively knows
the algorithm. Once John is sure of his beliefs about Mary’s information state, it then
becomes common knowledge that the distributed knowledge of the group contains the al-
gorithm. However, common knowledge of the algorithm has not been obtained. Mary may
then sell the algorithm to John, who is confident in it’s ability to unlock the doors.

I now consider one last example that poses a problem for a determinate context theorist,
originally attributed to Chris Potts (p.c):\textsuperscript{18}

Billy meets Alex at a conference and asks her:

\begin{quote}
(6.15) Where are you from?
\end{quote}
\textsuperscript{18}(von Fintel and Gillies 2011, 118).
That question is supposed, given a context, to partition answer-space according to how low-level in that context Billy wants his details about Alex to be. But notice that it’s not really clear whether Billy wants to know where Alex is currently on sabbatical or where Alex teaches or where Alex went to graduate school or where Alex grew up. And—the point for us—Billy might not know what he wants to know. He just wants to know a bit more about Alex and will decide after she answers whether he got an answer to his question or not. So context (or context plus Billy’s intentions) need not resolve the contextual ambiguity.

If Billy does not know what he wants to know, specifically, relative to a determinate contextual resolution, in this case, then I suspect the partition space contains multiple true answers, and all of the above possibilities mentioned should be distinguishable by the partition. Giving, then, a partial answer, could lead to further metalinguistic negotiation and a shift in context.

Before continuing, some definite choice points have been made regarding the delegation of duties between the modal base and (multiple) ordering source(s). I briefly highlight these points and provide some additional explanation. First and foremost I explicitly advocate an epistemic modal base and a sequence of ordering sources, ordered lexicographically by a context-sensitive order. I have covered the details of the modal base, and would now like to provide some background on ordering sources more generally. (Kaufmann 2013), when considering counterfactuals—where the modal base is empty—, conversational backgrounds
can have the following properties:\textsuperscript{19}

A conversational background $\beta$ is said to belong to the following classes iff the corresponding property holds of all $w, u, v \in W$:

**Consistent** $\beta(w)$ is consistent.

**Empty** $\beta(w)$ is the empty set.

**Realistic** all propositions in $\beta(w)$ are true at $w$.

**Unique** if all propositions in $\beta(w)$ are true at $v$ and $u$, then $v = u$.

**Totally Realistic** $\beta$ is realistic and unique.

Here I consider propositions sets of sequences of arbitrary and unbounded length. Traditionally, propositions are sets of worlds, and are conceived of as such here. This generalization is rudimentary. These ordering sources may be applied to counterfactual conditionals with empty modal bases. I contend here that when an epistemic distributed knowledge state is emptied, such a counterfactual intervention is made possible by context. This brings to light a more sophisticated view of epistemic modals, which would consider them weak modals on Kratzer’s account, due to the addition of their ordering sources. A weak epistemic modal would combine a modal base $F_{DB}$ and at least two ordering sources: $G_{\text{justification}}, G_{\text{realistic}}$. This combination of modal base and ordering sources generates the so-called ‘strong’ readings where ‘must’ seems to imply that the prejacent is true.

\textsuperscript{19}(Kaufmann 2013, 1142).
Here is a borderline case where my prediction of the data is *true*:

There is a murder case being studied by a group of detectives. The evidence is compatible with a number of different suspects, and therefore the distributed belief state and *justification* ordering source are both compatible with the same suspects being the murderer. The realistic ordering source, however, contains only one proposition: the single person who is the murderer. Bosch, based on his own instincts in dealing with the suspects and history of dealing with cases like this, makes a ‘leap of faith’ and states:

(6.16) Larissa must be the murder.

As it turns out, Larissa is in fact the murder, and so the realistic ordering source predicts the truth of (6.16). This could be considered a case where *intuitions* guide us to the truth.

This view contrasts with (von Fintel and Gillies 2010) in a number of ways, but agrees with several of their principles. They begin with Kartunnen’s Problem:

(6.17) Where are the keys?

a. They are in the kitchen drawer.

b. They must be in the kitchen drawer.

In this example the (a) response is more forceful than the (b) response. This is due to an *inferential* evidential marking carried by *must*. If one asserts (a), then one has direct
evidence of the truth of (a), whereas is one asserts (b), one only has inferential evidence of 
(b). My disagreement with von Fintel and Gillies is that the kernel, or modal base, of the 
epistemic modal models a reflexive accessibility relation. In other words, I do not think it
is directly truth entailing. I believe it is the place of the realistic ordering source to decide 
whether the must claim in the context entails the truth of the prejacent.

The justification order can possibly encode sources of beliefs that lead to the truth. This
could be turned into a linguistic account that made use of cross-linguistic evidence of eviden-
tial markings on modal expressions. It also corresponds with the traditional Justified True
Belief (JTB) account of Knowledge, in a linguistic setting. This differs from the knowledge
first approach made popular by (Williamson 2000) that has been adopted by linguistic the-
orists such as von Fintel and Gillies. Consider, however, the following modification of the 
rain example:

The weather report said it was going to rain today. Alex, Mary, and Billy have 
just come in from outside soaking wet. As they walk into the kitchen. John, 
preparing a meal, notices this and says:

(6.18) John: It must be raining.

(6.19) Mary: No, the neighbor’s sprinkler is on the fritz.

However, between the time it took Alex, Mary, and Billy to walk from the living 
room into the kitchen, it had in fact started raining. But no one had noticed.
Now, I believe this exchange to be felicitous. However, I don’t believe John could look outside the window, see that it had started raining, and exclaim "I was right!". His previous information state was not connected to the truth in the right way, and thus would have been filtered out of the restricted state by the $G_{justification}$ relation. On an account where the information state is reflexive, it is hard to see how John’s response would be ruled out:

Upon looking out the window and seeing that it had begun raining. John exclaims:

(6.20) ?? John: See, I was right!

Deontic modals have a variety of different ordering sources from which combinations may be drawn. We have already been introduced to stereotypical ordering sources, which encode the likelyhood of outcomes, as well as desirability ordering sources, which encode the general–or agent specific–desirability of specific outcomes. (Kratzer 1991, 646) splits deontic ordering sources into the following inexhaustive list:

1. what the law provides ...

2. what is good for you ...

3. what we aim at ...

4. what we hope ...

5. what is rational ...
Notice uses of the plural above. This is further evidence that even the ordering source could potentially take a group reading. Sorting out the specific precedence order of this complicated interplay of deontic sources is the realm of pragmatics proper, where context is adjudicated. I only propose that the framework developed makes available the ranking and integration of multiple such ordering sources, this increases the explanatory value by making explicit the structural framework wherein the sentence content is developed.

Let us consider the “Samaritan Paradox” by (Kratzer 1991, 643):

(6.21) No murder occurs.

(6.22) If a murder occurs, the murderer will go to jail.

(6.23) If a murder occurs, the murderer will be invited for dinner.

A standard semantics has a problem differentiating between the two conditionals. However, introducing an ordering source scale allows one to prioritize sets of source propositions easily. The modal interpretation can be broken down into stages. First the conditional takes the modal base of a covert embedded (epistemic) modal as a restrictor anaphorically as per (Frank 1996). However, this modal base is then neutralized to the antecedent according to
**subtraction**, before the antecedent is added to the modal base according to **addition**. This saves as much of the modal base as possible in the case of counter-epistemic antecedents. Then the modal in the consequent is evaluated by again neutralizing the modal base to the prejacent condition according to **subtraction**, before the ordering source restriction occurs. It is then the business of the ordering sources to determine if a consequence relation holds. This ordering relation is processed by **top**. The modal base should never generate this entailment directly. Factoring and weighing the different types of ordering sources according to the context generates entailments. It could be the case that many of the propositions in the modal base are shared by one or more of the ordering sources, and this should not be considered a problem or an overworking of the system. It is a specific division of labor that generates a general framework capable of handling a variety of edge cases. In this case, it is easy to see that a combination of ordering sources exist where murderers going to jail are prioritized over murderers going free and being available for dinner. It is the neutralization to the antecedent and consequent prejacent that ensures the modal base does not crash on the basis of contradictory information.

(Arregui 2010) argues that **should have** cannot be read as a **past should**: Consider John, an officer in a military parade, who is chosen randomly, by the toss of a coin, to be shown in a close-up on television. Looking at him, someone could utter (6.24):

(6.24) John should have shaved.
I now consider a pair of past-shifted counterfactuals. First, take a case from (Slote 1978, 27), originally attributed to Sidney Morgenbesser: A fair coin is tossed; a bet on heads is offered; the bet is declined. Consider (6.25):

(6.25) If you had bet heads, you would have won.

In this case the judgement is that (6.25) is true. Now consider the case introduced by (Tichý 1976). Suppose Fred always takes his hat when it rains, but takes it only half of the time when it doesn’t rain. Now consider (6.26):

(6.26) If it had not been raining, Fred would have taken his hat.

Contrasting with (6.25), (6.26) is judged to be false. This suggests the following pair of predictions:

(6.27) (a) If you had bet heads, you would have won. (⊤)

(b) If it had not been raining, Fred would have taken his hat. (⊥)

The contrast found in (6.27) suggests that there is some sensitivity to the facts between the time of antecedent tense (here past) and the time of the context. This particular contrast is drawn out in an example by (Edgington 2004, 12):

I am driving to the airport to catch a nine’o’clock flight to Paris. The car breaks down on the motorway. I sit there, gnashing my teeth, waiting for the breakdown

20I follow (Walters 2009) in my presentation of these cases.
service. Nine’o’clock passes: I’ve missed my flight. More time passes. ‘If I had caught the plane, I would be halfway to Paris by now.’, I say to the repairman who eventually shows up. ‘Which flight were you on?’ he asks. I tell him. ‘Well you’re wrong.’, he says. ‘I was listening to the radio. It crashed. If you had caught the plane, you would be dead by now.’

In this case there are two information contexts, by which past-shifted counterfactuals are evaluated. First, we have the pre-repairman post-flight context $C_1$:

$\text{(6.28)} \quad C_1 : \{\neg \text{catch}, \} $

(a) If I had caught the plane, I would be halfway to Paris by now. ($\top$)

Here, in the contemporary analysis, the modal base will have $\neg$catch removed. However, if there exists an ordering source $g$, contextually supplied, which models the ordinary or normal course of events, it would include her catching the plane and being halfway to Paris by now, since this is the world-line from which we have diverged. Thus, 6.28 is judged true. However, certain additional information is introduced about the actual world-line—that the plane crashed. The judgment shifts once this new information is introduced:

$\text{(6.29)} \quad C_2 : \{\neg \text{catch}, \text{crash} \} $

(a) If I had caught that plane, I would be halfway to Paris by now. ($\bot$)

(b) If I had caught that plane, I would be dead by now. ($\top$)
Now that the actual world is one in which the plane has crashed, the ordering source which supplies the normal course of events shifts. It no longer includes worlds where the plane is in the air and halfway to Paris, as this possibility has been pruned from the actual world line from which the ordering source is generated. This new ordering source predicts her death. The solution here is the work of the distribution of labor between the modal base, which oridinarilly and originally carries this information and the ordering sources which model and predict how things should be and what should occur.

In each of the Tichý and Morgenbesser conditionals in (6.27) above it is the time of the utterance that the modal base and ordering source are generated, and thus the facts about the past are fixed.

In the case of Fred taking his hat, note that in the example it has not been stated that it is commonly believed either that it is raining or that Fred has indeed taken his hat. Thus, only the conditional premise that \textit{if it rains, Fred takes his hat} should be considered part of the common ground. None of the facts about Fred taking his hat are fixed by the past. Further, it should be assumed that there are some worlds where it is not raining and Fred has not taken his hat, satisfying the second premise, that when it does not rain Fred only takes his hat half the time. Thus, no prediction can be made at the present time about whether Fred would have his hat if it had not been raining.
Appendix A

Derivation Comparison

Rules

The system of modality and conditionals proposed has had several moving parts introduced over this development. I pause now to review the essential pieces. We have developed two separate types of moves. First, we have compositional moves related to the specific compositional framework I have chosen to develop, for ancillary reasons. Second, we have the interpretation of the modal and conditional, adopted for this framework in minor ways. In this section I lay out the semantic and pragmatic rules required for the interpretation of a bare epistemic or deontic modal. In the next section, I compare two versions of composition and their corresponding conditional denotations. I end with proofs demonstrating the differences between the two rule-sets compositionally.
Modal Base

The modal base is populated by the *distributed knowledge* of the group. *Aggregation* and *distribution* enforce this constraint.

**Aggregation** Suppose a context $c$ determines a modal base $B$ by determining a group $G$ and $c'$ determines $B'$ by $G'$: if $G \subseteq G'$, then $[B']^{c'i} \subseteq [B]^{ci}$.

**Distribution** Fix a $c$ relevant group $G_c$: $[B]^{ci} = \bigcap_{x \in G} f_x(i)$.

The following pragmatic rules explain the conversational dynamics of this group operation:

**Defeasible Closure** If $H$ knows that $\varphi$ is compatible with what $x$ knows, for each $x \in G$, then it is reasonable for $H$ to defeasibly infer that $\varphi$ is compatible with what $G$ knows.

**Information Sharing** If a group $G$ distributively believes $\varphi$, then the group ought to attempt, through conversational moves, to attain a state of common belief that $\varphi$.

**Information Assessment** If a group $G$ commonly believes $\varphi$, then the group ought to attempt, through conversational moves, to attain a state of common knowledge that $\varphi$.

Here I aggregate (Frank 1996)'s interpretations of modal comparison, as well as a basic rule for negation. These rules are not the study of this inquiry, but paint a bigger picture.

**Least** $[p$ is at least as good a possibility as $q]^{h_i} = \{w \in W : P([p]^{h_i}) \geq P([q]^{h_i})\}$
Better $[p \text{ is a better possibility than } q]^h_i = \{w \in W : P([p]^h_i) > P([q]^h_i)\}$

Weak $[\text{weak necessity } p]^h_i = [p \text{ is a better possibility than } \neg p]^h_i$

Slight $[\text{slightly possible } p]^h_i = [\text{possible } p]^h_i \cap [\text{weak necessity } \neg p]^h_i$

Negation $\neg : \langle \langle \alpha, \beta \rangle, \langle \alpha, \beta \rangle \rangle := \lambda P_{(\alpha\beta)}\cdot[D_\alpha \times D_\beta/P]$

Here is the final definition of a weak modal, subsuming both epistemic and deontic modality.

Weak Modal $\lambda F_1 \lambda G_1 \ldots \lambda G_n \lambda P_p$. [$\mathcal{S}^U(c(n_S)(\top(Prem(G^1, \subseteq) \ast \ldots \ast Prem(G^n, \subseteq))))((\cap - (\langle F_1, \{P_C, P\} \rangle)) \subseteq P)$

Where $c(n_S)$ is the contextual selection function selecting the numerical value $n$ that corresponds to the selection of the element from $\top(\cdot)$ determined by context. $\top(\cdot)$ is defined:

Top Given $\langle \varphi, \leq_{lib} \rangle$, $\top(\langle \varphi, \leq_{lib} \rangle) := \{\bigcup! (x \in \varphi) : \forall y \in \varphi (x \leq_{lib} y \rightarrow y \leq_{lib} x)\}$.

CRU $f(w) \cup! g(w) = \{X \subseteq (f(w) \cup g(w)) : \text{consistent}(X) \land (\forall Y)X \subseteq Y : \text{inconsistent}(Y) \land f(w) \subseteq X\}$

$\top(\cdot)$ is a complex definition, I now break down its parts:

Sequence Structure Call a poset $\langle \varphi, \subseteq \rangle$, where $\varphi : \langle f, t \rangle$ a sequence structure.

Premise Structure Given a sequence structure $\langle \varphi, \subseteq \rangle$, define the premise structure:

$Prem(\varphi, \subseteq)$ is the poset $\langle \varphi', \subseteq' \rangle$ s.t. (1) $\varphi'$ is the set of sequences $\varphi' \in \varphi$ s.t. $\cap \varphi' \neq \emptyset$ and (2) $\subseteq'$ is the restriction of $\subseteq$ to $\varphi'$.
Lexicographic Order $\langle X, Y \rangle \leq_{lib} \langle X', Y' \rangle$ iff (1) $X \subseteq X'$ and (2) if $X = X'$ then $Y \subseteq Y'$.

If $\langle \varphi_1, \subseteq_1 \rangle$, $\langle \varphi_2, \subseteq_2 \rangle$ are sequence structures, then so is $\langle \varphi_1, \subseteq_1 \rangle * \langle \varphi_2, \subseteq_2 \rangle$, defined as $\langle \varphi_1 \times \varphi_2, \subseteq_1 \times \subseteq_2 \rangle$. For two arbitrary orderings over sets of sequences $\subseteq_1, \subseteq_2$, the operation $\subseteq_1 \times \subseteq_2$ defines the lexicographic order on $\varphi_1 \times \varphi_2$ s.t. $\forall \langle a, b \rangle, \langle a', b' \rangle \in \varphi_1 \times \varphi_2, \langle a, b \rangle \leq_{lib} \langle a', b' \rangle$ iff (1) $a \subseteq_1 a'$ and (2) if $a = a'$, then $b \subseteq_2 b'$.

Remark This order allows for stacks of ordering sources of arbitrary length to be factored into the meaning of a modal. Provided a contextually defined stack of ordering sources $G_1^1, \ldots, G^n_a$ and a contextually defined center $\langle a, w, t \rangle$, we take the order $Prem(G_1^1(a)(w)(t), \subseteq) * \ldots * Prem(G^n_a(a)(w)(t), \subseteq)$.

A conversational background $\beta$ is said to belong to the following classes iff the corresponding property holds of all $w, u, v \in W$:

**Consistent** $\beta(w)$ is consistent.

**Empty** $\beta(w)$ is the empty set.

**Realistic** all propositions in $\beta(w)$ are true at $w$.

**Unique** if all propositions in $\beta(w)$ are true at $v$ and $u$, then $v = u$.

**Totally Realistic** $\beta$ is realistic and unique.

In the manuscript I argue that, for epistemic modals at least, a stack of three ordering sources are required. First, there is a standard *stereotypical* ordering source, a *justification*
ordering source, and a *realistic* ordering source. This combination generates a view of epistem+
omic modality where an account of knowledge may be maintained—a JTB or JTB+ account,
depending on how you split hairs—, without requiring the entire concept to be contained
within the calculation of the modal base. I believe this to be a benefit, as the modal base
can be manipulated when it composes with a conditional. In this way, the effects of the
groups knowledge is not totally lost when calculating conditional interpretations, as it would
be if the entire package was forced into the modal base.

**VIFA and Combinatorial Logic**

To resolve the problem of *Frank-Zvolensky* conditionals, the following addition and subtract-
tion rules are introduced. These rules break the state of the modal base, as it is passed up
the syntactic tree anaphorically:

**Addition** \( \lambda F. \lambda A. [+ (F, A)] = F' : f \text{ s.t. for all } w : s, t : i, x : e, p : p, F'(w) = F(w) \text{ except } F'(w)(A) = 1. \)

**Subtraction** For \( \Gamma, \Delta : f, - (\Gamma, \Delta) \) is a function of type \( f := \text{ the contextually determined maximal set } \gamma : f \subseteq \Gamma : f \text{ s.t. there is no } \varphi : f \subseteq \gamma : f \text{ s.t. } \exists \delta : p \in \Delta : f(\cap \varphi \subseteq \delta). \)

In what follows I use standard rules more familiar to contemporary linguistics to generate
a derivation that may be easier to read and understand in comparison with my new
developments, which follow.
Let $\alpha$ be a branching node and $\beta, \gamma$ be $\alpha$’s daughters, where $\beta$ dominates only a numerical index $i$. Then, for any variable assignment $g$, $[\alpha]^g = \lambda x. [\gamma]^g[i:=x]$.

This rule is in the similar spirit of the VIFA:

**VIFA** If $\alpha$ is a branching node and $\{\beta, \gamma\}$ the set of its daughters, $\beta$ is of type $\langle \langle a, b \rangle, c \rangle$ and $\gamma$ is of type $b$ and contains a free variable indexed $i$ of type $a$, and $(\lambda x. [\gamma]^g[i:=x])$ is in the domain of $[\beta]^g$, then $[\alpha]^g = [\beta]^g(\lambda x. [\gamma]^g[i:=x])$.

This operation is build into the interpretation of *if* generally. In the same way, (von Stechow 2003) posits a feature on *only* that deletes * features on indexicals that must be bound, here *if* must likewise carry a feature that obligatorily deletes * features from the arguments $s, i, e$ in its restriction, allowing (VIFA) to bind them, creating a function of type $p$.

**IF** $\lambda A. \lambda F. \lambda C(f, t). [C (+(- (F, \{C\})), A)]$

We have now defined both an addition operation $+$ and a subtraction operation $-$ on the modal base. Putting this together, I provide a derivation of (A.2). For the purposes of simplifying an already complicated derivation, I eliminate the pronoun in the consequent for a name which we assume co-refers:

(A.1) If John$_1$ shaves, John$_1$ should shave.

$^1$See (Glanzberg and King m.s.) for a detailed comparison of the two rules.
\[
\begin{align*}
\text{if } & \text{ John shaves John should shave} \\
& \quad \text{max}(\lambda p. C(\langle p \rangle)(\text{\textit{John}\_\textit{t}})(\text{\textit{shave}}(\langle \text{\textit{f}} \rangle)) \subseteq \lambda s. \lambda m. \lambda x. \lambda w. \text{\textit{shave}}(\langle \text{\textit{f}} \rangle)(\langle \text{\textit{s}} \rangle)), \lambda s. \lambda m. \lambda x. \lambda w. \text{\textit{shave}}(\langle \text{\textit{f}} \rangle)(\langle \text{\textit{s}} \rangle)) \subseteq \lambda s. \lambda m. \lambda x. \lambda w. \text{\textit{shave}}(\langle \text{\textit{f}} \rangle)(\langle \text{\textit{s}} \rangle))
\end{align*}
\]
To provide an analysis of the conditional in the spirit of *direct compositionality*, the combinator $K$ must be introduced into the interpretation of *if*.

$$\text{If } \lambda A.p.\lambda x.s.\lambda y.i.\lambda z.e.\lambda F.f.\lambda C.(f.t).[K(C(+(-F,\{A^C\}),A)))](x,y,z)$$

The following new composition rules are required to circumvent the syncategorematic rule (VIFA):

**COORD1** If $\alpha$ is a branching node, $\{\beta, \gamma\}$ is the set of $\alpha$’s daughters, and $\beta : \langle x, x \rangle$, $\gamma : \langle xy, t \rangle$,

then $[\alpha] = \lambda x.\lambda y.[\gamma(\beta(x))(y)]$

**COORD2** If $\alpha$ is a branching node, $\{\beta, \gamma\}$ is the set of $\alpha$’s daughters, and $\beta : \langle \langle x, t \rangle, x, t \rangle$, $\gamma : \langle \langle x, t \rangle, x, t \rangle$, then $[\alpha] = \lambda x : \langle x, t \rangle.[\beta(x) = \gamma(x) = 1]$.

**DELAY** $\alpha : \langle xy, t \rangle$, then $\lambda x : x.\lambda y : y.[\alpha(x)(y)] = \lambda y : y.\lambda x : x.[\alpha(x)(y)]$. Thus, $\alpha : \langle xy, t \rangle = \alpha' : \langle yx, t \rangle$.

This generates an alternative derivation:

(A.2) If John$_1$ shaves, John$_1$ should$_2$ shave.
if John$_1$ shaves John$_1$ should shave
\[ \lambda F(x,e(y)) \cdot \lambda s. \lambda t. \lambda x. \lambda y. \lambda z. \lambda p_{(e(x),y,z) \cdot p_{(e(x),y,z)}}. \]

\[ [F(x,e(y)) \cdot \lambda s. \lambda t. \lambda x. \lambda y. \lambda z. \lambda p_{(e(x),y,z) \cdot p_{(e(x),y,z)}}]. \]

\[ \lambda g_{(e(x),y,z) \cdot p_{(e(x),y,z)}}. \]

\[ [g_{(x,y,z)(p)}]. \]

\[ \lambda P_{(e(x),y,z) \cdot p_{(e(x),y,z)}}. \]

\[ [P_{(x,y,z)}]. \]

\[ \lambda x. \lambda y. \lambda z. \lambda p_{(e(x),y,z) \cdot p_{(e(x),y,z)}}. \]

\[ [\text{shave}(x,y,z)]. \]

\[ \lambda P_{(e(x),y,z) \cdot p_{(e(x),y,z)}}. \]

\[ [\text{shave}(x,y,z)]. \]

\[ \lambda x. \lambda y. \lambda z. \lambda p_{(e(x),y,z) \cdot p_{(e(x),y,z)}}. \]

\[ [\text{shave}(x,y,z)]. \]

\[ \lambda f_{(e(x),y,z) \cdot p_{(e(x),y,z)}}. \]

\[ [f_{(x)(y)(z)(p)}]. \]

\[ \lambda x. \lambda y. \lambda z. \lambda p_{(e(x),y,z) \cdot p_{(e(x),y,z)}}. \]

\[ [f_{(x)(y)(z)(p)}]. \]

\[ \lambda P_{(e(x),y,z) \cdot p_{(e(x),y,z)}}. \]

\[ [\text{shave}(x,y,z)]. \]

\[ \lambda x. \lambda y. \lambda z. \lambda p_{(e(x),y,z) \cdot p_{(e(x),y,z)}}. \]

\[ [\text{shave}(x',y',z')]. \]

\[ w : s \quad t : i \quad x : i \quad \text{john} : e \quad \times i \]

\[ (w,t) : s \times i \quad \times i \]

\[ (w,t,john) : s \times i \times e \quad \times i \]
\[(w, t, john) : s \times i \times e \quad \lambda x, \lambda y, \lambda z, \lambda f_{\text{ss}}(g(x, y, z)) \cdot \lambda p_{\text{ss}}. \{ f(K(c(1))(x)(y)(z))(p) = f(x)(K(c(2))(y))(z)(p) = f(x)(y)(z)(p) = 1 \} \subseteq \lambda x, \lambda y, \lambda z, \lambda f_{\text{ss}}[\text{share}(x', y', z')]\]

\[f(K(c(1))(w))(t)(john)(p) = f(w)(K(c(2))(t))(john)(p) = f(w)(t)(john)(p) = 1 \} \subseteq \lambda x, \lambda y, \lambda z, \lambda f_{\text{ss}}[\text{share}(x', y', z')]\]
\[\lambda C((\alpha x^n_0 . \alpha y^n_0 . \alpha z^n_0 . \alpha p^n_{\mathit{last}, t} . [f(x^n_0, y^n_0, z^n_0)](p)) \circ \alpha f^n_{(g(w, t, john))\circ \alpha p^n_{\mathit{last}, t} . [f(K(c(1))(w))(t)(john)(p)] = f(w)(K(c(2))(t)(john)(p) = 1) \subseteq \lambda x^n_0 . \lambda y^n_0 . \lambda z^n_0 . [\mathit{shave}(x^n_0, y^n_0, z^n_0)]\]

\[\begin{align*}
\lambda y^n_0 . \lambda z^n_0 . [\mathit{shave}(x^n_0, y^n_0, z^n_0)]((t', w')) = \lambda y^n_0 . \lambda z^n_0 . [\mathit{shave}(x^n_0, y^n_0, z^n_0)]((t', w'))
\end{align*}\]
Bibliography


Glanzberg, M. and King, J. C.: m.s., Binding, compositionality, and semantic values, *m.s.*


Lewis, D.: 1979, Attitudes *De Dicto* and *De Se*, *The Philosophical Review*.


McCawley, J. D.: 1968a, Lexical insertion in a transformational grammar with deep structure, in B. Darden, C.-J. Bailey and A. Davidson (eds), *Papers from the Fourth Meeting of the Chicago Linguistics Society*, University of Chicago, Department of Linguistics.


Pierce, B. C. (ed.): 2005, *Advanced Topics in Types and Programming Languages*, MIT.


Quine, W.: 1953, *From a Logical Point of View*, Harvard U P.

Quine, W.: 1960b, *Word and Object*, MIT.


Vlach, F.: 1973, ‘Now’ and ‘then’: a formal study in the logic of tense and anaphora, PhD thesis, UCLA.


