

The Dynamics of Sense and Implicature

Dissertation

Presented in Partial Fulfillment of the Requirements for the Degree Doctor
of Philosophy in the Graduate School of The Ohio State University

By

Scott Martin, B.A., M.A.

Graduate Program in Linguistics

The Ohio State University

2013

Dissertation Committee:

Carl Pollard (co-advisor)

Craige Roberts (co-advisor)

Michael White

Copyright © by

Scott Martin

2013

Abstract

This thesis is a both a descriptive and theoretical examination of *implicatures*, parts of the contextual meanings of utterances that are separate from their *sense*, their main point. The empirical taxonomy I describe draws on the work of Grice (1975), but fleshes out two important subcategories of implicature that he did not discuss in detail. One of these subcategories is the *conventional implicatures*, which contains definite anaphora, iterative adverbs, honorifics, and Potts’s (2005) “CIs”: nominal appositives, nonrestrictive relative clauses, *as*-parentheticals, and expressives. The other is the *nonconventional implicatures* apart from Grice’s *conversational implicatures*, which contains lexical items often construed as bearing presuppositions, such as so-called *factive* verbs, aspectuals, and achievements. I offer evidence that what distinguishes the class of anaphora from other conventional implicatures is that the use of a definite must be anchored to the speaker, since it bears the implication that an antecedent is retrievable in the discourse context.

This new characterization of the landscape of implicatures holds significant consequences for semantic theory. The notion of *contextual felicity* usually assumed to apply to presuppositions is generalized to a mechanism that accounts for the (in)felicity of all conventional implicatures on the basis of whether their content conflicts with entailments present in the discourse context. For the nonanaphoric implicatures, the choice between anchoring their implicature to the speaker or to an embedded perspective is influenced in part by this new notion of contextual felicity. Since it places nonanaphoric lexical items usually thought to be presuppositional under the category of nonconventional implicatures, the terms *presupposition* and *anaphora* become synonymous. As a corollary, the process of presupposition *accommodation* (Lewis, 1979) takes on a more limited role than is usually thought.

I formalize insights from this new taxonomy of implicatures in a dynamic semantics that follows on the work of Heim (1982), Beaver (2001), and de Groote (2006), among others. This formal theory is *dynamic* in the sense that utterances are modeled as both updating the discourse context and depending on it for their own interpretation. It is also *compositional* in Montague's (1973) sense: meanings of phrases are built up based on the meanings of their component lexical items and the way they are syntactically combined. Rather than the single meaning level usually assumed by semantic theories, the dynamic semantics developed here uses a two-level scheme for separating the sense of expressions from their implicatures, following Karttunen and Peters (1979). This dynamic semantics is embedded within a categorial grammar that separates word order from combinatorics, based on ideas originally due to Curry (1961). The grammar is in turn built upon the solid, well-understood, mainstream mathematical foundations of dependent type theory and linear logic.

I then apply this semantics to build a robust account of anaphora that adopts the perspective that definites give rise to an implication of *retrievability* of their antecedent, rather than bearing presuppositions, as they are usually treated. The notion of *anaphoric accessibility* common to dynamic theories is extended by implementing Roberts's (2003) *weak familiarity* in the account, broadening its empirical coverage to instances in which a definite's antecedent is merely entailed to exist, but not overtly mentioned. The semantics is then extended to handle both the *weak* and *strong* readings of determiners in a contextually dependent way. Potts's (2005) CIs are also modeled by using essentially the same mechanism as the one that captures the implicatures associated with anaphora. I show how the formal theory I develop here represents a considerable advance with respect to Potts's, due largely to a more empirically adequate treatment of anaphoric links between CIs and the discourse in which they are situated.

I argue that the theory I develop in this thesis compares favorably with other attempts to treat similar phenomena. In addition to clarifying the empirical status of implicatures through a new meaning taxonomy that extends Grice's original, its formal rigor provides an explicit scientific theory of conventional implicature that is *falsifiable*, in the sense of making predictions to which counterexamples could in principle be given. I speculate about how the theory might be extended to account for even more phenomena, and argue that it is very well suited to computational implementation for various applications.

To Robin, Vera, and Xavier

Acknowledgments

It has been said that doing a Ph.D. is a monastic experience. While I think there is some truth in that statement, I also now have a deeper understanding of what John Donne meant when he said that no man is an island. I am deeply indebted to everyone who helped and influenced me along the way. I hope I can eventually offer more in return than this small gesture of thanks.

First and foremost, I am humbly and enduringly grateful to my wife Robin and our two amazing kids. There are significant risks and hardships that go along with writing a dissertation, and I know that to whatever extent these affected me, they affected you in equal measure. Your love and support kept me going, and dinners with you were a nightly reminder of what's really important.

I was honored to receive a dissertation year presidential fellowship from the Ohio State University graduate school, which I gratefully acknowledge. Without it, this thesis would not have been possible in its current form. I owe thanks to Chris Barker, David Beaver, Philippe de Groote, Jirka Hana, Jungmee Lee, Reinhard Muskens, Julia Papke, Carl Pollard, Craige Roberts, Anastasia Smirnova, and Bridget Smith for the role they played in helping me get the presidential. I am also thankful to have been the recipient of generous funding from the OSU linguistics department to present early versions of my thesis work at various conferences.

I trace the origins of this project to my presentation of Muskens's (1996) paper *Combining Montague Semantics and Discourse Representation Theory* to Carl Pollard's seminar in the spring of 2009. That fall, Carl, Craige Roberts, Elizabeth Smith and I started meeting informally as the *Findlay working group*, so named because Findlay, Ohio is about halfway between Columbus and Ann Arbor, Michigan, where Craige was visiting that term—and because *Upper Sandusky* was thought too verbose. I benefited greatly

from those early Findlay discussions, and had fun too. Our collaboration convinced me that this entire effort was worth undertaking.

It almost goes without saying that I owe a great deal to my committee members, all of whom strongly influenced both my education in linguistics and this thesis, and all of whom I am honored to call colleagues. I first got into this line of work after taking Carl Pollard's course in formal foundations when I arrived at Ohio State. I was intrigued to see how formal methods could be applied to linguistic analysis, even though I found the course itself so challenging at the time that I would not have imagined eventually becoming the grader for it. I have been profoundly influenced by Carl's view that in science, formal methods are not just valuable but indispensable. Carl spent many long afternoons with me at various coffee shops near campus, filling up legal pads with detailed notes, poring over the problems I was struggling with, and almost always giving pointers that aimed in the right direction. In addition to finding numerous errors and inconsistencies in my work, he was open minded enough to really listen to my ideas while maintaining the honesty to force me to confront the facts head on. I benefited greatly from Carl's ability to ask questions no one else had thought of forthrightly and without pretense.

I met Craige Roberts in her semantics course in the winter of 2007. In many ways, it was this class, which she co-taught with David Dowty and Judith Tonhauser, that got me interested in semantics as a focus of inquiry. From the beginning, it was heartening to find an area of linguistics where formal methods have a strong influence. But I have also been consistently impressed by Craige's seemingly limitless persistence in the face of complications, confusions, and unknowns. Her intimate awareness of the core issues in semantics is truly monumental, and her willingness to pose tough questions is admirable. It was often Craige who prompted me to get clear about not just *what* I was trying to do and *how*, but also ask *why*, and consider how it fit into the larger scheme of things. Since I started my thesis, Craige has really made an effort to bring me into the fold of semantics research, and I continue to be inspired by her view that semantics, at its best, is a science that tries to get at what people do when they communicate.

If not for Mike White, I probably would never have come to Ohio State in the first place. It seems that he, more than anyone else, thought it wise to gamble on admitting a bored software engineer with a background in languages. With a more computational focus, Mike often raised questions

that defied comfortable assumptions I had made. He was also observant enough to notice several pretty serious errors in my work that had eluded everyone else, including me. I could always count on him to offer unflinching, incisive comments that cut to the heart of the matter. I also have Mike to thank for helping me secure several research assistantships during my time at Ohio State, without which I would not have been able to explore the computational side of linguistics at anywhere near the same level of detail. Collaborating on software projects and research with him has been a welcome confirmation that my work in semantics, though distantly related, is rooted in some kind of practical reality.

Along the way I have had the privilege of sharing my ongoing work with audiences in a number of different venues. I thank Philippe de Groote and Sylvain Pogodalla for inviting Carl Pollard and me to give a talk at the CAuLD Workshop on Logical Methods for Discourse in 2009. There I benefited from many stimulating discussions, not only with Philippe and Sylvain, but also with David Beaver, Joey Frazee, Oleg Kiselyov, Florent Pompidine, Ken Shan, and others. Itamar Francez and Marcin Morzycki pointed out some oversights and oversimplifications in an early version of this work that I presented at the Semantics Workshop of the American Midwest and Prairies in 2010. Some helpful pointers on my presentation to the 2011 Formal Grammar conference made their way into this thesis, and thanks are due to Philippe de Groote, András Kornai, Sylvain Salvati, and Thomas Ede Zimmermann for their suggestions and for a lively and entertaining conference dinner. I had long and interesting talks with Chris Barker at Semantics and Linguistic Theory in 2012, not only about the work I presented there with Greg Kierstead, but also about the state of linguistics as a field. At that same conference, I also had good conversations with Dan Lassiter, who fortunately warned me to avoid Pullum's (1989) boojum.

Rough and overly dense drafts of these presentations were delivered to the OSU discussion groups on logic, language, information and computation (a.k.a. *Commies*), syntax and semantics (a.k.a. *Synners*), and pragmatics, which unfortunately lacks a cute nickname. Thanks to all who endured these presentations and offered suggestions for how they might be improved, especially to Jefferson Barlew, Greg Kierstead, Andy Plummer, Alex Wein, and Murat Yasavul, for criticism that was sometimes pointed and usually insightful. Over the years, I also benefited from conversations with Adriane Boyd, Jon Dehdari, Manjuan Duan, Dominic Espinosa, Jirka Hana, Dave Howcroft, Dahee Kim, Yusuke Kubota, Bob Levine, Det-

mar Meurers, Rajakrishnan Rajkumar, Liela Rotschy, Marty van Schijndel, William Schuler, Oxana Skorniakova, Anastasia Smirnova, Bridget Smith, Elizabeth Smith, and Abby Walker, among others. Some of you were students, some were colleagues or mentors who gave me good advice at a time when I needed to hear it, and I am grateful.

Friends and family helped keep me grounded during this project. Bill and Laurie Franz were beyond generous, in ways large and small, and I am glad I had the chance to get to know them well. I enjoyed many nights of animated discussions about linguistics, academic life, and almost anything else with Vedrana Mihaliček and Chris Worth, both of whom started graduate school the same year as me. Always being at the same point in our graduate careers, grappling with many of the same issues at about the same time, helped build a camaraderie between us that I truly value. The Harnettys were my constant companions. They were gracious and open enough to really share their lives with me, and made me feel almost like family. The Franzes, Gorelovs, Willamans, and FORS made the holidays times to remember, and I still look forward to Memorial weekends at Lake Hope with the Fomins and Shepards. I was also lucky enough to live in a place where neighbors became friends, and I enjoyed many birthday parties, block parties, Derby parties, Halloweens and hangouts with the Tibet Road social club, and with all of the other Clintonville neighbors I had the good fortune to get to know.

I thank my colleagues at Nuance's natural language understanding and artificial intelligence laboratory, which I am fortunate to have joined. Ron Kaplan and Joel Tetreault afforded me the space and time to defend my thesis, and after a presentation of my thesis work to our research group, Chris Brew and Dick Crouch delivered an unvarnished critique that helped me be much better prepared for my defense. Valeria de Paiva graciously took the time to read a draft of appendix [A](#), offering many helpful suggestions.

Lastly, I am grateful to my parents, Donna and John Martin, my grandparents, Bea and Bob Buck and Ragnheiður and Robert Martin, and my brothers Mike and Todd, for instilling in me a sense that I could do whatever I wanted if I just kept at it, worked hard, and never got discouraged. Without this self-confidence and persistence, I would never have dared to set such a distant and difficult goal for myself, and I doubt it would have been possible to achieve it.

Vita

M.A. in Linguistics, The Ohio State University, 2011.

B.A. *cum laude* in French and Philosophy, Florida State University, 1995.

Presidential Fellow, The Ohio State University, 2012–2013.

Graduate Research Associate, The Ohio State University Department of Linguistics, 2006–2012.

Graduate Teaching Associate, The Ohio State University Department of Linguistics, 2007–2011.

Publications

Journal Articles

Scott Martin and Carl Pollard. A higher-order theory of presupposition. *Studia Logica*, 100(4):727–751, 2012. doi:[10.1007/s11225-012-9427-6](https://doi.org/10.1007/s11225-012-9427-6).

In Peer-Reviewed Proceedings

Gregory Kierstead and Scott Martin. A multistratal account of the projective Tagalog evidential ‘daw’. In *Proceedings of the 22nd Conference on Semantics and Linguistic Theory*. CLC Publications, 2012.

Kapil Thadani, Scott Martin, and Michael White. A joint phrasal and dependency model for paraphrase alignment. In *Proceedings of the 24th International Conference on Computational Linguistics*, 2012.

Scott Martin. Weak familiarity and anaphoric accessibility in dynamic semantics. In Philippe de Groote and Mark-Jan Nederhof, editors, *Formal Grammar*, number 7395 in Lecture Notes in Computer Science, pages 287–306. Springer, 2012. doi:[10.1007/978-3-642-32024-8_19](https://doi.org/10.1007/978-3-642-32024-8_19).

Scott Martin and Carl Pollard. Hyperintensional dynamic semantics: Analyzing definiteness with enriched contexts. In Philippe de Groote and Mark-Jan Nederhof, editors, *Formal Grammar*, number 7395 in Lecture Notes in Computer Science, pages 114–129. Springer, 2012. doi:[10.1007/978-3-642-32024-8_8](https://doi.org/10.1007/978-3-642-32024-8_8).

Scott Martin and Michael White. Creating disjunctive logical forms from aligned sentences for grammar-based paraphrase generation. In *Proceedings of the Workshop on Monolingual Text-to-text Generation*. Association for Computational Linguistics, 2011.

Scott Martin, Rajakrishnan Rajkumar, and Michael White. Grammar engineering for CCG using Ant and XSLT. In *Proceedings of the Workshop on Software Engineering, Testing, and Quality Assurance for Natural Language Processing*. Association for Computational Linguistics, 2009.

D.J. Hovermale and Scott Martin. Developing an annotation scheme for English language learner spelling errors. In *Proceedings of the 5th Midwest Computational Linguistics Colloquium*, 2008.

Scott Martin. A proof-theoretic approach to French pronominal clitics. In *Proceedings of the 13th European Summer School in Logic, Language and Information Student Session*, 2008.

Michael White, Rajakrishnan Rajkumar, and Scott Martin. Towards broad coverage surface realization with CCG. In *Proceedings of the Workshop on Using Corpora for Natural Language Generation: Language Generation and Machine Translation*, 2007.

Fields of Study

Major Field: Linguistics.

Contents

Abstract	i
Acknowledgments	iv
Vita	viii
List of Tables	xiv
List of Figures	xv
1 Introduction	1
1.1 Thesis Outline	4
1.2 Some Conventions	8
2 The Empirical Domain	11
2.1 An Overview of Implicatures	12
2.2 Conventional Implicatures	20
2.2.1 Obligatory Speaker Commitments	21
2.2.1.1 Definite Anaphora	21
2.2.1.2 Iterative Adverbs	26
2.2.1.3 Possessives	28
2.2.1.4 Honorifics	31
2.2.2 Variable Speaker Commitment Status	32
2.2.2.1 Descriptive Content	32
2.2.2.2 Supplements and Expressives	35
2.3 Nonconventional Implicatures	39
2.3.1 Aspectuals	39
2.3.2 Achievements	41
2.3.3 Factives	43

2.4	Taking Stock	47
2.4.1	Felicity, Accommodation, and Variability	49
3	Curryesque Categorical Grammar	55
3.1	A Logic of Signs	56
3.1.1	Signs, Contexts, and the Lexicon	58
3.1.2	Grammar Rules	59
3.1.3	Axiomatizing the Pheno Logic	61
3.1.4	An Agnostic Semantic Theory	62
3.1.4.1	Two Notable Sects	67
3.2	A Small Fragment of English	69
3.2.1	Quantifier Scope Ambiguity	76
3.2.2	Peripheral and Medial Extraction	81
3.3	Summary	86
4	Dynamic Categorical Grammar	88
4.1	Motivation for a Dynamic Approach	89
4.2	A Compositional Dynamic Semantics	91
4.2.1	Preliminaries	91
4.2.1.1	Contexts, Contents, and Updates	92
4.2.2	Dynamicizing a Static Semantics	96
4.2.2.1	Dynamic Connectives and Quantifiers	99
4.2.2.2	Dynamic Entailment and Definiteness	107
4.3	A Dynamic Fragment	109
4.3.1	Dynamic Quantifier Scope	115
4.3.2	Donkey Anaphora	118
4.4	Grammar Rules for Modeling Discourse	123
4.5	Summary and Comparison with Other Theories	127
5	Anaphora	129
5.1	Extending Dynamic Categorical Grammar	131
5.1.1	Redefining the Dynamic Logic	132
5.2	Anaphora in a Two-Level Setting	134
5.2.1	Generalized Definiteness	136
5.2.1.1	Possessives	142
5.2.2	A Broader Notion of Contextual Felicity	145
5.2.3	The Iterative Adverb ‘Too’	149
5.3	Generalized Familiarity and Accessibility	153

5.3.1	Implementing Weak Familiarity	156
5.4	Determiner Strength	166
5.4.1	Strong and Weak Readings of Determiners	166
5.4.2	Kanazawa's Tonicity-Based Approach	168
5.4.3	Chierchia's Dynamic/E-Type Account	169
5.4.4	A Synthesized Dynamic Proposal	171
5.4.4.1	Weakening Dynamic Determiners	172
5.5	Summary and Comparison with Other Theories	180
6	Variable Conventional Implicatures	183
6.1	Supplements	184
6.1.1	Analyzing Supplements	185
6.1.1.1	Stacking	195
6.2	Expressives	199
6.3	The Interaction between Sense and Implicature	204
6.4	Variable Conventional Implicatures and Contextual Felicity	216
6.5	The Binding Problem Revisited	218
6.5.1	A Dynamic Analog of the Binding Problem?	219
6.5.2	An Alternative Analysis	222
6.5.3	The Problem of Quantified Supplements	223
6.5.4	Potential Solutions	225
6.6	Summary	229
7	Conclusions and Future Directions	231
7.1	Persistent Entailments in DyCG	233
7.2	Towards an Account of Anchoring	235
7.3	Computational Considerations	237
7.4	In Sum	240
A	Tensor-Implication Logic	243
A.1	Syntax	243
A.1.1	Proof Theory	244
A.2	Algebraic Semantics	245
B	Type Theory with Cartesian Products	251
B.1	Syntax	251
B.1.1	Term Identification and Reduction	255
B.1.2	Term Normalization	259
B.1.3	Proof Theory	260

B.2 Semantics	263
C Dependent Typing with Sums	267
C.1 An Enriched Typing Ontology	268
C.2 Syntax	269
C.2.1 Proof Theory	271
C.3 Applications	278
C.3.1 Von Neumann-style Natural Numbers	278
C.3.2 Vectors as n -ary Cartesian Products	280
C.3.3 Bit Vector Selection	285
C.4 Extending Pure Type Systems with Dependent Sums	287
Bibliography	290
Index of Citations	305

List of Tables

2.1 Summary characterization of implicatures 48

List of Figures

2.1	Taxonomic graph of implicatures.	49
3.1	Grammar rules of Curryesque Categorical Grammar	60
3.2	Full derivation of <i>Lance rode a bike</i>	75
4.1	Dynamic Categorical Grammar grammar rule for discourse .	123
6.1	Nonlogical rule converting generalized quantifiers to pred- icatives	188
A.1	Inference rule schemas for tensor-implication logic	244
B.1	Inference rule schemas for type theory	261
B.2	Structural rule schemas for type theory	262
C.1	Inference rule schemas for λP_{Σ}	272
C.2	Inference rules for pure type systems	288

Chapter 1

Introduction

Grice (1975) is well known for having investigated *conversational implicatures*, aspects of the contextual meanings of utterances that are not explicitly stated but are in some sense determined by the discourse context itself in combination with the interlocutors' goals and intentions. In fact, Grice is so well known for his work on conversational implicatures, the central focus of his 1975 paper, that the term *Gricean implicature* is sometimes found in the semantics literature, referring strictly to the conversational implicatures he explores.

But as Potts (2005) and others have recognized, there is much more to the Gricean story about implicature than just the conversational subcategory. Grice himself mostly just sketches a description of what I contend to be a very important subcategory of implicatures, the *conventional implicatures*. The following passage contains almost all of what Grice has to say regarding them:

In some cases the conventional meaning of the words used will determine what is implicated, besides helping to determine what is said. If I say (smugly), "He is an Englishman; he is, therefore, brave," I have certainly committed myself, by virtue of the meaning of my words, to its being the case that his being brave is a consequence of (follows from) his being an Englishman. But while I have said that he is an Englishman, and said that he is brave, I do not want to say that I have *said* [...] that it follows from his being an Englishman that he is brave, though I have certainly indicated, and so implicated, that this is so. I do not want to say that my utterance of this sentence

would be, *strictly speaking*, false should the consequence in question fail to hold. So *some* implicatures are conventional ... (Grice, 1975, page 167 in Martinich 2001)

He later follows with the following hint about how the conventional subcategory of implicatures is to be distinguished from the conversational one:

The presence of a conversational implicature must be capable of being worked out; for even if it can in fact be intuitively grasped, unless the intuition is replaceable by an argument, the implicature (if present at all) will not count as a *conversational* implicature; it will be a *conventional* implicature. (Grice, 1975, page 170 in Martinich 2001)

This thesis takes Grice's pointers about conventional implicatures as its starting point. Based first on the distinction between conventional and nonconventional implicatures, I develop a more fully fleshed out taxonomy of implicature that is rooted in Grice's (1975) work. This taxonomy adds the additional distinction of *anchoring to a point of view*, either the speaker's or otherwise, as discussed by Potts (2005), Amaral, Roberts, and Smith (2007), and Harris and Potts (2009). I make an effort to link this taxonomy with recent work on *projective meanings*, which escape the effects of semantic operators like negation and modals, due mainly to Roberts, Simons, Beaver, and Tonhauser (2009), Simons, Roberts, Beaver, and Tonhauser (2010), and Pollard and Smith (2011).

The result is that, starting from Grice's intuitions about implicatures, an empirical categorization of implicatures is possible in which the meaning types usually called *presupposition*, *anaphora*, and Potts's (2005) "CIs" can be subsumed under more general categories. This characterization sheds light on some old and important topics in semantics, such as the presuppositional nature (or lack thereof) of so-called *factive* verbs, aspectuals like *stop*, and achievements like *win*, and the role of *contextual felicity*, as conceived of by Stalnaker (1978), Lewis (1979), and Heim (1982), in discourse interpretation.

One upshot of this way of organizing implicatures is that there is no longer a place for the term *presupposition* separate from *anaphora*, as I argue at length that the nonanaphoric lexical items that are sometimes classified as presuppositions actually belong to a separate class, the *nonconventional implicatures*, which may or may not give rise to certain entailments. Another

is that the taxonomy I explore here presents a challenge to two of the central aspects of Potts's (2005) account: his idea that conventional implicatures should be forbidden from interacting with other types of content, and his claim that no lexical item can contribute to both the implicature of an expression and to its main point.

Still another implication of the generalized taxonomy I describe is that infelicity for anaphora, for example, when a definite is used without a proper antecedent, is modeled by exactly the same mechanism that captures infelicity for other instances of conventional implicatures, such as a nominal appositive whose content conflicts with entailments already present in the discourse context. This generalization of the mechanism by which (in)felicity is determined has implications for the notion of *presupposition accommodation* (Lewis, 1979): since many lexical items long thought to be presuppositional are reclassified as simply nonconventional implicatures, accommodation takes on a far more limited role than usually assumed. For lexical items that can take on a nonspeaker point of view, the threat of infelicity is also taken to inform the choice between anchoring to the speaker or to a nonspeaker perspective.

In the interest of making these empirical claims more concrete, I then give a compositional, dynamic semantics that makes formally explicit many of the factors distinguishing implicatures that I explore in the taxonomy I develop. My main aims for this formal semantics are twofold. First, I intend it to serve as a scientific theory in the sense of yielding predictions to which counterexamples could be provided, that is, a *falsifiable* theory. And second, its formal rigor is designed with the prospect of computational implementation in mind, with the goal of impacting a wide range of tasks in natural language processing and understanding, especially ones that involve or are influenced by aspects of discourse.

The dynamic semantics I pursue in this thesis is situated in the tradition of discourse semantics that runs through Karttunen (1974), Heim (1982), and Beaver (2001), but is also informed in large part by related formalisms, such as those due to Kamp (1981) and Groenendijk and Stokhof (1990, 1991) and their descendants: Chierchia (1995), Muskens (1994, 1996), Blackburn and Bos (1999), Bos (2003), de Groote (2006), and van Eijck and Unger (2010), among others. It uses a categorial syntax made up of two parts, an abstract syntactic component that models combinatorics, and a concrete syntax for modeling word order. In this way, it is similar to other frameworks that

adopt Curry's (1961) notion of the division between *phenogrammar* and *tectogrammar*, such as the work of de Groote (2001) or Muskens (2001, 2007).

I then extend this dynamic semantics equipped with a categorial syntax to a two-level semantics, following on ideas chiefly due to Karttunen and Peters (1979) but also developed in a different direction by Potts (2005). One level of the semantics captures the *sense* of expressions, their main point, and the other captures their implicatures. I show that this way of setting up a dynamic semantics allows a robust, explicit account of both anaphora and Pottsonian "CIs" like nominal appositives, parentheticals, nonrestrictive relatives, and expressives. This semantics has the desirable attribute of allowing anaphora to interact with other kinds of conventional implicatures in a very free manner, while still maintaining a separation between their contributions to the sense and to the implicature of expressions. I end this thesis by speculating about how nonconventional implicatures and anchoring to a point of view might be modeled in the formal theory I investigate for the conventional subclass of implicatures. I also offer promising pointers to ways in which the grammatical and semantic theory I develop in this thesis might be realized computationally.

In the rest of this chapter, I provide an outline of the chapters and appendices (§1.1), and in §1.2, I spell out some of the notational conventions used throughout the thesis.

1.1 Thesis Outline

The remainder of this thesis is organized as follows, with the main content given in chapters 2 through 7.

Chapter 2: The Empirical Domain makes a case for an extension of Grice's (1975) taxonomy of implicatures that more fully describes both the conventional and nonconventional subclasses. To help distinguish the senses of expressions from their implicatures, I discuss (§2.1) two diagnostics for **persistence**, a notion related to Simons et al.'s (2010) *projection*, and also make the case that point of view anchoring distinguishes between types of implicatures.

I then explore the taxonomy in more detail, discussing the subclass of conventional implicatures in §2.2. The descendants of this class are divided into

1. The obligatorily speaker-oriented (§2.2.1), such as definite anaphora (§2.2.1.1), iterative adverbs (§2.2.1.2), possessives (§2.2.1.3), and honorifics (§2.2.1.4), and
2. Those with variable orientation (§2.2.2), like the descriptive content implication (§2.2.2.1) and Potts’s “CIs” (§2.2.2.2).

In §2.3, I delve into the nonconventional implicatures besides the conversational implicatures: aspectuals (§2.3.1), achievements (§2.3.2), and factives (§2.3.3). Finally, §2.4 sums up the empirical characterization and discusses some of its implications for the way persistent meanings are usually conceived of, including the notions of *contextual felicity* and *accommodation*, and the variability in persistence of certain implicatures (§2.4.1).

Chapter 3: Curryesque Categorical Grammar begins the effort to explicitly state some of the empirical generalizations from chapter 2. The grammatical theory developed in this chapter embodies Curry’s (1961) idea that syntax is best modeled by separating the underlying combinatorics from the surface form. In §3.1, I formally devise the grammar, building it on three separate logics, two for the syntax and a third for a (static) semantics. Some foundational notions are defined in §3.1.1, and §3.1.2 gives the rules of the grammatical system. The concrete syntax, or *phenogrammar*, is axiomatized in §3.1.3, and the semantics itself (§3.1.4) is very general, built straightforwardly on a logic of propositions that has only the usual connectives and quantifiers, but without making a commitment on the way the notion of *possible worlds* should be implemented, following Plummer and Pollard (2012). Then, in §3.2, a minimal fragment demonstrating some of the grammar’s core capabilities is given, discussing several basic examples, quantifier scope ambiguity (§3.2.1), and extraction (§3.2.2). A summary of Curryesque Categorical Grammar is given in §3.3.

Chapter 4: Dynamic Categorical Grammar expands the grammar from chapter 3 to one with a dynamic, rather than a static, semantics, first motivating the dynamic approach in §4.1. The dynamic semantics itself is laid out in §4.2, and I show in §4.2.2 how it is essentially generated by a mapping from the generic static semantics in chapter 3. Next, §4.3 demonstrates how the new dynamic semantics works on some basic examples, and then shows that the treatment of quantifier scope from chapter

3 is maintained (§4.3.1). In §4.3.2, I show how the dynamic semantics can handle the familiar cases of *donkey anaphora* that motivated the early dynamic theories of Kamp (1981) and Heim (1982). The grammar rules of Curryesque Categorical Grammar are extended for modeling discourse in §4.4, along with a straightforward example of how they are used to put a discourse together from its component utterances. Lastly, the dynamic semantics is compared with some other approaches in §4.5.

Chapter 5: Anaphora deals with the conventional implicatures associated with anaphora. First (§5.1) the dynamic theory from chapter 4 is extended to have two meaning levels rather than a single one, with a level for the sense of an expression and a level for any implicatures it gives rise to, following Karttunen and Peters (1979).

I then offer a novel modeling of anaphora in the two-level setting in §5.2, with a generalized notion of definiteness as implicature (5.2.1), an account of possessives (5.2.1.1), and a generalized notion of felicity in context (5.2.2). I also offer an account of the iterative adverb *too* in §5.2.3. An implementation of Roberts’s (2003) *weak familiarity* is explored in §5.3. Then, in §5.4, a treatment of the *weak* and *strong* readings of determiners in a dynamic setting that blends some insights by Kanazawa (1994) and Chierchia (1995). Kanazawa’s approach is discussed in §5.4.2, and Chierchia’s in §5.4.3. Then §5.4.4 proposes a way of thinking about determiner strength that synthesizes some aspects of the two approaches, along with a way of formally redefining the Dynamic Categorical Grammar determiners (§5.4.4.1). To wrap up, §5.5 reflects on my approach to anaphora and compares it to some other theories.

Chapter 6: Variable Conventional Implicatures demonstrates how the two-level extensions to the dynamic semantics explored in chapter 5 are general enough to also capture the implicatures associated with Potts’s (2005) “CIs,” starting with supplements (§6.1). Beginning with an analysis of supplements in §6.1.1, I show that the theory developed in this chapter can handle *stacked supplements* in §6.1.1.1, which an alternative to Potts’s theory due to Nouwen (2007) is incapable of. Expressives are then treated in §6.2. I next show how this theory correctly allows sense content to interact with implicature content in §6.3, which Potts explicitly disallows.

In §6.4, I discuss how the formal definition of contextual felicity given in §5.2.2 in the context of an account of anaphora also generalizes to the case of the conventional implicatures for which nonspeaker point of view anchoring is possible, such as supplements and expressives. The *binding problem* of Karttunen and Peters (1979) is discussed in §6.5, and in §6.5.1, I show how no analog of the binding problem arises for this two-level dynamic semantics. An alternative analysis is then proposed in §6.5.2, in which the problem also does not arise. In §6.5.3, I turn to the problem of supplements with quantificational anchors, offering some potential solutions in §6.5.4. A summary of the chapter is given in §6.6.

Chapter 7: Conclusions and Future Directions wraps up the thesis, revisiting its main themes. I sketch an account of the nonconventional implicatures associated with *persistent entailments* (discussed in chapter 2) in §7.1, and then a sketch of an account of anchoring to a point of view is given in §7.2. In §7.3, some promising possibilities are discussed for implementing the dynamic theory developed here as software for practical applications to natural language processing, and some related applications are discussed (Blackburn and Bos 1999, Bos 2003, van Eijck and Unger 2010). Finally, §7.4 summarizes the thesis’s contributions.

I then provide three appendices (A, B, and C) as technical background that is not centrally relevant to the thesis itself.

Appendix A: Tensor-Implication Logic describes a subsystem of Girard’s (1987) linear logic that uses only the tensor product \otimes , its unit **1**, and its residual \multimap . The syntax of this system is given in §A.1, along with a natural deduction proof theory (§A.1.1). An algebraic semantics following Troelstra (1992) and de Paiva (2002) is provided in §A.2.

Appendix B: Type Theory with Cartesian Products lays out a version of Church’s (1940) *simple theory of types*, as elaborated by Henkin (1950, 1963) and Andrews (2002). It is extended straightforwardly with a unit type **1** and product types built with the constructor \times in addition to the usual implicative types built with \rightarrow . Its syntax and proof theory is given in §B.1, along with a discussion of term reduction and normalization (§B.1.1

and §B.1.2) and a proof theory (§B.1.3). An extension to Henkin’s (1950) semantics for the case of product types is given in §B.2.

Appendix C: Dependent Typing with Sums extends the type theory in appendix B to a system using *dependent types* based on the system λP discussed by Barendregt (1991, 1992), and extended with *dependent sum types*, which generalize the product types of simple type theory, following Aspinall and Hofmann (2005). The enhancements to the type system are introduced in §C.1, and then the extended syntax is discussed in §C.2, along with an elaborated proof theory (§C.2.1). Several example applications are then given in §C.3: an implementation of a type of natural numbers in the style of Von Neumann (§C.3.1), type constrained vectors (§C.3.2), and the type of a selection function for bit vectors (§C.3.3). Finally, I describe an adaptation of the extended dependent type system to the setting of *pure type systems* (Barendregt, 1992) in §C.4.

A bibliography is given, as usual. For convenience, I also provide an index of citations after the bibliography. I have made an effort to make the references in this thesis a useful tool for future research, providing *digital object identifiers* (DOIs) in the bibliography where available.

1.2 Some Conventions

Special typographic conventions are observed to signal that terminology is being used. Specifically, when a term is being used in this thesis for the first time that does not originate elsewhere, or is being used in a different way from previous usage, it is set in **boldface**. Terms that are well known, or originate from another source, are set in *italics*, and usually have an accompanying reference, especially if it is the first time the term occurs.

Numbered linguistic examples are given in the text in the following form:

(2.5) Lance, a cyclist, is from Texas.

The number 2.5 signals that this is the fifth example in the second chapter. Because this thesis uses a large number of equations, examples and equations are numbered using the same numbering scheme and sequence. This means that, for example, no equation can be numbered 2.5, because

an example has already used that number, and vice versa. Unnumbered examples or pieces of linguistic data that occur inline in the text are set in *italics*. Judgments about examples are expressed by prefixing symbols to the example in question, according to the following scheme:

* syntactic ungrammaticality

infelicity in the given context, or if no context is given, infelicity in any context

? the example is questionable in terms of felicity

Examples are sometimes accompanied by a context, either numbered or unnumbered, that signals the discourse context in which the example in question occurs for the purposes of the discussion. I also sometimes use subscripts in examples to disambiguate the intended reading, as in

(2.3) A woman_i walked in, and then she_i bought a ticket,

where the subscripts indicate that the pronoun *she* should be interpreted as being anteceded by *A woman*.

Mathematical definitions, lemmas, propositions, theorems, and examples are numbered using the same scheme as for linguistic examples and equations, for example ‘Theorem 3.13,’ except that they use their own numbering sequence. Various shorthands for the different mathematical formalisms used in this thesis are given where those formalisms are defined. For linear logic and type theory, they are discussed in the appendices; for the formal semantic theory that is a central subject of the thesis, they are discussed mainly in chapters 3 and 4.

References to an author are cited as ‘Higginbotham (2013),’ with the year in parentheses following the author’s name. If the work itself is being referenced, rather than the author, the citation takes the form ‘Higginbotham 2013,’ with no parentheses. If a work has three or more authors, they are all listed the first time the work is referenced, for example, ‘Higginbotham, Snicklefritz, and Snodgrass 2013.’ Subsequent references to a work with more than two authors abbreviate the author list, as in ‘Higginbotham et al. 2013.’ References to sections are abbreviated from ‘section 9.3’ to ‘§9.3.’ The format of references to axioms, examples, equations, and proofs is to write the number in parentheses, for example, ‘(3.11)’. References to definitions, lemmas, propositions, and theorems use only the number, without parentheses, as in ‘lemma 4.11.’ In the digital version of

this thesis, references to appendices, axioms, bibliography entries, chapters, contexts, definitions, equations, examples, figures, lemmas, pages, proofs, propositions, sections, tables, and theorems are hyperlinked, pointing to the item they refer to. Also, digital object identifiers in the bibliography are hyperlinked to a web service that resolves the object they point to.

Chapter 2

The Empirical Domain

Much has been written in the semantics literature on those components of the meaning of an utterance that are not central to its main point. In this chapter, I bring into focus some subclasses of these non-main-point implications, which are called *implicatures*, following Grice (1975). These implications have been variously described as *backgrounded* (Roberts et al., 2009), as *not-at-issue* (Potts, 2005; Roberts et al., 2009; Simons et al., 2010), as *implicatures* (Roberts, 2012e), and, by Pollard and Smith (2011), as the part of utterance meaning that is not *proffered* for acceptance or rejection, in the sense of Roberts (1996, 2012c). As I discuss in detail below, the implicatures form a more general class of meanings that includes what are usually called *presuppositions*, but implicatures in general do not require that their truth-conditional content be an entailment of the interlocutors' mutual knowledge prior to the utterance in which they occur.

One subclass of implicatures, the **conventional implicatures**, in turn subsumes two meaning classes that are traditionally categorized separately, and are distinguished by their status as *speaker commitments*, propositions whose truth the speaker is committed to in the context of interpretation. The class of *obligatorily speaker-anchored conventional implicatures* includes phenomena usually referred to as *anaphora*, while the class of *variable conventional implicatures*, which are not necessarily speaker-anchored, includes Potts's (2005) "CIs." Here I reclaim the term *conventional implicature* from Potts, generalizing it to the sense used by Grice (1975) and Karttunen and Peters (1979), which also contains the subclass of anaphora.

Another subclass, the **nonconventional implicatures**, is characterized by the property that the ability of its members to persist as mutually ac-

cepted information in a discourse is variable and contextually conditioned. Instances of nonconventional implicatures include the implications sometimes associated with verbs that are usually described as *factive*, aspectuals, and achievements with a preparatory phase.

Before launching into a discussion of my classification of implicatures and how it should be accounted for, I first give some general background on the topic.

2.1 An Overview of Implicatures

Frege's (1892) *Kepler* example is perhaps the first illustration of the phenomenon of content bearing a **persistent implication**, and it is certainly among the best known.

$$(2.1) \quad \text{Kepler} \left\{ \begin{array}{l} \text{died} \\ \text{did not die} \end{array} \right\} \text{ in misery.}$$

(Frege, 1892)

Frege noted that for both the negated and nonnegated variants of (2.1), "there is a presupposition that the name 'Kepler' designates something." In more modern parlance: by the use of the name *Kepler*, the speaker is implicating that the discourse context contains an *antecedent* discourse referent that is entailed to have the property of being named 'Kepler.'

This simple example demonstrates the two most basic facts about persistent implications. The first fact is that natural language utterances often have more than a single associated implication. Frege's example bears (at least) an implication about Kepler's state of mind upon his death as well as the implication that *Kepler* has a suitable antecedent in the context of interpretation. The second fact that (2.1) illustrates is that not all utterance implications are interpreted in the same way: the implication that Kepler was in a miserable state at the end of his life is sensitive to negation, while the implication that there is an antecedent corresponding to *Kepler* is not, and so it is said to *persist* through the negation.

I will call the main point of an utterance, the portion targeted by semantic operators like negation, its *sense*, and the portion not targeted by operators its *implicature*. Thus senses, as defined here, correspond to *what is said* in Grice's (1957) terminology. Simons et al. (2010) offer a compelling explanation of the sense/implicature distinction in terms of the *at-issue* nature of the content in question. Here I examine the distinction between

these two meaning types in great detail, but leave open the question of explaining the divide.

A standard empirical diagnostic for identifying presupposition persistence is the *family of sentences* tests. The name “family of sentences” is from Chierchia and McConnell-Ginet 1990, but as Roberts et al. (2009) note, the tests themselves have been in use at least since Langendoen and Savin 1971. In these diagnostics, a sentence that is suspected of containing a subexpression with a *persistent implication* (called the *triggering expression* or just the *trigger*) is altered so that the suspected trigger is embedded in the scope of an **entailment-modifying operator**.¹ Importantly, these tests apply to *atomic* sentences, which contain no semantic operators or attitude predicates, in order to provide a more clear diagnostic of persistence.

Persistence Diagnostic 2.1 (Family of Sentences). An implication *m* of an atomic sentence *s* associated with a potential trigger *t* *persists* if *m* remains an utterance implication of minimal variants of *s* with *t* embedded in the following contexts:

- Negation.
- Interrogation.
- A modal.
- The antecedent of a conditional.

If *c* is a family of sentences context and *m* persists when embedded in *c*, then *m persists through c in s*.

To demonstrate, the persistence tests are applied to Frege’s Kepler example in (2.2):

- (2.2)
- a. Kepler did not die in misery.
 - b. Did Kepler die in misery?
 - c. Maybe Kepler died in misery.
 - d. If Kepler died in misery, pursuing an academic career seems foolish.

¹Elsewhere in the literature, the term *entailment-canceling operator* is used with the same sense as here; I prefer the term *entailment-modifying* because the resulting entailments are based on the entailments in the scope of the operator.

The procedure for applying these tests to the suspected trigger *Kepler* in (2.1) is to examine each of the four associated minimal variants in (2.2), asking for each whether the implication associated with *Kepler* that it must designate something is retained. Since this requirement holds in each case, *Kepler* is deemed to persist by the family of sentences tests.

The implications associated with pronominal anaphora exhibit similar behavior under the persistence tests as do those associated with proper names, as demonstrated in the following example.

(2.3) A woman_{*i*} walked in, and then she_{*i*} bought a ticket.

Applying the four tests to (2.3) show that the pronoun *she* persists in each of them, since on the intended reading,² the implication that *she* has a suitable antecedent survives for every test case:

- (2.4) A woman walked in.
- a. She didn't buy a ticket.
 - b. Did she buy a ticket?
 - c. Maybe she bought a ticket.
 - d. If she bought a ticket, the train must be leaving soon.

Note that the relevant implication remains in effect whether or not (2.3) is read so that *A woman* is interpreted as the antecedent of *she*, and so the persistent behavior of *she* is not contingent on *where* its antecedent is found, just on whether one is available at all.

The family of sentences regime was first designed for detecting persistence associated with presuppositions, defined as felicity constraints on the context of interpretation (Karttunen, 1974; Stalnaker, 1978; Lewis, 1979; Heim, 1983b). But it also conclusively diagnoses persistence for (2.5), an instance of a CI in the sense of Potts (2005). In (2.5), the use of the nominal appositive *a cyclist*, a *supplement* in Potts's (2005) terminology, gives rise to the implication that Lance is one.

(2.5) Lance, a cyclist, is from Texas.

Applying the persistence diagnostic to (2.5) shows that the appositive persists:

²As I discuss in §1.2, examples in this thesis follow the convention, used in (2.3), that anaphoric relations are signaled by subscripting the anaphor and its antecedent with the same index variable, here *i*.

- (2.6) a. Lance, a cyclist, is not from Texas.
 b. Is Lance, a cyclist, from Texas?
 c. Maybe Lance, a cyclist, is from Texas.
 d. If Lance, a cyclist, is from Texas, I'll bet he lives in Austin.

In each of the four cases in (2.6), the implication that Lance is a cyclist survives, and the appositive *a cyclist* is considered persistent.

The family of sentences tests diagnose persistence because they examine whether an implication survives embedding a trigger beneath operators that modify entailments. However, they are not useful in diagnosing persistence when no entailment-modifying operator is present. While the tests in (2.6) show conclusively that *a cyclist* persists in each of the family of sentences contexts, they have nothing to say about the unembedded case in (2.5). All the persistence tests reveal about (2.5) itself is that the appositive *would* persist if the sentence was altered so that the appositive is embedded under an entailment-modifying operator. And so persistence, as defined here, refers to the potential to persist in case the implication is embedded. To diagnose the capability to persist even when a suspected trigger is not embedded beneath any operator, a separate diagnostic is needed.

Persistence Diagnostic 2.2 (Direct Denial). An implication *m* of an atomic sentence *s* associated with a potential trigger *t* persists past *t* if *m* survives the immediate, direct denial of an utterance of *s*.

Because it asks whether an implication survives direct denial, there is a sense in which this diagnostic is a variation on the family of sentences variant using negation in which the negation is external to the sentence under test. It successfully diagnoses persistence for both (2.3) and (2.5) as follows.

- (2.7) a. A woman walked in, and then she bought a ticket.
 b. No, she didn't. She asked for a train schedule.
- (2.8) a. Lance, a cyclist, is from Texas.
 b. No, that's not true—he's actually from northeast Arkansas.

As the example of the direct denial diagnostic in (2.7) shows, directly denying (2.3) has no effect on the implication associated with *she* that a suitable antecedent is available. And for (2.8), the implication that Lance is a cyclist persists even if (2.5) is directly denied.

Direct denial is useful as a diagnostic because implications that have a tendency to persist must be denied using other means, namely via *indirect rejection* (Roberts et al., 2009).

- (2.9) Lance, a cyclist, is from Texas.
- a. # No, he's not. He's actually a radon mitigation specialist.
 - b. Hold on a second. Lance might be from Texas, but he's definitely not a cyclist.

In (2.9a), the attempt to negate the implication associated with *a cyclist* results in infelicity, whereas the use of negation to deny that Lance is from Texas is perfectly acceptable, as (2.8) shows. To deny the content of the appositive, an indirect rejection like (2.9b) must be used instead.

However, consideration of a broader range of examples exposes a major limitation of both the family of sentences tests and the direct denial test for diagnosing persistent content more generally. As stated, neither diagnostic applies to *utterances*, but instead to *sentences*. Since context is not accounted for, both testing regimes make indeterminate diagnoses when a suspected trigger has an implication that persists in some contexts but not in others. For example, the aspectual verb *stop* has been taken to presuppose that at some past time its complement property applied to its subject, that is, as imposing a *felicity condition* that prior context must entail that its subject has the relevant property.

- (2.10) Kim stopped smoking.

In (2.10), the suspected persistent implication, called the *pre-state implication*, is that Kim used to smoke.

Applying the persistence tests to (2.10) suggests that this implication does persist in some cases but not in others:

- (2.11)
- a. Kim didn't stop smoking.
 - b. Did Kim stop smoking?
 - c. Maybe Kim stopped smoking.
 - d. If Kim stopped smoking, then I won't ask her for a cigarette.

The inconclusive results are best seen by considering how the persistent implication associated with *stop* in (2.10) behaves in the following contexts.

Context (2.12): It is mutual knowledge to the interlocutors that Kim is well known to be a lifelong smoker.

Context (2.13): It is unknown to the addressee whether Kim has ever smoked before.

Context (2.12) entails the purported pre-state implication that Kim smoked at some prior time. But note that, since (2.10) is acceptable in context (2.13), it cannot be characterized as imposing a felicity condition on the discourse context because Kim's status as a smoker is unknown to the addressee. Interpreted in context (2.13), the pre-state implication does not hold for any of the four tests. For (2.11a), it may be the case that Kim did not stop smoking simply because she never started. A similar situation holds for (2.11b): the answer may be *No, she never started*. And neither of (2.11c) or (2.11d) require Kim to have previously smoked, since both express hypotheticals. In context (2.12), the utterance of (2.11b) or (2.11c) might, for example, be taken as speculation about the reason Kim seems so out of sorts.

As the variability observed for the variant in (2.11a) implies, the results are parallel for the direct denial diagnostic.

- (2.14) a. Kim stopped smoking.
 b. No, that's not true.

The denial of (2.10) in (2.14) negates the proposition that Kim stopped smoking, but makes an indeterminate diagnosis with respect to *stop*'s pre-state implication. In context (2.12), the pre-state implication that Kim used to smoke persists, but in context (2.13), it does not.

So both the family of sentences and the direct denial testing regimes clearly diagnose persistence in the case of proper names, pronouns, and Potts's (2005) CIs, but are unable to provide a reliable diagnosis for the aspectual *stop*, because the persistence of the pre-state implication additionally depends on the discourse context. In his early work on presupposition persistence, Karttunen invokes persistence variability to distinguish between the *factive* and *semi-factive* verbs (1971), and also notes *stop*'s persistence variability in certain contexts (1973). More recently, Abusch (2010) discusses this contextually-conditioned variability observed for certain implicature triggers, which she calls *soft triggers*.

Both persistence tests are so useful for diagnosing persistence because they identify triggers as not being targeted by operators that modify entailments. Together, they help distinguish between persistent content, which is not targeted by entailment-modifying operators because it is not intended

by the utterance's author to figure into the compositional calculation of its meaning, and the sense content whose entailments are modified by these operators. Even when the diagnostics are inconclusive, they still provide the information that the trigger in question *can* persist in some circumstances, just not in all. So the term *persistent* refers to implications that have the ability to persist.

For the purposes of this thesis, I define *persistent implications* as those implications that are persistent as determined by either of the persistence diagnostics. The complement of this class, the *sense implications*, are those implications that are not found to persist by the diagnostics, that is, the implications that are targeted by operators. This distinction captures the fact that conversational participants actively place content in the background by choosing to construct utterances in a certain way, as opposed to other alternatives. Importantly, as the variation observed for (2.11) in different contexts shows, an implication's being persistent does not necessarily imply that it persists in a given utterance in a given context of interpretation. A persistent implication may persist only in certain contexts or only past certain operators.

The differing results observed for the persistence diagnostics with respect to certain triggers also suggest an empirical distinction. Following Pollard and Smith (2011), I note that examples (2.1), (2.3) and (2.5), for which the persistence diagnostics are conclusive, share the common property that their implicature content is conventionally signaled. This contrasts with example (2.10), where the persistence of the pre-state implication is *not* part of *stop*'s conventional meaning, and for which the diagnostics give a different verdict depending on the context.

A separate distinction between implicatures that cross-cuts the conventional/nonconventional divide has to do with whether or not they are required to be speaker commitments. Amaral et al. (2007) first pointed out that CIs are not invariably speaker-anchored, contrary to the claims in Potts 2005, and Harris and Potts (2009) give compelling evidence that CIs can indeed be anchored to nonspeaker points of view. Unfortunately, as (2.15) shows, giving a simple diagnostic for anchoring is difficult because detecting nonspeaker anchoring can necessitate the construction of a rich context of interpretation.

- (2.15) Joan believes that her chip, which she had installed last month, has a twelve year guarantee.
(Amaral et al., 2007, example 27)

In some contexts, the nonrestrictive relative clause *which she had installed last month* is interpreted as being anchored to the point of view of the speaker. But consider (2.15) in the following context:

Context (2.16): Joan is crazy. She's hallucinating that some geniuses in Silicon Valley have invented a new brain chip that's been installed in her left temporal lobe and permits her to speak any of a number of languages she's never studied.
(Amaral et al., 2007, example 25)

Interpreted in context (2.16), the proposition that Joan had her chip installed last month is part of the belief attributed to Joan, and not the speaker.

An example of a CI that must be a speaker commitment is the nominal appositive *a cyclist* in (2.17).

- (2.17) Dana believes that Lance, a cyclist, is not a cyclist.

In (2.17), the appositive is interpreted as anchored to the speaker's point of view rather than to Dana's because otherwise it would attribute inconsistent beliefs to Dana. Compare this anchoring variability for CIs with the persistent implication associated with *Kepler* in (2.1), above. In Frege's *Kepler* example, the implication that Kepler has an antecedent in prior discourse is necessarily a speaker commitment about the state of the discourse context. This commitment is simply built into the conventional meaning of the proper name *Kepler*.

Finally, the phenomenon of *modal subordination* (Roberts, 1989) demonstrates why the distinction between implicatures based on their status as speaker commitments cannot be made simply on the basis of their (speaker or nonspeaker) anchoring.

- (2.18) If a woman_i walks in and she_i buys a ticket_j, it_j's probably for the evening train to Cleveland.
a. She_i'll likely be a daily commuter on this line.
b. # It_j's in her pocket.
- (2.19) Kim sure seems on edge today.

- a. Maybe Kim quit smoking. Or maybe she never even started smoking at all, and she just hasn't yet had her coffee today.
- b. # Kim quit smoking. She never started smoking at all, she just hasn't yet had her coffee today.

On the intended reading, the use of the pronoun *It* in (2.18b) is infelicitous because its antecedent is made inaccessible: in Roberts's (1989) terminology, *a ticket* occurs as part of an utterance in *nonfactual mood*, while the pronoun that purportedly references it is uttered in *factual mood*. The pronoun *She* in (2.18a) is felicitous, because the second modal can be taken to have a domain restriction that makes an antecedent for the pronoun available. Both pronouns, however, bear an associated speaker commitment to the effect that a suitable antecedent can be found in the discourse context.

This behavior shows a strong contrast with (2.19). In (2.19a), the pre-state implication of *stop* that Kim used to smoke is clearly not a speaker commitment because it is uttered in nonfactual mood. Factual mood is used in (2.19b), with the result that the speaker is necessarily committed to *stop*'s pre-state implication, which is simply an entailment. The infelicity of (2.19b) then results from the fact that the entailment of the pre-state of the first sentence contradicts the denial in the first clause of the second. The point is that, since both (2.19a) and (2.19b) are anchored to the speaker's point of view, felicity hinges on whether the speaker is committed to an implication and not to which point of view the implication is anchored.

In the following two sections, I explore in greater detail the heterogeneity between implicatures evidenced by the persistence tests and their status as speaker commitments. Then, in §2.4, I draw some taxonomic distinctions between persistent content based on persistence variability and the conditions under which they result in infelicity.

2.2 Conventional Implicatures

The class of **conventional implicatures**, which Pollard and Smith (2011) call *warrants*, comprises all those persistent implications that are conventionally signaled as part of the meaning of their corresponding trigger. Among others, this class contains certain implications associated with anaphora (though not all, as I explain below in §2.2.2) and honorifics, as well as those associated with nominal appositives, nonrestrictive relative clauses, expressives, parentheticals, all instances of "CIs" in Potts's (2005)

sense. Though anaphora and CIs are often treated as both empirically and conceptually separate (see, for example, Potts 2005; Roberts et al. 2009; Tonhauser, Beaver, Roberts, and Simons 2013, among others), I argue that their associated persistent implications have more in common than not.

2.2.1 Obligatory Speaker Commitments

2.2.1.1 Definite Anaphora

The phenomenon of (definite) anaphora occurs when a *discourse referent* (Karttunen, 1976) is introduced and interpreted as the antecedent to a later definite: a proper name, demonstrative, pronoun, possessive, or a noun phrase whose determiner is *the*. Heim (1982, chapter 3) describes this behavior in the form of a condition on the felicitous use of an (in)definite noun phrase, called the *Extended Novelty-Familiarity-Condition*. This condition requires that when a definite is used, its antecedent must be *familiar* (already available in prior discourse). By contrast, the discourse referent introduced by the use of an indefinite comes with the condition of being *novel* (not familiar).

- (2.20) a. A donkey_i walked into a bar. It_i brayed.
 b. # It_i brayed. A donkey_i walked into a bar.

The example in (2.20b) is simply (2.20a) with the order of its sentences reversed, and yet the use of the definite *it* is infelicitous in (2.20b) because no discourse referent is available yet to serve at its antecedent. Example (2.20b) is also unacceptable for another reason: on the intended reading, the indefinite *A donkey* attempts to introduce a discourse referent that has already been referenced by *it*. Thus for Heim, these stipulations for the use of definites and indefinites constitute a felicity condition on their context of interpretation.

Carl Pollard (personal communication) points out the existence of examples like the following.

- (2.21) A donkey_i walked into a bar. It_i brayed. So $\left\{ \begin{array}{l} \text{It}_i \\ \text{a donkey} \end{array} \right\}$ brayed.
 Big deal.

Example (2.21) seems to present a counterexample to the novelty condition, because the indefinite *a donkey* is apparently used to refer to a previously mentioned donkey. I would argue that there actually is a difference between

the variant of (2.21) with It_i and the one with the indefinite. In the It variant, it is more clear that what the speaker is stating to be a big deal is the fact that the donkey mentioned before brayed. But for the variant with the indefinite, it seems that what the speaker takes to be a big deal is the fact that *some* donkey brayed, and the fact that the previously mentioned one, in particular, brayed is irrelevant.

The novelty of indefinites is an instance of a conventional implicature, because novelty is a part of the conventional meaning of an indefinite but it is persistent. As far as I am aware, the empirical characterization I give here is the first to classify novelty for indefinites as a conventionally signaled implication. The persistence of this **novelty implication** is illustrated in the following tests.

- (2.22) A donkey walked into a bar.
- a. A donkey didn't bray.
 - b. Did a donkey bray?
 - c. Maybe a donkey brayed.
 - d. If a donkey brayed, it probably wants bourbon.

In each of the family of sentence variants in (2.22), the second instance of the indefinite *a donkey* cannot be interpreted as introducing the same discourse referent as the one introduced by the prelude *A donkey walked into a bar*.

Note that the novelty implication is separate from the lifespan of the discourse referent introduced by an indefinite. It has been observed at least since Karttunen (1976), and formalized as early as Kamp (1981) and Heim (1982), that discourse referents introduced in certain contexts are only accessible within those contexts.

- (2.23) $\left\{ \begin{array}{c} A \\ \# \text{No} \end{array} \right\}$ donkey_{*i*} walked into a bar. It_{*i*} brayed.
- (2.24) If a donkey_{*i*} walks into a bar, it_{*i*} says hello to everyone before sitting down. # It_{*i*} brayed.
- (2.25) A donkey_{*i*} walked into a bar. Every farmer bought a donkey_{#*i*} a drink.

Basic accessibility is illustrated in (2.23), in which the negated variant with *No* creates a context that bounds the accessibility of the discourse referent introduced by *No donkey*. A more complex case is given in (2.24): the

discourse referent introduced by *a donkey_i* in the antecedent of the *if*-clause is available to the pronoun in its consequent, but not in the subsequent utterance. But (2.25) shows that even discourse referents within limited accessibility contexts (here, *Every*) still bear the novelty implication: the donkey that every farmer bought a drink for cannot be construed as the same donkey that walked into a bar.

The persistence diagnostics can be used to detect the persistent implication associated with *it* in a way similar to that used for (2.3). Applying the family of sentences tests to the second sentence of (2.20a) gives (2.26); the direct denial test for (2.20a) is illustrated in (2.27).

- (2.26) A donkey walked into a bar.
- a. It didn't bray.
 - b. Did it bray?
 - c. Maybe it brayed.
 - d. If it brayed, it probably wants bourbon.
- (2.27) a. A donkey walked into a bar. It brayed.
- b. No it didn't. Donkeys don't behave that way in bars.

In each case, what persists from the variants in (2.26) is the **familiarity implication** (Heim, 1982) that an antecedent is available in prior discourse. Similarly, in (2.27), the familiarity implication of a suitable antecedent for *It* survives direct denial. The familiarity implication is not only persistent, but arises as part of the conventional meaning of definites. Note that the familiarity implication, as defined here, is not a commitment to the existence of an entity in the world, but to the availability of discourse referent in prior context. As Tonhauser et al. (2013) discuss, the existence commitment can take on an embedded point of view, but since it is a commitment about the state of the discourse context, the familiarity implication cannot—it must be anchored to the speaker.

Example (2.28) clarifies the additional complication for familiarity, discussed above in connection with (2.23), that the familiar discourse referent must be available in the local context of interpretation.

- (2.28) $\left\{ \begin{array}{c} A \\ \# \text{ No} \end{array} \right\}$ farmer owned a donkey_i. It_i brayed.

The pronoun *It* in (2.28) bears the familiarity implication that its antecedent is available in prior context. In the indefinite variant, this implication

is satisfied, because the discourse referent introduced by *a donkey* is still accessible. But the variant with *No* provides no such discourse referent. For the *No* case, the accessibility of the discourse referent corresponding to *a donkey* is closed off, and so no accessible antecedent for *It* is available. I have more to say about accessibility in chapter 4, where I discuss a formal account of anaphora in dynamic semantics.

As evidence that the familiarity implication persists not just for pronouns but for definites more generally, consider (2.29).

(2.29) The donkey brayed.

The persistence tests show that the definite *The donkey* bears the same persistent familiarity implication as observed for the pronoun *it* in (2.20).

(2.30) A donkey walked into a bar.

- a. The donkey didn't bray.
- b. Did the donkey bray?
- c. Perhaps the donkey brayed.
- d. If the donkey brayed, the horse probably just arrived.

(2.31) a. A donkey walked into a bar. The donkey brayed.

- b. No, it couldn't have. It would've gotten kicked out for that.

Here again, the testing regime shows that the implication that an antecedent exists for *The donkey* survives both embedding and denial, and so it is persistent.

Heim (1982) and Roberts (2003), among others, maintain that for an addressee to be able to retrieve the intended antecedent of a definite, it must bear the definite's descriptive content in the sense that the context of interpretation must entail that the antecedent has the property associated with its descriptive content. More recently, it has been pointed out that this requirement is too strong (Pollard and Smith, 2011; Roberts, 2012d; Tonhauser et al., 2013). Definites give rise to a persistent implication that the context is consistent with the antecedent having the definite's descriptive content, but this implication is not a presupposition imposing a felicity constraint on the discourse context (see the discussion of (2.55), below). This *descriptive content implication* that the antecedent has suitable descriptive content is actually a separate implication from the familiarity implication, as I discuss below in connection with (2.50) and (2.51).

However, the descriptive content implication is insufficiently fine grained to rule out cases where the requisite property is predicated of *more than one* discourse referent:

(2.32) A farmer bought a brown donkey_{*i*} and a gray donkey_{*j*}.

$$\left\{ \begin{array}{l} \# \text{ It}_{i/j} \\ \# \text{ The donkey}_{i/j} \\ \text{The gray donkey}_{\#i/j} \end{array} \right\} \text{ snorted.}$$

(Martin and Pollard, 2012a, example N)

In (2.32), neither of the definite noun phrases *it* or *the donkey* are specific enough to disambiguate which of the brown or gray donkey is supposed to be associated with the antecedent.

Roberts (2003) calls this additional requirement the presupposition of *informational uniqueness* because the antecedent must be the *only* one in the context to bear the descriptive content associated with the definite it antecedes, a stronger requirement than Heim's familiarity. Stone and Webber (1998) contend that the requirement is even more general: that the antecedent must be the most plausible one with the descriptive content in question. Roberts's informational uniqueness is distinguished from Russell's (1905) referential uniqueness because it requires only that the antecedent be unique among the discourse referents *in the context of interpretation*, not a unique semantic entity in the world, to bear the relevant property. Finally, as Roberts (2005) argues, the intended antecedent must be salient to the addressee to a degree determined in part by the richness of its descriptive content.

In more recent work, Roberts (2010, 2012d) characterizes the familiarity, informational uniqueness, and salience implications associated with the use of a definite as being derivable from a notion of *retrievability*, which is in turn based on more general ideas about the nature of meaning and intention in cooperative conversation (Grice, 1957, 1975). Following Roberts, I adopt the term **retrievability implication** to refer to the combined implications of familiarity, informational uniqueness, and salience. The retrievability implication that the use of a definite gives rise to is the implication that a familiar, salient discourse referent is available that is informationally unique in the sense that the relevant descriptive content is not in conflict with what the context entails for the referent in question.

Example (2.33) serves as evidence that the retrievability implication is a speaker commitment.

- (2.33) Kim doesn't know that there's a donkey_i over there. She doesn't hear it_i braying.

Here, a discourse referent is introduced into the context of interpretation by *a donkey* that, on the intended reading, is later used as the antecedent of the pronoun *it*. But since (2.33) makes it clear that Kim is unaware the donkey in question exists, the pronoun *it* must be interpreted as a commitment by the speaker that its antecedent is retrievable given the current context of interpretation. The retrievability implication is necessarily a speaker commitment because retrievability is an implication about the discourse context itself, which is constructed as part of the cooperative conversational enterprise between the speaker and the addressees.

2.2.1.2 Iterative Adverbs

Iterative adverbs like *too* and *again* also have persistent implications that are similar to those associated with definite anaphora.

- (2.34) a. **Sam** is having dinner in New York tonight, too.
 (Kripke, 2009, example 14, emphasis as in original)
 b. Sam is having **dinner** in New York tonight, too.
 c. Sam is having dinner in **New York** tonight, too.

In each example in (2.34), the persistent *alternative implication* associated with *too* implicates that an alternative related to the intonationally focused part of the sentence (indicated by boldface) is familiar in the discourse context. This implication is *too*'s analog of the familiarity implication for definites. For (2.34a), the implication is that someone other than Sam is dining in New York. For (2.34b), that Sam is doing something other than dining in New York. And for (2.34c), that he is dining somewhere other than New York as well.

Heim (1990b) makes a similar observation for (2.35).

- (2.35) John went to Harvard too.
 (Heim, 1990b, example 14)

Heim notes that for (2.35), nearly any imaginable felicitous discourse context will have the required entailment that someone other than John attended Harvard. Heim's story about *too*, which is formalized by Beaver

(2001, definition D17), is that *too* requires that the context of interpretation must entail that some accessible discourse referent bears the relevant property.³

The following tests provide solid evidence for the persistence of *too*'s alternative implication.

- (2.36) a. It's not true that Sam is having dinner in New York tonight, too.
 b. Is Sam having dinner in New York tonight, too?
 c. It's possible that Sam is having dinner in New York tonight, too.
 d. If Sam is having dinner in New York tonight, too, then his plane must have already arrived.
- (2.37) a. Sam is having dinner in New York tonight, too.
 b. No, he's not. I saw him on his way to the airport this morning.

Regardless of how the intonation is placed, each of the family of sentence variants in (2.36) maintains the implication of a familiar alternative, as does the direct denial in (2.37).

One difference between definites and *too* is that there is no analog of the informational uniqueness implication associated with definites:

- (2.38) Kim and Sandy are both having dinner in New York tonight.
 Robin is having dinner in New York tonight, too.

In (2.38), both Kim and Sandy fulfill the alternative implication associated with *too*. So unlike the antecedent of a definite, informational uniqueness is not required for *too*'s antecedent.

Several authors (Kripke, 2009; Roberts et al., 2009; Roberts, 2010; Tonhauser et al., 2013) have pointed out that there is more to this story, since iterative adverbs like *too* also bear an implication that is similar to the salience implication of definites. This implication, which is obligatorily speaker-oriented in addition to being conventional and persistent, is the requirement that a salient alternative be established in the discourse context. Clearly, this **salience implication** must persist since the alternative implication does (see the persistence tests in (2.36) and (2.37)). Example (2.39)

³This is a simplification, since maximal salience is required in addition to accessibility. But since salience is arguably at least partly determined by pragmatics, I will make no effort to encode it in the formal theory of discourse pursued here. Chapter 5 offers more discussion on this point.

gives evidence that the salience implication is also a necessary speaker commitment.

- (2.39) Sam is having dinner in New York tonight. Dana is unaware that anyone is having dinner in New York tonight, and so Dana is unaware that Sandy is having dinner in New York tonight, too.

In this example, it cannot be Dana that is committed to the implication of salience of an alternative because she is unaware of any such alternative, as is explicitly stated. Rather, the salience implication in (2.39) is interpreted as being a proposition that the speaker is committed to.

In accordance with the data, I define the **retrievability implication** for *too* somewhat differently than for definite anaphora. For *too*, retrievability is the implication that a familiar, salient alternative to the relevant implication is available in the discourse context. As for the retrievability implication for definites, the commitment to the existence of an alternative may be anchored to an embedded point of view, but the implication that there is an alternative available in the discourse context cannot be. The familiarity and salience implications for definites and for *too* differ in that for definites, there is an additional requirement of informational uniqueness that is not required for *too*, as (2.38) shows. However, just as for definites, the retrievability implication associated with *too* is an implication made by the speaker about the state of the discourse context, and as such must be a speaker commitment.

2.2.1.3 Possessives

The situation is slightly more complicated for possessives than for other definites. Notice that (2.40) is felicitous even if uttered in a context where the hearer has no knowledge that the speaker owns any cat.

- (2.40) I can't come to the meeting—I have to pick up my cat at the veterinarian.
(Stalnaker, 1998, example 1)

For Stalnaker (1998), as for von Stechow (2008), who discusses the same example, (2.40) is a straightforward instance of *accommodation* in Lewis's (1979) sense. However, notice the difference between the following variants of (2.40):

- (2.41) I can't come to the meeting—I have to pick $\left\{ \begin{array}{l} \# \text{ it} \\ ? \text{ the cat} \\ \text{one of my cats} \\ \text{a cat of mine} \end{array} \right\}$
up at the veterinarian.

The definites *it* and *the cat* in the relevant variants of (2.41) are much more difficult, perhaps even impossible, to accommodate than *my cat* in (2.40).

On the other hand, either of the two indefinites *one of my cats* and *a cat of mine* are perfectly good stand-ins for paraphrasing (2.40), unless it is already common knowledge that the speaker only has a single cat. As a further empirical gauge of the definiteness of possessives, consider the following context.

Context (2.42): It is unknown to any of the participants at a meeting whether Kim, who was supposed to attend the meeting but has not shown up yet, has a cat.

Then the tests can be applied to a variant of (2.40) as follows:

- (2.43) a. Kim_i doesn't have to pick up his_i cat.
b. Does Kim_i have to pick up his_i cat?
c. Maybe Kim_i has to pick up his_i cat.
d. If Kim_i has to pick up his_i cat, he_i won't be here on time.
- (2.44) a. Kim_i has to pick up his_i cat.
b. No, that can't be why he's late. He doesn't have a cat!

If interpreted in context (2.42), none of the tests in (2.43) require a discourse referent for *his cat* to be retrievable in prior context the way that *the cat* or *it* would, nor does the direct denial test in (2.44), although they all do commit the speaker to the belief that Kim has a cat.

I argue that in light of the discrepancy in the availability of accommodation for possessives as opposed to other definites, the similarity of (2.40) with variants using an indefinite in place of *my cat*, and the persistence tests for possessive definiteness, placing possessives on a par with other definites is unwarranted. For Barker (1991) and Pollard and Smith (2011), the different behavior observed for possessives serves as evidence that possessives can be used both as definites and indefinites.

Context: A driver has just picked up a hitchhiker. In the context of asking where the hitchhiker would like to be dropped off, the driver is describing the errands she is in the process of running.

- (2.45) I can drop you on Summit. I have to go over there anyway to pick up my ferret from the vet.
(Pollard and Smith, 2011)

Given the context of utterance, specially concocted to force the interlocutors to have no mutual conversational history, the hitchhiker cannot be assumed to have prior knowledge that the driver owns a ferret. Instead, the noun phrase *my ferret* in (2.45) simply introduces a discourse referent just as an indefinite in the same position would.

Pollard and Smith also point out the following case in which a possessive behaves in a way inconsistent with characterization as a definite.

Context: Ten-year old Timmy is jealous of older brother Eddie who is real cool and has a motorcycle. Now he has something else to be jealous about and complains to Mom.

- (2.46) a. How come I don't get to skip a month of school like Eddie?
b. Timmy, you didn't crash your bike and break your neck.
(Pollard and Smith, 2011)

In (2.46), interpreted in the intended context, the possessive *your bike* cannot refer to the motorcycle that Timmy is known to have, since Timmy is only ten years old. Instead, its meaning could be paraphrased by *Timmy, there's no bike you own such that you crashed that bike and broke your neck*. That is, the possessive in (2.46) seems to be an instance of an existential quantification that is outscoped by the negation associated with *didn't*.

Although he does not use the term *retrievability*, Barker (2000) explains this apparent difference between possessives and other definites by linking the retrievability implication of the possessive as a whole to the retrievability of the possessor. I agree with Barker's characterization: as the diagnostics in (2.43) show, the retrievability implication that is in effect for possessive pronouns does not extend to the entire possessive noun phrase. In fact, the implication associated with possessives that the possessed bears a certain property is similar to a variable conventional implicature, a topic I turn to in section §2.2.2. This story needs to be extended, however, to account for examples like the following:

Context: The utterance in (2.45) occurred at the beginning of the exchange, and the hitchhiker is preparing to get out.

(2.47) Thanks for the ride! I hope your ferret is OK.

In this case, the possessive *your ferret* does seem to invoke the ferret mentioned at the beginning. So it seems that a condition is needed in which possessives behave as definites when possible, and as indefinites otherwise. I discuss this point in greater detail in §5.2.1.1 and §5.2.2.

It also bears acknowledging some ways in which the retrievability implication associated with definites is more subtle than it may seem at first blush. Roberts (2003, 2005) argues that Heim's novelty-familiarity condition is too strong, and suggests a generalized, more empirically adequate characterization of familiarity based on entailment. Then there is the problem of anaphoric *accessibility*, which dynamic theories have long grappled with (for example Kamp, 1981; Heim, 1982, 1983a; Groenendijk and Stokhof, 1990, and their descendants). Accessibility is the problem of saying how discourse referents sometimes have a limited "lifespan" (Karttunen, 1976) as antecedents for definite anaphora. Also, as Rooth (1987) first pointed out, the *unselective binding* approach to quantificational noun phrases in Kamp's and Heim's theories leads to empirical predictions that are too coarsely grained. Kanazawa (1994) and Chierchia (1995) offer attractive solutions to this problem, but neither of their accounts is without its own problems. I postpone these and other related issues until chapter 4, where I lay out a formal account of anaphora.

2.2.1.4 Honorifics

Potts (2005) characterizes Japanese honorifics among his "CIs" because they have the properties of being persistent (*not-at-issue*, in his terminology) and being obligatorily speaker-anchored. But the taxonomy I investigate here recognized the fact that Pottsonian CIs are not all invariably anchored to the speaker, as pointed out by Amaral et al. (2007) and acknowledged by Harris and Potts (2009). As a result, the honorifics are separated from the other Pottsonian CIs because they *do* have the property of obligatory speaker anchoring, as the following example shows for the French honorific *tu*.

(2.48) Est-ce que tu as mangé?
 Is it that you.FAM have eaten
 Did you eat?

In (2.48), the gloss of *tu* indicates that it is the familiar variant of the French counterpart of English *you*; there is also a formal variant, *vous*. In this example, the implication that the speaker believes that she is on familiar terms with the addressee is a speaker commitment. And this commitment status holds up under embedding beneath an attitude predicate, as (2.49) shows.

- (2.49) Bernard croît que tu as mangé.
 Bernard believes that you.FAM have eaten
 Bernard thinks you ate.

As for (2.48), the implication of familiarity in (2.49) cannot be attributed to Bernard, but only to the speaker, as it was the speaker's choice to use the familiar variant *tu* rather than *vous*.

2.2.2 Variable Speaker Commitment Status

2.2.2.1 Descriptive Content

In addition to the retrievability implication, the use of *it* in (2.20) also implies that the retrieved referent must not be entailed by the context to be human, as the following variant of (2.3) shows.

- (2.50) A woman_i walked in, and then it_{#i} bought a ticket.

This **descriptive content implication** extends from pronouns to definites more generally, as in (2.51).

- (2.51) A mule_i walks in. The $\left\{ \begin{array}{l} \text{creature}_i \\ \text{mammal}_i \\ \text{equine}_i \\ \text{donkey}_{\#i} \end{array} \right\}$ brays.

In this example, an indefinite introduces a discourse referent and ascribes the property of being a mule to it. Then, in one variant, the definite *the donkey* purportedly takes the introduced referent as its antecedent, but this is an impossible interpretation because being a donkey and being a mule are mutually exclusive properties. The other variants are intended to show that descriptive content that does not conflict with the property of being a mule is perfectly fine.

The descriptive content implication is clearly persistent, since for each case of (2.52) the implication remains in effect, just as it does for the denial in (2.53).

- (2.52) A donkey walked into a bar.
 a. The donkey didn't bray.
 b. Did the donkey bray?
 c. It's possible that the donkey brayed.
 d. If the donkey brayed, it must be hungry.
- (2.53) a. A donkey walked into a bar. The donkey brayed.
 b. No way, that's just impossible.

But the descriptive content implication cannot simply be rolled into the retrievability implication, because the descriptive content implication is not obligatorily a speaker commitment, as [Tonhauser et al. \(2013\)](#) point out for English demonstratives and the Paraguayan Guaraní third-person pronoun.

- (2.54) Sandy thinks a horse_i behind her house is a donkey, and she
 thinks $\left\{ \begin{array}{l} \text{the horse}_i \\ \text{the donkey}_i \end{array} \right\}$ keeps her up all night braying.

Example (2.54) shows that the descriptive content implication is not necessarily a speaker commitment. In the variant with *the horse* as the definite, the descriptive content implication is a speaker commitment. But for the variant that uses *the donkey*, it is a belief attributed to Sandy, since the speaker makes clear that the donkey is actually a horse, and is only considered to be a donkey in Sandy's embedded perspective. This contrasts with the retrievability implication, which must be a speaker commitment, as example (2.33) in §2.2.1.1 shows.

[Zeevat \(1992\)](#) also notices this difference between the retrievability implication and the descriptive content implication. [Zeevat's](#) "resolution triggers" are those that require a suitable discourse referent as an antecedent, while "lexical triggers" are preconditions in a more general sense.

A prime example [of lexical triggers] seems to be sortal information associated with verbs and nouns. The meaning of these words can typically be divided into a part which identifies the type of entity referred to and a part which actually describes the entity.

([Zeevat, 1992](#), page 397)

For Zeevat, a definite is both a resolution trigger requiring that it must be resolved to a familiar discourse referent, and also a lexical trigger that makes a further descriptive content requirement of whichever referent the resolution trigger locates as the definite's antecedent.

In addition to Tonhauser et al. (2013), who discuss both English and Guaraní data in their investigation, Pollard and Smith (2011) also provide evidence that the descriptive content implication is separate from the retrievability implication by noting that a pronoun's descriptive content can be new information, as in the following example.

- (2.55) A cop_i just walked over to my car. She_i wrote me a ticket for my busted tail light!

Since the information that the cop in question is female may be new to the hearer, it does not figure into the retrievability implication associated with the use of the pronoun *She* in (2.55). All that is required is that the pronoun's descriptive content not conflict with what the context entails about the antecedent discourse referent:

- (2.56) A cop_i just walked over to my car. The guy_i didn't care that I was speeding. # But she_i wrote me a ticket for my busted tail light!

Example (2.56) clearly shows that the pronoun *she* cannot be interpreted as taking an antecedent that is already entailed to be male.

Taken together, (2.52), (2.53) and (2.54) demonstrate that the descriptive content implication associated with definites is conventionally signaled, persistent, and displays variability with respect to its status as a speaker commitment. Finally, there is also an analog of the descriptive content implication for possessives, which (2.57) shows also has speaker commitment variability.

- (2.57) Chris believes that Kim_i's dog_j is a cat, and that Kim_i has to pick up his_i $\left\{ \begin{array}{c} \text{dog}_j \\ \text{cat}_j \end{array} \right\}$.

As for (2.54), on the intended interpretation of (2.57), either of *dog* or *cat* can be used to specify the descriptive content of the possessed associated with *his*. Since the pet Kim has to pick up is only believed to be a cat by Chris, the use of *cat* to specify the relevant descriptive content must be Chris's commitment and not the speaker's. For a similar reason, *dog* must be interpreted as a speaker commitment: Chris thinks Kim's pet is a cat, not a dog.

2.2.2.2 Supplements and Expressives

As discussed in connection with (2.6), (2.15) and (2.17) in §2.1, above, nominal appositives share the same properties as the descriptive content implication: the persistent implication of a nominal appositive is part of its conventional meaning, but it is not required to be a speaker commitment. In fact, the descriptive content implication has these properties in common with Potts's class of "CIs" as a whole.

The following are examples of an expressive and two supplements: a nonrestrictive relative clause, and an *as*-parenthetical.

- (2.58) That socialist Obama got re-elected.
- (2.59) The votes from Florida's election, which was held earlier this month, are still being counted.
- (2.60) The state of Ohio, as the Associated Press is now reporting, has been called for President Obama.

In (2.58), the expressive *That socialist* conventionally bears the implication that Obama's politics are viewed negatively from the point of view the expressive is anchored to. The commas in (2.59) and (2.60) demarcate the boundaries of the content set apart by *comma intonation*, which has the effect of making the demarcated content persistent as part of its conventional meaning (Potts, 2005). For (2.59), the persistent implication is that Florida's election was held earlier this month. For (2.60), the implication that the Associated Press is now reporting that Ohio has been called for Obama is made to persist.

The tests in (2.61)–(2.66) show that these implications are all indeed persistent. First, the expressives:

- (2.61) a. That socialist Obama didn't get re-elected.
b. Did that socialist Obama get re-elected?
c. It's likely that that socialist Obama got re-elected.
d. If that socialist Obama got re-elected, it's time to move to a country that respects liberty.
- (2.62) a. That socialist Obama got re-elected.
b. No he didn't. The media has just misconstrued the results.

In each of the tests in (2.61) and (2.62), the implication that the speaker is not fond of the putative leftward tilt of Obama's politics is present, just as it is for (2.58). Next, consider nonrestrictive relative clauses.

- (2.63) a. The votes from Florida's election, which was held earlier this month, are not still being counted.
 b. Are the votes from Florida's election, which was held earlier this month, still being counted?
 c. Maybe the votes from Florida's election, which was held earlier this month, are still being counted.
 d. If the votes from Florida's election, which was held earlier this month, are still being counted, then Florida should allow international election monitors next time.
- (2.64) a. The votes from Florida's election, which was held earlier this month, are still being counted.
 b. No they aren't. They finished counting yesterday, and Obama won narrowly.

Similarly to the tests for expressives, the nonrestrictive relative implicating that Florida's election was held earlier this month survives embedding in all of the family of sentences tests in (2.63), as it does for the denial test in (2.64). Lastly, the *as*-parentheticals:

- (2.65) a. The state of Ohio, as the Associated Press is now reporting, has not been called for President Obama.
 b. Has the state of Ohio, as the Associated Press is now reporting, been called for President Obama?
 c. Perhaps the state of Ohio, as the Associated Press is now reporting, has been called for President Obama.
 d. If the state of Ohio, as the Associated Press is now reporting, has been called for President Obama, then the election is effectively decided.
- (2.66) a. The state of Ohio, as the Associated Press is now reporting, has been called for President Obama.
 b. That can't be right. This guy Karl is on Fox News saying it's way too early to call Ohio.

The *as*-parenthetical *as the Associated Press is now reporting* has its effect untouched by each of the variants of (2.60) used in the persistence diagnostics (2.65) and by the denial test in (2.66). These tests provide strong evidence that the class of constructions Potts calls "CIs" really is persistent.

The following examples show that these implications can represent speaker commitments.

- (2.67) Kim actually thinks that socialist Obama is not a socialist!
- (2.68) Sandy believes that Florida's election, which was held earlier this month, actually took place two months ago and the votes are still being counted.
- (2.69) Karl thinks that Ohio—which, as the Associated Press is now reporting, has been called for Barack Obama—has not yet been called.

In each of (2.67)–(2.69), the persistent implication is a speaker commitment because the embedded point of view is inconsistent with it. In (2.67), Kim does not think Obama is a socialist. Similarly, for (2.68), Sandy cannot simultaneously believe that Florida's election was held two months ago and that it was held earlier this month. In (2.69), finally, the content of the *as*-parenthetical cannot be anchored to Karl's point of view because Karl does not think Ohio has been called by anyone.

After a detailed effort at describing the behavior of these constructions in various contexts, Potts (2005) concludes that they are all *invariably* anchored to the speaker's point of view on the basis of examples like (2.70).

- (2.70) Ed said, as Sue predicted, it is raining. # But in fact Sue didn't predict rain.
(Potts, 2005, example 2.47)

Countering Potts's claim, Amaral et al. (2007) give convincing counterexamples that appositives, expressives, and nonrestrictive relative clauses can all have nonspeaker anchoring given the right context. Harris and Potts (2009) follow on Amaral et al.'s work, confirming that supplements and expressives can be anchored to an embedded point of view as part of an in-depth experimental study of the relative frequency of speaker-anchored and nonspeaker-anchored readings.

For both Amaral et al. and Harris and Potts, the process for showing that these implications can be anchored to a nonspeaker point of view involves concocting a discourse context with just the right features to prompt a nonspeaker-anchored reading, similar to context (2.16), above. Beginning with a variant of the expressive in (2.58):

Context (2.71): The speaker was unaware of the outcome of the election until he walked past a tea party rally where the dejected participants were complaining about the result.

(2.58') Apparently, those tea partiers are upset that that socialist Obama got re-elected.

Assuming that the speaker is not herself a tea partier, and assuming that the speaker does not herself believe Obama to be a socialist, her utterance of (2.58') in the context (2.71) cause the expressive's implication of dissatisfaction with Obama's purported location on the political spectrum to be interpreted as anchored to the tea partiers rather than to the speaker.

Next, a context capable of causing the anchoring of (2.59) to be different than the speaker's point of view.

Context (2.72): Kim is aware that Florida's election was actually held several months in the past. She is describing her friend Robin, who is delusional and believes that it is currently November of the year 2000.

(2.59) The votes from Florida's election, which was held earlier this month, are still being counted.

In context (2.72), the nonrestrictive relative clause *which was held earlier this month* in (2.59) must be interpreted as anchored to the delusional Robin, since Kim does not believe Florida's election was held earlier this month.

Finally, a context that causes the *as*-parenthetical in (2.60) to be anchored not to the speaker but instead to Karl.

Context (2.73): The speaker has not heard of any reports from the Associated Press regarding the election. Karl believes that the Associated Press has called Ohio for Obama.

(2.60) The state of Ohio, as the Associated Press is now reporting, has been called for President Obama.

Since the speaker of (2.60), interpreted in context (2.73), does not know Ohio has been called, the content of the parenthetical *as the Associated Press is now reporting* must be anchored to Karl's point of view.

Based on the persistence tests applied to the examples in (2.5) and (2.58)–(2.60), together with the evidence in contexts (2.15) and (2.71)–(2.73) that their anchoring is sensitive to context, there is a solid case for categorizing supplements and expressives as conventional implicatures that are not necessarily speaker commitments.

2.3 Nonconventional Implicatures

As its name implies, the class of **nonconventional implicatures** is made up of those implicatures that result not solely from the conventional meaning of their trigger, but by a process of inference based on the entailments present in the discourse context along with those resulting from the trigger. Their nonconventional nature is the reason that this class of implicature displays some variability with respect to whether their associated persistent implication persists or not.

The nonconventional implicatures I discuss here does not include the *conversational implicatures* of Grice (1975). This meaning class corresponds roughly to Abusch's (2010) "soft triggers," Simons et al.'s (2010) and Tonhauser et al.'s (2013) "class C" persistent contents, and to Pollard and Smith's (2011) "persistent entailments." Unlike certain conventional implicatures, none of the members of this class are required to be speaker commitments, which has implications for persistence variability, as I discuss below.

The nonconventional implicatures include the so-called factive implications of verbs like *know*, *realize* and *regret*, aspectual verbs like *continue*, *quit*, *start*, *stop* and *switch to*, and verbs signaling achievements with a preparatory phase, such as *graduate* and *win*. Much of the early literature on these verbs construes them as presuppositional, as imposing a constraint that the context of interpretation must entail the relevant persistent implication (Karttunen, 1974; Stalnaker, 1978; Lewis, 1979; Heim, 1983b). But in this thesis, I follow a countervailing trend in the literature of treating them as persistent but not necessarily as imposing felicity constraints, exemplified by Boër and Lycan (1976), Abbott (2000), Simons (2001), Gauker (2008), Beaver (2010), and Roberts et al. (2009), Simons et al. (2010), among others.

2.3.1 Aspectuals

As the tests in (2.11) and (2.14) demonstrate, the aspectual verb *stop* can have a persistent implication, namely that its pre-state holds. In (2.74), the pre-state implication of

(2.10) Kim stopped smoking.

is explicitly negated by the speaker, but (2.10) is felicitously embedded beneath the predicate *thinks*, with the associated belief attributed to Robin.

- (2.74) Kim has never smoked in her life, but Robin thinks Kim stopped smoking.

In this example, the addressee infers that Robin thinks Kim used to smoke because it is not possible for Robin to think Kim stopped smoking without also thinking that she started smoking at some point. That is, Robin's thinking Kim used to smoke arises because it is an entailment of Robin thinking Kim stopped smoking.

Importantly, though, aspectuals do not impose any requirement that the context must entail their pre-state. As an example, consider (2.75).

- (2.75) I wonder why Kim is so on edge lately. Maybe she stopped smoking, or something.

If an interlocutor responded with *Yes, and she's having a tough time with it*, the speaker of (2.75) would be justified in inferring that Kim smoked before. But if the response instead was *No, that can't be it—she's never smoked in her life*, the inference of *stop*'s pre-state is not warranted. Thus the persistence of *stop*'s pre-state implication is at least partly determined by context, and therefore not part of the conventional meaning of *stop*, as it is for the retrievability implication associated with definites or the comma intonation signaling a nominal appositive.

The pattern these verbs demonstrate is that they have a persistent implication whose persistence is not dictated entirely by their conventional meaning, and that they can be anchored to an embedded, nonspeaker perspective. To take another example, consider *switch to* in the following.

- (2.76) Maybe Kim switched to drinking decaf.
 (2.77) If Kim switched to drinking decaf, she might be interested in getting rid of her caffeinated coffee supply.

The associated prior state in (2.76) is that there is some previous time at which Kim drank caffeinated coffee. But similarly as for (2.75), (2.76) is used in nonfactual mood, perhaps to offer an explanation for why Kim has seemed so much less on edge lately. That is, (2.76) is perfectly felicitous in contexts where nothing is known about her prior coffee consumption habits. The same is true of (2.77): the speaker may only be interested in obtaining some caffeinated coffee, and may have no knowledge of whether or not Kim ever drank it.

It is also felicitous to use *switch to* in contexts where her having previously consumed caffeinated coffee is known to be false, as (2.78) shows.

- (2.78) Why is Kim so relaxed? Well, she's not a coffee drinker, so it's not because she switched to drinking decaf.

The examples in (2.76)–(2.78) show that *switch to* does not strictly require its pre-state to be entailed by the discourse context as a presuppositional, because the pre-state implication could be new information or suppressed by the context.

Like *start*, *stop*, and *continue*, the pre-state implication of the verb *switch to* is both persistent and capable of being anchored to a nonspeaker point of view.

- (2.79) a. Kim didn't switch to drinking decaf.
 b. Did Kim switch to drinking decaf?
 c. It's possible Kim switched to drinking decaf.
 d. If Kim switched to drinking decaf, she'll probably be in a better mood.
- (2.80) a. Kim switched to drinking decaf.
 b. No she didn't. But she did quit smoking recently.

Just as in (2.11), for each of the tests in (2.79) and (2.80) the addressee may infer the pre-state implication that Kim used to drink caffeinated coffee as long as the context does not conflict with it. And *switch to* can clearly take on an embedded perspective, since in (2.81) the pre-state implication is necessarily anchored to Robin because the speaker makes clear it is false.

- (2.81) Kim never drank caffeinated coffee, but Robin believes that Kim switched to drinking decaf.

Importantly, the pre-state implication of *switch to* does not persist for (2.81) because it conflicts with the speaker's assertion. Thus *switch to* sometimes persists, but its persistence is not part of its conventional meaning, and it is not necessarily a speaker commitment.

2.3.2 Achievements

Verbs like *win* and *graduate* are associated with a prior period of time called a *preparatory phase*. They give rise to a **preparatory phrase implication** related to this prior time period that is similar to the pre-state implication for the aspectual verbs. For example, in (2.82), Lance's having participated in the Tour de France is entailed by his having won it.

(2.82) Lance won the Tour de France.

Just as aspectuals do not presuppose their pre-state, achievements do not presuppose their preparatory phase, as the tests in (2.83) and (2.84) demonstrate for *win*.

- (2.83) a. Lance didn't win the Tour de France.
 b. Did Lance win the Tour de France?
 c. Maybe Lance won the Tour de France.
 d. If Lance won the Tour de France, he'll be doing the talk show tour when he gets back home.

- (2.84) a. Lance won the Tour de France.
 b. No, he didn't—he was disqualified for doping.

The behavior of *win* is similar to that observed for the aspectual verb *switch to*: just as for the aspectuals, an addressee hearing (2.82) may infer that the preparatory phase is implicated by the speaker. And (2.85) shows that achievements like *win* can have a nonspeaker anchoring, since the speaker of (2.85) negates the relevant preparatory phase.

- (2.85) Lance didn't participate in the Tour de France, but Sandy believes Lance won the Tour de France.

For (2.85), the preparatory phase implication must be Sandy's belief and not the speaker's, and therefore it is not a speaker commitment. And similarly to (2.81), Lance's having participated in the Tour is not a persistent implication of (2.85) because it is contradicted by the speaker.

In a direct parallel to (2.10), consider (2.83a) in the following contexts:

Context (2.86): A sports website wants to run biographical pieces about participants in recent high-profile cycling races. One of the editors thinks the race winners would be the most amenable candidates.

Context (2.87): A sports website wants to interview winners of recent high-profile cycling races about doping in the sport. It is mutual information that Lance has retired from professional cycling.

In context (2.86), *win*'s preparatory phase in (2.83a) may persist because it is inferable by the interlocutors. By contrast, in context (2.87), the preparatory phase does not persist for the simple reason that the common

ground already entails its denial. Interpreting (2.83a) in these two contexts shows that nonspeaker anchoring is not the only reason that aspectuals and achievements fail to persist. They also fail to persist whenever their persistent implication is not consistent with entailments already present in the discourse context.

2.3.3 Factives

In the literature, the so-called *factive* verbs have often been characterized as presuppositional, constraining the context of interpretation to entail their complement. I adopted such a stance toward the emotive factive *suck* in Martin and Pollard (2012a). But at least since Karttunen (1971), who identifies a *semi-factive* subclass of factive verbs, it has been recognized that the persistent implications associated with many factive do not persist as rigidly as the implications associated with definites, for example.

Consider (2.89) in the following (veridical) context.

Context (2.88): It is not known whether the Riemann hypothesis, a famous open conjecture in mathematics, is true or false.

- (2.89) Louie doesn't $\left\{ \begin{array}{l} \text{know} \\ \# \text{ realize} \end{array} \right\}$ the Riemann hypothesis is true, he only thinks he does.
(Pollard and Smith, 2011)

In (2.89), the complement of *know* does not persist when *know* is embedded under negation, and so is not presuppositional, although interlocutors may sometimes infer that the truth of its complement (called its *factive implication*) is part of what is being implicated.

More generally, the complement of a factive is not even conventionally implicated, in addition to not being presupposed.

- (2.90) Sandy doesn't know Kim quit smoking, because he knows Kim never even started smoking in the first place.
(2.91) # Sandy doesn't know Kim, who he believes to have previously been a smoker, quit smoking, because he knows Kim never even started smoking in the first place.

These examples serve to contrast nonconventional implicatures from the conventional ones. In (2.90), the pre-state implication associated with *quit*

does not persist because it conflicts with the information that Sandy knows Kim never started smoking. But in (2.91), in which the pre-state implication is explicitly stated as a nonrestrictive relative, the situation is different: Sandy's conflicting knowledge gives rise to infelicity.

A similar example from Boër and Lycan 1976 shows that, in the case of *know*, the interlocutors may even know nothing about the factive complement.

Context: The interlocutors have no knowledge of the whether the Goldbach conjecture is true or false.

- (2.92) John doesn't know that Goldbach's conjecture is false.
(Boër and Lycan, 1976, example 92)

In (2.92), interpreted in the intended context, the addressee is not required to know that Goldbach's conjecture is false, or even to infer it after hearing (2.92) uttered.

Simons (2001) discusses this kind of persistence failure in connection with "explicit ignorance contexts," in which it is obvious to the interlocutors that a purported presupposition cannot be entailed by the context.

Context: The interlocutors are eating at a restaurant. At the neighboring table is a couple in a heated argument.

- (2.93) Perhaps she just discovered that he's having an affair.
(Simons, 2001, example 8)

Similarly to *know* in (2.89), the complement of the factive *discover* cannot be presupposed as mutual prior knowledge in (2.93), as the interlocutors are not acquainted with the arguing couple and have no idea what they are arguing about.

This behavior contrasts with that of *realize*, which, in addition to being anchored to the speaker rather than to Louie in (2.89), is infelicitous in a context that entails that the Riemann hypothesis is anything other than true. As the following tests show, the factive *prove*, like *know*, displays persistence variability in certain constructions.

- (2.94) a. No one proved the Riemann hypothesis.
b. Did someone prove the Riemann hypothesis?
c. Perhaps someone proved the Riemann hypothesis.

d. If someone proved the Riemann hypothesis, we should publish the proof.

- (2.95) a. Someone proved the Riemann hypothesis.
b. That's not true. If that had happened, I would've heard about it.

Just as for *know* in (2.89), the use of the verb *proved* in (2.94d) is completely acceptable in contexts that do not entail the truth of the Riemann hypothesis. The denial test in (2.95) illustrates similar behavior: the Riemann hypothesis's truth does not persist.

- (2.96) a. Louie doesn't realize the Riemann hypothesis is true.
b. Does Louie realize the Riemann hypothesis is true?
c. Maybe Louie realizes the Riemann hypothesis is true.
d. If Louie realizes the Riemann hypothesis is true, I'd like to see his proof.
- (2.97) a. Louie realizes the Riemann hypothesis is true.
b. No way. If he did, he would have attempted to publish a proof by now.

The tests in (2.96) and (2.97) point to another important distinction between factives. Notice that for each of the tests involving *realize*, the truth of the Riemann hypothesis is implicated. This contrasts with (2.98a), where the reason Louie does not know the Riemann hypothesis to be true may well be that it is an open conjecture, or false (for example, in a context like (2.88)).

- (2.98) a. Louie doesn't know the Riemann hypothesis is true.
b. Does Louie know the Riemann hypothesis is true?
c. Maybe Louie knows the Riemann hypothesis is true.
d. If Louie knows the Riemann hypothesis is true, I'd like to see his proof.
- (2.99) a. Louie knows the Riemann hypothesis is true.
b. No he doesn't, he only thinks he does.

The behavior of *realize* also different from the behavior exhibited by *know* in all of the tests in (2.98) and (2.99), none of which implicate that the Riemann hypothesis is true.

It is important to note the difference between the stative *realize* in (2.89) and its inchoative relative:

- (2.100) If Louie someday realizes the Riemann hypothesis is true, I'm sure he'll be thrilled.

The inchoative realization in (2.100) is taking place in the future, and therefore what is being realized (the truth of the Riemann hypothesis) is not entailed. Inchoative *realize* is synonymous with *find out*, *discover*, *come to know*, etc.

As the tests (2.96) and (2.97) demonstrate, the reason for the discrepancy between stative *realize* and the other factives (including inchoative *realize*) is that, when it is anchored to the speaker's point of view, stative *realize* always attributes belief in the truth of its factive implication to the speaker. That is, when speaker-anchored, the proposition expressed by stative *realize*'s complement is necessarily a speaker commitment. This contrasts with *know* and *prove*, for example, which can be interpreted as attributing belief in the relevant factive implication to an embedded perspective.

The following example further illustrates this contrast.

- (2.101) It's not raining. But Dana $\left\{ \begin{array}{l} \text{thinks} \\ \text{believes} \\ \text{is under the impression} \\ \text{is convinced} \\ \text{just knows} \\ \text{\# realizes} \end{array} \right\}$ that it's raining.

Even in the variant of (2.101) that uses *knows*, since the discourse context already contains the information that it is not raining, an addressee can interpret the belief that it is raining as anchored to Dana and the use of *know* in particular as ironically signaling her stubbornness. But the nonspeaker-anchored interpretation is simply not possible for *realize*, whose factive implication is always interpreted as a speaker commitment when *realize* is itself speaker-anchored.

However, when stative *realize* occurs in the scope of an attitude predicate or in a modal subordination context (Roberts, 1989), different behavior is observed.

- (2.102) Louie thinks the Riemann hypothesis is true, but that no one realizes it is.
- (2.103) It might be that the Riemann hypothesis is true, but that no one realizes it is.

Interpreted in context (2.88), in which the Riemann hypothesis has the status of a mere conjecture, belief in the factive implication of *realize* in (2.102) cannot be attributed to the speaker. Rather, the truth of the Riemann hypothesis is taken to be believed by Louie, the point of view to which *thinks* is anchored. Similarly, the use of the modal *It might be* in (2.103) in the given context requires that the speaker not believe the Riemann hypothesis is true.

Importantly, though, the factive implication of *realize* is not presupposed as part of its conventional meaning, in the sense of imposing a required entailment on the context of interpretation. This is illustrated in (2.102), because clearly, given the context, the speaker does not take the truth of the Riemann hypothesis to be mutual information. But *realize* does not require a presupposition of the truth of its complement even in unembedded cases, as the following example shows.

Context: A driver has just picked up a hitchhiker, who has a gum wrapper in his hair. They have not yet spoken.

(2.104) Do you realize there's a gum wrapper in your hair?

If (2.104) is uttered in the indicated context, the use of *realize* cannot be interpreted as requiring the discourse context to entail that the hitchhiker has a gum wrapper in his hair. For all the driver knows, the hitchhiker may be unaware of the gum wrapper's presence. The hitchhiker may still infer that the gum wrapper is in his hair based on hearing (2.104), so the factive implication can contribute new information.

Summing up, so-called factive verbs can give rise to a persistent implication, but the persistence of this implication displays variability depending on the discourse context. It is therefore not a part of the conventional meaning of a factive. The stative version of *realize* persists more rigidly in unembedded, but what persists is an entailment related to the use of *realize*, and is not a conventional implicature like anaphora, supplements, or expressives.

2.4 Taking Stock

The members of the empirical domain of implicatures, all persistent implications, are characterized according to two criteria. The first is whether the

Implication (Associated Trigger)	Conventional	Commitment
Novelty (indefinites)	Yes	Speaker
Retrievability (definite anaphora)	Yes	Speaker
Retrievability (possessor in possessive)	Yes	Speaker
Retrievability (<i>too</i>)	Yes	Speaker
Descriptive Content (definite anaphora)	Yes	Variable
Descriptive Content (possessed in possessive)	Yes	Variable
Conventional Implicatures	Yes	Variable
Pre-state (aspectuals)	No	Variable
Preparatory Phase (achievements)	No	Variable
Factive	No	Variable

Table 2.1: Summary characterization of implicatures based on conventionality and speaker commitment status.

persistence of their associated persistent implication is a part of their conventional meaning. The second is whether they must obligatorily give rise to a speaker commitment. Table 2.1 gives a summary of the implicatures discussed in this chapter, differentiated by these two criteria.

These criteria are useful for distinguishing implicatures because they shed light on some important notions related to persistent content. The taxonomic hierarchy of the implicatures investigated here, based on the criteria of the conventionality of their persistence and their speaker commitment status, is shown in figure 2.1. In this taxonomy, implicatures are first divided based on the whether or not they persist as part of their conventional meaning. As I discuss in §2.4.1, this distinction is important for a more general view of contextual felicity. The class of conventional implicatures bifurcates a second time into those that must be speaker commitments and those that do not have this requirement. By contrast, none of the nonconventional implicatures are obligatorily speaker commitments.

The top two levels of the taxonomic graph in figure 2.1 are of course reminiscent of Grice's (1975) taxonomy of implicatures. For Grice, implicatures are those implications that are not the main point of what is said, and the class of implicatures is split into the *conventional* and *nonconventional implicatures*, as here. I chose to reappropriate the term *conventional implicature* from Potts (2005) in order to signal the connection between anaphora,

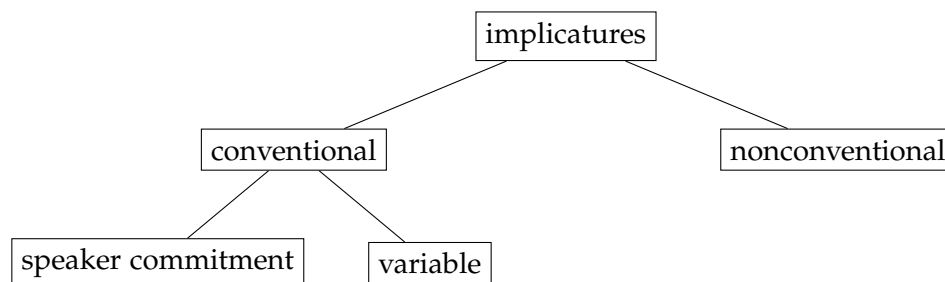


Figure 2.1: Taxonomic graph of implicatures.

Potts’s “CIs,” and the nonconventional implicatures, and also to connect it to an older tradition that dates to Grice and to Karttunen and Peters (1979). The taxonomy given here extends Grice’s original characterization of implicatures, which did not differentiate the conventional implicatures based on speaker commitment status, a central notion in my account.

2.4.1 *Felicity, Accommodation, and Variability*

One notion that this taxonomy serves to illuminate is the notion of *felicity*. Ever since Langendoen and Savin (1971) first identified the “projection problem” for presuppositions, there has been a strong tendency in the study of persistence to treat all of it as presuppositional. In this view, persistence occurs because a trigger imposes felicity constraints on the context of interpretation, and these constraints survive embedding within the scope of operators. Among many others, Karttunen (1973, 1974), Stalnaker (1973, 1978), Lewis (1979), Heim (1983b), van der Sandt (1992), Chierchia (1995), Geurts (1999) and Beaver (2001) have all adopted this view to one degree or another. I have also taken this stance toward persistence in my own previous work on dynamic semantics, anaphora, and factives (Martin, 2012; Martin and Pollard, 2012a,b).

But the traditional way of thinking about persistence has had the unfortunate effect that certain instances of implicatures, for example Potts’s (2005) CIs, are characterized as existing on a separate plane from anaphora, which figures among the implicature triggers usually considered presuppositional. Instead, characterizing anaphora and CIs as both giving rise to a conventionally signaled, persistent implication makes clear that they have much in common, as the parallel between (2.106) and (2.108) illustrates.

Context (2.105): There is no retrievable nonhuman antecedent.

(2.106) # It brayed.

The usual story about the unacceptability of (2.106), uttered in a context like (2.105) with no retrievable antecedent for *It*, is that the infelicity results from the speaker commitment that the addressee can retrieve a suitable antecedent. Potts (2005, page 111) describes the unacceptability of (2.108) as the property of CIs that their meaning is “nondeniable.”

Context (2.107): It is not known to the interlocutors whether Lance is or is not a cyclist.

(2.108) Lance, a cyclist, is from Texas. # Lance is not a cyclist.

Similarly to the unacceptability of (2.106), the speaker’s commitment to Lance not being a cyclist causes (2.108) to be unacceptable in the discourse context, in which it has already been established by the preceding utterance that Lance is a cyclist.

I would argue that both (2.106) and (2.108) constitute instances of a single, more general phenomenon: in each case, a persistent implication that is part of the conventional meaning of the trigger is at odds with the discourse context. For (2.106), the offending implication is that there is a retrievable antecedent for *It*. For (2.108), the nominal appositive’s content is the proposition that Lance is a cyclist, but this conflicts with the implication in the following utterance that, to the contrary, he is not a cyclist.

A similar parallel can be observed between the novelty implication associated with indefinites on the one hand and what Potts (2005, page 112) calls the “antibackgrounding” property of CI content.

(2.109) A donkey_i walked into a bar. A donkey_{#i} ordered a sandwich.

(2.110) Lance Armstrong survived cancer. # When reporters interview Lance, a cancer survivor, he often talks about the disease.
(Potts, 2005, example 4.46a)

On the intended reading, (2.109) is infelicitous because both indefinites give rise to the implication that the same discourse referent is being mentioned for the first time. In (2.110), the content of the nominal appositive *a cancer survivor* is already mutual knowledge to interlocutors who have accepted the first utterance.

And so rather than treating anaphora as having the special property of imposing felicity constraints on the discourse context, the infelicity that results when an anaphor's antecedent is not retrievable is simply a result of the way cooperative conversations are managed, in the sense of Grice (1975). The reason that (2.106) and (2.108) are unacceptable is that both threaten to leave the discourse context in an inconsistent state, with an implication that cannot be validated. This idea is already represented in the work of Karttunen (1974), Heim (1983b) and van der Sandt (1992) in the form of the requirement that presuppositions must be consistent with the discourse context, though they differ in how contexts are formally represented. The unacceptability of (2.109) and (2.110) arises from the fact that in both, a implicature trigger is used in a way that is not informative, and thus no implicature can occur. Geurts (1999, page 59) refers to the unacceptability of examples like (2.110) as the constraint of "informativeness" on discourse interpretation, invoking Grice's (1975) first maxim of quantity.

In addition to felicity, the taxonomy for implicatures I discuss above also has implications for the related notion of *accommodation* originally due to Lewis (1979). The characterization of anaphora as mere instances of the larger phenomenon of persistent contents allows a more open perspective on their supposed nature as constraint-imposers. Under the view held by that all persistence is presupposition, not only are anaphors and triggers like *too* construed as bearing presuppositions, but so are aspectual verbs, achievements, and factives. The "resolution" theories due to van der Sandt (1992) and Geurts (1999) exemplify this view, and propose that presupposition failure is always repaired by immediately making the required accommodation. A similar proposal is found in the treatment of names via *exception handling* found in de Groote and Lebedeva 2010. For accounts in this vein, cases like (2.104) in which the discourse context actually does not entail these triggers' purported presuppositions are construed as instances of *informative presupposition*, a seeming oxymoron.

In accounts that treat all persistence as presuppositional, the unfulfilled presuppositions of the class of meanings I call nonconventional implicatures are simply accommodated when the need arises. So theories like these actually posit that accommodation, which is traditionally thought of as a repair strategy that is invoked when the context becomes inconsistent (Stalnaker, 1978; Lewis, 1979), actually happen constantly during the course of ordinary discourse. A theory equipped with an unconstrained accommodation that is always available has to face the unfortunate consequence of

being scientifically unfalsifiable (Roberts, 2012a), becoming “vacuous, since no counterexamples could be raised against it” (Abbott, 2000, page 1426).

I would argue that it is more illuminating to simply treat implicatures as having an associated persistent implication that persists to a certain degree based on whether it is conventionally signaled and whether it is a speaker commitment, but is in no way presupposed in the sense of imposing a felicity constraint. This view of some persistent implications as not strictly constraining the context of interpretation has some recent precedent in the literature, see, for example, Abbott 2000, Simons 2001, Gauker 2008, Roberts et al. 2009, Abusch 2010, Beaver 2010, Simons et al. 2010, and Pollard and Smith 2011. Not only does this alternative empirical characterization better match the observed facts, it has the additional benefit of not requiring a special repair mechanism so that interpretation can take place in ordinary discourse. It also avoids the undesirable theoretical consequence that Karttunen (1974) called “part-time presupposition,” in which certain lexical items purportedly have a felicity constraint built into their conventional meaning but this constraint is frequently not observed. Under this view, the terms *presupposition* and *anaphora* become synonyms, because there are no triggers besides anaphora that are claimed to make constraints on the discourse context.

Although I do not attempt an explicit formal account of accommodation in this thesis, I would argue that there is still a place in a theory of implicature for a notion of accommodation, just not exactly as Lewis (1979) conceived of it. Theories in which accommodation comes to the rescue any time there is a threat of infelicity clearly have too broad a notion of accommodation. But accommodation still does occur in the course of conversations, because speakers sometimes use expressions bearing conventional implicatures that risk being interpreted as being in conflict with the context, as the examples in (2.106) and (2.108) show.

Several authors explore a more nuanced and constrained alternative to Lewis’s (1979) accommodation. Thomason (1990) construes accommodation as a process in which interlocutors add missing but intended content to the context as part of an effort to recognize the other interlocutors plans and goals. Von Stechow (2008) characterizes an accommodation mechanism that is more in line with what actually happens in natural language discourse. For von Stechow, when speakers accommodate, they “try to figure out which particular adjustment is [most] likely the one that the speaker intended” (page 162), however, sometimes accommodation simply cannot happen, as

when a definite is used in a context lacking a suitable antecedent. The view of accommodation as a form of goal-driven inference is also put forward by Beaver and Zeevat (2007), as it is by Thomason, Stone, and DeVault (2006) and Roberts (2012a), who each in different ways make a case for an accommodation process guided by intention recognition. These theories that are based on the interlocutors' intentions and common goals seem to offer a much more detailed and compelling story about accommodation than Lewis's original.

Finally, the taxonomy I present in this chapter also bears on persistence *variability*. The class of nonconventional implicatures displays foremost the characteristic that persistence fails when the potential persistent implication conflicts with the discourse context. And similarly, conventional implicatures like the anaphoric retrievability implication or the persistent implication of supplements normally result in unacceptability when they conflict with the discourse context.

The persistence variability for nonconventional implicatures and the exceptions to unacceptability for conventional implicatures are related in the following way. Following Roberts (2011, *to appear*), conventional implicatures that conflict with the discourse context give rise to unacceptability only when they are speaker commitments. An otherwise unacceptable conventional implicature can take on an embedded point of view as a strategy for avoiding unacceptability, for example, in (2.15), in which it cannot be the speaker who is committed to the existence of a chip behind Joan's ear. Nonconventional implicatures that are not consistent with the context cannot persist except when they are speaker commitments, as the following contrast shows:

- (2.85) Lance didn't participate in the Tour de France, but Sandy believes
Lance won the Tour de France.
- (2.111) Lance didn't participate in the Tour de France, # but Lance won
the Tour de France.

And so this taxonomy shows that unacceptability is a result of a implicature to which the speaker is committed that is also in conflict with the context.

In terms of explaining *why* implicatures behave the way they do, the story is simple. The conventional implicatures that must be speaker commitments (for example, the retrievability implication of a definite) all share the property that they are implications not about the state of the world but about the discourse context itself. Therefore, they cannot fail to be

speaker commitments because they represent information that the speaker is conveying to the addressee about the mutually accepted content they share for the purposes of the conversation. It is for this same reason that conventional implicatures that are not necessarily speaker commitments (for example, one of Potts's (2005) "CIs") can sometimes conflict with the discourse context yet not give rise to infelicity: they are not statements about the discourse context itself, and as such can be anchored to a non-speaker point of view under the right conditions. Then the difference between conventional implicatures that are not speaker commitments and the nonconventional implicatures is again simply that a nonconventional implicature does not persist as part of its conventional meaning, and thus persistence may sometimes fail.

It bears mentioning that I have not discussed *every* case of persistent content here, having left out conversational implicatures, the persistent implications associated with certain instances of intonational focus, clefts and pseudo-clefts, *too*, *even* and *almost*, to name just a few. But the empirical domain I delineate in this chapter represents not only a fairly large swath of persistent contents, but also reorganizes into the same subclass some persistent contents that are often viewed as distinct. In the ensuing chapters, I lay out a formal theory of some of the implicatures described here, and explore its predictions. The theory I propose is structured according to this taxonomy of implicatures: they are modeled as implications that, depending on whether their persistence is conventional, their status as speaker commitments, and their consistency with prior context, may update the context in a way that is not affected by entailment-modifying operators.

Chapter 3

Curryesque Categorical Grammar

The grammar formalism I describe in this chapter, called **Curryesque Categorical Grammar**, is a categorial grammar in the tradition of [Lambek 1958](#), as its name suggests. The term *Curryesque* refers to the fact that in this formalism, the mechanism for handling syntactic combinatorics is separate from the one that handles surface word order, following [Curry \(1961\)](#).

As such, this formalism is similar to other categorial frameworks that separate combinatorics and surface form. Two prominent examples are [de Groote’s \(2001\) *Abstract Categorical Grammars*](#) and the λ -grammars of [Muskens \(2001, 2007\)](#). It also represents an alternative to categorial frameworks like [Morrill’s \(1994\) *Type Logical Grammar*](#), [Moortgat’s \(1997\) *Categorical Type Logic*](#), and [Steedman’s \(2000\) *Combinatory Categorical Grammar*](#), all of which handle both combinatorics and word order using a single mechanism. The syntactic components of Curryesque Categorical Grammar resemble recent work on similar formalisms found in [Smith 2010](#), [Mihaliček 2012](#), and [Mihaliček and Pollard 2012](#). Since it is predicated by [Steedman’s](#) well-known formalism, I use the acronym **CyCG** to abbreviate *Curryesque Categorical Grammar*. I will use the term CyCG interchangeably to refer to the framework in a general sense and to an instance of a CyCG that describes a particular language.

Among the positive attributes of CyCG’s grammatical division of labor is the fact that it allows greater flexibility in writing lexical entries, since word order and combinatorics may diverge. Another feature is that a single CyCG syntax (word order/combinatorics pair) can correspond to multiple

semantics, in cases where semantic ambiguity can arise, such as multiple quantifier scope possibilities. The correspondence of a single surface word order with multiple semantic interpretations is achieved by an adaptation of a technique due to Oehrle (1994).

In §3.1, I discuss the motivations for the design choices behind CyCG, along with its formal underpinnings. The notion of a *sign*, a *context*, and a *lexicon* are discussed in §3.1.1, and the grammar rules of CyCG are given in §3.1.2. Then the following two sections discuss the word order component (§3.1.3) and semantics (§3.1.4) in more detail. Starting in §3.2, a fragment of English is laid out that demonstrates some of CyCG’s core capabilities. Quantifier scope ambiguities are discussed in §3.2.1, and extraction (both peripheral and medial) in §3.2.2. Finally, §3.3 gives a summary of this chapter.

3.1 A Logic of Signs

A CyCG is a deductive system for deriving representations of linguistic signs for a given natural language. CyCG proofs consist of triples of parallel proofs in the following three component systems:

1. A theory modeling the surface phonological forms of the language, expressed in (simple) type theory. This component is referred to interchangeably as the *phenogrammar*, *pheno logic*, or *concrete syntax*,
2. A tensor-implication logic of syntactic categories, called the *tectogrammar*, which captures the language’s abstract combinatorics, such as verb subcategorization, also called the *tecto logic* or *abstract syntax*, and
3. A theory that models the semantics of the language, expressed in type theory like the pheno logic. This component is called the *semantic logic* or simply the *semantics*.

Terms of the pheno and semantic logics are sometimes referred to respectively as *pheno terms* and *semantic terms*, with their types respectively *pheno types* and *semantic types*. Formulas in the tecto logic are sometimes called *tecto types*.

The respective type theories for the pheno logic and the semantics are distinct from one another in the sense that they use different sets

of nonlogical types and nonlogical constants (although they inherit the same *logical* type t and constants $*$ and $=_A$ from the underlying type theory). They may also have different axioms that control the behavior of the nonlogical constants. Appendix B contains a thorough overview of type theory, which is similar to the typed lambda calculus except that the relations of $\alpha\beta\eta$ -conversion are derived from axioms of the system rather than being stipulated in the metalanguage.

The tecto logic is an instantiation of tensor-implication logic with an arbitrary set of atomic formulas corresponding to the basic syntactic categories of the language being modeled. Tensor-implication logic, a subsystem of what is sometimes called the *multiplicative fragment of linear logic*, is described in detail in appendix A. This logic can be thought of as an expression of the *syntactic calculus* of Lambek 1958 in linear logic, except that there is only a single, undirected implication (\multimap) instead of the directed implications $/$ and \backslash . It can therefore be thought of as *unilinear logic*. Tensor-implication logic is a related system to intuitionistic propositional logic, but whereas a proof of the intuitionistic implication $A \rightarrow B$ is interpreted as a process for transforming any proof of A into a proof of B , the linear implication $A \multimap B$ maps any proof of A to a proof of B that only uses A once. Intuitionistic proofs are thought of as facts, linear proofs as pieces of data.

I observe the notational conventions for both tensor-implication logic and type theory that are spelled out in appendices A and B. In the tecto logic, types formed with \multimap are written right associatively, with $A \multimap B \multimap C$ abbreviating $A \multimap (B \multimap C)$. For the pheno logic and semantics, types formed with \rightarrow are written right-associatively but those formed with \times are written left-associatively (see §B.1). However, *terms* of application types, written $(f a)$, are left associative: $(f a b)$ abbreviates $((f a) b)$. Also, outermost parentheses in applications are often dropped. In abstractions, parentheses are abbreviated in the usual way, so that $\lambda_{x:A}.f$ is often written instead of $(\lambda_{x:A}.f)$, and the typing information is often dropped when it is available from context, so that just $\lambda_x.f$ is written instead. Finally, multiple variable bindings are sometimes collapsed onto a single λ , so that $\lambda_x\lambda_y.f$ becomes simply $\lambda_{xy}.f$. (Definition B.4 discusses term formation in type theory.)

3.1.1 Signs, Contexts, and the Lexicon

In this section I discuss the basic metalanguage notions of CyCG.

Definition 3.1 (Signs). A *sign* is represented in CyCG as a triple, written as

$$(3.1) \quad a : A ; B ; c : C ,$$

where $a : A$ is a declaration in the pheno logic with term a and type A , B is a tecto type, and $c : C$ is a semantic logic declaration of a term c having type C .

When the types are clear from context so that no confusion can arise, I sometimes suppress the type information and abbreviate signs of the form in (3.1) as simply

$$(3.2) \quad a ; B ; c .$$

When a sign is written in the form in (3.1), it is said to be in **long form**; a sign with elided type information, like the one in (3.2), is said to be in **short form**.

Definition 3.2 (Contexts). A *context* in CyCG is a (possibly empty) finite set of signs of the form

$$\{x_1 : A_1 ; B_1 ; y_1 : C_1, \dots, x_n : A_n ; B_n ; y_n : C_n\} .$$

The members of a CyCG context are called *hypotheses*. Every legal CyCG context Γ satisfies the following conditions:

1. The pheno and semantic term of each hypothesis in Γ is a variable in its respective instantiation of type theory, and
2. The set of variables is disjoint in the sense that each pheno variable and each semantic variable occur at most once in all the hypotheses in Γ .

Similarly to the contexts in type theory and tensor-implication logic, the context $\Gamma \dot{\cup} \Delta$ is the disjoint union of Γ and Δ , defined only if the variables occurring in the hypotheses in Γ and Δ are disjoint.

As for type-theoretic and tensor-implication contexts, $\Gamma \dot{\cup} \Delta$ is written Γ, Δ , and $\Gamma, a : A ; B ; c : C$ is shorthand for $\Gamma, \{a : A ; B ; c : C\}$. The curly brackets surrounding a context are very often elided.

As its name implies, a CyCG context is a combination of two type-theoretic contexts and one tensor-implication context in the form of a set of triples (CyCG signs). But here a conflict must be resolved: in type theory, a context is a *set* of variable declarations (definition B.5), while contexts in tensor-implication logic are *multisets* of formulas (definition A.2). CyCG takes a middle path between these. Its contexts can be permuted, but analogs of the type-theoretic structural rules of *weakening* and *contraction* are not available because the pheno and semantic variables must be disjoint. CyCG, as a result, keeps with the linear character of its tecto logic. In this way, it reflects the fact that natural language syntactic combinatorics are *resource-sensitive*: an English verb phrase requires exactly one subject to form a sentence.

CyCG grammaticality judgments are reminiscent of the derivability judgments of type theory and tensor-implication logic. They are notated in the form

$$\Gamma \vdash a : A ; B ; c : C ,$$

and are interpreted as saying that the sign $a : A ; B ; c : C$ is derivable in the context Γ . As for type theory and tensor-implication logic, when Γ is empty, the corresponding judgment is written $\vdash a : A ; B ; c : C$. If Γ is nonempty, the judgment is said to have an *undischarged hypothesis*.

Definition 3.3 (Lexicons). A CyCG *lexicon* is a finite set of nonlogical axioms satisfying the following:

1. The pheno term is a closed term of the pheno logic, and
2. The semantic term is a closed term of the semantic logic.

The members of a lexicon are called the *lexical entries*.

3.1.2 Grammar Rules

The CyCG lexicon is the starting point for the CyCG grammar rules in figure 3.1, which allow more complex signs to be derived based on the contents the lexicon. The Entry rule simply states that lexical entries can be used in derivations. The Trace rule differs from Entry in that lexical entries are nonlogical axioms, but instances of Trace introduce variables that are stored in the context. So, for the pheno and semantic components, Entry and Trace are just the respective CyCG analogs of the Const and Var

$$\begin{array}{c}
\vdash a : A ; B ; c : C \quad (\text{Entry}) \\
\\
x : A ; B ; y : C \vdash x : A ; B ; y : C \quad (\text{Trace}) \\
\\
\frac{\Gamma, x : A ; B ; y : C \vdash d : D ; E ; f : F}{\Gamma \vdash (\lambda_x d) : A \rightarrow D ; B \multimap E ; (\lambda_y f) : C \rightarrow F} \quad (\text{Extract}) \\
\\
\frac{\Gamma \vdash f : A \rightarrow B ; C \multimap D ; g : E \rightarrow F \quad \Delta \vdash a : A ; C ; b : E}{\Gamma, \Delta \vdash (f a) : B ; D ; (g b) : F} \quad (\text{Combine})
\end{array}$$

Figure 3.1: Grammar rules of Curryesque Categorical Grammar in natural deduction presentation. The sign in the Entry rule must be a lexical entry. The symbols a , c , d , f , and g are metavariables over terms, x and y are metavariables over type-theoretic variables, while A , B , C , D , E , and F range over types.

rules from type theory. The Trace rule’s tecto component is also analogous to the Id rule from tensor-implication logic, whereas the tecto component of a lexical entry effectively adds an atomic type to the underlying logic, but leaving it unmentioned in the context. The Extract rule combines the rule \rightarrow I from type theory and the rule \multimap I from tensor-implication logic, applying them in parallel in a way similar to the *pointwise abstraction* of Muskens (2001, 2007). Similarly, the Combine rule applies a functional sign with its argument, based on both the \rightarrow E and \multimap E rules of type theory and tensor-implication logic, respectively. Combine resembles Muskens’s *pointwise application*.

In CyCG, the definition of a *proof* is very similar to the way it is defined for its component logics (see definitions A.3 and B.23). A CyCG proof of the sign $\Gamma \vdash a : A ; B ; c : C$ is a natural deduction proof tree in which the root node’s label is $\Gamma \vdash a : A ; B ; c : C$, each leaf is an instance of either Entry or Trace, and every mother node is derived from its daughters by one of the rules in figure 3.1. As for type theoretic and tensor-implication proof trees, rule labels are optional. The management of contexts most resembles that of tensor-implication logic, since the rule of *permutation* is available, but neither of the rules of *weakening* or *contraction* from type theory can be used. CyCG contexts diverge from tensor-implication contexts in that a CyCG context is a *set* of hypotheses, whereas a tensor-implication context

is a *multiset* of formulas (see definitions A.2 and B.5). CyCG contexts are sets because occurrences of the same tecto formula are labeled by distinct variables.

As figure 3.1 shows, the grammar rules for CyCG are very general and simple. The only rules that do any real deductive work are Combine and Extract, and these are just analogs to lambda abstraction and application in type theory, or to \rightarrow introduction and elimination in tensor-implication logic. Note that there is no proof rule for handling products like the ones for tensor products in tensor-implication logic (\otimes I and \otimes E) or for cartesian products in type theory (\times I and \times E). However, nothing strictly rules out the use of the connectives \otimes and \times . For example, transitive verbs could be modeled by the (uncurried) tecto type $(NP \otimes NP) \rightarrow S$. But since this type is equivalent to its curried variant $NP \rightarrow NP \rightarrow S$, there is no immediate need to introduce the added complexity of products.

All of the descriptive richness of the grammar is located in the lexicon, where lexical entries may be arbitrarily complex as long as they can interact with the grammar rules. Because of this, CyCG qualifies as a *lexicalist* theory of grammar. With a notion of the lexicon so defined and the grammar rules, all the ingredients required to specify a CyCG are now available. They are as follows: a pheno logic (with nonlogical constants, types, and optional axioms governing the constants), a set of basic types for the tecto logic, a semantic logic (also with its own nonlogical constants, types, and axioms), and a lexicon.

3.1.3 Axiomatizing the Pheno Logic

The phenogrammatical component, which models surface forms, is a type-theoretic encoding of a logic of string concatenation. Accordingly, it makes use of a single nonlogical type: the type s of phonological strings.

To handle the concatenation, some constants and axioms are needed.

Definition 3.4 (Phenogrammatical Concatenation). The constant $\cdot : s \rightarrow s \rightarrow s$ denotes an operation that takes two strings to a third string, and the null phonological string constant is denoted by $e : s$. For readability, \cdot is written in infix rather than prefix notation, so that for all $s : s$ and $t : s$, I write $(s \cdot t)$ instead of $((\cdot s) t)$.¹ The following axioms govern phenogrammatical

¹Below, functions that are declared to be infix when they are defined are all written analogously.

concatenation.

$$(3.3) \quad \vdash \forall_{s:s} \forall_{t:s} \forall_{u:s} (s \cdot t) \cdot u = s \cdot (t \cdot u)$$

$$(3.4) \quad \vdash \forall_{s:s} (s \cdot \mathbf{e}) = s$$

$$(3.5) \quad \vdash \forall_{s:s} (\mathbf{e} \cdot s) = s$$

The β -normal forms of concatenated strings are stipulated as follows. Assuming the strings s , t , and u are in normal form, the concatenation $s \cdot (t \cdot u)$ is in normal form. The string s is the normal form of both $(s \cdot \mathbf{e})$ and $(\mathbf{e} \cdot s)$.

The first axiom, in (3.3), states that \cdot is associative. Because of this associativity, I usually suppress parentheses in pheno terms. The axioms in (3.4) and (3.5) together state that \mathbf{e} is a two-sided identity for \cdot , since concatenating any string with the empty string has no effect. The pheno logic, in essence, is an encoding of a free monoid on the set of phonological strings (*inhabitants* of type s in the sense used in §B.1).

3.1.4 An Agnostic Semantic Theory

The CyCG semantics is based on the possible worlds semantics in [Plummer and Pollard 2012](#), which is *agnostic* in the following sense. It is intentionally uncommitted with respect to whether propositions are defined as sets of possible worlds, or vice versa, or even in some other way. And relatedly, it is not committed to the conditions under which a proposition is true at a given world.

By its agnosticism, this semantics generalizes both [Montague's \(1973\)](#) well-known possible worlds semantics and offers a strategy for repairing some of its considerable foundational problems, which are discussed by [Thomason \(1980\)](#), [Muskens \(2005\)](#), and [Pollard \(2008a,b\)](#). However, although it is weaker than both [Montague](#) semantics and [Pollard's](#) hyper-intensional semantics, it remains strong enough for a theory of grammar because neither the internals of a proposition nor its truth at a given world is central to the task of modeling linguistic meanings.

Like the pheno logic, the semantics is an instantiation of type theory, but its types, constants, and axioms are different. In addition to the logical type t , the semantics additionally uses the nonlogical types e , p , and w , the types of individuals, propositions, and worlds, respectively. An important preliminary is the notion of meaning types and their extensions at a world.

Definition 3.5 (Meaning Types and Extensions). The set of *meaning types* is recursively defined as follows:

1. The types 1, e and p are meaning types, and
2. If A and B are meaning types, then so are $A \rightarrow B$ and $A \times B$.

The *extension type* of a meaning type A is denoted $\text{Ext}(A)$.

$$\begin{aligned}\text{Ext}(1) &=_{\text{def}} 1 \\ \text{Ext}(e) &=_{\text{def}} e \\ \text{Ext}(p) &=_{\text{def}} t \\ \text{Ext}(A \rightarrow B) &=_{\text{def}} A \rightarrow \text{Ext}(B) \\ \text{Ext}(A \times B) &=_{\text{def}} \text{Ext}(A) \times \text{Ext}(B)\end{aligned}$$

For each meaning type A , the nonlogical constant

$$@_A : A \rightarrow w \rightarrow \text{Ext}(A),$$

called the *extension function for A* , denotes the extension of an inhabitant of A at a world. The extension functions $@$ are written infix, similarly to phenogrammatical concatenation, and are subject to the following axioms.

$$(3.6) \quad \vdash \forall_{w:w}. (* @_1 w) = *$$

$$(3.7) \quad \vdash \forall_{x:e} \forall_{w:w}. (x @_e w) = x$$

$$(3.8) \quad \vdash \forall_{f:A \rightarrow B} \forall_{w:w}. (f @_{A \rightarrow B} w) = \lambda_{x:A}. (f x) @_B w$$

$$(3.9) \quad \vdash \forall_{c:A \times B} \forall_{w:w}. (c @_{A \times B} w) = \langle (\pi_1 c) @_A w, (\pi_2 c) @_B w \rangle$$

When the type is clear from context, the subscript is sometimes dropped from the extension function $@$ to reduce clutter.

In the case of propositions, the type of the extension function $@_p$ is $p \rightarrow w \rightarrow t$. That is, $@_p$ tests whether a proposition is true at a given world. Note that no axiom is given corresponding to the extension function $@_p$ for propositions. This is because, in the general case, agnostic semantics does not take a position on how the constant $@_p$ is axiomatized, only that it is available as part of the theory. Below, in §3.1.4.1, I discuss two possible specializations that define $@_p$ in different ways, with important empirical consequences.

Definition 3.6 (Meaning Equivalence). If A is a meaning type, two meanings $a : A$ and $b : A$ are equivalent if they have the same extension at every world. The relation of equivalence between meanings is written $\equiv_A : A \rightarrow A \rightarrow t$, and is subject to the axiom

$$(3.10) \quad \vdash \forall_{a:A} \forall_{b:A}. (a \equiv_A b) \Leftrightarrow \forall_{w:w}. ((a @_A w) = (b @_A w)).$$

With a constant for testing a proposition's truth at a world, the relation of propositional entailment can be defined.

Definition 3.7 (Entailment). Propositional entailment is encoded in the constant entails : $p \rightarrow p \rightarrow t$, which is subject to the axiom

$$(3.11) \quad \vdash \forall_{p:p} \forall_{q:p}. (p \text{ entails } q) \Leftrightarrow \forall_{w:w}. ((p @ w) \Rightarrow (q @ w)).$$

And so in this semantics, as usual, a proposition p entails another proposition q if q is true at every world where p is true.

Because of the way meaning equivalence is defined, equivalence of propositions can be characterized as mutual entailment, as the following theorem shows.

Theorem 3.8 (Propositional Equivalence as Mutual Entailment). *For all propositions $p : p$ and $q : p$, we have*

$$\vdash ((p \text{ entails } q) \wedge (q \text{ entails } p)) \Leftrightarrow (p \equiv_p q).$$

Proof. The proof is straightforward based on the definitions of [Meaning Equivalence](#) and [Entailment](#) (3.6 and 3.7), along with the [Identity of Biimplication](#) theorem (B.14). \square

The logic of propositions is then defined in terms of $@_p$.

Definition 3.9 (Propositional Connectives and Quantifiers). The following constants define the propositional connectives and quantifiers, where the

type A is a meaning type:

$\text{true} : p$	(a necessary truth)
$\text{false} : p$	(a necessary falsehood)
$\text{not} : p \rightarrow p$	(negation)
$\text{and} : p \rightarrow p \rightarrow p$	(conjunction)
$\text{implies} : p \rightarrow p \rightarrow p$	(implication)
$\text{or} : p \rightarrow p \rightarrow p$	(disjunction)
$\text{forall} : (A \rightarrow p) \rightarrow p$	(universal quantifier)
$\text{exists} : (A \rightarrow p) \rightarrow p$	(existential quantifier)

These constants are subject to the following axioms, which ensure that the propositional connectives behave analogously to their boolean counterparts. As above, the schematic metavariable A ranges over meaning types.

$$(3.12) \quad \vdash \forall_{w:w}. \text{true} @ w$$

$$(3.13) \quad \vdash \forall_{w:w}. \neg(\text{false} @ w)$$

$$(3.14) \quad \vdash \forall_{p:p} \forall_{w:w}. ((\text{not } p) @ w) \Leftrightarrow \neg(p @ w)$$

$$(3.15) \quad \vdash \forall_{p:p} \forall_{q:p} \forall_{w:w}. ((p \text{ and } q) @ w) \Leftrightarrow ((p @ w) \wedge (q @ w))$$

$$(3.16) \quad \vdash \forall_{p:p} \forall_{q:p} \forall_{w:w}. ((p \text{ implies } q) @ w) \Leftrightarrow ((p @ w) \Rightarrow (q @ w))$$

$$(3.17) \quad \vdash \forall_{p:p} \forall_{q:p} \forall_{w:w}. ((p \text{ or } q) @ w) \Leftrightarrow ((p @ w) \vee (q @ w))$$

$$(3.18) \quad \vdash \forall_{P:A \rightarrow p} \forall_{w:w}. ((\text{forall } P) @ w) \Leftrightarrow \forall_{x:A}. ((P x) @ w)$$

$$(3.19) \quad \vdash \forall_{P:A \rightarrow p} \forall_{w:w}. ((\text{exists } P) @ w) \Leftrightarrow \exists_{x:A}. ((P x) @ w)$$

(This axiomatization draws directly on [Plummer and Pollard 2012](#).) I observe the following notational shorthand for binding operators such as the quantifiers exists and forall . For A and B types, and for all terms $O : (A \rightarrow B) \rightarrow B$ and $\lambda_x.P : A \rightarrow B$ with $x : A$ a variable,

$$(3.20) \quad (O_{x:A} P) =_{\text{def}} (O \lambda_{x:A}. P).$$

This shorthand is intended to evoke the notation associated with the type-theoretic binding operators \forall and \exists . As for other binding operators, the type of the bound variable is often dropped, so that simply $(O_x P)$ is written in place of $(O_{x:A} P)$ when the type of the variable either is irrelevant or can be reconstructed from the surrounding terms. The abbreviation for

abstracts that uses $.$ in place of the outer parentheses is extended to the shorthand in (3.20), so that $O_x.P$ abbreviates $(O_x P)$.

For convenience, I also define the following operations on properties.

Definition 3.10 (Operations on Properties). The *property conjunction*

$$\text{that} : (e \rightarrow p) \rightarrow (e \rightarrow p) \rightarrow (e \rightarrow p)$$

is defined as

$$(3.21) \quad \text{that} =_{\text{def}} \lambda_{PQx}.(P x) \text{ and } (Q x) .$$

Similarly, *property negation*, whose type is

$$\text{non} : (e \rightarrow p) \rightarrow (e \rightarrow p)$$

gives the negated form of a specified property:

$$(3.22) \quad \text{non} =_{\text{def}} \lambda_{Px}.\text{not } (P x)$$

The property conjunction *that* is written infix, while property negation *non* is written prefix like other unary operators.

Due to the axioms in (3.12)–(3.19), together with the axiomatization of entails in (3.11), the type of propositions is interpreted as forming a preboolean algebra. Importantly, the interpretation of propositions does not form a full boolean algebra because the entails relation is not necessarily antisymmetric. So mutually entailing propositions do not have to be identical, although this may be the case in a particular instantiation of agnostic semantics (see §3.1.4.1, below).

Finally, the logic of propositions in definition 3.9 also implies that the set of propositions that are true at a given world is maximal consistent. (Here, and below, I do not distinguish between sets and their corresponding characteristic functions.)

Definition 3.11 (Facts at a World). The facts at a world w , denoted by the constant facts : $w \rightarrow p \rightarrow t$, are simply the propositions true at w :

$$(3.23) \quad \vdash \forall_{w:w}.(\text{facts } w) = \lambda_{p:p}.p @ w$$

Definition 3.12 (Maximal Consistency). The properties of being closed under entailment (ec) and conjunction (ac) are lambda-definable for a given set of propositions, as is the property of being the **ultimate arbiter** (ua).

$$\begin{aligned} \text{ec} &=_{\text{def}} \lambda_{s:p \rightarrow t} \forall_{p:p} \forall_{q:p}. ((s\ p) \wedge (p \text{ entails } q)) \Rightarrow (s\ q) \\ \text{ac} &=_{\text{def}} \lambda_{s:p \rightarrow t} \forall_{p:p} \forall_{q:p}. ((s\ p) \wedge (s\ q)) \Rightarrow (s\ (p \text{ and } q)) \\ \text{ua} &=_{\text{def}} \lambda_{s:p \rightarrow t} \forall_{p:p}. ((s\ p) \vee (s\ (\text{not } p))) \wedge \neg((s\ p) \wedge (s\ (\text{not } p))) \end{aligned}$$

Here, characteristic functions of sets of propositions, of type $p \rightarrow t$, are used in place of the sets themselves. The ua property says of a set of propositions that for any proposition p , it has either p or its denial as a member, but not both. The property of maximal consistency, written maxcons, is defined as the conjunction of these three:

$$(3.24) \quad \text{maxcons} =_{\text{def}} \lambda_{s:p \rightarrow t}. (\text{ec } s) \wedge (\text{ac } s) \wedge (\text{ua } s)$$

Then we have the following:

Theorem 3.13 (Maximal Consistency of Facts). *For any world w , we have*

$$\vdash \text{maxcons} (\text{facts } w).$$

That is, the set of facts at w is maximal consistent.

Proof. The properties of closure under entailment and propositional conjunction are immediate consequences of the axioms in (3.11) and (3.15). The property that for some world w , the set (facts w) is the ultimate arbiter for all propositions follows from the axiomatization of not in (3.14) and the fact that the term logic of the underlying type theory is classical. \square

Theorem 3.13 implies that the set (facts w) for some w is interpreted as an ultrafilter on the preboolean algebra of propositions.

3.1.4.1 Two Notable Sects

Plummer and Pollard (2012) also show two possibilities for strengthening the agnostic semantics discussed above by defining the extension function $@_p$ that says which propositions are true at which worlds. The first possibility is to redefine the type p as the type

$$p =_{\text{def}} w \rightarrow t$$

of sets of worlds. Then $@_p$ is defined by the *montagovian axiom*

$$(3.25) \quad \vdash \forall_{p:p} \forall_{w:w}. (p @ w) \Leftrightarrow (p w),$$

which implies that a proposition p is true at a world w if and only if w is a member of the set p .

Strengthened by the montagovian axiom, agnostic semantics becomes the type-theoretic encoding of Montague's (1973) possible worlds semantics due to Gallin (1975). Because the extension function $@_p$ is essentially just set membership, the logical connectives are interpreted as the familiar set-theoretic binary operations and entailment as the subset inclusion relation on the powerset of the set of worlds. As such, the entails relation gains the property of antisymmetry, and the logic of propositions is strengthened to a full boolean algebra. But then the semantics suffers from the well-known *granularity problem*: since mutually entailing propositions are identical sets of worlds, knowing that $2 = 2$ implies knowledge of every necessary truth, among other unwelcome implications (see Pollard, 2008b, for discussion).

An alternative is to strengthen agnostic semantics into one that does not identify mutually entailing propositions. This is accomplished by instead leaving p as a basic nonlogical type, and then adding axioms ensuring that the set of worlds and the set of maximal consistent sets of propositions is in one-to-one correspondence. The following axioms achieve this correspondence:

$$(3.26) \quad \vdash \forall_{v:w} \forall_{w:w}. ((\text{facts } v) = (\text{facts } w)) \Rightarrow v = w$$

$$(3.27) \quad \vdash \forall_{s:p \rightarrow t}. (\text{maxcons } s) \Rightarrow \exists_{w:w}. s = (\text{facts } w)$$

Axiom (3.26) says that two worlds are identical if the (maximal consistent) sets of facts associated with them are identical. The axiom in (3.27) states that every maximal consistent set is the set of facts at some world. Note that these two axioms, which Plummer and Pollard call the *tractarian axioms*, govern the behavior of the extension function $@_p$ because facts is defined in terms of it (see (3.23), above).

The resulting theory is essentially the hyperintensional semantics discussed by Pollard (2008a,b). The logic of propositions still functions as desired, but entailment remains merely a preorder, and therefore the granularity problem that plagues Montague semantics does not arise.

There is one technical difference between this strengthening of agnostic semantics and Pollard’s hyperintensional semantics. Pollard’s theory is built on the categorical type theory in Lambek and Scott 1986, in which subtyping is available. For Pollard, the type of worlds is defined as the subtype of $p \rightarrow t$ consisting of the maximal consistent sets of propositions. Here, as for Plummer and Pollard, the underlying type theory is not equipped with subtyping, but the tractarian axioms achieve the same result as Pollard’s axiomatization.

3.2 A Small Fragment of English

To show how CyCG works in practice, I start by analyzing the following simple examples:

(3.28) It snowed.

(3.29) Kim sneezed.

My aim in this section is to give some familiarity with CyCG equipped with a basic static semantics before moving on to a more involved theory in chapter 4. Below, §3.2.1 and §3.2.2 discuss quantifier scope ambiguity and some slightly more complicated instances of extraction.

The CyCG for handling (3.28) and (3.29), and the other examples in this section, is specified following §3.1.3 and §3.1.4, above, with a tecto logic that has the basic types NP, N, and S of noun phrases, common nouns, and sentences, respectively. The pheno logic has the single nonlogical type s of strings, along with the axiomatization of the concatenation operator \cdot in definition 3.4. The semantic theory is the agnostic semantics discussed in §3.1.4 above, with nonlogical types e , p , and w , and the logic of propositions as described in definition 3.9.

For (3.28), the lexicon needs to contain entries for both *It* and *snowed*. First, the basic type *It* is added to the tectogrammar, corresponding to the pleonastic *it* in English. Using this new type instead of the type NP of noun phrases avoids allowing the ungrammatical **There rains*.

(3.30) $\vdash \lambda_f.f \text{ it} : (s \rightarrow s) \rightarrow s ; (\text{It} \multimap S) \multimap S ; \lambda_P.P * : (1 \rightarrow p) \rightarrow p$

(3.31) $\vdash \lambda_{s,s}.s \cdot \text{snowed} : s \rightarrow s ; \text{It} \multimap S ; \lambda_u.\text{snow} : 1 \rightarrow p$

This lexicon models the pleonastic *It* as having the empty semantics $*$, the only inhabitant of the type-theoretic unit 1. Then *snowed* is treated

semantically as expecting a single argument of the unit type to yield the proposition *snow*. With the CyCG proof rules in place, this lexicon allows the following derivation of (3.28):

$$(3.32) \quad \frac{\vdash \lambda_f.f \text{ it}; (\text{It} \multimap S) \multimap S; \lambda_P.P * \quad \vdash \lambda_s.s \cdot \text{snowed}; \text{It} \multimap S; \lambda_u.\text{snow}}{\vdash ((\lambda_f.f \text{ it}) \lambda_s.s \cdot \text{snowed}); S; ((\lambda_P.P *) \lambda_u.\text{snow})} \text{ (Co)}$$

The root of this proof β -reduces to the normal form

$$\vdash \text{it} \cdot \text{snowed} : s; S; \text{snow} : p.$$

Note also that, since the tecto type of *snowed* is *It* and its semantics is $\lambda_u.\text{snow} : 1 \rightarrow p$, not just any noun phrase will do as the subject argument to *snowed*; it must be a CyCG sign that is tectogrammatically a pleonastic *it* and semantically the unit constant $*$.

The lexicon for *Kim sneezed* is also appropriately simple. The proper name *Kim* corresponds to the lexical entry

$$\vdash \text{Kim} : s; \text{NP}; k : e,$$

where *Kim* and *k* are nonlogical constants in their respective logics. Then the lexical entry for the intransitive verb *sneezed* is

$$\vdash \lambda_s.s \cdot \text{sneezed} : s \rightarrow s; \text{NP} \multimap S; \text{sneeze} : e \rightarrow p.$$

(Here the semantic term *sneeze* is in η -reduced form; see theorem B.15).

With this lexicon, CyCG can generate a sign corresponding to (3.29) as follows, using the short form for signs described in (3.2).

$$(3.33) \quad \frac{\vdash \lambda_s.s \cdot \text{sneezed}; \text{NP} \multimap S; \text{sneeze} \quad \vdash \text{Kim}; \text{NP}; k}{\vdash ((\lambda_s.s \cdot \text{sneezed}) \text{Kim}); S; (\text{sneeze } k)} \text{ (Co)}$$

This proof introduces the notational convention for proof labels that *Co* is written as an abbreviation for *Combine*. After performing β -reduction on the pheno term of the derived sign, the long normal form of the root label is

$$\vdash \text{Kim} \cdot \text{sneezed} : s; S; (\text{sneeze } k) : p.$$

In this analysis, the string in (3.29) is derived with the tecto type *S* of sentences, expressing the proposition that *Kim sneezed*, as desired.

Getting slightly more complicated, consider

(3.34) Lance rode a bike.

Similar to *Kim* and *sneezed*, above, the lexical entries corresponding to the proper name *Lance* and the common noun *bike* are as follows:

$$\begin{aligned} &\vdash \lambda_f.f \text{ Lance} : (s \rightarrow s) \rightarrow s; NP; \lambda_p.(Pl) : (e \rightarrow p) \rightarrow p \\ &\vdash \text{bike} : s; N; \text{bike} : e \rightarrow p \end{aligned}$$

Here, *Lance* is treated semantically as being *type raised*, applying a property to the constant $l : e$, with the semantic type of a generalized quantifier, like other noun phrases will be handled in CyCG. The transitive verb *rode* takes two NP arguments to form a sentence, so its lexical entry is

$$\vdash \lambda_{st}.t \cdot \text{rode} \cdot s : s \rightarrow s \rightarrow s; NP \multimap NP \multimap S; \text{rode} : e \rightarrow e \rightarrow p,$$

with the semantics *rode* in η -normal form. Lastly, *a* is semantically a generalized determiner, taking the denotation of a common noun to form a generalized quantifier (Barwise and Cooper, 1981), defined by analogy to Montague 1973 for the agnostic setting.

(3.35) $a =_{\text{def}} \lambda_{pQ}.\text{exists}_x.(Px) \text{ and } (Qx) : (e \rightarrow p) \rightarrow (e \rightarrow p) \rightarrow p$

Here, the shorthand exists_x is as defined in (3.20). Writing

$$QP =_{\text{def}} (NP \multimap S) \multimap S$$

to abbreviate the combinatorial type of a generalized quantifier, the lexical entry for *a* is

(3.36) $\vdash \lambda_{sf}.f(a \cdot s) : s \rightarrow (s \rightarrow s) \rightarrow s; N \multimap QP; a : (e \rightarrow p) \rightarrow (e \rightarrow p) \rightarrow p.$

This is in essence the same treatment of generalized determiner word order found in Oehrle 1994, in which the pheno of a generalized quantifier is ‘lowered’ into a λ -bound position in some function from strings to strings.

For readability, I split the proof corresponding to (3.34) into two parts: first the pheno, and then the tecto and semantics (but see figure 3.2 on page 75 for the entire proof with all three components). The pheno part of the

proof starts by forming the phonology for the generalized quantifier *a bike*:

$$(3.37) \quad \frac{\vdash \lambda_{sf}.f(a \cdot s) : s \rightarrow (s \rightarrow s) \rightarrow s \quad \vdash \text{bike} : s}{\vdash \lambda_f.f(a \cdot \text{bike}) : (s \rightarrow s) \rightarrow s} \text{ (Co)}$$

Here, and in the proofs that follow, I engage in the mild notational abuse that β -reduction is performed whenever possible, with the reduced form replacing its corresponding redex.

Because *a bike* needs to quantify into the verb's object position, the verb *rode* is first combined with a variable representing its object. The following proof shows this step.

$$(3.38) \quad \frac{\vdash \lambda_{st}.t \cdot \text{rode} : s \rightarrow s \rightarrow s \quad s : s \vdash s : s}{s : s \vdash \lambda_t.t \cdot \text{rode} \cdot s : s \rightarrow s} \text{ (Co)}$$

The sign derived in (3.38) is of the right type to be taken as *a bike*'s argument, however, making this combination would result in *a bike* becoming the subject of *rode*, not the object. So the next proof step is to provide *rode* with a variable representing its subject. Next, the object argument is extracted so that *a bike* can take the proper scope.

$$(3.39) \quad \frac{\begin{array}{c} (3.38) \\ \vdots \\ s : s \vdash \lambda_t.t \cdot \text{rode} \cdot s : s \rightarrow s \end{array} \quad t : s \vdash t : s}{\begin{array}{c} s : s, t : s \vdash t \cdot \text{rode} \cdot s : s \\ t : s \vdash \lambda_s.t \cdot \text{rode} \cdot s : s \rightarrow s \end{array}} \text{ (Co) (Ex)}$$

(Here the notational convention for proof labels is extended, with Ex abbreviating Extract.) Then *a bike* is quantified in.

$$(3.40) \quad \frac{\begin{array}{c} (3.37) \\ \vdots \\ \vdash \lambda_f.f(a \cdot \text{bike}) : (s \rightarrow s) \rightarrow s \end{array} \quad \begin{array}{c} (3.39) \\ \vdots \\ t : s \vdash \lambda_s.t \cdot \text{rode} \cdot s : s \rightarrow s \end{array}}{t : s \vdash t \cdot \text{rode} \cdot a \cdot \text{bike} : s} \text{ (Co)}$$

Then the subject argument is extracted so that *Lance* can be incorporated.

(3.41)

$$\begin{array}{c}
 \vdots \\
 \text{(3.40)} \\
 \frac{\vdots}{t : s \vdash t \cdot \text{rode} \cdot a \cdot \text{bike} : s} \text{ (Ex)} \\
 \frac{\vdash \lambda_f.f \text{ Lance} : (s \rightarrow s) \rightarrow s \quad \vdash \lambda_t.t \cdot \text{rode} \cdot a \cdot \text{bike} : s \rightarrow s}{\vdash \text{Lance} \cdot \text{rode} \cdot a \cdot \text{bike} : s} \text{ (Co)}
 \end{array}$$

This proof yields the surface form of (3.34).

Turning to the corresponding tecto and semantic parts of the proof, we start with the generalized quantifier *a* combining with its common noun argument *bike*, as before.

(3.42)

$$\frac{\vdash N \multimap QP; a : (e \rightarrow p) \rightarrow (e \rightarrow p) \rightarrow p \quad \vdash N; \text{bike} : e \rightarrow p}{\vdash QP; (a \text{ bike}) : (e \rightarrow p) \rightarrow p} \text{ (Co)}$$

Here, as above, the tecto type QP is an abbreviation for the generalized quantifier type $(NP \multimap S) \multimap S$. Next *rode* is passed its object argument.

(3.43)

$$\frac{\vdash NP \multimap NP \multimap S; \text{rode} : e \rightarrow e \rightarrow p \quad NP; x : e \vdash NP; x : e}{NP; x : e \vdash NP \multimap S; (\text{rode } x) : e \rightarrow p} \text{ (Co)}$$

And then its subject argument is provided, before an abstract is formed using the variable in object position.

(3.44)

$$\begin{array}{c}
 \text{(3.43)} \\
 \vdots \\
 \frac{NP; x : e \vdash NP \multimap S; (\text{rode } x) : e \rightarrow p \quad NP; y : e \vdash NP; y : e}{NP; x : e, NP; y : e \vdash S; (\text{rode } x y) : p} \text{ (Co)} \\
 \frac{\vdots}{NP; y : e \vdash NP \multimap S; \lambda_x.(\text{rode } x y) : e \rightarrow p} \text{ (Ex)}
 \end{array}$$

Next the generalized quantifier *a bike* takes scope (here the semantic type $(e \rightarrow p) \rightarrow p$ of *a bike* is omitted to save space).

(3.45)

$$\begin{array}{c}
 \text{(3.42)} \quad \quad \quad \text{(3.44)} \\
 \vdots \quad \quad \quad \vdots \\
 \frac{\vdash QP; (a \text{ bike}) \quad NP; y : e \vdash NP \multimap S; \lambda_x.(\text{rode } x y) : e \rightarrow p}{NP; y : e \vdash S; (a \text{ bike})_x.(\text{rode } x y) : p} \text{ (Co)}
 \end{array}$$

Finally, the subject trace is extracted so that *Lance* can take the resulting abstract as argument. In the following proof, the rule labels and the semantic type $(e \rightarrow p) \rightarrow p$ of *Lance* are elided as a space-saving measure.

(3.46)

$$\begin{array}{c}
 \text{(3.45)} \\
 \vdots \\
 \text{NP} ; y : e \vdash S ; (\text{a bike } \lambda_x.(\text{rode } x \ y)) : p \\
 \hline
 \vdash \text{QP} ; \lambda_P.(Pl) \quad \vdash \text{NP} \multimap S ; \lambda_y.(\text{a bike})_x.(\text{rode } x \ y) : e \rightarrow p \\
 \hline
 \vdash S ; (\text{a bike})_x.(\text{rode } x \ l) : p
 \end{array}$$

This proof shows that, as required, the combinatorial type of (3.34) is *S*.

This proof derives correct truth conditions for (3.34), since after expanding the definition of *a* and performing β -reduction, the term in the root label of (3.46) reduces to

$$\vdash \text{exists}_x.(\text{bike } x) \text{ and } (\text{rode } x \ l) : p .$$

The full derived CyCG sign analyzing (3.34) combines the pheno (3.41) and tecto/semantic (3.45) portions of the proof:

(3.47)

$$\vdash \text{Lance} \cdot \text{rode} \cdot \text{a} \cdot \text{bike} : s ; S ; (\text{a bike})_x.(\text{rode } x \ l) : p$$

This sign, which is in β -reduced form, has as its pheno the concatenated string *Lance rode a bike*, the surface form of (3.34), which is analyzed as having the syntactic type *S* of sentences and suitable truth conditions for its semantic proposition. Figure 3.2 on page 75 shows the full, combined CyCG proof for (3.34), which is partly β -expanded.

One thing to notice about the pheno and semantic proofs is that, since they are both terms of type theory, the CyCG proof rules that were needed to derive them can be reconstructed based on the structure of the terms themselves. For example, the fully β -expanded pheno term modeling (3.34) is

$$\vdash ((\lambda_f(f \text{ Lance})) \lambda_t.(((\lambda_{sf} f (\text{a} \cdot s)) \text{ bike}) \lambda_s.t \cdot \text{rode} \cdot s)) : s .$$

In this term, every application not occurring in a lexical entry term corresponds to an instance of the Combine rule, while every abstraction term that is not part of a lexical entry corresponds to an instance of Extract. The β -expanded semantic term in (3.46) is similar, with applications corre-

$$\begin{array}{c}
\frac{\vdash \lambda_{st}.f(a \cdot s); N \multimap QP; a \quad \vdash \text{bike}; N; \text{bike}}{\vdash \lambda_{st}.f(a \cdot \text{bike}); QP; (a \text{ bike})} \text{(Co)} \\
\\
\frac{\vdash \lambda_{st}.t \cdot \text{rode} \cdot s; NP \multimap NP \multimap S; \text{rode} \quad s; NP; x \vdash s; NP; x}{s; NP; x \vdash \lambda_t.t \cdot \text{rode} \cdot s; NP \multimap S; (\text{rode } x)} \text{(Co)} \\
\frac{s; NP; x, t; NP; y \vdash t \cdot \text{rode} \cdot s; S; (\text{rode } x y)}{t; NP; y \vdash \lambda_s.t \cdot \text{rode} \cdot s; NP \multimap S; \lambda_x.\text{rode } x y} \text{(Ex)} \\
\\
\vdots \\
\frac{\vdash \lambda_{st}.f(a \cdot \text{bike}); QP; (a \text{ bike}) \quad t; NP; y \vdash \lambda_s.t \cdot \text{rode} \cdot s; NP \multimap S; \lambda_x.\text{rode } x y}{t; NP; y \vdash ((\lambda_{st}.f(a \cdot \text{bike})) \lambda_s.t \cdot \text{rode} \cdot s); S; (a \text{ bike})_x.(\text{rode } x y)} \text{(Ex)} \\
\frac{\vdash \lambda_{st}.f \text{ Lance}; NP; \lambda_P.Pl \quad \vdash \lambda_t.((\lambda_{st}.f(a \cdot \text{bike})) \lambda_s.t \cdot \text{rode} \cdot s); S; ((\lambda_P.Pl) \lambda_y.(a \text{ bike})_x.(\text{rode } x y))}{\vdash ((\lambda_{st}.f \text{ Lance}) \lambda_t.((\lambda_{st}.f(a \cdot \text{bike})) \lambda_s.t \cdot \text{rode} \cdot s); S; ((\lambda_P.Pl) \lambda_y.(a \text{ bike})_x.(\text{rode } x y)))} \text{(Co)}
\end{array}$$

Figure 3.2: Full derivation of (3.34), in partially β -expanded short form. The proof tree is split into three subtrees to economize horizontal space, with the topmost proof as in (3.37) and (3.42), and the middle one as in (3.39) and (3.44). As above, the tecto type QP is an abbreviation for $(NP \multimap S) \multimap S$.

sponding to Combine and nonlexical abstractions to Extract:

$$\vdash ((\lambda_P(P)) \lambda_y.\text{exists } \lambda_x.(\text{bike } x) \text{ and } ((\lambda_x(\text{rode } x y)) x)) : p$$

The tecto logic, however, has no terms, and so no analogous reconstruction is available. Accordingly, for the remaining examples I sometimes show only tecto logic proofs, providing only the term structure as a reflection of the pheno and semantic proofs. Also, labels for proof rules are suppressed in what follows, since it is always possible to tell which rule was used: if the rule application is unary, it is an instance of Extract; the only binary rule application is the Combine rule.

Note that there is an alternative scoping for (3.34), one in which the subject trace is withdrawn first and *Lance* takes scope before *a bike*. But this other scoping produces the same sign as in (3.47), and is therefore not a true instance of ambiguity. I turn to quantifier scope ambiguities, in which a single syntax is paired with multiple semantics, in the next section.

3.2.1 Quantifier Scope Ambiguity

Consider the quantifier scope ambiguity in

(3.48) Every cyclist rode a bike.

This example is like (3.34), except that there are two possible truth conditions, depending on whether it is interpreted as saying each cyclist rode the same bike, or if the bike was possibly different for each one. As a preliminary, the generalized determiner *Every* is defined, like *a*, above, as an agnostic analog to Montague's (1973) definition.

$$(3.49) \text{ every} =_{\text{def}} \lambda_{PQ}.\text{forall}_x.(P x) \text{ implies } (Q x) : (e \rightarrow p) \rightarrow (e \rightarrow p) \rightarrow p$$

Then the additional lexical entries needed to analyze (3.48) are

$$(3.50) \quad \vdash \lambda_{sf}.f(\text{every} \cdot s) ; N \multimap QP ; \text{every}$$

for *Every*, and

$$(3.51) \quad \vdash \text{cyclist} : s ; N ; \text{cyclist} : e \rightarrow p$$

for *cyclist*. In these lexical entries, again, the tecto type QP abbreviates the combinatorial generalized quantifier type $(NP \multimap S) \multimap S$, and the semantic

terms are in η -normal form. Also, to save horizontal space, the types for the pheno and semantic terms of *Every* in (3.50) are not shown, but they are the same as the types for the generalized determiner a in (3.36), above.

These type definitions allow a derivation that proves (3.48) has the tecto type S. Starting again with the pheno, as above for (3.34), the transitive verb *rode* first takes its two trace arguments, and then its object trace is extracted.

$$(3.52) \quad \frac{\frac{\frac{\vdash \lambda_{st}.t \cdot \text{rode} \cdot s : s \rightarrow s \rightarrow s \quad s : s \vdash s : s}{s : s \vdash \lambda_t.t \cdot \text{rode} \cdot s : s \rightarrow s} \quad t : s \vdash t : s}{s : s, t : s \vdash t \cdot \text{rode} \cdot s : s} \quad t : s \vdash \lambda_s.t \cdot \text{rode} \cdot s : s \rightarrow s$$

Next the generalized quantifiers are formed. The pheno proof for *a bike* is in (3.37), and the proof for *Every cyclist* is as follows.

$$(3.53) \quad \frac{\vdash \lambda_{sf}.f(\text{every} \cdot s) : s \rightarrow (s \rightarrow s) \rightarrow s \quad \vdash \text{cyclist} : s}{\vdash \lambda_f.f(\text{every} \cdot \text{cyclist}) : (s \rightarrow s) \rightarrow s}$$

Then the generalized quantifier pheno term corresponding to *a bike* takes the object-extracted verb phrase as its argument, and the subject trace is extracted.

$$(3.54) \quad \frac{\frac{\frac{(3.37) \quad \vdots}{\vdash \lambda_f.f(a \cdot \text{bike}) : (s \rightarrow s) \rightarrow s} \quad \frac{(3.52) \quad \vdots}{t : s \vdash \lambda_s.t \cdot \text{rode} \cdot s : s \rightarrow s}}{t : s \vdash t \cdot \text{rode} \cdot a \cdot \text{bike} : s} \quad t : s \vdash \lambda_t.t \cdot \text{rode} \cdot a \cdot \text{bike} : s \rightarrow s$$

Lastly, the generalized quantifier *every* takes scope.

$$(3.55) \quad \frac{\frac{(3.53) \quad \vdots}{\vdash \lambda_f.f(\text{every} \cdot \text{cyclist}) : (s \rightarrow s) \rightarrow s} \quad \frac{(3.54) \quad \vdots}{\vdash \lambda_t.t \cdot \text{rode} \cdot a \cdot \text{bike} : s \rightarrow s}}{\vdash \text{every} \cdot \text{cyclist} \cdot \text{rode} \cdot a \cdot \text{bike} : s}$$

It is also possible to derive the surface form of (3.48) by inverting the order of the proofs in (3.53) and (3.53), extracting the subject trace first, rather than the object, and combining *every cyclist* with the subject-extracted verb phrase pheno before *a bike* takes the object-extracted pheno as its argument.

Either proof strategy is equivalent with respect to the derived β -normal pheno term because the positions of the subject and object are fixed in the pheno corresponding to *rode*, but as I discuss below, the scoping order has implications in the semantics.

In the tecto/semantic proof, *rode* is first provided with its two arguments, just as for the pheno. In the following proof, the semantic type $e \rightarrow e \rightarrow p$ of *rode* is omitted to save horizontal space.

(3.56)

$$\frac{\frac{\vdash NP \multimap NP \multimap S; \text{rode} \quad NP; x : e \vdash NP; x : e}{NP; x : e \vdash NP \multimap S; (\text{rode } x) : e \rightarrow p} \quad NP; y : e \vdash NP; y : e}{NP; x : e, NP; y : e \vdash S; (\text{rode } x y) : p}$$

The generalized determiners then combine with their common noun arguments to form generalized quantifiers, similarly to the pheno proofs in (3.53) and (3.37). The tecto/semantic proof for *a bike* is given above in (3.42), and the one for *Every* is below.

(3.57)

$$\frac{\vdash N \multimap QP; \text{every} : (e \rightarrow p) \rightarrow (e \rightarrow p) \rightarrow p \quad \vdash N; \text{cyclist} : e \rightarrow p}{\vdash QP; (\text{every cyclist}) : (e \rightarrow p) \rightarrow p}$$

Now, there is a choice in terms of the order in which to extract the traces in preparation for the generalized quantifiers *Every cyclist* and *a bike*. Extracting the object-position trace gives the ‘surface scope’ reading of (3.48), by first combining the generalized quantifier *a bike* with the object-extracted verb phrase:

(3.58)

$$\frac{\begin{array}{c} (3.42) \\ \vdots \\ \vdash QP; (\text{a bike}) \end{array} \quad \frac{\begin{array}{c} (3.56) \\ \vdots \\ NP; x : e, NP; y : e \vdash S; (\text{rode } x y) : p \end{array}}{NP; y : e \vdash NP \multimap S; \lambda_x. \text{rode } x y : e \rightarrow p}}{NP; y : e \vdash S; (\text{a bike})_x. \text{rode } x y : p}$$

And then the subject is extracted so that *Every cyclist* can take scope.

(3.59)

$$\frac{\begin{array}{c} (3.57) \\ \vdots \\ \vdash QP; (\text{every cyclist}) \end{array} \quad \frac{\begin{array}{c} (3.58) \\ \vdots \\ NP; y : e \vdash S; (\text{a bike})_x. \text{rode } x y : p \end{array}}{\vdash NP \multimap S; \lambda_y. (\text{a bike})_x. \text{rode } x y : e \rightarrow p}}{\vdash S; (\text{every cyclist})_y. (\text{a bike})_x. \text{rode } x y : p}$$

On the other hand, extracting the subject trace first allows the ‘reverse scope’ reading.

(3.60)

$$\begin{array}{c}
 \text{(3.57)} \quad \vdots \\
 \vdots \\
 \vdots
 \end{array}
 \quad
 \begin{array}{c}
 \text{(3.56)} \quad \vdots \\
 \vdots \\
 \vdots
 \end{array}
 \quad
 \frac{
 \text{NP} ; x : e, \text{NP} ; y : e \vdash S ; (\text{rode } x \ y) : p
 }{
 \text{NP} ; x : e \vdash \text{NP} \multimap S ; \lambda_y. \text{rode } x \ y : e \rightarrow p
 }$$

$$\text{NP} ; x : e \vdash S ; (\text{every cyclist})_y. \text{rode } x \ y : p$$

The object is extracted second, in preparation for *a bike*.

(3.61)

$$\begin{array}{c}
 \text{(3.42)} \quad \vdots \\
 \vdots \\
 \vdots
 \end{array}
 \quad
 \begin{array}{c}
 \text{(3.60)} \quad \vdots \\
 \vdots \\
 \vdots
 \end{array}
 \quad
 \frac{
 \text{NP} ; x : e \vdash S ; (\text{every cyclist})_y. \text{rode } x \ y : p
 }{
 \vdash \text{NP} \multimap S ; \lambda_x. (\text{every cyclist})_y. \text{rode } x \ y : e \rightarrow p
 }$$

$$\vdash S ; (\text{a bike})_x. (\text{every cyclist})_y. \text{rode } x \ y : p$$

And so the grammar proves that (3.48) has the combinatorial type of a sentence, and generates the two possible semantic scopings, but only a single surface form and tecto proof.

Note that in order to derive the correct surface form, the generalized quantifiers (*every cyclist*) must take the abstract in which y is the λ -bound variable. This is because the semantic variable y is paired with the pheno variable t ; the sign used in the instance of Trace in the proof that combines all three components of the grammar is $t : s ; \text{NP} ; y : e$. Similarly, *a bike* must take as its semantic argument an abstract in which x is bound, because the pheno variable corresponding to x is s . The implication is that, in order to derive the correct surface form for (3.48), the only truth conditions that can be derived are either the ‘surface’ or ‘reverse’ scope readings demonstrated in the proofs in (3.59) and (3.61). For example, with the lexicon as given above, no proof for (3.48) is available whose semantics are truth conditionally equivalent to those for *A bike rode every cyclist*.

Just as the tecto logic does not change depending on quantifier scope, the pheno string derived is also the same in both cases. This partially β -expanded pheno term corresponding to the semantic scoping in (3.59) shows why:

$$(3.62) \quad \vdash ((\lambda_f f(\text{every} \cdot \text{cyclist})) \lambda_t. ((\lambda_f f(\text{a} \cdot \text{bike})) \lambda_s. t \cdot \text{rode} \cdot s)) : s,$$

while the pheno term for the scoping in (3.61) is

$$(3.63) \quad \vdash ((\lambda_f f(a \cdot \text{bike})) \lambda_s.((\lambda_f f(\text{every} \cdot \text{cyclist})) \lambda_t.t \cdot \text{rode} \cdot s)) : s.$$

Both of these pheno terms have the same normal form

$$\vdash \text{every} \cdot \text{cyclist} \cdot \text{rode} \cdot a \cdot \text{bike} : s,$$

the correct analysis of the surface form in (3.48). The reason this surface form is the only one that can be derived is the same reason only two semantic scopings are available: the semantic variable x is paired with the pheno variable s , and y with t .

And so the two possible CyCG signs that are derivable for (3.48) are the following, in short form, with fully β -reduced pheno terms:

$$(3.64) \quad \vdash \text{every} \cdot \text{cyclist} \cdot \text{rode} \cdot a \cdot \text{bike} ; S ; (\text{every cyclist})_y.(a \text{ bike})_x.\text{rode } x \ y$$

$$(3.65) \quad \vdash \text{every} \cdot \text{cyclist} \cdot \text{rode} \cdot a \cdot \text{bike} ; S ; (a \text{ bike})_x.(\text{every cyclist})_y.\text{rode } x \ y$$

These signs point to a noteworthy difference between CyCG and categorial frameworks in which surface form and combinatorics are combined into a single grammar component. For CyCG, the same pheno term and tecto proof can correspond to different quantifier scopings, whereas single-component syntactic frameworks need two separate syntactic analyses to get the two scopings. The reason is that, for CyCG, the only possibilities for deriving the correct surface form involve both generalized quantifiers *Every cyclist* and *a bike* being ‘lowered’ in to the same surface position.

I note briefly that a second proof is available for the reverse scope case in (3.61), corresponding to the following tecto proof, where QP abbreviates $(NP \multimap S) \multimap S$, as before.

$$(3.57) \quad \begin{array}{c} \vdots \\ (3.42) \quad \vdash QP \end{array} \quad \frac{\frac{\vdash NP \multimap NP \multimap S \quad NP \vdash NP}{NP \vdash NP \multimap S}}{NP \vdash S} \quad \frac{\vdash QP \quad NP \vdash S}{\vdash NP \multimap S} \quad \frac{}{\vdash S}$$

But in this proof, the first generalized quantifier to scope *must* be the one corresponding to *Every cyclist*; otherwise, the incorrect surface form *A bike*

every cyclist rode results, which does not correspond to (3.48).² So this alternative tecto proof does not allow semantic ambiguity the way the proofs in (3.59) and (3.61) do, in which traces are used to fill both of the verb *rode*'s argument positions.

Note that, because CyCG quantifiers can take scope at any node whose tecto type is $NP \multimap S$, quantification in CyCG is essentially unconstrained. An examination of whether and how the possible scopings for quantifiers should be limited is beyond the scope of this thesis.

3.2.2 Peripheral and Medial Extraction

CyCG also handles complex noun phrases involving peripheral extraction, such as the relative clause

(3.66) A bike that Lance rode.

For (3.66), the only extension needed to the lexicon used above for (3.48) is for the relativizer *that*. To make this extension, we first define the semantics of *that* to use the property conjunction defined in (3.21). The lexical entry corresponding to *that* is

(3.67) $\vdash \lambda_{sf}.s \cdot \text{that} \cdot (f\ e) : s \rightarrow (s \rightarrow s) \rightarrow s; N \multimap (NP \multimap S) \multimap N; \text{that},$

where the semantic type $(e \rightarrow p) \rightarrow (e \rightarrow p) \rightarrow (e \rightarrow p)$ of *that* has been omitted to save horizontal space.

Based on this additional lexical entry, a proof for the relative clause in (3.66) is available. The proof starts by supplying a trace for the object argument to *rode*, then *Lance* as the subject, then extracting the trace. First, the pheno:

(3.68)

$$\frac{\vdash \lambda_f.f \text{ Lance} : (s \rightarrow s) \rightarrow s \quad \frac{\vdash \lambda_{st}.t \cdot \text{rode} \cdot s : s \rightarrow s \rightarrow s \quad s : s \vdash s : s}{s : s \vdash \lambda_t.t \cdot \text{rode} \cdot s : s \rightarrow s}}{s : s \vdash \text{Lance} \cdot \text{rode} \cdot s : s} \quad \frac{}{\vdash \lambda_s.\text{Lance} \cdot \text{rode} \cdot s : s \rightarrow s}$$

The tecto/semantic proof proceeds in parallel, where as before, the constant $l : e$ corresponds to *Lance* for simplicity, and *Lance*'s generalized quantifier

²While this surface form is incorrect for the example currently being considered, it might be desirable if the construction in question were a reduced relative or a topicalization.

semantic type is elided to save space.

$$(3.69) \quad \frac{\frac{\frac{\vdash \text{NP} \multimap \text{NP} \multimap \text{S}; \text{rode} \quad \text{NP}; x : e \vdash \text{NP}; x : e}{\vdash \text{QP}; \lambda_P.P \mid \quad \text{NP}; x : e \vdash \text{NP} \multimap \text{S}; (\text{rode } x) : e \rightarrow p}}{\text{NP}; x : e \vdash \text{S}; (\text{rode } x \mid) : p}}{\vdash \text{NP} \multimap \text{S}; \lambda_x.(\text{rode } x \mid) : e \rightarrow p}$$

Next, *that* takes *bike* as its first argument, and the proof term derived in (3.69) as its second. Afterward, the generalized determiner *a* takes *bike that Lance rode* as its argument. The pheno proof shows how the concrete syntax assigned to *that* ends up passing the empty string *e* to the function derived in (3.68). (Here the semantic type for that is omitted to conserve horizontal space.)

$$(3.70) \quad \frac{\frac{\vdash \lambda_{sf}.s \cdot \text{that} \cdot (f \text{ e}) \quad \vdash \text{bike} : s}{\vdash \lambda_f.\text{bike} \cdot \text{that} \cdot (f \text{ e}) : (s \rightarrow s) \rightarrow s} \quad \frac{\begin{array}{c} (3.68) \\ \vdots \end{array} \quad \vdash \lambda_s.\text{Lance} \cdot \text{rode} \cdot s : s \rightarrow s}{\vdash \text{bike} \cdot \text{that} \cdot \text{Lance} \cdot \text{rode} \cdot \text{e} : s}}$$

And then the pheno of the indefinite determiner combines with the pheno derived in (3.70).

$$(3.71) \quad \frac{\vdash \lambda_{sf}.f(a \cdot s) : s \rightarrow (s \rightarrow s) \rightarrow s \quad \frac{\begin{array}{c} (3.70) \\ \vdots \end{array} \quad \vdash \text{bike} \cdot \text{that} \cdot \text{Lance} \cdot \text{rode} \cdot \text{e} : s}{\vdash \lambda_f.f(a \cdot \text{bike} \cdot \text{that} \cdot \text{Lance} \cdot \text{rode} \cdot \text{e}) : (s \rightarrow s) \rightarrow s}}{\vdash \lambda_{sf}.f(a \cdot \text{bike} \cdot \text{that} \cdot \text{Lance} \cdot \text{rode} \cdot \text{e}) : (s \rightarrow s) \rightarrow s}$$

On the tecto/semantic side, the relativizer *that* first combines with the common noun *bike*. Here again, that's semantic type is elided.

$$(3.72) \quad \frac{\vdash \text{N} \multimap (\text{NP} \multimap \text{S}) \multimap \text{N}; \text{that} \quad \vdash \text{N}; \text{bike} : e \rightarrow p}{\vdash (\text{NP} \multimap \text{S}) \multimap \text{N}; (\text{bike that}) : (e \rightarrow p) \rightarrow (e \rightarrow p)}$$

Next, the relativizer combines with the result of the proof in (3.69). The semantic type of (bike that) is omitted here; it is $(e \rightarrow p) \rightarrow (e \rightarrow p)$.

$$(3.73) \quad \frac{\begin{array}{c} (3.72) \\ \vdots \\ \vdash (\text{NP} \multimap \text{S}) \multimap \text{N}; (\text{bike that}) \end{array} \quad \begin{array}{c} (3.69) \\ \vdots \\ \vdash \text{NP} \multimap \text{S}; \lambda_x.(\text{rode } x \text{ I}) : e \rightarrow p \end{array}}{\vdash \text{N}; (\text{bike that } \lambda_x.(\text{rode } x \text{ I})) : e \rightarrow p}$$

And finally, the indefinite determiner, whose semantic type is elided, takes the relativized common noun as its argument.

$$(3.74) \quad \frac{\begin{array}{c} (3.73) \\ \vdots \\ \vdash \text{N} \multimap \text{QP}; a \quad \vdash \text{N}; (\text{bike that } \lambda_x.(\text{rode } x \text{ I})) : e \rightarrow p \end{array}}{\vdash \text{QP}; (a (\text{bike that } \lambda_x.(\text{rode } x \text{ I}))) : (e \rightarrow p) \rightarrow p}$$

Thus, the proof yields a sign with the tecto type QP of generalized quantifiers.

With all three components derived, a β -reduced sign modeling (3.66) is

$$(3.75) \quad \vdash \lambda_f.f(a \cdot \text{bike} \cdot \text{that} \cdot \text{Lance} \cdot \text{rode} \cdot e); \text{QP}; a (\text{bike that } \lambda_x.(\text{rode } x \text{ I})),$$

written in short form. Recalling the discussion of normal forms in definition 3.4, the pheno component of this sign could be reduced further, to

$$\vdash \lambda_f.f(a \cdot \text{bike} \cdot \text{that} \cdot \text{Lance} \cdot \text{rode}) : (s \rightarrow s) \rightarrow s,$$

while the semantics could be further β -reduced to

$$\vdash \lambda_p.\text{exists}_x.((\text{bike } x) \text{ and } (\text{rode } x \text{ I}) \text{ and } (P x)) : (e \rightarrow p) \rightarrow p,$$

by expanding the definition of a in (3.35).

The analysis of (3.66) is used in the CyCG treatment of the instance of medial extraction in

$$(3.76) \quad \text{Every cyclist wanted a bike that Lance rode today.}$$

The lexicon extensions needed to begin analyzing (3.76) are straightforward. The transitive verb *wanted* is modeled similarly to *rode*, with the lexical

entry

(3.77)

$$\vdash \lambda_{st}.t \cdot \text{wanted} \cdot s : s \rightarrow s \rightarrow s ; NP \multimap NP \multimap S ; \text{wanted} : e \rightarrow e \rightarrow p ,$$

and the lexical entry corresponding to the sentential modifier *today* is

$$(3.78) \quad \vdash \lambda_s.s \cdot \text{today} : s \rightarrow s ; S \multimap S ; \lambda_p.(\text{today } p) : p \rightarrow p .$$

The pheno proof in (3.68) is modified, allowing *today*'s pheno to be incorporated.

(3.79)

$$\begin{array}{c} \vdash \lambda_{st}.t \cdot \text{rode} \cdot s : s \rightarrow s \rightarrow s \quad s : s \vdash s : s \\ \vdash \lambda_f.f \text{ Lance} \quad \hline s : s \vdash \lambda_t.t \cdot \text{rode} \cdot s : s \rightarrow s \\ \vdash \lambda_s.s \cdot \text{today} \quad \hline s : s \vdash \text{Lance} \cdot \text{rode} \cdot s : s \\ \hline s : s \vdash \text{Lance} \cdot \text{rode} \cdot s \cdot \text{today} : s \\ \hline \vdash \lambda_s.\text{Lance} \cdot \text{rode} \cdot s \cdot \text{today} : s \rightarrow s \end{array}$$

Here, *Lance*'s pheno type $(s \rightarrow s) \rightarrow s$ and *today*'s pheno type $s \rightarrow s$ is elided to save space.

Then the tecto/semantic proof in (3.69) is modified slightly to the following, which accounts for the presence of the sentential modifier *today*. First a sentence depending on the hypothesis $NP ; x : e$ is derived. (In this proof, the semantic types of *Lance* and *rode* are omitted.)

(3.80)

$$\begin{array}{c} \vdash NP \multimap NP \multimap S ; \text{rode} \quad NP ; x : e \vdash NP ; x : e \\ \vdash QP ; \lambda_p.Pl \quad \hline NP ; x : e \vdash NP \multimap S ; (\text{rode } x) : e \rightarrow p \\ \hline NP ; x : e \vdash S ; (\text{rode } x \text{ l}) : p \end{array}$$

And then *today*, whose semantic term is written in η -normal form, combines with the derived sentence, similarly to the pheno proof in (3.79).

$$(3.81) \quad \begin{array}{c} \vdash S \multimap S ; \text{today} : p \rightarrow p \quad \vdash \text{NP ; } x : e \vdash S ; (\text{rode } x \text{ l}) : p \\ \hline \text{NP ; } x : e \vdash S ; \text{today } (\text{rode } x \text{ l}) : p \\ \hline \vdash NP \multimap S ; \lambda_x.\text{today } (\text{rode } x \text{ l}) : e \rightarrow p \end{array}$$

The next part of the analysis is a proof resembling (3.58), with a subproof like (3.72) modified to contain the subproof (3.81) instead of (3.69).

First, the modified subproof, with semantic types omitted:

$$(3.82) \quad \frac{\frac{\vdash N \multimap (NP \multimap S) \multimap N; \text{that} \quad \vdash N; \text{bike}}{\vdash (NP \multimap S) \multimap N; (\text{bike that})} \quad \frac{\vdash NP \multimap S; \lambda_x.(\text{rode } x \text{ l})}{\vdash N; (\text{bike that } \lambda_x.\text{today } (\text{rode } x \text{ l}))}}{\vdash N; (\text{bike that } \lambda_x.\text{today } (\text{rode } x \text{ l}))} \quad (3.80)$$

Next the generalized quantifier *a* takes the whole relativized common noun as argument.

$$(3.83) \quad \frac{\vdash N \multimap QP : a \quad \vdash N; (\text{bike that } \lambda_x.(\text{rode } x \text{ l}))}{\vdash QP ; a (\text{bike that } \lambda_x.\text{today } (\text{rode } x \text{ l}))} \quad (3.82)$$

Finally, the following tecto proof shows the two quantifiers taking scope in the ‘surface scope’ configuration, with *Every cyclist* outscoping *a bike that Lance rode today*. The proof starts by scoping the generalized quantifier derived in (3.83), then extracting the subject-position trace. Here the proof of $(\text{wanted } x \text{ } y)$ that depends on two traces is similar to the proof in (3.56), and the semantic term derived in (3.83) is abbreviated to save space.

$$(3.84) \quad \frac{\frac{\vdash QP ; a (\text{bike that } \dots) \quad \frac{\vdash NP ; x, NP ; y \vdash S ; (\text{wanted } x \text{ } y)}{\vdash NP ; y \vdash NP \multimap S ; \lambda_x.(\text{wanted } x \text{ } y)}}{\vdash NP ; y \vdash S ; (a (\text{bike that } \lambda_x.\text{today } (\text{rode } x \text{ l})))_x.\text{wanted } x \text{ } y}}{\vdash NP \multimap S ; \lambda_y.(a (\text{bike that } \lambda_x.\text{today } (\text{rode } x \text{ l})))_x.\text{wanted } x \text{ } y} \quad (3.83)$$

Lastly, the generalized quantifier *Every cyclist* takes scope, with the term derived in (3.84) again abbreviated to save space.

$$(3.85) \quad \frac{\vdash QP ; (\text{every cyclist}) \quad \vdash NP \multimap S ; \lambda_y.(a (\text{bike that } \dots))}{\vdash S ; (\text{every cyclist})_y.(a (\text{bike that } \lambda_x.\text{today } (\text{rode } x \text{ l})))_x.\text{wanted } x \text{ } y} \quad \begin{matrix} (3.57) \\ \vdots \end{matrix} \quad \begin{matrix} (3.84) \\ \vdots \end{matrix}$$

And so the sentence in (3.76) has the tecto type of a sentence.

The semantic term derived in (3.85) builds on (3.75), adding in the semantics for *today*, and then scoping the generalized quantifiers *a bike that*

Lance rode today and *Every cyclist* over the transitive verb. Depending on the which takes scope over the verb first, two readings are available. The first reading corresponds to the scope configuration in (3.85); the second corresponds to the ‘reverse scope’ reading in which *a bike that Lance rode* outscopes *Every cyclist*. The tecto/semantic proof for the ‘reverse’ reading is just like the one in (3.85), except that the generalized quantifier term derived in (3.57) takes scope before the one derived in (3.83).

$$\begin{aligned} &\vdash (\text{every cyclist})_y. (a (\text{bike that } \lambda_x. \text{today} (\text{rode } x \mid)))_x. \text{wanted } x \mid y : p \\ &\vdash (a (\text{bike that } \lambda_x. \text{today} (\text{rode } x \mid)))_x. (\text{every cyclist})_y. \text{wanted } x \mid y : p \end{aligned}$$

There are also two possibilities for the pheno term that is derived, which builds on a pheno term similar to the one in (3.75), depending on the scoping order:

$$\begin{aligned} &\vdash (\lambda_f. f (\text{every} \cdot \text{cyclist}) \lambda_t. (\lambda_f. f (a \cdot \text{bike} \cdot \text{that} \cdot \text{Lance} \cdot \text{rode} \cdot \mathbf{e} \cdot \text{today}) \\ &\quad \lambda_s. t \cdot \text{wanted} \cdot s)) : s \\ &\vdash (\lambda_f. f (a \cdot \text{bike} \cdot \text{that} \cdot \text{Lance} \cdot \text{rode} \cdot \mathbf{e} \cdot \text{today}) \lambda_s. (\lambda_f. f (\text{every} \cdot \text{cyclist}) \\ &\quad \lambda_t. t \cdot \text{wanted} \cdot s)) : s \end{aligned}$$

But both of these reduce to the same form

$$\vdash \text{every} \cdot \text{cyclist} \cdot \text{wanted} \cdot a \cdot \text{bike} \cdot \text{that} \cdot \text{Lance} \cdot \text{rode} \cdot \mathbf{e} \cdot \text{today} : s,$$

the surface form of (3.76), with the null string marking the medial extraction site. So just as for the analysis of (3.48), there are two different semantic terms corresponding to the available scopings, but the surface string and the tecto proof are the same for both.

3.3 Summary

CyCG is a categorial grammar for analyzing natural language signs that adopts the strategy, dating back to Curry (1961), of splitting the syntax into surface form and underlying combinatorics. The component for deriving surface forms is fairly simple, consisting of a type-theoretic implementation of a monoid on the set of strings. The combinatorial component is also simple, represented in the multiplicative fragment of linear logic with a small number of basic types. I have also shown here how CyCG can be equipped

with an expressive semantics that is a generalization of both [Montague's \(1973\)](#) semantics and [Pollard's \(2008a\)](#) hyperintensional semantics.

There are, of course, differences between CyCG's approach to syntax and the approaches used by categorial grammar frameworks that are not in the Curryesque tradition. Whereas CyCG lets the pheno logic handle word order and leaves the combinatorics to the tecto, non-Curryesque categorial grammars combine all of the machinery for both into a single component, employing various tactics to get things to work smoothly. For Type Logical Grammars in the tradition of [Morrill \(1994\)](#), various structural rules and indexing schemes are often used. [Moortgat's \(1997\)](#) Categorical Type Logic, which is based on [Lambek's \(1958\)](#) syntactic calculus, can handle peripheral extraction but needs to specify special operations to handle cases of medial extraction. Combinatory Categorical Grammar ([Steedman, 2000](#)) employs an array of combinators to treat extraction.

I do not investigate here the comparative coverage of Curryesque categorial grammars versus the non-Curryesque ones, and note that in many cases the non-Curryesque methods likely represent alternative means of achieving similar results. But CyCG does have some attractive properties as a theory, as [Mihaliček and Pollard \(2012\)](#) have argued for a related framework. Its pheno and semantic components are implemented in a mainstream type theory. In the case of the pheno, the monoidal theory is extremely simple. As for the semantics, it is no more complicated in essence than the [Montague](#) semantics that enjoys so much familiarity among semanticists.

Lastly, the design choice of separating word order from combinatorics allows the grammar rules to be very austere, with just two rules (Combine and Extract) invoked to perform all of the computational work. This move also has the positive consequence of allowing grammar writers a great deal of freedom in terms of composing their lexical entries.

Chapter 4

Dynamic Categorical Grammar

To model natural language discourse, I supplant the semantic component of the Curryesque Categorical Grammar discussed in chapter 3 with a discourse semantics. The resulting extended formalism is called **Dynamic Categorical Grammar**, also referred to via the acronym **DyCG**.

As its name implies, DyCG is *dynamic* in the sense that utterances not only update the context of interpretation but also rely on it for their own interpretation. It therefore figures into the tradition of dynamic theories that dates back to Discourse Representation Theory (Kamp, 1981; Kamp and Reyle, 1993) and File Change Semantics (Heim, 1982, 1983a, 1992, also known as Context Change Semantics). But it also traces its lineage to the branch of dynamic theories that are additionally compositional in Montague’s (1973) sense, for example, the dynamic semantics due to Groenendijk and Stokhof (1990, 1991), Chierchia (1992, 1995), Muskens (1994, 1996), Beaver (2001), and de Groote (2006). DyCG is most similar to Muskens’s, Beaver’s, and de Groote’s semantics because it is expressed in pure type theory.

The core idea behind DyCG’s semantics is that the meanings of declaratives are not simply propositions, but functions that take an input context and produce an output context updated with their content. Contexts themselves are also functions, from a vector of entities to a proposition. Then discourse referents are modeled as indices into these entity vectors. The notion of *anaphoric accessibility* is handled by dynamic negation (definition 4.9), which limits the availability of discourse referents in its scope, because the other quantifiers are defined in terms of it. Indefinites introduce new

discourse referents by extending the domain of the entity vectors that are the domains of contexts.

In §4.1, I discuss some pretheoretical motivations for adopting a dynamic approach to semantics. The fundamentals of DyCG are given in §4.2, starting with some preliminary definitions (§4.2.1), and then giving the DyCG definitions of discourse contexts, the meanings of declaratives, and discourse updates (§4.2.1.1). In §4.2.2, I show how a static semantics like the one discussed in chapter 3 can be turned into a dynamic semantics via some simple translations. The dynamic counterparts of the static semantic operators are defined in §4.2.2.1, and then in §4.2.2.2, some dynamic notions of entailment are defined, along with an operator that selects the antecedents of anaphors. With these definitions in place, §4.3 demonstrates a small DyCG fragment of English, showing some basic examples first, then how quantifier scope (§4.3.1) and donkey anaphora (§4.3.2) are handled in DyCG. Some extensions to the CyCG deduction rules for modeling discourse are discussed in §4.4, and then §4.5 gives a chapter summary and a comparison between DyCG and other approaches to semantics.

4.1 Motivation for a Dynamic Approach

Adopting a dynamic semantics entails a more complicated formal theory than is necessary for the static, propositional semantics described for CyCG in chapter 3, for example. So some discussion of the motivations for taking on this additional complexity is warranted. Rather than characterizing meanings in terms of truth in some model of the world, dynamic semantics opts to model them in terms of their interaction with the context of interpretation. This philosophical take on meaning, pioneered by Karttunen (1974), Lewis (1979), Kamp (1981), and Heim (1982), among others, is more in line with the way speakers actually use language in context.

At the empirical level, static semantics suffers from the famous defect that it cannot handle so-called *donkey anaphora* (Geach, 1962) of the form

(4.1) If a farmer_{*i*} owns a donkey_{*j*}, he_{*i*} beats it_{*j*}.

Static semantic theories are hard pressed to capture the anaphora indicated in (4.1) because the scope of the variables representing the farmer and the donkey is ‘closed off’ by the clause subordinate to *If*. Rather than modeling them as scope-bound individual variables, the dynamic solution is to introduce *discourse referents* (Karttunen, 1976) representing the farmer

and the donkey into the context of interpretation. These discourse referents are then available for later reference by anaphoric pronouns.

For a broader consideration of implicatures in general, such as the one undertaken in this thesis, it is advantageous to have a formal model of the discourse context in the semantics itself. The use of a pronoun bears an implication about the state of the discourse context, namely, that a suitable antecedent is retrievable (see §2.2.1.1, above). But other implicatures also interact with the discourse context. For example, nominal appositives update the discourse context directly without being subject to entailment-modifying operators. And one reason that nonconventional implicatures can fail to persist is that their entailments are in conflict with the entailments present in the discourse context. Beaver (2001), following Heim (1983b, 1992), comes to a similar conclusion, that a notion of the discourse context is required for the proper modeling of presuppositions.

A notion of the context also allows a richer formal model than possible with a prominent alternative to dynamic semantics, exemplified in the work of Cooper (1979) and the *E-Type pronouns* of Evans (1980). In these theories, a pronoun's meaning is taken to be available from the ambient discourse context in an unspecified way, because they are couched in purely static semantics that do not formally account for the context. But not only are these theories necessarily less fine-grained in terms of what they can say about the discourse context, it is unclear how they would be extended to model implicatures in general.

Schlenker (2007), following on Geurts (1999, chapter 4), argues against a dynamic approach to implicature that follows Heim (1982, 1983b) on the grounds that the meanings of connectives and quantifiers are arbitrarily stipulated, and thus not explanatory. Instead, Schlenker argues for a purely static semantics in which the persistence of implicatures is derived from over-arching principles of *transparency*, *nontriviality*, and *constancy*. But unless static propositional logic is taken to have some kind of ontological primacy, the alternatives to Schlenker's assessment found in Dekker 2005 and Rothschild 2011 are compelling. For Dekker and Rothschild, connectives and quantifiers in a Heim-style dynamic semantics are not stipulative, but rather derivable from their truth conditions in concert with a notion of *linearity*: in an utterance of the form S_1 and S_2 , the content of S_1 updates the context *before* the content of S_2 . Static conjunction has no way to capture this, so an argument can be made that any static semantics is simply too coarse-grained to faithfully model language use in context.

4.2 A Compositional Dynamic Semantics

The dynamic semantics I will use to model senses and implicatures is an outgrowth of earlier work (Martin, 2012; Martin and Pollard, 2012a,b; Kierstead and Martin, 2012), which is in turn greatly based on de Groote’s (2006) compositional dynamic theory. It is most similar to Kierstead and Martin 2012 in that de Groote’s *continuations* are not used, either explicitly or by reducing a continuized type system to a direct-style syntax, as I explore in Martin 2010 using a variant of the $\lambda\mu$ -calculus (Parigot, 1992, 2000).

The type system, however, is equally complex as de Groote’s, but as I discuss below, this way of setting up a dynamic semantics is more in line with the way dynamic interpretation has been formally treated since Kamp (1981) and Heim (1982), with utterances updating an input context to produce an output context. Because of this treatment of utterances, the semantics I present here most closely resembles the formal theories in Muskens 1994, 1996, Beaver 2001, and de Groote 2006.

4.2.1 Preliminaries

The underlying type theory for this dynamic semantics is the dependent type theory λP_Σ discussed in appendix C, which allows functional types $\Pi_{x:A}.B$ and pair types $\Sigma_{x:A}.B$ in which the type B may depend on the variable $x : A$. As appendix C discusses in detail, these types are generalizations of the function and pair types in ordinary type theory (appendix B). Type-theoretic functions and pairs are still available as the special cases $A \rightarrow B$ and $A \times B$ in which B does not depend on an inhabitant of A (see equations (C.3) and (C.13)).

Since the grammar rules for DyCG, given in figure 3.1, use a different scheme for managing contexts than the ones for dependent type theory (figure C.1), there are many occurrences of the Weak and Conv rule from dependent type theory that are suppressed in DyCG derivations. Many proofs involving type formation are not shown, and the Abs rule is abbreviated for clarity by omitting the proof of the type’s derivability. The interested reader is referred to appendix C for technical details.

The dynamic elaboration is built on the static agnostic semantics in chapter 3, so the nonlogical types e and p , of individuals and propositions, are still used in addition to the logical type t . The dynamic semantics

additionally makes use of the type ω discussed in §C.3.1. The inhabitants of this type will be used as discourse referents. Accordingly, definition 3.5 is extended to add ω as a meaning type, with extension type

$$\text{Ext}(\omega) =_{\text{def}} \mathbf{n},$$

and an extension function $@_{\omega_n}$ for each type ω_n , axiomatized as

$$\vdash \forall_{n:\mathbf{n}} \forall_{i:\omega_n} \forall_{w:w}. (i @_{\omega_n} w) = (\text{nat } i),$$

where nat , defined in §C.3.1, maps each inhabitant of ω_n to its corresponding natural number.

4.2.1.1 Contexts, Contents, and Updates

The context of interpretation for a discourse consists of the propositions that the interlocutors mutually accept, parameterized by the entities that the propositions involve, whose identities are not necessarily specified or even known. Accordingly, contexts in DyCG are modeled as mappings from vectors of entities of arbitrary arity to propositions, with vectors defined as in definition C.3. For $n : \mathbf{n}$ a natural number, the type of n -**contexts** is

$$(4.2) \quad c_n =_{\text{def}} e^n \rightarrow \mathbf{p},$$

the type of functions taking an n -ary entity vector to a proposition. The entity vector that is the argument to a context can be seen as an analog to Heim's (1982) "sequences" and to the set of discourse referents accessible from a given discourse representation structure (Kamp, 1981). The **arity** of a context $c : c_n$, obtained via the function $|\cdot|_n : \Pi_{n:\mathbf{n}} \Pi_{c:c_n} . \mathbf{n}$, is simply the arity of its input vector. For each natural number n ,

$$(4.3) \quad |\cdot|_n =_{\text{def}} \lambda_{c:c_n} . n.$$

This function's natural number subscript is almost always dropped in practice.

The **empty context** $\tau : c_0$ is a special case of a context whose input vector is the empty entity vector and whose propositional content is just

the necessarily true proposition:

$$(4.4) \quad \mathbf{T} =_{\text{def}} \lambda_{\mathbf{x}^0}.\text{true}$$

This context is useful for evaluating the effects of discourse updates, as I discuss below.

A useful shorthand for writing variables of a vector type is to superscript the variable with the vector's arity the first time the variable appears bound. That is, abstracts of the form $\lambda_{\mathbf{x}:e^n}.a$ are written simply $\lambda_{\mathbf{x}^n}$, and then as just \mathbf{x} in the body a of the abstract. An example context is the following 2-context:

$$\lambda_{\mathbf{x}^2}.\text{(cyclist } \mathbf{x}_0 \text{) and (bike } \mathbf{x}_1 \text{) and (own } \mathbf{x}_1 \mathbf{x}_0 \text{)} : c_2$$

This context contains a conjoined proposition with the information that the first member \mathbf{x}_0 of the 2-ary vector parameter \mathbf{x} is a cyclist and that its second member \mathbf{x}_1 is a bike, and that the first member owns the second member. This is the context that might be used to model an utterance of the sentence *A cyclist owns a bike*, for example. This example clarifies the sense in which entity vectors serve the purpose of Karttunen's (1976) *discourse referents*: nothing needs to be known about the actual identities of the coordinates of \mathbf{x} , but any suitable vector must be of arity 2.

Definition 4.1 (Vectors of Meaning Types). For the case of vectors of type A^n whose factors are all of some meaning type A , the two clauses describing the set of meaning types in definition 3.5 is extended with the following:

3. If A is a meaning type and $n : n$, then A^n is a meaning type.

The extension type of a vector of meaning types is given by extending the Ext function, as follows, where A is a meaning type and n a natural number.

$$\text{Ext}(A^n) =_{\text{def}} \text{Ext}(A)^n$$

Finally, the axiomatization of the extension function $@$ is extended with a recursively defined axiom for vectors of meaning types, where the functions head and tail are as defined in (C.25) and (C.26), and the vector concatenation function \bullet is as defined in definition C.7.

$$\vdash \forall_{n:n} \forall_{\mathbf{x}:A^n} \forall_{w:w}. (\mathbf{x} @_{A^n} w) = ((\text{tail } \mathbf{x}) @_{A^{n-1}} w) \bullet ((\text{head } \mathbf{x}) @_A w)$$

It is sometimes useful to speak of contexts in general when the arity is not necessarily known or relevant. The corresponding type of contexts of

any arity is

$$c =_{\text{def}} \Sigma_{n:n} \cdot c_n ,$$

the dependent sum whose second component, of type c_n , depends on the natural number n that is its first component (see §C.2.1 for a discussion of dependent sum types). To keep the notation standard, we make the arity available for every $c : c$ by defining the function $|\cdot| : c \rightarrow n$ as simply the natural number that is its first component:

$$(4.5) \quad |\cdot| =_{\text{def}} \lambda_{c:c} . (\pi_1 c)$$

And so the arity of every $c : c$ is the arity $|(\pi_2 c)|$ of the n -context that is its second component, since the type c is the type of pairs of a natural number and a context of that arity.

The meanings of declarative utterances are modeled as **contents**, which for every natural number n have the type

$$(4.6) \quad k_n =_{\text{def}} \Pi_{c:c} \cdot c_{|c|+n} ,$$

where n denotes the number of discourse referents introduced by the content. For some n , this is the type of functions from contexts to contexts where the output context's arity is the arity of the input context plus n , the number of newly introduced discourse referents.

The number of discourse referents a content introduces is called its **degree**, and is available via the function $|\cdot|_n : \Pi_{n:n} \Pi_{k:k_n} \cdot n$, defined similarly to the arity function for contexts in (4.3), for n a natural number:

$$(4.7) \quad |\cdot|_n =_{\text{def}} \lambda_{k:k_n} \cdot n$$

Just as for the arity function for contexts, the subscript on the degree function is dropped in practice. Also similarly to the type c of contexts of any arity, the type

$$(4.8) \quad k =_{\text{def}} \Sigma_{n:n} \cdot k_n$$

is the type of contents of any degree, with the corresponding degree function $|\cdot| : k \rightarrow n$, defined as

$$(4.9) \quad |\cdot| =_{\text{def}} \lambda_{k:k} \cdot (\pi_1 k) .$$

(Compare with the arity function for contexts of any arity in (4.5).) Note that, because of the type constraints on k_n in (4.6), we have

$$|(k\ c)| = |k| + |c|$$

for every $k : k$ and $c : c$, as desired. In chapter 5, the type of contents will be redefined in order to separate the sense and implicature parts of utterance meaning.

To reduce notational clutter, shorthands for the application of a content to a context are defined as follows, similarly to (4.5):

$$\begin{aligned} (k\ c) &=_{\text{def}} (k\ (\pi_2 c)) && \text{for } k : k_n \text{ and } c : c, \\ (k\ c) &=_{\text{def}} ((\pi_2 k)\ c) && \text{for } k : k \text{ and } c : c_n, \\ (k\ c) &=_{\text{def}} ((\pi_2 k)\ (\pi_2 c)) && \text{for } k : k \text{ and } c : c. \end{aligned}$$

These definitions remove the need to always first invoke the second projection function π_2 when applying a content to an argument. Whether $k : k$, $k : k_n$, $c : c$, or $c : c_n$ is irrelevant, because the application $(k\ c)$ is always available.

The type u_n is the type of n -ary *updates*, also known as *context changes* (following Heim (1982)). For n a natural number, an update $u : u_n$ takes a context and returns another context whose arity is increased by n over the input context.

$$u_n =_{\text{def}} \Pi_{c:c} . c_{|c|+n}$$

Intuitively, an update is a content that has been proffered and accepted by the interlocutors. Accordingly, the grammar rules for composing discourses in §4.4, below, operate on updates, not on contents. Just as for contexts and contents, the type u is the type of updates of any arity:

$$u =_{\text{def}} \Sigma_{n:n} . u_n$$

Similar notational shorthand to that used for contents is also used for updates:

$$\begin{aligned} (u\ c) &=_{\text{def}} (u\ (\pi_2 c)) && \text{for } u : u_n \text{ and } c : c, \\ (u\ c) &=_{\text{def}} ((\pi_2 u)\ c) && \text{for } u : u \text{ and } c : c_n, \\ (u\ c) &=_{\text{def}} ((\pi_2 u)\ (\pi_2 c)) && \text{for } u : u \text{ and } c : c. \end{aligned}$$

As for contents, the intention is that applications can be used interchangeably for updates whether the arity is specified as a type parameter or not. In the following, I will sometimes drop the natural number subscript from the types for contexts, contents, and updates when it is either irrelevant or can be inferred from contexts.

4.2.2 Dynamicizing a Static Semantics

To give a mapping from the static CyCG semantics in chapter 3, it is first useful to recursively define, for every natural number n , the type of n -ary static properties as follows:

$$\begin{aligned} p_0 &=_{\text{def}} p \\ p_{n+1} &=_{\text{def}} e \rightarrow p_n \end{aligned}$$

This definition ensures that each type p_n is the type of functions from n entity arguments to a proposition. For example, the type of transitive verb meanings is $p_2 = e \rightarrow e \rightarrow p$. Clearly a nullary static property is just a static proposition.

The corresponding dynamic type hierarchy uses contents rather than propositions as the meanings of declaratives, and discourse referents (i.e., natural numbers) as arguments rather than entities. To begin defining the dynamic counterparts of static properties, we need to ensure that a dynamic property takes a natural number n to a content whose input context is of an arity large enough to map n to an entity. To this end, I define, for each natural number $m : n$, the type of contexts of arity at least m as

$$(4.10) \quad c_{\geq m} =_{\text{def}} \sum_{n:n} c_{m+n} ,$$

and then the type of contexts strictly greater than m is defined in terms of $c_{\geq m}$, as

$$(4.11) \quad c_{> m} =_{\text{def}} c_{\geq m+1} .$$

Finally, to find the greatest discourse referent among several arguments, I define the following function.

Definition 4.2 (Greatest Discourse Referent). For every natural number i , the function

$$\max_i : \omega^{i+1} \rightarrow \omega$$

selects the greatest coordinate from a list of $i + 1$ discourse referents, which cannot be empty because of the type constraints. Its natural number parameter is often dropped, and it is subject to the following axioms:

$$\begin{aligned} &\vdash \forall_{\mathbf{n}:\omega^1}.(\max \mathbf{n}) = \mathbf{n}_0 \\ &\vdash \forall_{i:\mathbf{n}} \forall_{\mathbf{n}:\omega^{i+2}}. \max(\text{tail } \mathbf{n}) < (\text{head } \mathbf{n}) \Rightarrow (\max \mathbf{n} = (\text{head } \mathbf{n})) \\ &\vdash \forall_{i:\mathbf{n}} \forall_{\mathbf{n}:\omega^{i+2}}. (\neg(\max(\text{tail } \mathbf{n}) < (\text{head } \mathbf{n}))) \Rightarrow ((\max \mathbf{n}) = \max(\text{tail } \mathbf{n})) \end{aligned}$$

In definition 4.2, the head and tail functions are as defined in (C.25) and (C.26), respectively. The first axiom is the base case, stating the obvious fact that the greatest coordinate in a one-element vector is the sole element. The other two axioms simply state that when the list of discourse referents is longer than one, the greatest coordinate is found by recursively comparing the head of the list with the greatest coordinate of the tail of the list.

Definition 4.3 (Dynamic Properties). The types of **dynamic properties** are defined as follows.

$$\begin{aligned} \mathbf{d}_{0,i} &=_{\text{def}} \mathbf{k}_i \\ \mathbf{d}_{*,i} &=_{\text{def}} 1 \rightarrow \mathbf{k}_i \\ \mathbf{d}_{1,i} &=_{\text{def}} \prod_{n:\omega} \prod_{c:\mathbf{c}_{>n}} \cdot \mathbf{c}_{|c|+i} \\ \mathbf{d}_{2,i} &=_{\text{def}} \prod_{m:\omega} \prod_{n:\omega} \prod_{c:\mathbf{c}_{>(\max m,n)}} \cdot \mathbf{c}_{|c|+i} \\ \mathbf{d}_{3,i} &=_{\text{def}} \prod_{k:\omega} \prod_{m:\omega} \prod_{n:\omega} \prod_{c:\mathbf{c}_{>(\max k,m,n)}} \cdot \mathbf{c}_{|c|+i} \end{aligned}$$

Here I engage in the notational abuse discussed in §C.3.1 of writing an inhabitant of ω interchangeably with the natural number it corresponds to. I also write, for example, k, m, n to denote the vector $\langle \langle *, k \rangle, m \rangle, n \rangle$. Informally, the general pattern for the types $\mathbf{d}_{m,i}$, where $m : \mathbf{n}$, is

$$\mathbf{d}_{m,i} =_{\text{def}} \prod_{n_1:\omega} \cdots \prod_{n_m:\omega} \prod_{c:\mathbf{c}_{>(\max n_1, \dots, n_m)}} \cdot \mathbf{c}_{|c|+i}.$$

That is, a dynamic property is a function from some number of discourse referents that returns a content whose input context must be at least large enough to have the greatest among those discourse referents in its

domain. For example, the dynamic type of transitive verbs is $d_{2,i}$ for some i . A nullary dynamic property, that is, a content, is called a **dynamic proposition**.

In practice, the subscript i representing the degree of the content is sometimes omitted when it is irrelevant, as is the information about the minimum size of the input context.

With a mapping from the static type hierarchy to the dynamic one in place, we can define the dynamic counterparts of n -ary static propositions.

Definition 4.4 (Dynamic n -ary Propositions). The dyn_n functions from p_n to $d_{n,i}$ handle the translation from static to dynamic recursively for each n . For the base case, define for static propositions p

$$\text{dyn}_0 =_{\text{def}} \lambda_{p:p_0} \lambda_{c:c} \lambda_{\mathbf{x}|\mathbf{c}|} . p ,$$

and informally, for each $n : \mathbf{n}$, define for relations $r : p_{n+1}$

$$\begin{aligned} \text{dyn}_{n+1} =_{\text{def}} & \lambda_{r:p_{n+1}} \lambda_{m:\omega} \lambda_{i_1:\omega \dots i_{n-1}:\omega} \lambda_{c:c_{>(\max m, i_1, \dots, i_{n-1})}} \lambda_{\mathbf{x}|\mathbf{c}|} . \\ & (\text{dyn}_n (r \mathbf{x}_m)) i_1 \dots i_{n-1} c \mathbf{x} . \end{aligned}$$

For meaning types $P : 1 \rightarrow p$ that expect an argument of the unit type 1, define

$$\text{dyn}_* =_{\text{def}} \lambda_{P:1 \rightarrow p} \lambda_{u:1} . \text{dyn}_0 (P u) .$$

As examples, we have

$$\begin{aligned} (\text{dyn}_* \text{snow}) &= \lambda_u \lambda_c \lambda_{\mathbf{x}|\mathbf{c}|} . (\text{snow } u) , \\ (\text{dyn}_0 \text{true}) &= \lambda_c \lambda_{\mathbf{x}|\mathbf{c}|} . \text{true} , \\ (\text{dyn}_1 \text{cyclist}) &= \lambda_n:\omega \lambda_{c:c_{>n}} \lambda_{\mathbf{x}|\mathbf{c}|} . (\text{cyclist } \mathbf{x}_n) , \text{ and} \\ (\text{dyn}_2 \text{ride}) &= \lambda_m:\omega \lambda_n:\omega \lambda_{c:c_{>(\max m, n)}} \lambda_{\mathbf{x}|\mathbf{c}|} . (\text{ride } \mathbf{x}_m \mathbf{x}_n) \end{aligned}$$

as the dynamic counterparts of *snow*, the necessarily true proposition *true*, the unary static property *cyclist*, and the binary static property *ride*. The notational convention of writing the dynamic counterpart of a static meaning in small caps is observed throughout. For example, for the

dynamic properties above, the following abbreviations are used:

$$\begin{aligned}\text{SNOW} &=_{\text{def}} (\text{dyn}_* \text{snow}) \\ \text{CYCLIST} &=_{\text{def}} (\text{dyn}_1 \text{cyclist}) \\ \text{RIDE} &=_{\text{def}} (\text{dyn}_2 \text{ride})\end{aligned}$$

4.2.2.1 Dynamic Connectives and Quantifiers

The dynamic counterparts of the static connectives and quantifiers must account for the way contextual interpretation functions, while still preserving the proposition-level effects of their static incarnations. The dynamic meanings of indefinites need to introduce discourse referents, and the dynamic version of conjunction must exhibit the sequential nature of discourse. Dynamic connectives and quantifiers also have to model the limited lifespan observed for discourse referents: a discourse referent introduced in the scope of a quantifier or negation is unavailable for later reference except under special circumstances.

As a preliminary to defining promotion to an update and dynamic conjunction, I first introduce a notational convenience for vectors. Rather than littering up the semantic terms with vector coordinate selections and prefixations, I write $\mathbf{x}^m, \mathbf{y}^n$ as a shorthand for variables of type e^{m+n} , denoting vectors that are the concatenation $\mathbf{x} \bullet \mathbf{y}$ of some $\mathbf{x} : e^m$ and $\mathbf{y} : e^n$. As a further shorthand, I sometimes write simply y to represent a vector of the form \mathbf{y}^1 where $\mathbf{y}_0 = y$. Finally, in view of proposition C.8, if $\mathbf{x} : e^0$ is the empty e-vector and \mathbf{y}^n is a vector variable, then I write just $\lambda_{\mathbf{y}^n}$ rather than $\lambda_{\mathbf{x}^0, \mathbf{y}^n}$ or $\lambda_{\mathbf{y}^n, \mathbf{x}^0}$.

The function cc is defined to promote a content to an update, also known as a context change.

Definition 4.5 (Context Change). The **context change function** cc , whose type is

$$\text{cc} : \Pi_{k:k} \cdot \mathbf{u}_{|k|} ,$$

is defined as

$$(4.12) \quad \text{cc} =_{\text{def}} \lambda_{k:k} \lambda_{c:c} \lambda_{\mathbf{x}^{[c]}, \mathbf{y}^{[k]}} \cdot (c \mathbf{x}) \text{ and } (k \ c \ \mathbf{x}, \mathbf{y}) .$$

This function takes a content and transforms it into an update that also incorporates the information in the input context into its output. It is used to promote a content to a discourse update after the corresponding utterance is accepted by an interlocutor. Although the input vector $\mathbf{x} : e^{|c|+|k|}$ to the resulting context contains all of the discourse referents already in the context c along with any introduced by the content k , only the prefix containing c 's discourse referents is provided as argument to c , as shown in equation (4.12), which conjoins two contexts asymmetrically.

Definition 4.6 (Dynamic Conjunction). The dynamic counterpart of the static conjunction and : $p \rightarrow p \rightarrow p$ is

$$(4.13) \quad \text{AND} : \Pi_{h:k} \Pi_{c:k} \cdot k_{|h|+|k|} ,$$

which reflects the fact the conjunction in discourse is asymmetric in the general case: the information in the second conjunct is interpreted in a context updated with the information in the left conjunct.

$$(4.14) \quad \text{AND} =_{\text{def}} \lambda_{h:k} \lambda_{c:k} \lambda_{c:c} \lambda_{\mathbf{x}^{|c|}, \mathbf{y}^{|h|}, \mathbf{z}^{|k|}} \cdot (h \ c \ \mathbf{x}, \mathbf{y}) \text{ and } (k \ (cc \ h \ c) \ \mathbf{x}, \mathbf{y}, \mathbf{z})$$

Note that k is evaluated in the context $(cc \ h \ c)$ and then applied to the full list of discourse referents $\mathbf{x}, \mathbf{y}, \mathbf{z} : e^{|c|+|h|+|k|}$, whereas h is evaluated in the input context, applied afterward to the vector \mathbf{x}, \mathbf{y} containing only the first $|c| + |h|$ discourse referents (since $|(h \ c)| = |c| + |h|$ by the definition of the type k_n in (4.6)).

To give an example of AND, first define the content

$$\text{RAIN} =_{\text{def}} (\text{dyn}_* \text{rain}) = \lambda_u \lambda_c \lambda_{\mathbf{x}^{|c|}} \cdot \text{rain} : d_*$$

similarly to the definition of SNOW : d_* , above, where rain : $1 \rightarrow p$ takes the unit $*$ to the proposition that it rained. Then the conjunction of $(\text{RAIN} \ *)$ with $(\text{SNOW} \ *)$, is as follows, where RAIN and SNOW are the corresponding

$\beta\eta$ -reduced forms:

$$\begin{aligned}
& \vdash (\text{RAIN AND SNOW}) \\
& = (\text{RAIN } c \mathbf{x}, \mathbf{y}) \text{ and } (\text{SNOW } (\text{cc RAIN } c) \mathbf{x}, \mathbf{y}, \mathbf{z}) \\
& = \lambda_c \lambda_{\mathbf{x}^{[c]}, \mathbf{y}^{[\text{RAIN}], \mathbf{z}^{[\text{SNOW}]}} \cdot (\text{RAIN } c \mathbf{x}, \mathbf{y}) \text{ and } ((\lambda_{c\mathbf{x}^{[c]}} \text{SNOW}) (\text{cc RAIN } c) \mathbf{x}, \mathbf{y}, \mathbf{z}) \\
& = \lambda_c \lambda_{\mathbf{x}^{[c]}, \mathbf{y}^{[\text{RAIN}], \mathbf{z}^{[\text{SNOW}]}} \cdot (\text{RAIN } c \mathbf{x}, \mathbf{y}) \text{ and } ((\lambda_{\mathbf{x}^{[(\text{cc RAIN } c)]}} \text{SNOW}) \mathbf{x}, \mathbf{y}, \mathbf{z}) \\
& = \lambda_c \lambda_{\mathbf{x}^{[c]}, \mathbf{y}^{[\text{RAIN}], \mathbf{z}^{[\text{SNOW}]}} \cdot (\text{RAIN } c \mathbf{x}, \mathbf{y}) \text{ and snow} \\
& = \lambda_c \lambda_{\mathbf{x}^{[c]}, \mathbf{y}^{[\text{RAIN}], \mathbf{z}^{[\text{SNOW}]}} \cdot ((\lambda_{c\mathbf{x}^{[c]}} \text{rain}) c \mathbf{x}, \mathbf{y}) \text{ and snow} \\
& = \lambda_c \lambda_{\mathbf{x}^{[c]}, \mathbf{y}^{[\text{RAIN}], \mathbf{z}^{[\text{SNOW}]}} \cdot \text{rain and snow} : k_0
\end{aligned}$$

This example demonstrates how the input context to the second conjunct `SNOW` is determined by `(cc RAIN c)`.

In the case of the dynamicization of the static existential quantifier exists, first we define the following.

Definition 4.7 (Extension Function). The **extension function**

$$(\cdot)^+ : \Pi_{c:c \cdot c_{|c|+1}},$$

which extends a context to the next higher arity. This function is defined as

$$(\cdot)^+ =_{\text{def}} \lambda_{c:c} \lambda_{\mathbf{x}^{[c]}, \mathbf{y}} \cdot c \mathbf{x}.$$

That is, the context c^+ is the context just like c except that its vector argument has an extra coordinate. When multiple applications of the extension function are needed, the shorthand c^{++} is often written for $(c^+)^+$, the result of applying the extension function to the context c^+ .

Definition 4.8 (Dynamic Existential). For each $i : n$, the **dynamic existential quantifier**

$$(4.15) \quad \text{EXISTS} : \Pi_{D:d_{1,i}} \cdot k_{i+1},$$

the dynamic counterpart of the static existential exists, takes a unary dynamic property to a dynamic proposition (that is, a content) of degree i . Since **EXISTS** introduces a discourse referent, it is defined to ensure that the resulting dynamic proposition is type-constrained to increase the arity of the context by one: note that the domain of the resulting content is a context of arity $|c| + 1$.

$$(4.16) \quad \text{EXISTS} =_{\text{def}} \lambda_D \lambda_c. D \mid c \mid c^+$$

Since the natural number $|c|$, the arity of c , is the index of the newly-added discourse referent in the context c^+ , **EXISTS** has the effect of simultaneously extending its context argument c and passing the new discourse referent to its dynamic property argument D . In this way, the **EXISTS** captures the *novelty condition* associated with indefinites that is discussed in §2.2.1.1.

As an example, consider the dynamic existential quantifier applied to the dynamic property **CYCLIST**, whose bound variables have been renamed via α -conversion to avoid confusion:

$$\begin{aligned} \vdash \text{EXISTS CYCLIST} &= \lambda_c. \text{CYCLIST} \mid c \mid c^+ \\ &= \lambda_c. (\lambda_n \lambda_{d:c>n} \lambda_{\mathbf{x}|d|} (\text{cyclist } \mathbf{x}_n) \mid c \mid c^+) \\ &= \lambda_c. (\lambda_{d:c>|c|} \lambda_{\mathbf{x}|d|} (\text{cyclist } \mathbf{x}_{|c|})) c^+ \end{aligned}$$

Note that, although the λ -bound context d does not appear free in the term $(\text{cyclist } \mathbf{x}_{|c|})$, the supplied argument c^+ still has the effect of constraining the arity of the vector \mathbf{x} to be $|c^+| = |c| + 1$. This is due to the dependently typed definition of k_n in (4.6), and is also required by the type constraints for **EXISTS** in (4.15). Then the reduced form is on the right side of the equality symbol in

$$\vdash \text{EXISTS CYCLIST} = \lambda_{c:c} \lambda_{\mathbf{x}|c|} \lambda_{y}. \text{cyclist } y : k_1 .$$

Applying **EXISTS CYCLIST** to the empty context τ yields

$$\begin{aligned} \vdash ((\text{EXISTS CYCLIST}) \tau) &= \lambda_{\mathbf{x}^0} \lambda_{y}. (\text{cyclist } y) \\ &= \lambda_{y^1}. (\text{cyclist } \mathbf{y}_0) : c_1 , \end{aligned}$$

since $|\tau| = 0$. Note that by proposition C.6, the term $\lambda_{y^1}. (\text{cyclist } \mathbf{y}_0)$ is interderivable with **cyclist**.

In addition to negating the propositional content in its scope, dynamic negation must also limit the accessibility of any discourse referents introduced in its scope. This second requirement is accounted for in DyCG negation using the propositional existential quantifier defined in equation (3.19), in a way reminiscent of Heim's (1982) "existential closure."

Definition 4.9 (Dynamic Negation). The dynamic negation

$$\text{NOT} : k \rightarrow k_0 ,$$

defined as

$$(4.17) \quad \text{NOT} =_{\text{def}} \lambda_{k:k} \lambda_{c:c} \lambda_{x:|c|} . \text{not exists}_{y^{|k|}} . (k \text{ c } x, y) ,$$

existentially binds all of the discourse referents introduced within its scope, then negates the resulting proposition.

Notice that the requirement of the type of NOT that its output content be of degree 0 is observed by this axiom. In the case when $|k| > 0$, all of the discourse referents introduced by k are bound by exists, implying the following.

Theorem 4.10. *For every $k : k$, we have $\vdash |(\text{NOT } k)| = 0$.*

Proof. This follows immediately from the definition of NOT in 4.9, since for any $k : k$, $(\text{NOT } k)$ is a content that takes a context to another context of the same arity. \square

In case $|k| = 0$, we have theorem 4.12, based on the lemma below.

Lemma 4.11. *If A is a meaning type, then for every $p : p$ in which $x : A$ is not free, we have $\vdash (\text{exists}_x p) \equiv p$.*

Proof. Let $a : t$ such that there is no free occurrence of x in a . Then we have

$$\vdash (\exists_x a) = a$$

by "Rule C" (Andrews, 2002, theorem 5245). Note that by axiom (3.19), universal instantiation, and universal generalization,

$$\vdash \forall_{p:p} \forall_{w:w} . ((\text{exists}_x p) @ w) \Leftrightarrow \exists_y . ((\lambda_x . p \ y) @ w) .$$

Now let $p : \mathbf{p}$ where $x : A$ is not free in p , for some meaning type A . By β -reduction, the substitution rule, universal instantiation, and theorem B.14, it follows that

$$\vdash \forall_{w:w}. ((\text{exists}_x p) @ w) = (p @ w).$$

Then we have

$$\vdash (\text{exists}_x p) \equiv p$$

by meaning equivalence (definition 3.6). \square

Theorem 4.12. *If $k : k_0$, then $\vdash (\text{NOT } k) \equiv \lambda_c \lambda_{\mathbf{x}|c|}. \text{not } (k \text{ c } \mathbf{x})$.*

Proof. Let $k : k_0$. Then by (4.17), we have

$$(\text{NOT } k) = \lambda_c \lambda_{\mathbf{x}|c|}. \text{not exists}_{\mathbf{y}^0}. (k \text{ c } \mathbf{x}, \mathbf{y}).$$

Invoking the substitution rule, we arrive at

$$\begin{aligned} & \vdash \lambda_c \lambda_{\mathbf{x}|c|}. \text{not exists}_{\mathbf{y}^0}. (k \text{ c } \mathbf{x}, \mathbf{y}) \\ & = \lambda_c \lambda_{\mathbf{x}|c|}. \text{not exists}_{\mathbf{y}^0}. (k \text{ c } \mathbf{x}) && \text{(by proposition C.8)} \\ & \equiv \lambda_c \lambda_{\mathbf{x}|c|}. \text{not } (k \text{ c } \mathbf{x}) && \text{(by lemma 4.11)} \end{aligned}$$

\square

Theorem 4.13. *If $k : k_0$, then $\vdash (\text{NOT } (\text{NOT } k)) \equiv k$.*

Proof. Let $k : k_0$. Then by theorem 4.12,

$$\vdash \text{NOT } (\text{NOT } k) \equiv \lambda_c \lambda_{\mathbf{x}|c|}. \text{not } (\text{not } (k \text{ c } \mathbf{x})).$$

Now, for any $p : \mathbf{p}$ and $w : w$, we have

$$\begin{aligned} & \vdash (\text{not } (\text{not } p)) @ w = (\neg (\neg (p @ w))) && \text{(by axiom (3.14))} \\ & = p @ w, \end{aligned}$$

since the underlying type theory is classical (see theorem B.17). Then by meaning equivalence (definition 3.6),

$$\vdash \forall_{p:\mathbf{p}}. (\text{not } (\text{not } p)) \equiv p.$$

Therefore for $k : k_0$,

$$\begin{aligned} \vdash \text{NOT} (\text{NOT } k) &\equiv \lambda_c \lambda_{\mathbf{x}|c|}. (k \text{ } c \text{ } \mathbf{x}) \\ &\equiv k \end{aligned}$$

by substitution and η -reduction. \square

Some examples illustrate the effect of dynamic negation on a content. For a simple example, applying NOT to (SNOW *) gives

$$\begin{aligned} \vdash \text{NOT} (\text{SNOW } *) &= \lambda_c \lambda_{\mathbf{x}|c|}. \text{not exists}_{\mathbf{y}^0}. (\text{SNOW } c \text{ } \mathbf{x}, \mathbf{y}) \\ &= \lambda_c \lambda_{\mathbf{x}|c|}. \text{not exists}_{\mathbf{y}^0}. ((\lambda_d \lambda_{\mathbf{z}|d|}. \text{snow}) c \text{ } \mathbf{x}, \mathbf{y}) \\ &= \lambda_c \lambda_{\mathbf{x}|c|}. \text{not exists}_{\mathbf{y}^0}. \text{snow} \\ &\equiv \lambda_c \lambda_{\mathbf{x}|c|}. \text{not snow} : k_0, \end{aligned}$$

by theorem 4.12.

A slightly more complicated example demonstrates how NOT limits the accessibility of discourse referents in its scope. The dynamic negation of (EXISTS CYCLIST) : k_1 is as follows, where, as above, α -conversion has been used to avoid confusion:

$$\begin{aligned} \vdash \text{NOT} (\text{EXISTS CYCLIST}) &= \lambda_c \lambda_{\mathbf{x}|c|}. \text{not exists}_{\mathbf{y}^1}. ((\text{EXISTS CYCLIST}) c \text{ } \mathbf{x}, \mathbf{y}) \\ &= \lambda_c \lambda_{\mathbf{x}|c|}. \text{not exists}_{\mathbf{y}^1}. ((\lambda_d \lambda_{\mathbf{z}|d^+|}. \text{cyclist } \mathbf{z}|d|) c \text{ } \mathbf{x}, \mathbf{y}) \\ &= \lambda_c \lambda_{\mathbf{x}|c|}. \text{not exists}_{\mathbf{y}^1}. ((\lambda_{\mathbf{z}|c^+|}. \text{cyclist } \mathbf{z}|c|) \mathbf{x}, \mathbf{y}) \\ &= \lambda_c \lambda_{\mathbf{x}|c|}. \text{not exists}_{\mathbf{y}^1}. \text{cyclist } (\mathbf{x}, \mathbf{y})|c| \\ &= \lambda_c \lambda_{\mathbf{x}|c|}. \text{not exists}_{\mathbf{y}^1}. \text{cyclist } \mathbf{y}_0 : k_0 \end{aligned}$$

This implies that NOT (EXISTS CYCLIST) is interderivable with

$$\lambda_c \lambda_{\mathbf{x}|c|}. \text{not (exists cyclist)} : k_0,$$

by proposition C.6. And so the argument to cyclist, the only coordinate of the 1-ary vector \mathbf{y} , is not available outside the scope of NOT because it is bound by the instance of exists. The type constraints associated with EXISTS in (4.15) are preserved because the arity of \mathbf{x}, \mathbf{y} is $|\mathbf{x}| + 1$.

The definitions of AND, EXISTS, and NOT allow a full dynamic version of the logic of propositions defined in definition 3.9 to be defined, with

the remaining connectives and quantifier are defined in terms of these first three.

Definition 4.14 (Dynamicization of the Static Logic of Propositions). The types of the dynamicized connectives and quantifiers are as follows:

$$\begin{aligned} \text{NOT} &: k \rightarrow k_0 \\ \text{AND} &: \Pi_{h:k} \Pi_{k:k} k_{|h|+|k|} \\ \text{IMPLIES} &: k \rightarrow k \rightarrow k_0 \\ \text{OR} &: k \rightarrow k \rightarrow k_0 \\ \text{FORALL} &: d_1 \rightarrow k_0 \\ \text{EXISTS} &: d_1 \rightarrow k_1 \end{aligned}$$

Then the correspondence between the static and dynamic logics is defined below.

- (4.18) $(\text{dyn not}) =_{\text{def}} \text{NOT}$
- (4.19) $(\text{dyn and}) =_{\text{def}} \text{AND}$
- (4.20) $(\text{dyn implies}) =_{\text{def}} \text{IMPLIES} =_{\text{def}} \lambda_{hk}. \text{NOT} (h \text{ AND } (\text{NOT } k))$
- (4.21) $(\text{dyn or}) =_{\text{def}} \text{OR} =_{\text{def}} \lambda_{hk}. \text{NOT} ((\text{NOT } h) \text{ AND } (\text{NOT } k))$
- (4.22) $(\text{dyn forall}) =_{\text{def}} \text{FORALL} =_{\text{def}} \lambda_D. \text{NOT EXISTS}_n. \text{NOT} (D \ n)$
- (4.23) $(\text{dyn exists}) =_{\text{def}} \text{EXISTS}$

The reason that the types of the dynamic connectives and quantifiers are somewhat more complicated is that they have the additional task of managing the discourse referents in the context, and so, for example, the type of **EXISTS** is not the dynamic generalized quantifier type $d_1 \rightarrow k$, but a dependent variant of this type in which the output content is constrained to extend the arity of the input context by 1. Similarly, the type of **AND** is not $k \rightarrow k \rightarrow k$, and the type of **NOT** is not $k \rightarrow k$. In the case of **AND**, its type reflects the fact that the resulting content must be extended by the sum of the number of discourse referents introduced by its conjuncts. For **NOT**, the dependent typing requires that the output content not introduce any discourse referents, since it renders inaccessible any discourse referents introduced in its scope.

I define some convenience operators on dynamic properties, by analogy to the ones defined in 3.10.

Definition 4.15 (Operations on Dynamic Properties). The DyCG analog of the static property conjunction that in (3.21) is

$$\text{THAT} : d_1 \rightarrow d_1 \rightarrow d_1 ,$$

defined by analogy to its static counterpart (repeated below) as follows:

$$(3.21) \quad \text{that} =_{\text{def}} \lambda_{PQx}.(P x) \text{ and } (Q x)$$

$$(4.24) \quad \text{THAT} =_{\text{def}} \lambda_{DEN}.(D n) \text{ AND } (E n)$$

Likewise, the dynamic version of property negation

$$\text{NON} : d_{1,i} \rightarrow d_{1,0} ,$$

for each natural number i , is analogous to static property negation.

$$(3.22) \quad \text{non} =_{\text{def}} \lambda_{Px}.\text{not } (P x)$$

$$(4.25) \quad \text{NON} =_{\text{def}} \lambda_{Dn}.\text{NOT } (D n)$$

4.2.2.2 Dynamic Entailment and Definiteness

DyCG also makes use of a contextualized version of entailment, and a contextualized description operator for selecting anaphoric antecedents. As a preliminary for defining these, I first define a propositional notion of entailment based on entails : $p \rightarrow p \rightarrow t$ from definition 3.7.

Definition 4.16 (Propositional Entailment). The relation

$$\text{Entails} : p \rightarrow p \rightarrow p$$

encodes entailment between propositions as a proposition, subject to the axiom

$$\vdash \forall_{p:p} \forall_{q:p} \forall_{w:w}. (p \text{ Entails } q) @ w \Leftrightarrow (p \text{ entails } q) .$$

That is, Entails encodes a form of entailment that holds whenever one proposition entails another at every world.

Then contextual entailment is defined in terms of Entails.

Definition 4.17 (Contextual Entailment). The notion of **contextual entailment**, encoded in the function

$$\text{c-entails} : \Pi_{c:c} \Pi_{d:c \geq |c|} \cdot \mathbf{p} ,$$

which is written infix and defined as

$$(4.26) \quad \text{c-entails} =_{\text{def}} \lambda_{c:c} \lambda_{d:c \geq |c|} . \text{forall}_{\mathbf{x}^{|c|}} . (c \mathbf{x}) \text{ Entails exists}_{\mathbf{y}^{|d|-|c|}} . (d \mathbf{x}, \mathbf{y})$$

defines contextual entailment as holding between two contexts c and d , where d is of an arity at least as great as c .

Intuitively, this definition requires that the first context must entail the existence of entities satisfying the content that is found in the second but not in the first.

Entailment between contents is then defined based on contextual entailment.

Definition 4.18 (Entailment between Contents). To determine whether a content's potential update would be entailed by a given context, the function

$$\text{k-entails} : c \rightarrow k \rightarrow \mathbf{p} ,$$

is used. It is defined as

$$\text{k-entails} =_{\text{def}} \lambda_{c:c} \lambda_{k:k} . c \text{ c-entails } (cc \ k \ c) ,$$

and written infix.

Then we can determine whether a content is contextually consistent as follows.

Definition 4.19 (Consistency for Contents). The function

$$\text{k-cons} : c \rightarrow k \rightarrow \mathbf{p} ,$$

also written infix and defined as

$$\text{k-cons} =_{\text{def}} \lambda_{c:c} \lambda_{k:k} . \text{not } (c \text{ k-entails } (\text{NOT } k)) ,$$

yields the proposition that a context c does not contextually entail the dynamic negation of a content k .

Finally, the function defined below is used to retrieve the antecedents of anaphors.

Definition 4.20 (Generalized Definiteness). The **generalized definiteness function**

$$\text{the} : \Pi_{D:d_1} \Pi_{c:c} \omega_{|c|}$$

selects the unique discourse referent entailed by a given context to have a specified dynamic property. It is defined as

$$(4.27) \quad \text{the} =_{\text{def}} \lambda_{D:d_1} \lambda_{c:c} \iota_{n:\omega_{|c|}}.c \text{ k-entails } (D n),$$

where $\iota_{n:\omega_{|c|}}$ is the description operator for natural numbers less than $|c|$ (see definition B.8 and theorem B.9).

Definiteness, by the definition in (4.27), requires that the context entails a certain discourse referent has a certain property. But for pronouns, as I discuss in chapter 2 in connection with (2.55) and (2.56), all that is required is that the discourse not be *inconsistent with* the descriptive content. Accordingly, pronouns use a different function to select their antecedents.

Definition 4.21 (Definiteness for Pronouns). The **definiteness function for pronouns** has the same type as the in definition 4.20:

$$\text{pro} : \Pi_{D:d_1} \Pi_{c:c} \omega_{|c|}$$

However, its definition is based on k-cons rather than invoking k-entails directly.

$$(4.28) \quad \text{pro} =_{\text{def}} \lambda_{D:d_1} \lambda_{c:c} \iota_{n:\omega_{|c|}}.c \text{ k-cons } (D n)$$

These definitions of definiteness will be greatly extended in chapter 5.

4.3 A Dynamic Fragment

Turning to some English examples that demonstrate how the dynamic semantics replaces the static semantics in chapter 3, consider first the

dynamic version of the simple examples in (3.28) and (3.29), repeated below.

(3.28) It snowed.

(3.29) Kim sneezed.

For (3.28), the lexicon needs to be modified so that *snowed* corresponds to the entry

$$\vdash \lambda_s.s \cdot \text{snowed} ; It \multimap S ; \lambda_u.\text{SNOW} : d_* .$$

As part of the broader DyCG strategy of treating all signs with tecto type NP semantically as dynamic generalized quantifiers (see the discussion of *KIM*, below), the dynamic version of the pleonastic *It* that is the subject of weather predicates like *snowed* is $\text{IT}_{\text{pleo}} : d_* \rightarrow k$, defined as

$$\text{IT}_{\text{pleo}} =_{\text{def}} \lambda_D.(D *) .$$

Then the DyCG lexical entry corresponding to *It* uses similar pheno and identical tecto to the static version in (3.30).

$$\vdash \lambda_f.(f \text{ it}) : (s \rightarrow s) \rightarrow s ; (It \multimap S) \multimap S ; \text{IT}_{\text{pleo}} : d_* \rightarrow k$$

Aside from the different semantic typing assigned to *It*, the DyCG proof of (3.28) is very similar to the CyCG one in (3.32).

$$(4.29) \quad \frac{\vdash \lambda_f.(f \text{ it}) ; (It \multimap S) \multimap S ; \text{IT}_{\text{pleo}} \quad \vdash \lambda_s.s \cdot \text{snowed} ; It \multimap S ; \lambda_u.\text{SNOW}}{\vdash ((\lambda_f(f \text{ it})) (\lambda_s s \cdot \text{snowed})) ; S ; (\text{IT}_{\text{pleo}} \lambda_u.\text{SNOW})}$$

(This proof uses the Combine rule, and as before, the rule label is omitted to save space.) As intended, the DyCG version has the same concrete and abstract syntax, with the only difference being that the semantics uses *SNOW* rather than *snow*. The root label in (4.29) reduces to the long normal form

$$\vdash (\text{it} \cdot \text{snowed}) : s ; S ; \lambda_{cx|c|}.\text{snow} : k .$$

To handle (3.29), the semantics of proper names needs to be modified from the simple entity constants used in chapter 3 to give proper names the type $d_1 \rightarrow k$ of dynamic generalized quantifiers, imitating Montague's (1973) static semantics and Barwise and Cooper's (1981) treatment of noun phrases. For example, define *KIM* : $d_1 \rightarrow k$ as follows, where $\text{NAMED-KIM} =_{\text{def}} (\text{dyn}_1 \text{ named-kim})$, and *named-kim* is the static property of

being named *Kim*:

$$(4.30) \quad \begin{aligned} \text{KIM} &=_{\text{def}} \lambda_{Dc}.D(\text{the NAMED-KIM } c) c \\ &= \lambda_{Dc}.D(\iota_{n:\omega|c} c \text{ k-entails } (\text{NAMED-KIM } n)) c \end{aligned}$$

(And similarly for other proper names.) This definition simply passes the unique discourse referent entailed by the context to be named *Kim* to a specified dynamic property. In chapter 5, I discuss how DyCG handles situations in which there is no such unique discourse referent, in the context of dynamically modeling senses and implicatures.

Defining SNEEZE : d₁ as

$$\text{SNEEZE} =_{\text{def}} (\text{dyn}_1 \text{ sneeze}) = \lambda_{n:\omega} \lambda_{c:c>n} \lambda_{\mathbf{x}|c}.(\text{sneeze } \mathbf{x}_n),$$

and the required DyCG lexical entries as

$$\begin{aligned} &\vdash \lambda_f.(f \text{ Kim}) : (s \rightarrow s) \rightarrow s; (\text{NP} \multimap S) \multimap S; \text{KIM} : (\omega \rightarrow k) \rightarrow k \\ &\vdash \lambda_{s.s} \cdot \text{sneezed} : s \rightarrow s; \text{NP} \multimap S; \text{SNEEZE} : \omega \rightarrow k, \end{aligned}$$

a DyCG proof of (3.29) is available.

$$(4.31) \quad \frac{\vdash \text{Kim}; (\text{NP} \multimap S) \multimap S; \text{KIM} \quad \vdash \lambda_{s.s} \cdot \text{sneezed}; \text{NP} \multimap S; \text{SNEEZE}}{\vdash ((\lambda_{s.s} \cdot \text{sneezed}) \text{ Kim}); S; (\text{KIM SNEEZE})}$$

A reduced form of the root of this proof is

$$\vdash \text{Kim} \cdot \text{sneezed} : s; S; \lambda_{c:c>(\text{the NAMED-KIM } c)} \lambda_{\mathbf{x}|c}.(\text{sneeze } \mathbf{x}_{(\text{the NAMED-KIM } c)}) : k,$$

where (the NAMED-KIM *c*) is the unique discourse referent named *Kim* per definition 4.20. And so, as for the proof in (4.29), the proof in (4.31) yields a sign with the identical pheno string and tecto type as its static counterpart, with the difference that the semantics is dynamic.

To show how DyCG handles quantification, we start with the example

(3.34) Lance rides a bike.

In keeping with the strategy of constructing a dynamic semantics based on its static counterparts, the dynamic generalized determiner $A : d_1 \rightarrow d_1 \rightarrow k$ is defined as follows, with its static counterpart a repeated alongside for

comparison.

$$(3.35) \quad a =_{\text{def}} \lambda_{PQ}.\text{exists}_x.(P x) \text{ and } (Q x)$$

$$(4.32) \quad A =_{\text{def}} \lambda_{DE}.\text{EXISTS}_n.(D n) \text{ AND } (E n)$$

Thus the dynamic indefinite determiner A in (4.32) can equivalently be written as

$$\lambda_{DE}.\text{EXISTS} (D \text{ THAT } E) .$$

Then the DyCG semantics for the proper name *Lance* is defined similarly to KIM , based on the dynamic property

$$(4.33) \quad \text{NAMED-LANCE} =_{\text{def}} (\text{dyn}_1 \text{ named-lance}) ,$$

as

$$(4.34) \quad \text{LANCE} =_{\text{def}} \lambda_{Dc}.D (\text{the NAMED-LANCE } c) c ,$$

and the dynamic semantics for the transitive verb *rides* and common noun *bike* as:

$$\text{RIDE} =_{\text{def}} (\text{dyn}_2 \text{ ride})$$

$$\text{BIKE} =_{\text{def}} (\text{dyn}_1 \text{ bike})$$

Lastly, the DyCG lexicon necessary to prove a sign corresponding to (3.34) has the following entries, where $\text{QP} =_{\text{def}} (\text{NP} \multimap \text{S}) \multimap \text{S}$, as in chapter 3.

$$\begin{aligned} &\vdash \lambda_f.(f \text{ Lance}) : (s \rightarrow s) \rightarrow s ; \text{NP} ; \text{LANCE} : d_1 \rightarrow k \\ &\vdash \lambda_{st.t} . \text{rides} \cdot s : s \rightarrow s \rightarrow s ; \text{NP} \multimap \text{NP} \multimap \text{S} ; \text{RIDE} : d_2 \\ &\vdash \lambda_{sf}.f (a \cdot s) : s \rightarrow (s \rightarrow s) \rightarrow s ; \text{N} \multimap \text{QP} ; A : d_1 \rightarrow d_1 \rightarrow k \\ &\vdash \text{bike} : s ; \text{N} ; \text{BIKE} : d_1 \end{aligned}$$

Now everything necessary for a DyCG proof of (3.34) is in place.

Since both *Lance* and *a bike* are typed as (dynamic) generalized quantifiers, the proof starts by providing two trace arguments to the transitive verb *rides*. Unlike CyCG traces, where the semantic variable is of type e of entities, DyCG semantic traces have the type of natural numbers, that is,

the type of discourse referents. First the object trace:

$$(4.35) \quad \frac{\vdash \lambda_{st}.t \cdot \text{rides} \cdot s; \text{NP} \multimap \text{NP} \multimap \text{S}; \text{RIDE} \quad s; \text{NP}; m \vdash s; \text{NP}; m}{s; \text{NP}; m \vdash \lambda_t.t \cdot \text{rides} \cdot s; \text{NP} \multimap \text{S}; (\text{RIDE } m)}$$

Next the subject trace is provided.

$$(4.36) \quad \frac{\begin{array}{c} (4.35) \\ \vdots \\ s; \text{NP}; m \vdash \lambda_t.t \cdot \text{rides} \cdot s; \text{NP} \multimap \text{S}; (\text{RIDE } m) \end{array} \quad t; \text{NP}; n \vdash t; \text{NP}; n}{s; \text{NP}; m, t; \text{NP}; n \vdash t \cdot \text{rides} \cdot s; \text{S}; (\text{RIDE } m \ n)}$$

Before *a bike* can take scope, it first must be formed as a generalized quantifier. The proof subtree in (4.37) shows the generalized determiner *a* combining with its argument *bike*.

$$(4.37) \quad \frac{\vdash \lambda_{sf}.f(a \cdot s); \text{N} \multimap \text{QP}; \text{A} \quad \vdash \text{bike} : s; \text{N}; \text{BIKE}}{\vdash \lambda_f.f(a \cdot \text{bike}); \text{QP}; (\text{A BIKE})}$$

Then, starting from the proof in (4.36), the trace in object position is bound so that *a bike* can take the proper scope.

$$(4.38) \quad \frac{\begin{array}{c} (4.36) \\ \vdots \\ s; \text{NP}; m, t; \text{NP}; n \vdash t \cdot \text{rides} \cdot s; \text{S}; (\text{RIDE } m \ n) \end{array}}{t; \text{NP}; n \vdash \lambda_s.t \cdot \text{rides} \cdot s; \text{NP} \multimap \text{S}; \lambda_m.\text{RIDE } m \ n}$$

Here, for reasons of horizontal space, I split the proof into its two ‘halves,’ the first showing the pheno and tecto components only. The proof in (4.39) shows *a bike* taking scope in the pheno.

$$(4.39) \quad \frac{\begin{array}{c} (4.37) \\ \vdots \\ \vdash \lambda_f.f(a \cdot \text{bike}); \text{QP} \end{array} \quad \begin{array}{c} (4.38) \\ \vdots \\ t; \text{NP} \vdash \lambda_s.t \cdot \text{rides} \cdot s; \text{NP} \multimap \text{S} \end{array}}{t; \text{NP} \vdash ((\lambda_f.f(a \cdot \text{bike})) \lambda_s.t \cdot \text{rides} \cdot s); \text{S}}$$

The following proof shows *a bike* taking scope on the semantic side.

$$(4.40) \quad \frac{\begin{array}{c} (4.37) \\ \vdots \\ \vdash \text{QP}; (\text{A BIKE}) \end{array} \quad \begin{array}{c} (4.38) \\ \vdots \\ \text{NP}; n \vdash \text{NP} \multimap \text{S}; \lambda_m.\text{RIDE } m \ n \end{array}}{\text{NP}; n \vdash \text{S}; (\text{A BIKE})_m.\text{RIDE } m \ n}$$

Finally, the proper name *Lance* takes its scope, with the proof again split into two for reasons of space. First the pheno side:

$$(4.41) \quad \frac{\begin{array}{c} (4.39) \\ \vdots \\ t; \text{NP} \vdash t \cdot \text{rides} \cdot a \cdot \text{bike}; \text{S} \end{array} \quad \vdash \lambda_f.(f \text{ Lance}); \text{QP}}{\vdash ((\lambda_f(f \text{ Lance})) \lambda_t.t \cdot \text{rides} \cdot a \cdot \text{bike}); \text{S}}$$

Then the semantics of *Lance* taking scope:

$$(4.42) \quad \frac{\begin{array}{c} (4.40) \\ \vdots \\ \text{NP}; n \vdash \text{S}; (\text{A BIKE})_m.\text{RIDE } m \ n \end{array} \quad \vdash \text{QP}; \text{LANCE}}{\vdash \text{S}; \text{LANCE}_n.(\text{A BIKE})_m.\text{RIDE } m \ n}$$

And so the β -reduced long-form root label of the final proof is

$$(4.43) \quad \vdash \text{Lance} \cdot \text{rides} \cdot a \cdot \text{bike} : s; \text{S}; \text{LANCE}_n.(\text{A BIKE})_m.\text{RIDE } m \ n : k_1 ,$$

which analyzes (3.34) as a sentence with the correct surface form. The dynamic semantics of (4.43) reduces as follows.

$$\begin{aligned} & \vdash \text{LANCE}_n.(\text{A BIKE})_m.\text{RIDE } m \ n \\ &= \text{LANCE}_n.\text{EXISTS}_m.(\text{BIKE } m) \text{ AND } (\text{RIDE } m \ n) \\ &= ((\lambda_{Dc}(D(\text{the NAMED-LANCE } c) \ c)) \\ &\quad \lambda_n \lambda_{c:c} \lambda_{\mathbf{x}|c^+|.}(\text{BIKE } |c|) \text{ AND } (\text{RIDE } |c| \ n)) \\ &= \lambda_{c:c} \lambda_{\mathbf{x}|c^+|.}(\text{BIKE } |c|) \text{ AND } (\text{RIDE } |c| \ (\text{the NAMED-LANCE } c)) : k_1 . \end{aligned}$$

Note that there is another semantic scoping available, namely the one that results from withdrawing the subject and object traces in the other order,

yielding

$$\vdash (\text{A BIKE})_m.\text{LANCE}_n.\text{RIDE } m \ n : k_1 .$$

However, this alternative scoping does not result in a distinct reading, since LANCE simply passes the relevant discourse referent to its argument in both scopings.

4.3.1 Dynamic Quantifier Scope

DyCG straightforwardly provides a dynamic version of the CyCG account of (3.48), repeated below.

(3.48) Every cyclist rides a bike.

To get started, we need a dynamic *every* that is based on its static counterpart in a similar way to the parallel between *a* and *A*. As for *A*, the definition of $\text{EVERY} : d_1 \rightarrow d_1 \rightarrow k_0$ in (4.44) is presented alongside its static counterpart *every*.

$$(3.49) \quad \text{every} =_{\text{def}} \lambda_{PQ}.\text{forall}_x.(P \ x) \text{ implies } (Q \ x)$$

$$(4.44) \quad \text{EVERY} =_{\text{def}} \lambda_{DE}.\text{FORALL}_n.(D \ n) \text{ IMPLIES } (E \ n)$$

To give DyCG versions of the two scoping alternatives in (3.64) and (3.65), all that remains is to specify the dynamic meanings of *cyclist*.

$$\text{CYCLIST} =_{\text{def}} (\text{dyn}_1 \text{ cyclist})$$

Then the DyCG lexical entries for *Every* and *cyclist*, written in short form in (4.45) and (4.46), are just like the corresponding CyCG ones, with the dynamic semantics replacing the static one.

$$(4.45) \quad \vdash \lambda_{sf}.f(\text{every} \cdot s) ; N \multimap QP ; \text{EVERY}$$

$$(4.46) \quad \vdash \text{cyclist} ; N ; \text{CYCLIST}$$

(These lexical entries are written in short form to save horizontal space.) For clarity, in the following DyCG proofs for (3.48), proofs, only the semantics is shown because the pheno and tecto parts of the proof are identical to their static counterparts in (3.64) and (3.65).

For the dynamic surface scope reading, we start with the proof in (4.38), then apply the semantics of *a bike* from (4.37) to it. Then the discourse

referent variable n is bound to prepare for *Every cyclist*'s scope taking.

$$(4.47) \quad \frac{\frac{\frac{(4.37) \quad \vdots}{\vdash (\text{A BIKE}) : d_1 \rightarrow k} \quad \frac{(4.38) \quad \vdots}{n : \omega \vdash \lambda_m.\text{RIDE } m \ n : d_1}}{n : \omega \vdash (\text{A BIKE})_m.\text{RIDE } m \ n : k}}{\vdash \lambda_n.(\text{A BIKE})_m.\text{RIDE } m \ n : d_1}$$

The proof that *Every cyclist* is a dynamic generalized quantifier is similar to the proof for *a bike* in (4.37).

$$(4.48) \quad \frac{\vdash \text{EVERY} : d_1 \rightarrow d_1 \rightarrow k_0 \quad \vdash \text{CYCLIST} : d_1}{\vdash (\text{EVERY CYCLIST}) : d_1 \rightarrow k_0}$$

Finally, (EVERY CYCLIST) takes its scope over the proof in (4.47).

$$(4.49) \quad \frac{\frac{(4.48) \quad \vdots}{\vdash \text{EVERY CYCLIST} : d_1 \rightarrow k_0} \quad \frac{(4.47) \quad \vdots}{\vdash \lambda_n.(\text{A BIKE})_m.\text{RIDE } m \ n : d_1}}{\vdash (\text{EVERY CYCLIST})_n.(\text{A BIKE})_m.\text{RIDE } m \ n : k_0}$$

The reverse scope reading starts with the semantic part of the proof in (4.36), because in the reverse scope proof, (EVERY CYCLIST) needs to take scope first instead of (A BIKE). Then the object-positions discourse referent m is extracted to ready the proof so that (A BIKE) can take scope.

$$(4.50) \quad \frac{\frac{(4.37) \quad \vdots}{\vdash (\text{A BIKE})} \quad \frac{\frac{(4.48) \quad \vdots}{\vdash (\text{EVERY CYCLIST})} \quad \frac{(4.36) \quad \vdots}{m, n \vdash (\text{RIDE } m \ n)}}{m \vdash \lambda_n.\text{RIDE } m \ n}}{m \vdash (\text{EVERY CYCLIST})_n.\text{RIDE } m \ n} \quad \frac{\vdash (\text{A BIKE}) \quad m \vdash (\text{EVERY CYCLIST})_n.\text{RIDE } m \ n}{\vdash \lambda_m.(\text{A BIKE})_m.(\text{EVERY CYCLIST})_n.\text{RIDE } m \ n}$$

(This proof suppresses all typing information so that the entire proof can be displayed.)

And so the signs derived by DyCG for (3.48) are very similar to those derived by CyCG. These signs are given below in parallel to display their

similarity.

(3.64)

$$\vdash \text{every} \cdot \text{cyclist} \cdot \text{rides} \cdot a \cdot \text{bike} ; S ; (\text{every cyclist})_y. (a \text{ bike})_x. \text{ride } x \ y$$

(4.51)

$$\vdash \text{every} \cdot \text{cyclist} \cdot \text{rides} \cdot a \cdot \text{bike} ; S ; (\text{EVERY CYCLIST})_n. (\text{A BIKE})_m. \text{RIDE } m \ n$$

(3.65)

$$\vdash \text{every} \cdot \text{cyclist} \cdot \text{rides} \cdot a \cdot \text{bike} ; S ; (a \text{ bike})_x. (\text{every cyclist})_y. \text{ride } x \ y$$

(4.52)

$$\vdash \text{every} \cdot \text{cyclist} \cdot \text{rides} \cdot a \cdot \text{bike} ; S ; (\text{A BIKE})_m. (\text{EVERY CYCLIST})_n. \text{RIDE } m \ n$$

Note that, although the combined DyCG proofs are not shown, the surface forms in (4.51) and (4.52) are the ones derived by the grammar for the same reason they are the ones derived by the CyCG variants: the string variable s is linked to the discourse referent variable n via the Trace rule, just as the string variable t is linked to the discourse referent m .

The dynamic semantic terms for the surface scope reading of (3.48) reduces as follows.

$$\begin{aligned} & \vdash (\text{EVERY CYCLIST})_n. (\text{A BIKE})_m. \text{RIDE } m \ n \\ &= \text{FORALL}_n. (\text{CYCLIST } n) \text{ IMPLIES } (\text{A BIKE})_m. \text{RIDE } m \ n \\ &= \text{NOT EXISTS}_n. \text{NOT } ((\text{CYCLIST } n) \text{ IMPLIES } (\text{A BIKE})_m. \text{RIDE } m \ n) \\ &= \text{NOT EXISTS}_n. \text{NOT NOT } ((\text{CYCLIST } n) \text{ AND NOT } (\text{A BIKE})_m. \text{RIDE } m \ n) \\ &= \text{NOT EXISTS}_n. \text{NOT NOT } ((\text{CYCLIST } n) \text{ AND} \\ & \quad \text{NOT EXISTS}_m. (\text{BIKE } m) \text{ AND } (\text{RIDE } m \ n)) : k_0 \end{aligned}$$

The reverse scope reading, on the other hand, reduces to:

$$\begin{aligned} & \vdash (\text{A BIKE})_m. (\text{EVERY CYCLIST})_n. \text{RIDE } m \ n \\ &= \text{EXISTS}_m. ((\text{BIKE } m) \text{ AND FORALL}_n. (\text{CYCLIST } n) \text{ IMPLIES } (\text{RIDE } m \ n)) \\ &= \text{EXISTS}_m. ((\text{BIKE } m) \text{ AND NOT EXISTS}_m. \text{NOT} \\ & \quad ((\text{CYCLIST } n) \text{ IMPLIES } (\text{RIDE } m \ n))) \\ &= \text{EXISTS}_m. ((\text{BIKE } m) \text{ AND NOT EXISTS}_n. \text{NOT} \\ & \quad \text{NOT } ((\text{CYCLIST } n) \text{ AND NOT } (\text{RIDE } m \ n))) : k_1 \end{aligned}$$

By theorem 4.13, these readings are equivalent to

$$\text{NOT EXISTS}_n.(\text{CYCLIST } n) \text{ AND NOT EXISTS}_m.(\text{BIKE } m) \text{ AND } (\text{RIDE } m \ n)$$

and

$$\text{EXISTS}_m.(\text{BIKE } m) \text{ AND NOT EXISTS}_n.(\text{CYCLIST } n) \text{ AND NOT } (\text{RIDE } m \ n)$$

respectively, so that, by proposition C.6, we finally arrive at

$$\lambda_{c|x|c|}.\text{not exists}_y.(\text{cyclist } y) \text{ and not exists}_z.(\text{bike } z) \text{ and } (\text{ride } z \ y) : k_0$$

for the surface scope reading, and

$$\lambda_c \lambda_{x|c|y} . (\text{bike } y) \text{ and not exists}_z.(\text{cyclist } z) \text{ and not } (\text{ride } y \ z) : k_1$$

for the reverse scope reading. Notice that for the reverse scope reading, a discourse referent is introduced, but no discourse referent is introduced for the surface scope reading.

4.3.2 Donkey Anaphora

To demonstrate a central motivation for replacing the CyCG static semantics, I examine the following example, an instance of *donkey anaphora* (Geach, 1962, chapter 5):

(4.53) Every cyclist_{*i*} that owns a bike rides it_{*i*}.

The DyCG lexical entries for all of the words in (4.53) are already defined besides *owns* and *it*. Starting with *owns*, it is defined based on its static counterpart *own* using the dynamicizer *dyn*₂:

$$\text{OWN} =_{\text{def}} (\text{dyn}_2 \text{ own})$$

The pronoun *it* needs to select its antecedent from prior context, and its antecedent must meet the condition of not being human. It is modeled by the dynamic generalized quantifier *IT* : *d*₁ → *k*, defined as

$$(4.54) \quad \text{IT} =_{\text{def}} \lambda_D \lambda_c . D (\text{pro NONHUMAN } c) c ,$$

which passes the unique discourse referent that has descriptive content consistent with being nonhuman. Here $\text{NONHUMAN} =_{\text{def}} (\text{dyn}_1 \text{ nonhuman})$, where *nonhuman* is the static property of being nonhuman. Note that the definition of *IT* is just like the definition of the proper names *KIM* and *LANCE* in (4.30) and (4.34), above, with the exception that *pro* is used in place of the *and* and the dynamic property passed to *pro* is different.

The extensions to the DyCG lexicon needed to account for (4.53) are the following.

$$\begin{aligned} &\vdash \lambda_{st}.s \cdot \text{owns} \cdot t : s \rightarrow s \rightarrow s ; \text{NP} \multimap \text{NP} \multimap S ; \text{OWN} : d_1 \\ &\vdash \lambda_{sf}.s \cdot \text{that} \cdot (f \mathbf{e}) : s \rightarrow (s \rightarrow s) \rightarrow s ; N \multimap (\text{NP} \multimap S) \multimap N ; \text{THAT} \\ &\vdash \lambda_f.(f \text{ it}) : (s \rightarrow s) \rightarrow s ; (\text{NP} \multimap S) \multimap S ; \text{IT} : d_1 \rightarrow k \end{aligned}$$

The DyCG lexical entry corresponding to *owns* is straightforwardly a transitive verb, similar to *ride*, above. The dynamic property conjunction *THAT* has the same abstract and concrete syntax as its static counterpart *that*, and its type is omitted here for readability. The lexical entry for *IT* is similar to the one used for other generalized quantifiers, such the dynamic proper name semantics *KIM*, discussed above.

Since *EVERY* needs to take as its arguments the dynamic properties corresponding to *cyclist that owns a bike* and *rides it*, we first derive the property arguments. The one headed by *cyclist* begins by hypothesizing arguments for *owned*, similarly to the proof in (4.36), starting with the object trace.

$$(4.55) \quad \frac{\vdash \lambda_{st}.t \cdot \text{owns} \cdot s ; \text{NP} \multimap \text{NP} \multimap S ; \text{OWN} \quad s ; \text{NP} ; m \vdash s ; \text{NP} ; m}{s ; \text{NP} ; m \vdash \lambda_t.t \cdot \text{owns} \cdot s ; \text{NP} \multimap S ; (\text{OWN } m)}$$

And next the trace for the subject position. (Here, and in the following proofs, I perform β -reduction in the concrete syntactic and semantic terms when possible.)

$$(4.56) \quad \frac{\begin{array}{c} (4.55) \\ \vdots \\ s ; \text{NP} ; m \vdash \lambda_t.t \cdot \text{owns} \cdot s ; \text{NP} \multimap S ; (\text{OWN } m) \quad t ; \text{NP} ; n \vdash t ; \text{NP} ; n \end{array}}{s ; \text{NP} ; m, t ; \text{NP} ; n \vdash t \cdot \text{owns} \cdot s ; S ; (\text{OWN } m n)}$$

Next the object trace is extracted so that *a bike* (from the proof in (4.37)) can take scope, after which the subject trace is extracted. (Here and below,

for reasons of horizontal space, the proofs are split, showing the abstract syntax and semantics separately from the concrete syntax.)

$$\begin{array}{c}
 (4.57) \quad \frac{\frac{\frac{(4.37) \quad \vdots}{\vdash \text{QP}; (\text{A BIKE})} \quad \frac{(4.56) \quad \vdots \quad \text{NP}; m, \text{NP}; n \vdash S; (\text{OWN } m \ n)}{\text{NP}; n \vdash \text{NP} \multimap S; \lambda_m.\text{OWN } m \ n}}{\text{NP}; n \vdash S; (\text{A BIKE})_m.\text{OWN } m \ n}}{\vdash \text{NP} \multimap S; \lambda_n.(\text{A BIKE})_m.\text{OWN } m \ n}
 \end{array}$$

The concrete syntactic part of the proof in (4.57) is as follows.

$$\begin{array}{c}
 (4.58) \quad \frac{\frac{(4.37) \quad \vdots}{\vdash \lambda_f.f(a \cdot \text{bike}) : (s \rightarrow s) \rightarrow s} \quad \frac{(4.56) \quad \vdots \quad s : s, t : s \vdash t \cdot \text{owns} \cdot s : s}{t : s \vdash \lambda_s.t \cdot \text{owns} \cdot s : s \rightarrow s}}{t : s \vdash t \cdot \text{owns} \cdot a \cdot \text{bike} : s} \\
 \vdash \lambda_t.t \cdot \text{owns} \cdot a \cdot \text{bike} : s \rightarrow s
 \end{array}$$

The property conjunction **THAT** first combines with **CYCLIST**.

$$(4.59) \quad \frac{\vdash \text{N} \multimap (\text{NP} \multimap S) \multimap (\text{NP} \multimap S); \text{THAT} \quad \vdash \text{N}; \text{CYCLIST}}{\vdash (\text{NP} \multimap S) \multimap (\text{NP} \multimap S); (\text{CYCLIST THAT})}$$

The pheno portion of this proof is given below.

$$(4.60) \quad \frac{\vdash \lambda_{sf}.s \cdot \text{that} \cdot (f \ e) : s \rightarrow (s \rightarrow s) \rightarrow s \quad \vdash \text{cyclist} : s}{\vdash \lambda_f.\text{cyclist} \cdot \text{that} \cdot (f \ e) : (s \rightarrow s) \rightarrow s}$$

Then the proof for *cyclist that ...* combines with the one for *owns a bike*, below, with only the short-form semantic terms shown.

$$\begin{array}{c}
 (4.61) \quad \frac{\frac{(4.59) \quad \vdots}{\vdash (\text{CYCLIST THAT})} \quad \frac{(4.57) \quad \vdots}{\vdash \lambda_n.(\text{A BIKE})_m.\text{OWN } m \ n}}{\vdash \text{CYCLIST THAT } \lambda_n.(\text{A BIKE})_m.\text{OWN } m \ n}
 \end{array}$$

The abstract syntax for (4.61) is then:

$$(4.62) \quad \frac{\begin{array}{c} (4.60) \\ \vdots \\ \vdash \lambda_f.\text{cyclist} \cdot \text{that} \cdot (f \mathbf{e}) : (s \rightarrow s) \rightarrow s \end{array} \quad \begin{array}{c} (4.58) \\ \vdots \\ \vdash \lambda_t.t \cdot \text{owns} \cdot a \cdot \text{bike} : s \rightarrow s \end{array}}{\vdash \text{cyclist} \cdot \text{that} \cdot \mathbf{e} \cdot \text{owns} \cdot a \cdot \text{bike} : s}$$

These proofs yield the DyCG sign for *cyclist that owns a bike* below.

$$(4.63) \quad \vdash \text{cyclist} \cdot \text{that} \cdot \mathbf{e} \cdot \text{owns} \cdot a \cdot \text{bike}; \\ \text{NP} \multimap \text{S}; \text{CYCLIST THAT } \lambda_n.(\text{A BIKE})_m.\text{OWN } m \ n$$

As for the dynamic property corresponding to *rides it*, we start with the proof in (4.38).

$$(4.64) \quad \frac{\begin{array}{c} (4.38) \\ \vdots \\ \vdash (\text{NP} \multimap \text{S}) \multimap \text{S}; \text{IT} \quad \text{NP}; n \vdash \text{NP} \multimap \text{S}; \lambda_m.\text{RIDE } m \ n \end{array}}{\frac{\text{NP}; n \vdash \text{S}; \text{IT}_m.\text{RIDE } m \ n}{\vdash \text{NP} \multimap \text{S}; \lambda_n.\text{IT}_m.\text{RIDE } m \ n}}$$

And the concrete syntax for (4.64) is in (4.65).

$$(4.65) \quad \frac{\begin{array}{c} (4.38) \\ \vdots \\ \vdash \lambda_f.(f \text{ it}) : (s \rightarrow s) \rightarrow s \quad t : s \vdash \lambda_s.t \cdot \text{rides} \cdot s : s \rightarrow s \end{array}}{\frac{t : s \vdash t \cdot \text{rides} \cdot \text{it} : s}{\vdash \lambda_t.t \cdot \text{rides} \cdot \text{it} : s \rightarrow s}}$$

With the dynamic meanings for the two properties *cyclist that owns a bike* and *rides it* derived, all that remains is to pass both of these to the dynamic meaning of *Every*, finally giving the following DyCG sign.

$$(4.66) \quad \vdash \text{every} \cdot \text{cyclist} \cdot \text{that} \cdot \mathbf{e} \cdot \text{owns} \cdot a \cdot \text{bike} \cdot \mathbf{e} \cdot \text{rides} \cdot \text{it}; \\ \text{S}; (\text{EVERY} (\text{CYCLIST THAT } \lambda_n.(\text{A BIKE})_m.\text{OWN } m \ n))_n.\text{IT}_m.\text{RIDE } m \ n$$

To see that the semantics captures the meaning of (4.53), we first reduce the semantic term in (4.66).

$$\begin{aligned}
& \vdash (\text{EVERY } (\text{CYCLIST THAT } \lambda_n.(\text{A BIKE})_m.\text{OWN } m \ n))_n.\text{IT}_m.\text{RIDE } m \ n \\
& = (\text{EVERY}_n.((\text{CYCLIST } n) \text{ AND } (\text{A BIKE})_m.\text{OWN } m \ n))_n.\text{IT}_m.\text{RIDE } m \ n \\
& = \text{FORALL}_n.(((\text{CYCLIST } n) \text{ AND } (\text{A BIKE})_m.\text{OWN } m \ n) \\
& \quad \text{IMPLIES IT}_m.\text{RIDE } m \ n) \\
& = \text{NOT EXISTS}_n.\text{NOT } (((\text{CYCLIST } n) \text{ AND } (\text{A BIKE})_m.\text{OWN } m \ n) \\
& \quad \text{IMPLIES IT}_m.\text{RIDE } m \ n) \\
& = \text{NOT EXISTS}_n.\text{NOT NOT } (((\text{CYCLIST } n) \text{ AND } (\text{A BIKE})_m.\text{OWN } m \ n) \\
& \quad \text{AND NOT IT}_m.\text{RIDE } m \ n) \\
& = \text{NOT EXISTS}_n.\text{NOT} \\
& \quad \text{NOT } (((\text{CYCLIST } n) \text{ AND EXISTS}_m.(\text{BIKE } m) \text{ AND } (\text{OWN } m \ n)) \\
& \quad \text{AND NOT IT}_m.\text{RIDE } m \ n)
\end{aligned}$$

Because of the property BIKE, the context passed to the dynamic property containing IT contains a discourse referent whose descriptive content is consistent with being nonhuman. And so this term can be applied to the empty context τ , defined above in (4.4), to yield the following term by proposition C.6.

$$\begin{aligned}
& \vdash \lambda_{x^0}.\text{not exists}_y.\text{not not exists}_z.(\text{cyclist } y) \text{ and } (\text{bike } z) \text{ and} \\
& \quad (\text{own } z \ y) \text{ and not } (\text{ride } z \ y)
\end{aligned}$$

And by eliminating the double negation, we get

$$\begin{aligned}
& \equiv \lambda_{x^0}.\text{not exists}_y.\text{exists}_z.(\text{cyclist } y) \text{ and } (\text{bike } z) \text{ and} \\
& \quad (\text{own } z \ y) \text{ and not } (\text{ride } z \ y) : c_0 .
\end{aligned}$$

For the empty entity vector x^0 , this semantics gives the truth conditions intuitively associated with (4.53), because it says that there is no cyclist that owns some bike but does not ride it.

In chapter 5, I discuss anaphora in much greater detail, in the context of extending the dynamic semantics presented here to a full theory of conventional implicature. There I discuss Roberts's (2003) *weak familiarity* and its impacts on anaphoric accessibility and the so-called *proportion*

$$\frac{\vdash s : s ; D ; u : u \quad \vdash t : s ; S ; k : k}{\vdash s \cdot t : s ; D ; (cc k) \circ u : u} \text{ (Continue)}$$

Figure 4.1: DyCG natural deduction rule for discourse, where \circ is the composition function for updates defined in (4.67).

problem, along with a solution to it that involves redefining **EVERY** and other determiners. The intent of this chapter is simply to lay out the basic underpinnings of DyCG and show that it is capable of handling the kinds of phenomena that other dynamic theories are capable of.

4.4 Grammar Rules for Modeling Discourse

DyCG is intended to model discourse-level meanings, whereas CyCG stops at the level of sentences. Because of this, some extensions to the CyCG grammar rules in figure 3.1 are needed, which use the new nonlogical type D , of discourses. The necessary extension to the grammar rules is shown in figure 4.1.

A discourse is continued using the *Continue* rule, which takes an update and merges it with the content of an utterance into a new discourse. The resulting string is simply the concatenation $s \cdot t$ of the two discourses' concrete syntaxes, and the semantic content is simply the composition of the updates $(cc k)$ and u , where cc is the update function defined in (4.12) and $\circ : u \rightarrow u \rightarrow u$ is the standard infix function composition

$$(4.67) \quad \circ =_{\text{def}} \lambda_{u:u} \lambda_{v:u} \lambda_{c:c} . u (v c) .$$

Example B.24 discusses function composition in type theory in more detail.

The connection between dynamic conjunction, the context change function, and composition of updates is expressed in the following.

Theorem 4.22. *If h and k are contents, then $\vdash cc(h \text{ AND } k) = (cc k) \circ (cc h)$.*

Proof. Let $h : k$ and $k : k$. Then we have the following term identities by the definitions of **AND** (4.14), cc (4.12), and \circ (4.67), and the β -conversion

axiom (B.5):

$$\begin{aligned}
& \vdash \text{cc} (h \text{ AND } k) \\
& = \text{cc} (\lambda_{\mathbf{x}^{[c]}, \mathbf{y}^{[h]}, \mathbf{z}^{[k]}} (h \text{ c } \mathbf{x}, \mathbf{y}) \text{ and } (k (\text{cc } h \text{ c}) \mathbf{x}, \mathbf{y}, \mathbf{z})) \\
& = \lambda_c \lambda_{\mathbf{x}^{[c]}, \mathbf{y}^{[h]}, \mathbf{z}^{[k]}} \cdot (c \mathbf{x}) \text{ and } (h \text{ c } \mathbf{x}, \mathbf{y}) \text{ and } (k (\lambda_{\mathbf{x}^{[c]}, \mathbf{y}^{[h]}} (c \mathbf{x}) \text{ and } (h \text{ c } \mathbf{x}, \mathbf{y})) \mathbf{x}, \mathbf{y}, \mathbf{z}) \\
& = \lambda_c \cdot ((\lambda_c \lambda_{\mathbf{x}^{[c]}, \mathbf{y}^{[k]}} (c \mathbf{x}) \text{ and } (k \text{ c } \mathbf{x}, \mathbf{y})) \lambda_{\mathbf{x}^{[c]}, \mathbf{y}^{[h]}} (c \mathbf{x}) \text{ and } (h \text{ c } \mathbf{x}, \mathbf{y})) \\
& = \lambda_c \cdot (\text{cc } k) \lambda_{\mathbf{x}^{[c]}, \mathbf{y}^{[h]}} \cdot (c \mathbf{x}) \text{ and } (h \text{ c } \mathbf{x}, \mathbf{y}) \\
& = \lambda_c \cdot (\text{cc } k (\text{cc } h \text{ c})) \\
& = (\text{cc } k) \circ (\text{cc } h) : \mathbf{u}
\end{aligned}$$

Then it follows that $\vdash \forall_{hk} \cdot \text{cc} (h \text{ AND } k) = (\text{cc } k) \circ (\text{cc } h)$ from universal instantiation. \square

It is important to note about the Continue rule that it requires the variable contexts of its premises to be empty. This is done to prevent operators from taking scope over an entire discourse. As a result, all traces must be resolved at the sentence level, before the sentential sign is promoted to an update and used as in an instance of Continue.

We then define the **empty discourse** as

$$(4.68) \quad \vdash \mathbf{e} : \mathbf{s}; D; \lambda_{c:c} \cdot c : \mathbf{u},$$

whose pheno is the empty string \mathbf{e} and whose semantics is simply the identity function on contexts. Then we have the following derived rule:

$$(4.69) \quad \frac{\vdash s : \mathbf{s}; S; k : \mathbf{k}}{\vdash s : \mathbf{s}; D; (\text{cc } k) : \mathbf{u}} \text{ (Start)}$$

The Start rule says that any term of tecto type S whose semantics is a content can be promoted to a discourse with the same pheno and its semantics promoted to an update by cc . This rule is derived by invoking Continue using the empty discourse, for some sign $\vdash s : \mathbf{s}; S; k : \mathbf{k}$, as follows.

$$(4.70) \quad \frac{\vdash \mathbf{e} : \mathbf{s}; D; \lambda_{c:c} \cdot c : \mathbf{u} \quad \vdash s : \mathbf{s}; S; k : \mathbf{k}}{\vdash \mathbf{e} \cdot s : \mathbf{s}; D; (\text{cc } k) \circ \lambda_{c:c} \cdot c : \mathbf{u}} \text{ (Cn)}$$

(Here the label Cn abbreviates Continue.) Noting that $\mathbf{e} \cdot s = s$ by the axiom in (3.5) and the fact that

$$\begin{aligned} \vdash ((\mathbf{cc} k) \circ \lambda_{c:c}.c) &= \lambda_{c:c}.(\mathbf{cc} k) ((\lambda_{c:c}c) c) \\ &= \lambda_{c:c}.(\mathbf{cc} k) c \\ &= (\mathbf{cc} k) : \mathbf{u} \end{aligned}$$

by $\beta\eta$ -reduction, the root label of the proof in (4.70) reduces to the root label of the Start rule in (4.69).

As an example, consider the very simple discourse

(4.71) A cyclist_{*i*} arrived. She_{*i*} sneezed.

With the lexical entries for *A*, *cyclist*, and *sneezed* already defined, we need to state entries for the intransitive *arrived* and the pronoun *She*. Taking

$$\text{ARRIVE} =_{\text{def}} (\text{dyn}_1 \text{ arrive})$$

and

$$\text{SHE} =_{\text{def}} \lambda_{Dc}.D (\text{pro FEMALE } c) c ,$$

these lexical entries are straightforward:

$$\begin{aligned} \vdash \lambda_s.s \cdot \text{arrived} ; \text{NP} \multimap S ; \text{ARRIVE} : d_1 \\ \vdash \lambda_f.(f \text{ she}) ; (\text{NP} \multimap S) \multimap S ; \text{SHE} : d_1 \rightarrow k \end{aligned}$$

(The meanings for the proper names and pronouns will be generalized below in §5.2.1.) As for IT, note that the only difference between SHE and LANCE is the use of pro and dynamic property passed to pro. For the case of SHE, the dynamic property used is FEMALE =_{def} (dyn₁ female), where female is the static property of being female.

The DyCG sign for the first utterance in (4.71) is promoted to a discourse in (4.72) via the derived rule Start from (4.69), above.

$$(4.72) \quad \frac{\vdash a \cdot \text{cyclist} \cdot \text{arrived} : s ; S ; (\text{A CYCLIST ARRIVE}) : k}{\vdash a \cdot \text{cyclist} \cdot \text{arrived} : s ; D ; \mathbf{cc} (\text{A CYCLIST ARRIVE}) : \mathbf{u}}$$

This discourse is combined with the second utterance using Continue, as in (4.73), which shows just the abstract syntax and semantics.

$$(4.73) \quad \frac{\vdash D; \text{cc}(\text{A CYCLIST ARRIVE}) : u \quad \vdash S; (\text{SHE SNEEZE}) : k}{\vdash D; (\text{cc}(\text{SHE SNEEZE})) \circ \text{cc}(\text{A CYCLIST ARRIVE}) : u}$$

The concrete syntax corresponding to (4.73) is below.

$$(4.74) \quad \frac{\vdash a \cdot \text{cyclist} \cdot \text{arrived} : s \quad \vdash \text{she} \cdot \text{sneezed} : s}{\vdash a \cdot \text{cyclist} \cdot \text{arrived} \cdot \text{she} \cdot \text{sneezed} : s}$$

And so the DyCG discourse rules in figure 4.1 give an appropriate syntax and dynamic semantics that models (4.71). This can be seen by reducing the semantic term derived in (4.73):

$$\begin{aligned} & \vdash (\text{cc}(\text{SHE SNEEZE})) \circ \text{cc}(\text{A CYCLIST ARRIVE}) \\ &= \lambda_c.(\text{cc}(\text{SHE SNEEZE})) \text{cc}(\text{A CYCLIST ARRIVE}) c \\ &= \lambda_c.(\text{cc}(\text{SHE SNEEZE})) \lambda_{x|c|,y^1}.(c|c|(\mathbf{x}, \mathbf{y})) \text{ and } (\text{cyclist } y_0) \text{ and } (\text{arrive } y_0) : u \end{aligned}$$

This term can be further reduced and shorthanded to

$$\vdash \lambda_c.(\text{cc}(\text{SHE SNEEZE})) \lambda_{x|c|,y}.(c \mathbf{x}) \text{ and } (\text{cyclist } y) \text{ and } (\text{arrive } y) : u$$

Noting that

$$\vdash (\text{SHE SNEEZE}) = \lambda_{c\mathbf{x}|c|}.(\text{sneeze } \mathbf{x}_{(\text{pro FEMALE } c)}) : k,$$

the normal form of the semantic term derived in (4.73) is:

$$\begin{aligned} & \vdash (\text{cc}(\text{SHE SNEEZE})) \circ (\text{cc}(\text{A CYCLIST ARRIVE})) \\ &= \lambda_c \lambda_{\mathbf{x}|c|,y}.(c \mathbf{x}) \text{ and } (\text{cyclist } y) \text{ and } (\text{arrive } y) \text{ and } (\text{sneeze } \mathbf{x}_{(\text{pro FEMALE } c')}) : k_1 \end{aligned}$$

where

$$c' = \lambda_{\mathbf{x}|c|,y}.(c \mathbf{x}) \text{ and } (\text{cyclist } y) \text{ and } (\text{arrive } y).$$

Applying this semantics to the empty context τ , defined in equation (4.4), we get

$$\begin{aligned} & \vdash ((cc(SHE SNEEZE)) \circ cc(A CYCLIST ARRIVE)) \tau \\ & = \lambda_y.\text{true and (cyclist } y) \text{ and (arrive } y) \text{ and (sneeze } x_{(\text{pro } c'' \text{ FEMALE}))} \\ & \equiv \lambda_y.(\text{cyclist } y) \text{ and (arrive } y) \text{ and (sneeze } y) : c_1 \end{aligned}$$

by proposition C.8, where $c'' = \lambda_y.\text{true and (cyclist } y) \text{ and (arrive } y)$. The discourse referent y is selected as the argument to sneeze because it is the one that is consistent with being female, per the definition of *pro* in (4.21). Note that the context that results from applying the DyCG semantics for (4.71) to the empty context is interderivable with

$$\vdash \text{cyclist that arrive that sneeze} : e \rightarrow p$$

by proposition C.6.

4.5 Summary and Comparison with Other Theories

The dynamic semantics presented in this chapter is an encoding in pure (dependent) type theory of a semantics that models the meanings of declaratives as functions from discourse contexts to discourse contexts. In this way, it has much in common with the tradition of dynamic semantics that can be traced back to Heim (1982), especially with the work of Beaver (2001). Its approach to modeling the semantics of utterances also resembles the dynamic tradition dating to Groenendijk and Stokhof (1990, 1991). It also captures many central notions from Kamp's (1981) DRT, with the arguments to predicates modeled as discourse referents, and negation and quantifiers 'closing off' the accessibility of discourse referents introduced within their scope, and has much in common with Muskens's (1996) compositional, type-theoretic version of DRT and the recent computational dynamic semantics discussed by van Eijck and Unger (2010).

DyCG perhaps has the most in common with the work of de Groote (2006). Like de Groote, I give a straightforward mapping from a static semantics to a dynamic one based principally on redefinitions of the static operators and, exists, and not (although for de Groote, the underlying static semantics is not agnostic). And although, as I mention in the introduction to this chapter, DyCG does not explicitly employ the technique of *continua-*

tions, the type system it uses is underlyingly quite similar to [de Groote](#)'s. [De Groote](#)'s contexts, of type γ , are lists of entities, essentially identical to the entity vectors used here. For [de Groote](#), the meanings of declarative utterances are modeled by the type

$$\gamma \rightarrow (\gamma \rightarrow o) \rightarrow o,$$

where o is the truth value type, [de Groote](#)'s type for propositions. Thus the analog, in [de Groote](#)'s theory, of the DyCG type c of contexts is the type $\gamma \rightarrow o$. And since the order of the arguments in [de Groote](#)'s type of declaratives can be permuted to give

$$(\gamma \rightarrow o) \rightarrow \gamma \rightarrow o,$$

the type of declaratives in his theory is the analog of the type $c \rightarrow c$, which, modulo dependent typing information, is exactly the type of DyCG contents and updates. And so there is a deep similarity between [de Groote](#)'s work and other compositional dynamic theories that model declaratives as functions from contexts to contexts, including DyCG.

Finally, DyCG differs from my own previous work ([Martin, 2012](#); [Martin and Pollard, 2012a,b](#)) in that total functions are used rather than partial functions, although in [Kierstead and Martin \(2012\)](#) I opt for total functions as well. The reason, which will be greatly elaborated in chapters 5 and 6, is that DyCG as formulated here takes a completely different approach to anaphora. In this thesis, following the empirical discussion in chapter 2, anaphora is modeled not as constraining the input context but as giving rise to an implicature, which may conflict with the context in case there is no retrievable antecedent. As I will show in the next two chapters, this treatment of anaphora means that a general mechanism can also be used to model other conventional implicatures in a dynamic setting.

Chapter 5

Anaphora

Extending the compositional dynamic semantics in chapter 4 to handle implicatures more generally requires a way for the grammar to state which parts of the meanings of constructions are senses and which are implicatures. In this chapter I lay out a way to extend Dynamic Categorical Grammar that keeps these two meaning types separate. The approach is inspired by earlier work by [Karttunen and Peters \(1979\)](#) and [Nouwen \(2007\)](#), following the basic idea that the dynamic meanings of utterances determine a pair of propositions: one for its sense, and one for its implicature.

As chapter 2 discusses, the subclass of implicatures that must be anchored to the speaker is the subclass that includes anaphora, honorifics, and iterative adverbs. I give a DyCG account of many instances of anaphora, including proper names, pronouns, and definites using *the*, possessives, and iterative adverbs in the current chapter. In the literature, these implicatures are often referred to as bearing presuppositions, that is, contextual felicity constraints. But since the other subclasses of implicatures besides the anaphoric ones do not consistently place constraints on their context of interpretation, I simply use the term *anaphora* to refer to the ones that *do* constrain the context. Based on the new approach to anaphora explored here, I define a generalized notion of contextual felicity that is designed to account not only for the felicity conditions associated with anaphora, but for the felicity of the entire class of conventional implicatures, including the ones that can have a nonspeaker anchoring, discussed in chapter 6.

This chapter explores a robust account of anaphora that maintains a separation between sense and implicature. It models anaphoric triggers as giving rise to an implicature that a suitable antecedent is available in the

context of interpretation, with infelicity arising when this implicature is in conflict with the context because no such antecedent is available. This chapter also offers an expanded notion of the *accessibility* of an anaphoric antecedent compared with the usual accessibility relation in Discourse Representation Theory (Kamp, 1981; Kamp and Reyle, 1993), theories that are based on Heim (1982) such as Beaver 2001, and the mechanism for anaphora resolution found in the work of Groenendijk and Stokhof (1990, 1991). This expanded accessibility is achieved by implementing Roberts's (2003) *weak familiarity*, and is based in part on the account of weak familiarity in Martin 2012.

For reasons I discuss below, the definition of the determiner *EVERY* given in chapter 4, which is known as the *strong* reading, is not finely-grained enough. Drawing on examples of the use of various determiners in context, the account of anaphora I develop here, also makes *weak* determiner readings available, and I explore the conditions under which the strong or weak reading is chosen. Lastly, by basing anaphora in part on contextual entailments, this account represents an advance from de Groote 2006 and de Groote and Lebedeva 2010, which simply treat anaphora as an unspecified resolution function. In the case of proper names, de Groote and Lebedeva treat them as always requiring accommodation via *exception handling*. Instead, I model proper names as giving rise to anaphoric retrievability implications just like other instances of anaphora.

I start with the extensions to DyCG that are required for modeling both sense and implicature in §5.1, redefining some of the connectives and functions (§5.1.1). Then in §5.2, I lay out an account of anaphora using the newly defined two-level approach, giving a generalized formal treatment of definiteness (§5.2.1), including an account of possessives (§5.2.1.1). Next, §5.2.2 discusses a more general formal notion of contextual consistency and felicity than the one usually assumed by theories that treat presuppositions. §5.2.3 gives an account of the iterative adverb *too*. The notions of *familiarity* and *accessibility* are given a generalized treatment in the form of an implementation of *weak familiarity* in §5.3. Then §5.4 turns the focus to the *strong* and *weak* determiner readings, giving evidence for the two readings (§5.4.1) and then discussing two earlier approaches (§5.4.2 and §5.4.3) before proposing a synthesis of them in §5.4.4, including a general formal mechanism for yielding weak readings (§5.4.4.1). Finally, §5.5 offers a summary of this chapter and a comparison with some other

approaches, such as the work of Potts (2005) and the theory of anaphora due to van der Sandt (1992).

5.1 Extending Dynamic Categorical Grammar

To extend the DyCG developed in chapter 4 to a dynamic semantics that keeps sense and implicature separate, some retrofitting is required. Specifically, we need to redefine the type k of contents, the dynamic connectives AND and NOT (including some of the functions they rely on), and the dynamic quantifier EXISTS. For convenience, I also define the n -ary **empty contexts** $T_n : c_n$ for each natural number n , as follows:

$$(5.1) \quad T_n =_{\text{def}} \lambda x^n. \text{true}$$

And so the empty context defined in (4.4) can be redefined as $T =_{\text{def}} T_0$.

Starting with the type of contents, we use a dynamic version of the account in Karttunen and Peters 1979, which models meanings as pairs of expressions in Montague's (1973) *intensional logic*. Accordingly, for a natural numbers n , the type k_n , defined in equation (4.6), is supplanted by the type

$$(5.2) \quad k_n =_{\text{def}} \Pi_{c:c} \cdot \Sigma_{d:c|c|+n} \cdot c|d| \cdot$$

Intuitively, this is the type of functions that take a context as argument and bifurcates it into a pair of contexts, with n new discourse referents added that are available to both the first component and the second component. Following Karttunen and Peters, the first context is called the **sense context** (or just the **sense**), and the second is called the **implicature context** (or simply **implicature**). (Karttunen and Peters refer to these as the *extension expression* and *implicature expression*, respectively.) Drawing on Karttunen and Peters's notation, define the functions $(\cdot)^s$ and $(\cdot)^i$, both of type $\Pi_{n:n} \Pi_{p:\Sigma_{d:c_n} \cdot c|d|} \cdot c_n$, as follows, for n a natural number:

$$\begin{aligned} (\cdot)_n^s &=_{\text{def}} \lambda_{p:\Sigma_{d:c_n} \cdot c|d|} \cdot (\pi_1 p) \\ (\cdot)_n^i &=_{\text{def}} \lambda_{p:\Sigma_{d:c_n} \cdot c|d|} \cdot (\pi_2 p) \end{aligned}$$

As shorthand, I omit the natural number subscript on these functions, and write $c \mid d$ to abbreviate the pair of contexts $\langle c, d \rangle$.

$$\text{dyn}_0 =_{\text{def}} \lambda_{p:p} \lambda_{c:c} . \lambda_{\mathbf{x}^{|c|}} . p \mid \mathbf{T}_{|c|} ,$$

5.1.1 Redefining the Dynamic Logic

$$(5.3) \quad \text{cc} =_{\text{def}} \lambda_{k:k} \lambda_{c:c} \lambda_{\mathbf{x}^{[c]}, \mathbf{y}^{[k]}}. (c \mathbf{x}) \text{ and } ((kc)^s \mathbf{x}, \mathbf{y}) \text{ and } ((kc)^i \mathbf{x}, \mathbf{y}).$$

Turning to the dynamic connectives and quantifier, dynamic conjunction must now operate on contents as redefined above in (5.2). Its definition in (4.14) is replaced by

which may seem a bit involved. Intuitively, it conjoins both the sense and implicature contexts of k and h in the same way as the previous definition, but it takes care to ensure that the two content types are kept separate.

$$\begin{aligned} & \vdash \text{RAIN AND SNOW} \\ &= \lambda_{\mathbf{c}}. \lambda_{\mathbf{x}^{|\mathbf{c}|}}. \text{rain and snow} \mid \lambda_{\mathbf{x}^{|\mathbf{c}|}}. \text{true and true} \\ &\equiv \lambda_{\mathbf{c}}. \lambda_{\mathbf{x}^{|\mathbf{c}|}}. \text{rain and snow} \mid \mathbf{T}_{|\mathbf{c}|} : \mathbf{k}_0 \end{aligned}$$

(Recall that $|\text{RAIN}| = 0$ and $|\text{SNOW}| = 0$.) Thus the sense context of RAIN AND SNOW is the conjoined proposition rain and snow , but its impli-

cature context is essentially empty, containing only the necessarily true proposition true.

Nothing essentially changes for the definition of the dynamic existential **EXISTS**. Dynamic negation, on the other hand, needs a redefinition, because it must target only the sense proposition and not the implicature proposition of its content argument. Its type does not change, but its new definition supplants the one in (4.17).

$$(5.5) \quad \text{NOT} =_{\text{def}} \lambda_{kc}.\lambda_{x|c|}.\text{not exists}_{y|k|}.(kc)^s x, y \mid \lambda_{x|c|}.\text{exists}_{y|k|}.(kc)^i x, y$$

In this version of dynamic negation, the discourse referents introduced become existentially bound in both the sense and implicature parts of the resulting content. But importantly, only the sense proposition is negated.

To demonstrate the way the redefined dynamic negation, the negation of **EXISTS CYCLIST** is shown below.

$$\begin{aligned} & \vdash \text{NOT} (\text{EXISTS CYCLIST}) \\ & = \text{NOT } \lambda_c.\lambda_{x|c|}.\lambda_{y^1}.\text{cyclist } y_0 \mid T_{|c^+|} \\ & = \lambda_c.\lambda_{x|c|}.\text{not exists}_{y^1}.\text{cyclist } y_0 \mid \lambda_{x|c|}.\text{exists}_{y^1}.\text{true} : k_0 \end{aligned}$$

Applying the shorthand notation, η -reduction, lemma 4.11, and proposition C.6, this term can be written as

$$\vdash \lambda_c.\lambda_{x|c|}.\text{not exists cyclist} \mid T_{|c|} : k_0.$$

In what follows, I will engage in a minor abuse of notation in that terms that are interderivable according to proposition C.6 are sometimes substituted for each other when it is convenient, for example, to reduce clutter.

Analog of theorems 4.12 and 4.13 are available for the two-level setting.

Theorem 5.1. *If $k : k_0$, then*

$$\vdash (\text{NOT } k) \equiv \lambda_c.\lambda_{x|c|}.\text{not } ((kc)^s x) \mid (kc)^i.$$

Proof. Note that, for $k : k_0$, we have

$$\vdash (\text{NOT } k) = \lambda_c.\lambda_{x|c|}.\text{not exists}_{y^0}.(kc)^s x, y \mid \lambda_{x|c|}.\text{exists}_{y^0}.(kc)^i x, y$$

by equation (5.5). Similarly to the proof of theorem 4.12, invoke lemma 4.11, proposition C.8, substitution, and η -reduction to obtain

$$\begin{aligned} & \vdash \lambda_c. \lambda_{x|c}. \text{not exists}_{y^0}. (k\ c)^s\ x, y \mid \lambda_{x|c}. \text{exists}_{y^0}. (k\ c)^i\ x, y \\ & \equiv \lambda_c. \lambda_{x|c}. \text{not} ((k\ c)^s\ x) \mid (k\ c)^i. \end{aligned}$$

□

Theorem 5.2. *If $k : k_0$, then $\vdash (\text{NOT} (\text{NOT } k)) \equiv k$.*

Proof. For $k : k_0$, by an argument similar to the proof of theorem 4.13, we have

$$\begin{aligned} \vdash (\text{NOT} (\text{NOT } k)) & \equiv \lambda_c. \lambda_{x|c}. \text{not} (\text{not} ((k\ c)^s\ x)) \mid (k\ c)^i \\ & \equiv \lambda_c. (k\ c)^s \mid (k\ c)^i \\ & \equiv k, \end{aligned}$$

invoking axiom B.8.

□

An analog of theorem 4.22 is also available, by systematically replacing the instances of *cc* and *AND* with their corresponding redefinitions.

5.2 Anaphora in a Two-Level Setting

Giving an account of anaphora in this multi-level dynamic semantics allows a rethinking of both Potts's (2005) characterization of the behavior of conventional implicatures as well as the notion of *infelicity* from the literature on presuppositions. In the first case, viewing anaphora as a special case of conventional implicature allows us to take issue with Potts's empirical generalization that no lexical item can have both a sense contribution and an implicature contribution. In the second, we can go beyond the view of infelicity as the failure of a presupposition to be satisfied in a given context, toward a broader view of infelicity based on contextual entailments and Grice's (1975) *cooperative principle*. The relevant contextual entailments used to model anaphora should reflect the intuition discussed in §2.2.1, that the implications associated with anaphora are obligatorily anchored to the speaker because they are implications about the discourse context itself.

To explore these ideas further, I start with an examination of the dynamic meanings of proper names. For example, in chapter 4, I treated the

proper name *Lance* as having the DyCG translation

$$(4.34) \quad \text{LANCE} =_{\text{def}} \lambda_{Dc}. D (\text{the NAMED-LANCE } c) c .$$

That is, LANCE is a dynamic generalized quantifier that passes to its dynamic property argument the unique discourse referent in the input context that is entailed to have the name *Lance*. But this definition has nothing to offer in cases where there *is* no such discourse referent. The definiteness function the is defined in terms of the description operator ι , and because of the way ι is defined (see definition B.8 and theorem B.9), if there is no unique discourse referent meeting the relevant condition, then an arbitrary (but fixed) inhabitant of $\omega_{|c|}$ is selected. As it stands, where most theories would predict presupposition failure, DyCG would simply predict indeterminate behavior.

In Martin and Pollard (2012a,b), contents are modeled as partial functions from contexts to contexts, so that in case of presupposition failure, the offending context is simply not in the presupposing content's domain. But for DyCG, which uses only total functions, I propose a different treatment that extends the definition in (4.34) to the multi-level case. A prerequisite for giving a revised definition of LANCE is a definition of the propositional unique existential quantifier.

Definition 5.3 (Unique Existential). For each meaning type A , the quantifier

$$\text{exists!}_A : (A \rightarrow p) \rightarrow p$$

is axiomatized as

$$(5.6) \quad \vdash \forall_{P:A \rightarrow p} \forall_{w:w}. ((\text{exists!}_A P) @ w) \Leftrightarrow \exists!_{x:A}. ((P x) @ w),$$

where $\exists!$ is as in definition B.7.

As usual, the type parameter subscript A is often suppressed when it is irrelevant or inferable from context.

Then a definition of $\text{LANCER} : d_1 \rightarrow k$ that takes the conventional implicature perspective is

$$(5.7) \quad \text{LANCER} =_{\text{def}} \lambda_{Dc}. (D (\text{the NAMED-LANCER } c) c)^s \mid (D (\text{the NAMED-LANCER } c) c)^i \\ \text{and exists!}_{n:\omega_{|c|}}. (c \text{ k-entails } (\text{NAMED-LANCER } n)) .$$

The sense proposition of LANCER is just as before. What is new is the implicature proposition that there exists a unique natural number $n < |c|$ such that the context entails that the entity at coordinate n has the property of being named *Lance*. As I discuss below in §5.3, discourse referents may be explicitly introduced into DyCG by the use of an indefinite, or introduced via a pragmatic entailment-driven inference process.

5.2.1 Generalized Definiteness

As discussed in §2.2.2.1, especially in connection with examples (2.55) and (2.56), the descriptive content implications are different depending on the definite. For example, consider a version of (2.55) that uses a proper name instead of a pronoun:

(2.55) A cop_{*i*} just walked over to my car. She_{*i*} wrote me a ticket for my busted tail light!

(5.8) A cop_{*i*} just walked over to my car. # Lance_{*i*} wrote me a ticket for my busted tail light!

The difference between (2.55) and (5.8) is that proper names require their antecedent to be contextually entailed to have the property of being so named, while the use of a pronoun only require that their antecedent's descriptive content not *conflict* with the pronoun's descriptive content. In (2.55), the use of the pronoun *She* may be the first time the addressee learns that the cop is female, but for (5.8), the use of *Lance* cannot be the first time the addressee learns the cop's name is *Lance*.

In DyCG, this difference between pronouns and other definites is handled by the two functions *the* and *pro* from definitions 4.20 and 4.21. These functions are repeated below.

$$(4.27) \quad \text{the} =_{\text{def}} \lambda_{D:d_1} \lambda_{c:c} \lambda_{n:\omega_{|c|}}. c \text{ k-entails } (D n)$$

$$(4.28) \quad \text{pro} =_{\text{def}} \lambda_{D:d_1} \lambda_{c:c} \lambda_{n:\omega_{|c|}}. c \text{ k-cons } (D n)$$

Seeing these functions side by side makes it clear that the only difference between the way nonpronominal definites are treated and the way pronouns are treated is that for pronouns, all that is required is that the context be consistent with the relevant descriptive content, while other definites make the stronger requirement of (contextual) entailment of the descriptive content. As [Roberts \(2005\)](#) explains, there is more to this story, because pronouns additionally require maximal salience among the discourse referents entailed to have nonconflicting properties. But since salience is mainly a pragmatic phenomenon, I leave it unaccounted for in the DyCG semantic theory, with the assumption that anaphora resolution for pronouns incorporates an unspecified check for maximal salience.¹

With this difference in mind, two general functions can be used to encode definiteness in a dynamic setting.

Definition 5.4 (Generalized Definitizers). The **generalized definitizer functions** **THE** and **PRO**, both with the type $d_1 \rightarrow d_1 \rightarrow k$ of dynamic generalized determiners, are defined as follows:

(5.9)

$$\begin{aligned} \text{THE} =_{\text{def}} & \lambda_{DEc}. (E (\text{the } D c) c)^s \mid \\ & \lambda_{x|c}. (((D \text{ THAT } E) (\text{the } D c) c)^i x) \text{ and exists!}_{n:\omega|c}. (c \text{ k-entails } (D n)) \end{aligned}$$

(5.10)

$$\begin{aligned} \text{PRO} =_{\text{def}} & \lambda_{DEc}. ((D \text{ THAT } E) (\text{pro } D c) c)^s \mid \\ & \lambda_{x|c}. (((D \text{ THAT } E) (\text{pro } D c) c)^i x) \text{ and exists!}_{n:\omega|c}. (c \text{ k-cons } (D n)) \end{aligned}$$

The implicature proposition in a dynamic generalized determiner generated by either of these functions embodies the retrievability implication discussed in §2.2.1.1: the implication is that a contextually unique discourse referent must be available that bears the relevant property. Note that any implicature content associated with either *D* or *E* is inherited by the term that results from applying either of **THE** or **PRO**, in a way reminiscent of [Karttunen and Peters 1979](#).

Whether the discourse referent must be entailed to have the relevant property or merely not entailed to have a conflicting property depends

¹As [Stone and Webber \(1998\)](#) point out, even this unspecified salience check may not be enough in certain situations, in which a further notion of *plausibility* is required to disambiguate which is the intended antecedent. I leave plausibility this beyond the DyCG theory of anaphora as well.

on whether THE or PRO is used. For the definitizer function THE, note that in this generalized determiner's sense proposition, the unique discourse referent that the context entails to have the property D is selected. Part of its implicature proposition is the requirement that such a discourse referent is available among the ones in the context's input vector. However, for the case of PRO, the discourse referent selected must only have descriptive content that does not conflict with the property D , and its implicature part contains the relevant proposition. For PRO, the content of D is also incorporated into the sense of the resulting determiner, since, as (2.55) shows, the pronoun's sortal information may be new.

This definition of THE is somewhat distantly related to the one given by Beaver (2001, definition 54). To see how THE works, consider its application to the dynamic property NAMED-LANCE $=_{\text{def}} (\text{dyn}_1 \text{ named-lance})$.

$$\begin{aligned} & \vdash (\text{THE NAMED-LANCE}) \\ & = \lambda_{Dc}. (D (\text{the NAMED-LANCE } c) c)^s \mid \\ & \quad \lambda_{x|c|}. (((\text{NAMED-LANCE THAT } D) (\text{the NAMED-LANCE } c) c)^i x) \text{ and} \\ & \quad \text{exists!}_{n:\omega|c|}. c \text{ k-entails } (\text{NAMED-LANCE } n) : d_1 \rightarrow k \end{aligned}$$

Note that (THE NAMED-LANCE) has a meaning that is closely similar to the one given for LANCE in (5.7). Any implicature content associated with either property NAMED-LANCE or D is inherited by the result, and the implication is added that there is a retrievable antecedent that is contextually entailed to be named *Lance*. The dynamic meaning of the proper name *Lance* can then be defined instead as

$$(5.11) \quad \text{LANCE} =_{\text{def}} \text{THE NAMED-LANCE},$$

and similarly for other proper names.

The inanimate pronoun can be given a redefinition based on PRO that supplants the one in (4.54), as follows:

$$(5.12) \quad \text{IT} =_{\text{def}} \text{PRO NONHUMAN}$$

With PRO as defined in (5.10), this definition of IT expands to

$$\begin{aligned}
 & \vdash \text{PRO NONHUMAN} \\
 & = \lambda_{Dc}.((\text{NONHUMAN THAT } D) (\text{pro NONHUMAN } c) c)^s \mid \\
 & \quad \lambda_{\mathbf{x}|c}.(((\text{NONHUMAN THAT } D) (\text{pro NONHUMAN } c) c)^i \mathbf{x}) \text{ and} \\
 & \quad \text{exists!}_{n:\omega|c}.c \text{ k-cons } (\text{NONHUMAN } n) : d_1 \rightarrow k,
 \end{aligned}$$

where $\text{NONHUMAN} =_{\text{def}} (\text{dyn}_1 \text{ nonhuman})$ is the dynamic property of being nonhuman. Similarly, with the dynamic properties

$$\begin{aligned}
 \text{FEMALE} &=_{\text{def}} (\text{dyn}_1 \text{ female}) & \text{and} \\
 \text{MALE} &=_{\text{def}} (\text{dyn}_1 \text{ male}),
 \end{aligned}$$

where $\text{female} : e \rightarrow p$ and $\text{male} : e \rightarrow p$ are respectively the property of being female and the property of being male, dynamic meanings for the gendered pronouns *she* and *he* are available.

$$(5.13) \quad \text{SHE} =_{\text{def}} \text{PRO FEMALE}$$

$$(5.14) \quad \text{HE} =_{\text{def}} \text{PRO MALE}$$

So proper names and the pronouns are all defined in terms of THE and PRO. By way of illustrating that definites in DyCG behave as expected, consider the discourse

$$(5.15) \quad \text{A cyclist commutes to work. A driver commutes to work. The cyclist rides a bike.}$$

To model (5.15) in DyCG, first define $\text{DRIVER} =_{\text{def}} (\text{dyn}_1 \text{ driver})$ as the dynamic meaning of *driver* and $\text{COMMUTE} =_{\text{def}} (\text{dyn}_1 \text{ commute})$ as the meaning of *commutes to work*, where *driver* and *commute* are the static properties of being a driver and commuting to work, respectively. Then add the following lexical entries:

$$\begin{aligned}
 & \vdash \lambda_{sf}.f(\text{the } \cdot s) ; N \multimap (NP \multimap S) \multimap S ; \text{THE} \\
 & \vdash \text{driver} ; N ; \text{DRIVER} \\
 & \vdash \lambda_s.s \cdot \text{commutes} ; NP \multimap S ; \text{COMMUTE}
 \end{aligned}$$

(Lexical entries for the remaining words in (5.15) are defined as in chapter 4.)

Given these extensions to the lexicon, signs for the first two utterances in (5.15) are as follows.

$$\begin{aligned} &\vdash a \cdot \text{cyclist} \cdot \text{commutes} ; S ; (\text{A CYCLIST COMMUTE}) \\ &\vdash a \cdot \text{driver} \cdot \text{commutes} ; S ; (\text{A DRIVER COMMUTE}) \end{aligned}$$

After promoting the first utterance using the Start rule, it is combined with the second utterance via Continue, to yield

$$(5.16) \quad \vdash a \cdot \text{cyclist} \cdot \text{commutes} \cdot a \cdot \text{driver} \cdot \text{commutes} ; D ; \\ (\text{cc} (\text{A DRIVER COMMUTE})) \circ (\text{cc} (\text{A CYCLIST COMMUTE})) .$$

A sign modeling the third utterance of (5.15) is

$$\vdash \text{the} \cdot \text{cyclist} \cdot \text{rides} \cdot a \cdot \text{bike} ; S ; (\text{THE CYCLIST})_n . (\text{A BIKE})_m . \text{RIDE } m \ n ,$$

which, after combination with the discourse formed from the first two utterances, gives

$$\begin{aligned} &\vdash a \cdot \text{cyclist} \cdot \text{commutes} \cdot a \cdot \text{driver} \cdot \text{commutes} \cdot \text{the} \cdot \text{cyclist} \cdot \text{rides} \cdot a \cdot \text{bike} ; \\ &D ; (\text{cc} ((\text{THE CYCLIST})_n . (\text{A BIKE})_m . \text{RIDE } m \ n)) \circ \\ &(\text{cc} (\text{A DRIVER COMMUTE})) \circ (\text{cc} (\text{A CYCLIST COMMUTE})) . \end{aligned}$$

Since the semantics of (5.16) reduces to a term equivalent to

$$\begin{aligned} &\vdash \lambda_c . \lambda_{x|c|, y, z} . (c \ x) \text{ and } (\text{cyclist } y) \text{ and } (\text{commute } y) \text{ and} \\ &(\text{driver } z) \text{ and } (\text{commute } z) : u_2 , \end{aligned}$$

the semantics of the third utterance is able to select the unique discourse referent with the entailment of being a cyclist. Applying this semantics to the empty context τ_0 , we arrive at an equivalent β -normal form of the semantics of this discourse.

$$\begin{aligned} &\vdash ((\text{cc} (\text{THE CYCLIST}_n . (\text{A BIKE})_m . \text{RIDE } m \ n)) \circ \\ &(\text{cc} (\text{A DRIVER COMMUTE})) \circ (\text{cc} (\text{A CYCLIST COMMUTE}))) \text{ T} \\ &\equiv \lambda_{x, y, z} . (\text{cyclist } x) \text{ and } (\text{commute } x) \text{ and } (\text{driver } y) \text{ and } (\text{commute } y) \text{ and} \\ &(\text{bike } z) \text{ and } (\text{ride } z \ x) \text{ and exists!}_{n: \omega|c'|} . c' \text{ k-entails } (\text{CYCLIST } n) : c_3 \end{aligned}$$

Here,

$$c' = \lambda_{x,y}.\text{true and (cyclist } x) \text{ and (commute } x) \text{ and} \\ \text{(driver } y) \text{ and (commute } y) : c_2$$

is the context passed to the third utterance by the preceding discourse.

To see how the DyCG treatment of pronouns differs from its treatment of the definite determiner *the*, consider the discourse

(5.17) A cyclist_i commutes to work. She_i arrives on time.

Allowing the definition of *arrives* from chapter 4 as the lexical entry for *arrives on time*, all of the lexical entries needed to derive a DyCG sign for (5.17) are already defined. Using the rules Start and Continue as before, the sign corresponding to (5.17) is

$$\vdash a \cdot \text{cyclist} \cdot \text{commutes} \cdot \text{she} \cdot \text{arrives} ; D ; \\ (\text{cc (SHE ARRIVE)}) \circ (\text{cc (A CYCLIST COMMUTE)})$$

Since the dynamic meaning of the first utterance is

$$\vdash \text{A CYCLIST COMMUTE} \\ \equiv \lambda_c.\lambda_{x|c|,y}.\text{(cyclist } y) \text{ and (commute } y) \mid \tau_{|c^+|} : k_1 ,$$

the resulting context is passed by AND as the argument to the second utterance's meaning, which is

$$\vdash \text{SHE ARRIVE} = \lambda_c.\lambda_{x|c|}.\text{(female } x_{(\text{pro FEMALE } c)}) \text{ and (arrive } x_{(\text{pro FEMALE } c)}) \mid \\ \lambda_{x|c|}.\text{exists!}_{n:\omega_{|c|}}.c \text{ k-cons (FEMALE } n) : k_0 .$$

The second utterance's meaning selects the unique discourse referent that is not entailed to be nonfemale, namely, the cyclist. Then the β -reduced meaning of (5.17) is

$$\vdash (\text{cc (SHE ARRIVE)}) \circ (\text{cc (A CYCLIST COMMUTE)}) \\ = \lambda_c.\lambda_{x|c|,y}.(c \text{ } x) \text{ and (cyclist } y) \text{ and (commute } y) \text{ and} \\ \text{(female } x, y_{(\text{pro FEMALE } c')}) \text{ and (arrive } x, y_{(\text{pro FEMALE } c')}) \text{ and} \\ \text{exists!}_{n:\omega_{|c'|}}.c' \text{ k-cons (FEMALE } n) : u_1 ,$$

where, similarly as above,

$$c' = \lambda_{x|c|,y}.(c\ x) \text{ and } (\text{cyclist } y) \text{ and } (\text{commute } y) : c_{|c|+1}$$

is the context passed to $(cc\ (\text{SHE ARRIVE}))$.

As before for (5.15), we apply this meaning to the empty context, yielding

$$\begin{aligned} & \vdash (((cc\ (\text{SHE ARRIVE})) \circ cc\ (\text{A CYCLIST COMMUTE}))\ T) \\ & \equiv \lambda_y.(\text{cyclist } y) \text{ and } (\text{commute } y) \text{ and } (\text{female } y) \text{ and } (\text{arrive } y) \text{ and} \\ & \quad \text{exists!}_{n:\omega_{|c'|}}.c''\ \text{k-cons } (\text{FEMALE } n) : c_1, \end{aligned}$$

Here the context

$$c'' \equiv \lambda_y.(\text{cyclist } y) \text{ and } (\text{commute } y) : c_1$$

is the context passed to $(cc\ (\text{SHE ARRIVE}))$. With c'' as input, SHE is able to select the unique discourse referent whose entailments do not conflict with the property of being female, namely the cyclist discourse referent y . Note that SHE adds the information that the cyclist is female.

5.2.1.1 Possessives

The definitions of definites can also be used to define dynamic meanings for possessive determiners. As above for the definitizer function, I want to define a function that describes the meanings of all possessives in the most general way possible. Clearly, such a function must factor out the genderedness of possessive pronouns in order to generalize dynamic meanings for *her* and *his*. But recalling the discussion of possessives in §2.2.1.3, possessive determiners can sometimes function as definites and other times as indefinites. And so a general possessivizer function should factor out the variable behavior of possessives as definites and as indefinites as well.

With these requirements in mind, define the possessivizer function possessive with the type

$$\text{possessive} : (d_1 \rightarrow d_1 \rightarrow k) \rightarrow (d_1 \rightarrow k) \rightarrow d_1 \rightarrow d_1 \rightarrow k,$$

which takes a dynamic generalized determiner and a dynamic generalized quantifier to a dynamic generalized determiner. Its definition is inspired by the definition of HIS_s in [Martin 2012](#), equation (16):

$$(5.18) \quad \text{possessive} =_{\text{def}} \lambda_{ABDE}.A \ (D \text{ THAT } \lambda_m.(B_n.\text{HAVE } m \ n)) \ E$$

Here $\text{HAVE} =_{\text{def}} (\text{dyn}_2 \text{ have})$ is the dynamicization of the static binary property $\text{have} : p_2$ of possession. The typing of *possessive* says that A is a dynamic generalized determiner, B is a dynamic generalized quantifier, and D and E are dynamic properties.

The definition of the possessivizer in (5.18) allows definite versions of the possessive determiners *her*, *his*, and *its*, as follows.

$$(5.19) \quad \text{HER}_{\text{def}} =_{\text{def}} \text{possessive THE SHE}$$

$$(5.20) \quad \text{HIS}_{\text{def}} =_{\text{def}} \text{possessive THE HE}$$

$$(5.21) \quad \text{ITS}_{\text{def}} =_{\text{def}} \text{possessive THE IT}$$

Expanding these definitions show how they work to model their respective possessive determiners.

$$\vdash \text{HER}_{\text{def}} = \lambda_D.\text{THE} \ (D \text{ THAT } \lambda_m.(\text{SHE}_n.\text{HAVE } m \ n))$$

$$\vdash \text{HIS}_{\text{def}} = \lambda_D.\text{THE} \ (D \text{ THAT } \lambda_m.(\text{HE}_n.\text{HAVE } m \ n))$$

$$\vdash \text{ITS}_{\text{def}} = \lambda_D.\text{THE} \ (D \text{ THAT } \lambda_m.(\text{IT}_n.\text{HAVE } m \ n))$$

(Here, the semantic terms are written in η -reduced form to reduce clutter.) Similarly, indefinite versions of the possessive determiners are available by passing the indefinite A to *possessive* rather than the definite *THE*.

$$(5.22) \quad \text{HER}_{\text{indef}} =_{\text{def}} \text{possessive A SHE}$$

$$(5.23) \quad \text{HIS}_{\text{indef}} =_{\text{def}} \text{possessive A HE}$$

$$(5.24) \quad \text{ITS}_{\text{indef}} =_{\text{def}} \text{possessive A IT}$$

These give indefinite versions of the definite possessive determiners above, again in η -reduced form:

$$\begin{aligned} \vdash \text{HER}_{\text{indef}} &= \lambda_D.A (D \text{ THAT } \lambda_m.(\text{SHE}_n.\text{HAVE } m \ n)) \\ \vdash \text{HIS}_{\text{indef}} &= \lambda_D.A (D \text{ THAT } \lambda_m.(\text{HE}_n.\text{HAVE } m \ n)) \\ \vdash \text{ITS}_{\text{indef}} &= \lambda_D.A (D \text{ THAT } \lambda_m.(\text{IT}_n.\text{HAVE } m \ n)) \end{aligned}$$

The indefinite versions of the possessives capture the observations of [Barker \(2000\)](#) and [Pollard and Smith \(2011\)](#) discussed in §2.2.1.3.

To see how these definitions give different results in practice, apply both in turn to model the simple utterance

(5.25) His bike arrives.

The possessive *His* corresponds to two different lexical entries, which differ only in their semantics:

$$\begin{aligned} \vdash \lambda_{sf}.(f(\text{his} \cdot s)) ; N \multimap (NP \multimap S) \multimap S ; \text{HIS}_{\text{def}} \\ \vdash \lambda_{sf}.(f(\text{his} \cdot s)) ; N \multimap (NP \multimap S) \multimap S ; \text{HIS}_{\text{indef}} \end{aligned}$$

Depending on which lexical entry is chosen, the two semantics corresponding to *His* can yield one of the following two signs.

(5.26) $\vdash \text{his} \cdot \text{bike} \cdot \text{arrives} ; S ; (\text{HIS}_{\text{def}} \text{BIKE ARRIVE})$

(5.27) $\vdash \text{his} \cdot \text{bike} \cdot \text{arrives} ; S ; (\text{HIS}_{\text{indef}} \text{BIKE ARRIVE})$

First, we reduce the meaning of the definite version of *His*, to yield:

$$\begin{aligned} &\vdash \text{HIS}_{\text{def}} \text{BIKE ARRIVE} \\ &= \text{THE} (\text{BIKE THAT } \lambda_m.\text{HE}_n.\text{HAVE } m \ n) \text{ ARRIVE} \\ &\equiv \lambda_c.\lambda_{\mathbf{x}|c}.(\text{ARRIVE} (\text{the} (\text{BIKE THAT } \lambda_m.\text{HE}_n.\text{HAVE } m \ n) c) c)^s \mathbf{x} \mid \\ &\quad \lambda_{\mathbf{x}|c}.(\text{exists!}_{m:\omega|c} c \text{ k-entails } ((\text{BIKE } m) \text{ AND } \text{HE}_n.\text{HAVE } m \ n)) \text{ and} \\ &\quad \text{exists!}_{n:\omega|c}.c \text{ k-cons } (\text{MALE } n) : k \end{aligned}$$

In this version, the implicature proposition requires that there be both a male discourse referent and a bike that he has. But for the indefinite

version, only the requirement of the male discourse referent is present:

$$\begin{aligned}
& \vdash \text{HIS}_{\text{indef}} \text{BIKE ARRIVE} \\
& = \text{A (BIKE THAT } \lambda_m.\text{HE}_n.\text{HAVE } m\ n) \text{ ARRIVE} \\
& = \text{EXISTS}_m.(\text{BIKE } m) \text{ AND } (\text{HE}_n.\text{HAVE } m\ n) \text{ AND } (\text{ARRIVE } m) \\
& \equiv \lambda_c.\lambda_{\mathbf{x}^{[c]},y}.(\text{bike } y) \text{ and } (\text{male } \mathbf{x}, y_{(\text{pro MALE } c')}) \text{ and} \\
& \quad (\text{have } y\ \mathbf{x}, y_{(\text{pro MALE } c')}) \text{ and } (\text{arrive } y) \mid \\
& \quad \lambda_{\mathbf{x}^{[c]},y}.\text{exists!}_{n:\omega_{[c']}.c'} \text{ k-cons (MALE } n) : \text{k}
\end{aligned}$$

In this case, the context passed to $\lambda_m.\text{HE}_n.\text{HAVE } m\ n$ is

$$c' \equiv \lambda_{\mathbf{x}^{[c]},y}.(c\ \mathbf{x}) \text{ and } (\text{bike } y) : c_{[c]+1},$$

which contains the information that the newly-introduced discourse referent is a bike.

5.2.2 A Broader Notion of Contextual Felicity

With two possible readings for every possessive determiner, DyCG as a theory now embodies the hypothesis that possessives are always ambiguous between definite and indefinite. In this section, I propose a generalized notion of contextual felicity to explain how speakers perform the disambiguation in the course of interpreting discourse. This new notion of felicity is generalized in the sense that presupposition satisfaction is a special case of it.

To motivate this notion of felicity, I briefly recall Grice's (1975) *cooperative principle*, especially as it pertains to his *maxim of quality*. Grice defines the cooperative principle as follows:

We might then formulate a rough general principle which participants will be expected (*ceteris paribus*) to observe, namely: Make your conversational contribution such as is required, at the stage at which it occurs, by the accepted purpose or direction of the talk exchange in which you are engaged. (Grice, 1975, page 167 in Martinich 2001)

More relevant to the task of disambiguating the possessive determiners is this passage about quality:

Under the category of *quality* falls a supermaxim—“Try to make your contribution one that is true”—and two more specific maxims:

1. Do not say what you believe to be false.
2. Do not say that for which you lack adequate evidence.

(Grice, 1975, page 168 in Martinich 2001)

Following Grice, what serves to disambiguate the definite and indefinite readings of possessive determiners is the addressee’s assumption that the speaker is cooperative and, therefore, observing the maxim of quality. And so if the definite reading of the determiner would end up making the context inconsistent (in the sense of entailing a contradiction), then the indefinite reading is preferred. On the other hand, if the definite reading is felicitous, then it is preferred on the grounds that otherwise the speaker would have used the indefinite *a* because it clarifies that the speaker does not intend for an antecedent to be retrievable. Similarly, I follow the standard assumption in dynamic semantics since Kamp (1981) and Heim (1982) that an indefinite introduces a discourse referent, and thus implicates that the speaker did not intend for the addressee to retrieve an already familiar one.

To make these ideas more concrete, I define the notions of *consistency* and *felicity* as DyCG semantic terms.

Definition 5.5 (Contextual Consistency). A context c is *consistent* if and only if it is consistent with the content $\lambda_{c:c.T} | T$, where $T = T_0$ is as defined above. The function $\text{consistent} : c \rightarrow t$, defined as

$$(5.28) \quad \text{consistent} =_{\text{def}} \lambda_c.(c \text{ k-cons } \lambda_{c:T} | T) \equiv \text{true} ,$$

can be used as a test for consistency.

To see how the consistent function works, let $c : c$ be a context. Then by lemma 4.11 and definitions 4.16, 4.17, 4.18 and 4.19, we have

$$\begin{aligned}
& \vdash \text{consistent } c \\
& = (c \text{ k-cons } \lambda_c.T \mid T) \equiv \text{true} \\
& = (\text{not } (c \text{ k-entails } (\text{NOT } \lambda_c.T \mid T))) \equiv \text{true} \\
& = (\text{not } (c \text{ c-entails } (cc (\text{NOT } \lambda_c.T \mid T) c))) \equiv \text{true} \\
& = (\text{not } (c \text{ c-entails } \lambda_{x|c}.(c \ x) \text{ and } (\text{not true}) \text{ and true})) \equiv \text{true} \\
& = (\text{not forall}_{x|c}.(c \ x) \text{ Entails } ((c \ x) \text{ and false})) \equiv \text{true} \\
& = (\text{not forall}_{x|c}.(c \ x) \text{ Entails false}) \equiv \text{true}
\end{aligned}$$

And so the test for the consistency of a context c is a test whether c applied to every vector of the required arity entails the necessarily false proposition at every world.

Felicity is in turn based on contextual consistency.

Definition 5.6 (Felicity). A content k is felicitous in a context c if and only if the context obtained by promoting k to an update and then applying it to c is consistent in the sense defined in 5.5. This property is encoded in the function $\text{felicitous-in} : k \rightarrow c \rightarrow t$, written infix:

$$(5.29) \quad \text{felicitous-in} =_{\text{def}} \lambda_{kc}.\text{consistent } (cc \ k \ c)$$

The notions of consistency and felicity are not encoded in the grammar because they are part of the pragmatic processes used by speakers in the course of interpreting discourse, and not strictly part of the semantics of utterances themselves. But the definition of felicity in (5.29) is intended to reflect the fact that speakers consider the effects of content that is *proffered* for acceptance or rejection (Roberts, 1996) before integrating the new content into the discourse context. It is imaginable that speakers apply something like the felicitous-in function in the course of the invoking Continue rule (figure 4.1), before drawing the conclusion. As I discuss further in chapter 6, the definitions in equations (5.28) and (5.29) are general enough to also describe cases when conventional implicatures with variable commitment status are used infelicitously.

By way of illustrating how this process works, consider the two variant analyses corresponding to (5.25) in (5.26) and (5.27). In a situation in which (5.25) is uttered ‘out of the blue,’ with little or no prior context shared

by the interlocutors, the felicity test would fail for the definite reading (5.26), since the empty context does not provide enough information for an antecedent for *his bike* to be retrieved:

$$\vdash \neg ((\text{HIS}_{\text{def}} \text{BIKE ARRIVE}) \text{felicitous-in } \tau)$$

One reason is that a portion of the implicature part of the meaning of (5.25), namely the proposition

$$(5.30) \quad \text{exists!}_{m:\omega|c}.c \text{ k-entails } ((\text{BIKE } m) \text{ AND } \text{HE}_n.\text{HAVE } m \ n),$$

where c is the input context, is not consistent with the empty context τ . As a result, the indefinite reading (5.26) would be preferred in this case. (A second reason that the definite reading is infelicitous in an ‘out-of-the-blue’ context is that there is no antecedent for the pronoun HE .)

But consider (5.25) interpreted in a discourse context containing the utterance

$$(5.31) \quad \text{A cyclist owns a bike.}$$

Note that the DyCG semantics of (5.31) is as follows:

$$\begin{aligned} & \vdash (\text{A CYCLIST})_n.(\text{A BIKE})_m.\text{OWN } m \ n \\ & = \text{EXISTS}_n.(\text{CYCLIST } n) \text{ AND } \text{EXISTS}_m.(\text{BIKE } m) \text{ AND } (\text{OWN } m \ n) \\ & \equiv \lambda_c.\lambda_{x|c}.y.z.(\text{cyclist } y) \text{ and } (\text{bike } z) \text{ and } (\text{own } z \ y) \mid \tau_{|c^{++}} : k \end{aligned}$$

Therefore, the definite reading (5.26) is felicitous in the context updated by (5.31), since

$$\vdash (\text{HIS}_{\text{def}} \text{BIKE ARRIVE}) \text{felicitous-in } (\text{cc } ((\text{A CYCLIST})_n.(\text{A BIKE})_m.\text{OWN } m \ n) \tau).$$

The reason the definite reading is felicitous in this case is that both the implicature propositions are consistent with the context that results from applying the dynamic meaning of (5.31) to the empty context, and it is therefore preferred to the indefinite reading. The proposition in (5.30) then becomes consistent due to the presence of a suitable bike discourse referent and the entailment between owning and having. As for the other implicature proposition

$$\text{exists!}_{n:\omega|c'}.c' \text{ k-cons } (\text{MALE } n),$$

where

$$c' \equiv \lambda_{y,z}.(\text{cyclist } y) \text{ and } (\text{bike } z) \text{ and } (\text{own } z \ y) ,$$

it is also consistent with the relevant context, since being a cyclist does not entail being nonmale.

5.2.3 The Iterative Adverb *Too*

The DyCG approach to anaphora is general enough to extend to instances of anaphora that do not involve discourse referents in the same way that proper names, pronouns, and other definites do. As discussed in §2.2.1.2, example (2.35) (when it bears focus on *John*) bears the retrievability implication the context entails that a salient individual other than John went to Harvard.

(2.35) John went to Harvard, too.
(Heim, 1990b, example 14)

In this section, I discuss how many of the same mechanisms used to model definite anaphora in DyCG can be harnessed to model the dynamic meaning of *too*, including its attendant retrievability implication.

First note that the following discourse is infelicitous when *John* bears the focus:

(5.32) John went to Harvard. # John went to Harvard, too.

We clearly need a way to determine when two discourse referents are the same as a preliminary to giving an analysis of examples like (2.35) and (5.32). We start by giving a way to compare intensions for equality.

Definition 5.7 (Intensional Equality). The infix **intensional equality** function

$$\text{equals}_A : A \rightarrow A \rightarrow p ,$$

for each meaning type A , is axiomatized by

$$\vdash \forall_{a:A} \forall_{b:A} \forall_{w:w}. (a \text{ equals } b) @ w \Leftrightarrow a = b .$$

As usual, the type subscript is usually dropped when it is irrelevant. Based on intensional equality, we can then define the following function to handle the duty of comparing discourse referents.

Definition 5.8 (Contextually Entailed Equality). The **contextually entailed equality** function $c\text{-equals} : A \rightarrow A \rightarrow p$, written infix, compares two inhabitants of a meaning type A for equality as follows:

$$(5.33) \quad c\text{-equals} =_{\text{def}} \lambda_{a:A} \lambda_{b:A} \lambda_{c:c} \cdot \text{forall}_{\mathbf{x}|c|} \cdot (c \mathbf{x}) \text{ Entails } (a \text{ equals } b)$$

Here, Entails is as defined in 4.16.

With a way to compare discourse referents, we can give a definition of the dynamic meaning of *too*, with the type

$$\text{too} : d_1 \rightarrow d_1 ,$$

defined as

$$(5.34) \quad \text{too} =_{\text{def}} \lambda_{Dnc} \cdot (D \ n \ c)^s \mid \lambda_{\mathbf{x}|c|} \cdot ((D \ n \ c)^i \ \mathbf{x}) \text{ and } \text{exists}_{m:\omega|c|} \cdot (c \ k\text{-entails } (D \ m)) \text{ and not } (m \ c\text{-equals } n) .$$

The only contribution of *too* as defined in (5.34) is to add the implicature proposition that there must exist a discourse referent besides the one to which the property D applies that also has the property in question. This dynamic treatment of *too*, inspired by Heim 1990b, is similar to the one proposed by Beaver (2001, definition D17). Note that this definition is somewhat simplified in that focus is not taken into account.

Giving a lexicon for modeling (2.35) and (5.32) is straightforward. The name *John* is modeled similarly to *Lance* in (5.11), as

$$(5.35) \quad \text{JOHN} =_{\text{def}} \text{THE NAMED-JOHN} ,$$

where $\text{NAMED-JOHN} =_{\text{def}} (\text{dyn}_1 \text{ named-john})$ and $\text{named-john} : e \rightarrow p$ is the static property of being named *John*. And for simplicity, the property $\text{GO-HARVARD} =_{\text{def}} (\text{dyn}_1 \text{ go-harvard})$ is the dynamic property of going to

Harvard, where $\text{go-harvard} : e \rightarrow p$ is its static counterpart.

$$\begin{aligned} &\vdash \lambda_f.(f \text{ John}) ; (\text{NP} \multimap \text{S}) \multimap \text{S} ; \text{JOHN} \\ &\vdash \lambda_s.s \cdot \text{went} \cdot \text{to} \cdot \text{Harvard} ; \text{NP} \multimap \text{S} ; \text{GO-HARVARD} \\ &\vdash \lambda_{fs}.(f s) \cdot \text{too} ; (\text{NP} \multimap \text{S}) \multimap (\text{NP} \multimap \text{S}) ; \text{TOO} \end{aligned}$$

This lexicon is presented in short form, with the pheno and semantic types elided.

The concrete syntactic proof starts by combining *too* with the verb phrase.

$$(5.36) \quad \frac{\vdash \lambda_{fs}.(f s) \cdot \text{too} : (s \rightarrow s) \rightarrow s \rightarrow s \quad \vdash \lambda_s.s \cdot \text{went} \cdot \text{to} \cdot \text{Harvard} : s \rightarrow s}{\vdash \lambda_s.s \cdot \text{went} \cdot \text{to} \cdot \text{Harvard} \cdot \text{too} : s \rightarrow s}$$

Next, *John* takes the new verb phrase with *too* appended as its argument.

$$(5.37) \quad \frac{\vdash \lambda_f.(f \text{ John}) : (s \rightarrow s) \rightarrow s \quad \vdash \lambda_s.s \cdot \text{went} \cdot \text{to} \cdot \text{Harvard} \cdot \text{too} : s \rightarrow s}{\vdash \text{John} \cdot \text{went} \cdot \text{to} \cdot \text{Harvard} \cdot \text{too} : s} \quad \begin{array}{c} (5.36) \\ \vdots \end{array}$$

So the lexical entry for *too* given above yields the correct surface string for (2.35).

Turning to the combined abstract syntax and semantics, we begin in the same way, by combining *too* with the verb phrase.

$$(5.38) \quad \frac{\vdash (\text{NP} \multimap \text{S}) \multimap (\text{NP} \multimap \text{S}) ; \text{TOO} \quad \vdash \text{NP} \multimap \text{S} ; \text{GO-HARVARD}}{\vdash \text{NP} \multimap \text{S} ; (\text{TOO GO-HARVARD})}$$

Then as above, *John* takes the modified verb phrase as argument.

$$(5.39) \quad \frac{\vdash (\text{NP} \multimap \text{S}) \multimap \text{S} ; \text{JOHN} \quad \vdash \text{NP} \multimap \text{S} ; (\text{TOO GO-HARVARD})}{\text{S} ; (\text{JOHN} (\text{TOO GO-HARVARD}))} \quad \begin{array}{c} (5.38) \\ \vdots \end{array}$$

Reducing the semantic term, we have the following:

$$\begin{aligned}
 & \vdash \text{JOHN} (\text{TOO GO-HARVARD}) \\
 & = \text{THE NAMED-JOHN} (\text{TOO GO-HARVARD}) \\
 & \equiv \lambda_c. \lambda_{x|c|.} (\text{go-harvard } x_{(\text{the NAMED-JOHN } c)}) \mid \\
 & \quad \lambda_{x|c|.} (\text{exists!}_{n:\omega|c|.} c \text{ k-entails } (\text{NAMED-JOHN } n)) \text{ and} \\
 & \quad \text{exists}_{m:\omega|c|.} (c \text{ k-entails } (\text{GO-HARVARD } m)) \text{ and} \\
 & \quad \text{not } (m \text{ c-equals } (\text{the NAMED-JOHN } c)) : k
 \end{aligned}$$

And so the sense proposition of the dynamic meaning of (2.35) is simply that John went to Harvard, but the implicature proposition contains the propositions that there is someone named *John* in the context, and that someone other than John is contextually entailed to have gone to Harvard. Also, note that we have

$$\vdash \neg (\text{JOHN} (\text{TOO GO-HARVARD}) \text{ felicitous-in } (\text{cc JOHN GO-HARVARD } \tau)),$$

since the implicature proposition that some other discourse referent has the property of going to Harvard is in conflict with a discourse context containing only the information that John went to Harvard. Therefore, this DyCG definition of *too* captures the infelicity in (5.32) as well.

Finally, this DyCG definition of the meaning of *too* captures Heim’s (1990b) reason for presenting (2.35), namely that it is not enough simply for there to be a general entailment that someone not contextually entailed to be John went to Harvard. Presumably, this requirement would never fail to be satisfied, as Kripke (2009) notes for the similar example (2.34a). The implicature associated with *too* is stronger, that someone that has been previously mentioned in the current discourse context bears the relevant entailment. The definition of *TOO* in (5.34) captures this requirement, because

$$\vdash \neg (\text{JOHN} (\text{TOO GO-HARVARD}) \text{ felicitous-in } \tau).$$

The reason (2.35) is not felicitous in the empty context τ is that part of its implicature context, namely the proposition

$$\begin{aligned}
 & \text{exists}_{m:\omega|c|.} (c \text{ k-entails } (\text{GO-HARVARD } m)) \text{ and} \\
 & \text{not } (m \text{ c-equals } (\text{the NAMED-JOHN } c)),
 \end{aligned}$$

is in conflict with the empty context, since there does not exist a discourse referent that is contextually entailed to go to Harvard.

On the other hand, the DyCG semantics

$$\vdash (\text{KIM GO-HARVARD}) \text{ AND } (\text{JOHN (TOO GO-HARVARD)}) : k,$$

which would model an utterance like

(5.40) Kim went to Harvard, and John went to Harvard too ,

is felicitous in contexts in which antecedents for both *Kim* and *John* are retrievable. The reason is that there is a discourse referent in the context passed to *too*, besides the one corresponding to *John*, that is contextually entailed to have gone to Harvard.

5.3 Generalized Familiarity and Accessibility

The relation of *accessibility* is one of the central features of the early dynamic theories of Discourse Representation Theory (DRT, [Kamp, 1981](#); [Kamp and Reyle, 1993](#)), File Change Semantics (FCS, [Heim, 1982, 1992](#)), Dynamic Predicate Logic (DPL, [Groenendijk and Stokhof, 1990](#)) and Dynamic Montague Grammar (DMG, [Groenendijk and Stokhof, 1991](#)). In these theories, a potential antecedent is accessible from a definite such as a pronoun only if the antecedent does not occur within the scope of a semantic operator that does not also extend to the definite itself. The following examples, drawn from [Martin 2012](#), examples A–C, illustrate accessibility.

(5.41) If Pedro owns $\left\{ \begin{array}{c} a \\ \# \text{ every} \end{array} \right\}$ donkey_{*i*} he beats it_{*i*}.
([Kamp 1981](#), examples 1, 17)

(5.42) Everybody found a cat_{*i*} and kept it_{*i*}. # It_{*i*} ran away.
([Heim 1983a](#), example 5)

(5.43) $\left\{ \begin{array}{c} A \\ \# \text{ No} \end{array} \right\}$ donkey_{*i*} brays. Its_{*i*} name is ‘Chiquita.’

In (5.41)–(5.43), the variants in which a potential antecedent falls within the scope of a quantifier demonstrate limited accessibility for anaphoric linking. Constructions involving negation and conditionals also limit accessibility, as (5.44) and (5.45) show.

(5.44) A farmer doesn't own any donkey_{*i*}. # It_{*i*}'s brown.

(5.45) If a farmer_{*i*} owns a donkey_{*j*}, he_{*i*} beats it_{*j*}. # The farmer_{*i*} is from Ohio.

In (5.44), the donkey discourse referent introduced inside the negation cannot serve as the antecedent to the pronoun in the second utterance. Similarly, in (5.45), the intended antecedent to *The farmer_i* is within the antecedent of the conditional in the first utterance, and is inaccessible as a result.

This notion of accessibility is pervasive in the dynamic semantics tradition for two main reasons. The first is that essentially all dynamic theories can be characterized as descendants of one of the three main branches. For DRT, the list includes the work of Muskens (1994, 1996), Bos (2003, 2005), Blackburn and Bos (2005, 1999), van Eijck and Unger (2010), and Venhuizen, Bos, and Brouwer (2013); for FCS, Beaver (2001), Martin and Pollard (2012a,b), Martin (2012), and this thesis; for DPL/DMG, Chierchia (1992, 1995), de Groote (2006), Nouwen (2007), AnderBois, Brasoveanu, and Henderson (2010), de Groote and Lebedeva (2010), and Lebedeva (2012). The second reason is that the idea that semantic operators limit the accessibility of potential antecedents is built into all three of the main branch theories (and their descendants) at a deep level. In the case of FCS and DPL/DMG, accessibility is encoded into the meaning of certain determiners and the *If ... , then ...* construction. In the case of DRT, accessibility runs even deeper, as Chierchia and Rooth (1984) point out: DRT accessibility is a necessary consequence of the models required for interpreting its discourse representation structures.

Unfortunately, the notion of accessibility that is so tightly coupled with all dynamic theories is too strict, causing the model of anaphora presented by these theories to undergenerate. Perhaps the most famous counterexample to classic accessibility is the 'bathroom' example

(5.46) Either this house doesn't have a bathroom_{*i*}, or it_{*i*}'s in a funny place.

(Roberts 1989, example 18, attributed to Barbara Partee)

Since the discourse referent introduced by *a bathroom_i* in (5.46) is within the scope of a negation, accessibility of dynamic theories treat it on a par with (5.44), predicting that it is inaccessible outside the scope of the enclosing negation. However, on the intended interpretation, clearly the pronoun

it_i means the bathroom mentioned in the *Either ...* clause. As another counterexample, consider the following simple discourse.

- (5.47) Every cyclist owns a bike. One cyclist rides the bike he owns every day.

Following dynamic accessibility, the discourse referent *a bike* in (5.47), which could potentially antecede *the bike he owns*, is trapped within the scope of *Every*. But here the definite *the bike he owns* is completely felicitous; it seems to have been inferred, in some sense, by the information in the preceding utterance.

Several proposed fixes for the undergeneration caused by the overly restrictive nature of the accessibility relation in dynamic theories. Groenendijk and Stokhof (1991) give variants of the dynamic quantifiers and connectives in DMG which allow the anaphoric link evidenced in (5.46) by allowing discourse referents to escape the scope of their enclosing operators. However, these variant DPL/DMG quantifiers and connectives are too permissive, failing to rule out the inaccessibility demonstrated in (5.41)–(5.45).

The *E-Type pronoun* approach, prominently exemplified in the work of Evans (1977, 1980), Cooper (1979), Heim (1990a), Neale (1990), and Chierchia (1995), treats pronouns as disguised definite descriptions in Russell's (1905) sense. This approach suffers from a problem similar to the one that Groenendijk and Stokhof's scope-extension proposal suffers from. E-Type theories do not countenance a notion of anaphoric accessibility, since pronouns are modeled as borrowing their descriptive content from a previously occurring noun phrase, with no constraints on what counts as 'previously occurring.' Thus, in the general case, E-Type theories share an inability to rule out infelicitous examples like (5.44), just like Groenendijk and Stokhof's proposal for DMG. A second deficiency of E-Type theories is their requirement that the antecedent be *semantically* unique, in the sense of being the only entity in the world bearing the relevant descriptive content, rather than *informationally* unique, in the sense of being the only discourse referent in the current context of interpretation that is entailed to bear the description in question. For lengthy and convincing discussion of the deficiencies of E-Type theories, see Roberts 2005.

5.3.1 Implementing Weak Familiarity

Here I explore an alternative proposed by [Roberts \(2003, 2005\)](#) called *weak familiarity*. The approach [Roberts](#) spells out is already foreshadowed by [Groenendijk and Stokhof \(1991\)](#), who do not develop it further.

[...] [(5.48)] and [(5.49)] [...] seem to express the same thing:

- (5.48) It is not the case that John doesn't own a car. It is red and parked in front of the house.
- (5.49) John owns a car. It is red and parked in front of the house.

These examples suggest that at least in cases such as these, a sentence and its double negation are fully equivalent, i.e., that they do not just have the same truth conditions, but also the same dynamic properties. [...]

It may be worthwhile, though, to briefly point out another line of reasoning, which does not strike us as altogether untenable. It seems possible to argue that in [(5.48)] the pronoun *it* is not directly anaphorically related to the indefinite term *a car* in the preceding sentence, but only indirectly, mediated through a unique object, the existence of which can be inferred from the previous discourse and the context.

([Groenendijk and Stokhof, 1991](#), §7; examples (5.48) and (5.49) are respectively numbered 42 and 43.)

[Roberts's](#) approach generalizes the *Extended Novelty-Familiarity-Condition* of [Heim \(1982\)](#) discussed in §2.2.1.1 in the following way. The familiarity part of [Heim's](#) condition is loosened so that rather than requiring the explicit prior introduction of a discourse referent with the relevant descriptive content, only the contextual entailment of a referent matching the relevant content is needed. The benefit of the weak familiarity approach is that examples like (5.46) and (5.47), in which an antecedent is merely entailed, can be accounted for, while true instances of inaccessible antecedents like those in (5.41)–(5.45) are still ruled out.

In [Martin 2012](#) I pursued a strategy for implementing weak familiarity in which pronouns and other definites were allowed to take as their antecedents discourse referents that were not explicitly present but merely

entailed. Unfortunately, this strategy is not without considerable problems. These problems are centered around the fact that, while this approach correctly allows anaphoric links that are ruled out in other dynamic theories, anaphors and their antecedents are still in some sense disconnected. This occurs because the weakly familiar versions of definites are essentially indefinites that introduce discourse referents with impoverished descriptive content associated with them.

By way of illustrating this problem, note that the truth conditions for (5.46) can be paraphrased as *Either there's no bathroom in this house, or there is a bathroom in this house and that bathroom is in a funny place*. But by defining a weakly familiar version of *it* as an entailment-licensed indefinite, the descriptive content that the antecedent is entailed to have, namely the property of being a bathroom, is mostly lost: we are left only with the relatively poor descriptive content associated with the pronoun *it*.

The upshot of this loss of descriptive content is that the [Martin 2012](#) approach predicts different behavior for the following definites:

- (5.50) a. Either this house doesn't have a bathroom_{*i*}, or the bathroom_{*i*}'s in a funny place and I really wonder where it_{*i*} could be.
 b. Either this house doesn't have a bathroom_{*i*}, or it_{*i*}'s in a funny place and I really wonder where the bathroom_{*i*} could be.

The only difference between (5.50a) and (5.50b) is the order in which the definites *the bathroom_{*i*}* and *it_{*i*}* occur. For (5.50a), the [Martin 2012](#) implementation of weak familiarity essentially does the right thing: the weakly familiar *the bathroom_{*i*}* introduces a new bathroom discourse referent that in turn serves as the antecedent to the (now strongly familiar) pronoun *it_{*i*}*. But for (5.50b), both the pronoun *it_{*i*}* and the later definite *the bathroom_{*i*}* introduce separate discourse referents, and the semantics does not link these two discourse referents in any way. The reason is that the weakly familiar *it_{*i*}* in (5.50b) introduces a nonhuman discourse referent, but this discourse referent's descriptive content is not suitable to serve as the antecedent to a strongly familiar version of *the bathroom_{*i*}*. As a result, the weakly familiar *the bathroom_{*i*}* is required, and because of the entailment of the existence of a bathroom, it introduces its own bathroom discourse referent. The resulting context contains separate discourse referents for a nonhuman and for a bathroom, but this is clearly not the intended reading of (5.50b).

Here, rather than trying to encode entailment-based reasoning in the DyCG semantics as I did in [Martin 2012](#), I instead adopt the tack of

assuming a pragmatic process that introduces discourse referents into the context based on other entailments that are present, in cases where no corresponding overt discourse referent is available. This process constitutes a second way in which the arity of a context can be extended besides the use of an indefinite, and I assume that it exists on the same plane as the process I posited that handles the encoding of maximal salience for the antecedents of pronouns. This pragmatic mechanism functions as an auxiliary to the process of discourse interpretation, with explicitly stated content giving rise to other potential antecedents for anaphora during the course of interpreting the explicit content.

To see how this pragmatic implementation of weak familiarity works, consider a simplified variant of the ‘bathroom’ example (5.46):

(5.51) The house has no bathroom_{*i*}, or it_{*i*}’s in a funny place.

To start with, we need to define dynamic meanings for *house* and *bathroom*. These common nouns are straightforwardly dynamic properties, defined as follows.

$$\begin{aligned}\text{HOUSE} &=_{\text{def}} (\text{dyn}_1 \text{ house}) \\ \text{BATHROOM} &=_{\text{def}} (\text{dyn}_1 \text{ bathroom})\end{aligned}$$

where $\text{house} : p_1$ and $\text{bathroom} : p_1$ are respectively the properties of being a house and being a bathroom. Next, for simplicity, we define the property of being in a funny place as

$$\text{IN-FUNNY-PLACE} =_{\text{def}} (\text{dyn}_1 \text{ in-funny-place}) ,$$

where $\text{in-funny-place} : p_1$ is a static simplification of the property in question. And lastly, the dynamic generalized determiner

$$\text{NO} : d_1 \rightarrow d_1 \rightarrow k_0$$

is defined as

$$(5.52) \quad \text{NO} =_{\text{def}} \lambda_{DE} . \text{NOT} (A D E) .$$

Assuming that THE and HAVE are as defined in (4.27) and (5.18), an analysis of the first disjunct in (5.51) is available.

$$\begin{aligned}
& \vdash (\text{THE HOUSE})_n.(\text{NO BATHROOM})_m.\text{HAVE } m \ n \\
& \equiv \lambda_c.\lambda_{x|c}.(((\text{NO BATHROOM})_m.\text{HAVE } m \ (\text{the HOUSE } c)))^s \mid \\
& \quad \lambda_{x|c}.(((\text{HOUSE THAT } \lambda_n.(\text{NO BATHROOM})_m.\text{HAVE } m \ n) \ (\text{the HOUSE } c)))^i \\
& \quad \text{and exists!}_{n:\omega|c}.c \text{ k-entails } (\text{HOUSE } n) \\
& \equiv \lambda_c.\lambda_{x|c}.\text{not exists}_y.(\text{bathroom } y) \text{ and } (\text{have } y \ x_{(\text{the HOUSE } c)}) \mid \\
& \quad \lambda_{x|c}.\text{exists!}_{n:\omega|c}.c \text{ k-entails } (\text{HOUSE } n) : k
\end{aligned}$$

The dynamic meaning of the first disjunct is straightforward, stating that there is no bathroom that the unique house in the context has. We can also derive a dynamic meaning for the second utterance, as follows:

$$\begin{aligned}
& \vdash \text{IT IN-FUNNY-PLACE} \\
& \equiv \lambda_c.\lambda_{x|c}.(\text{nonhuman } x_{(\text{pro NONHUMAN } c)}) \text{ and } (\text{in-funny-place } x_{(\text{pro NONHUMAN } c)}) \mid \\
& \quad \lambda_{x|c}.\text{exists!}_{n:\omega|c}.c \text{ k-cons } (\text{NONHUMAN } n) : k
\end{aligned}$$

Recalling that the dynamic disjunction $\text{OR} : k \rightarrow k \rightarrow k_0$ is given the DeMorgan-inspired definition

$$(4.21) \quad \text{OR} =_{\text{def}} \lambda_{hk}.\text{NOT} ((\text{NOT } h) \text{ AND } (\text{NOT } k)),$$

the dynamic meaning of (5.51) is modeled as

$$\begin{aligned}
& \vdash ((\text{THE HOUSE})_n.(\text{NO BATHROOM})_m.\text{HAVE } m \ n) \text{ OR} \\
& \quad (\text{IT IN-FUNNY-PLACE}) \\
& = \text{NOT} ((\text{NOT} ((\text{THE HOUSE})_n.(\text{NO BATHROOM})_m.\text{HAVE } m \ n) \text{ AND} \\
& \quad \text{NOT } (\text{IT IN-FUNNY-PLACE}))) : k.
\end{aligned}$$

Unpacking this semantic term shows that the use of *it* is licensed because its implicature is consistent with entailments present in the context of interpretation. Note that, by the definition of OR in (4.21), the first disjunct of the meaning of (5.51) is negated. As a result, the context passed to the

(negated) second disjunct is equivalent to

$$\lambda_{x|c}.(c \ x) \text{ and } (\text{not not exists}_y.(\text{bathroom } y) \text{ and } (\text{have } y \ x_{(\text{the HOUSE } c)})) \text{ and} \\ \text{exists!}_{n:\omega|c}.c \text{ k-entails } (\text{HOUSE } n) : c$$

where c is the original input context to (5.51). After eliminating the double negation, this context is equivalent to

$$(5.53) \quad \lambda_{x|c}.(c \ x) \text{ and } (\text{exists}_y(\text{bathroom } y) \text{ and } (\text{have } y \ x_{(\text{the HOUSE } c)})) \text{ and} \\ \text{exists!}_{n:\omega|c}.c \text{ k-entails } (\text{HOUSE } n) : c.$$

And so the implicature associated with (IT IN-FUNNY-PLACE), namely that its input context is consistent with the existence of a discourse referent that is nonhuman, is felicitous in this context.

In cases of weak familiarity such as this one, Roberts (2003, 2005) refers to the antecedent *a bathroom_i* as the *licensing noun phrase*. I make the assumption that the pragmatic process of weak familiarity extends the context in (5.53), in which the licensing noun phrase has given rise to an existential entailment of the bathroom's existence, to the context

$$(5.54) \quad \lambda_{x|c}.y.(c \ x) \text{ and } (\text{exists}_z(\text{bathroom } z) \text{ and } (\text{have } z \ x_{(\text{the HOUSE } c)})) \text{ and} \\ (\text{bathroom } y) \text{ and } (\text{have } y \ x_{(\text{the HOUSE } c)})) \text{ and} \\ \text{exists!}_{n:\omega|c}.c \text{ k-entails } (\text{HOUSE } n) : c|c|+1.$$

Note that, in this context, a new discourse referent y is made available that has the properties of the discourse referent that is entailed to exist in (5.53). (Note that the bound variable has been renamed to avoid confusion.) More generally, for P a property and q a proposition, this pragmatic process converts a context of the form

$$\lambda_{x,y}.q : c_n$$

such that $\lambda_{x,y}.q$ c-entails $\lambda_{x,y}.(\text{exists } P)$, to a context of the form

$$\lambda_{x,y}.q \text{ and } (P \ y) : c_{n+1},$$

making explicitly accessible a discourse referent y that is merely entailed to exist.

With the context so extended to (5.54), the semantics of (5.51) is

$$\begin{aligned}
 & \vdash ((\text{THE HOUSE})_n.(\text{NO BATHROOM})_m.\text{HAVE } m\ n) \text{ OR} \\
 & \quad (\text{IT IN-FUNNY-PLACE}) \\
 & \equiv \lambda_c.\lambda_{\mathbf{x}|c|,y}.\text{(exists}_z(\text{bathroom } z) \text{ and (have } z\ \mathbf{x}_{(\text{the HOUSE } c)})\text{)) and} \\
 & \quad (\text{bathroom } y) \text{ and (have } y\ \mathbf{x}_{(\text{the HOUSE } c)}) \text{ and} \\
 & \quad (\text{nonhuman } \mathbf{x}, y_{(\text{pro NONHUMAN } c')}) \text{ and (in-funny-place } \mathbf{x}, y_{(\text{pro NONHUMAN } c')}) \mid \\
 & \quad \lambda_{\mathbf{x}|c|,y}.\text{(exists!}_{n:\omega|c|}c \text{ k-entails (HOUSE } n)\text{)) and} \\
 & \quad \text{exists!}_{n:\omega|c'|}c' \text{ k-cons (NONHUMAN } n) : k_1 .
 \end{aligned}$$

Here c' is the context (5.54) that is extended by weak familiarity before being passed to the second disjunct. Thus the discourse referent introduced via weak familiarity on the basis of the existential entailment triggered by the licensing noun phrase *a bathroom_i* is accessible to, and compatible with, the pronoun *it_i* in the second disjunct, as desired. In a context c containing no other maximally salient discourse referent compatible with being nonhuman, the weakly familiar bathroom discourse referent y is selected by $(\text{pro NONHUMAN } c')$, yielding

$$\begin{aligned}
 & \lambda_{\mathbf{x}|c|,y}.\text{(exists}_z(\text{bathroom } z) \text{ and (have } z\ \mathbf{x}_{(\text{the HOUSE } c)})\text{)) and} \\
 & \quad (\text{bathroom } y) \text{ and (have } y\ \mathbf{x}_{(\text{the HOUSE } c)}) \text{ and} \\
 & \quad (\text{nonhuman } y) \text{ and (in-funny-place } y) : c_1
 \end{aligned}$$

as its sense. Clearly this corresponds to the intuitively correct truth conditions associated with the sense of (5.51).

This example shows that weak familiarity succeeds where Heim's (1982) stronger version of familiarity fails. However, we have to be careful to assume that weak familiarity also interacts with the pragmatic process for limiting the discourse referents under consideration for the antecedents of anaphora to only the most salient one(s), following Roberts (2005). Without additionally specifying that the salience ranking mechanism interacts with the process of introducing of weakly familiar, entailed discourse referents, DyCG would analyze (5.51) to be infelicitous. This is because Π 's associated implicature

$$\text{exists!}_{n:\omega|c|}c \text{ k-cons (NONHUMAN } n)$$

would be inconsistent with the context if the house discourse referent was taken as most salient. The discourse referent for the house in (5.51) is the only overt, strongly familiar discourse referent whose descriptive content is consistent with being nonhuman, and thus the intended reading of (5.51) would fail if the salience filter did not also make the entailed discourse referent for the bathroom the most salient one.

For an example of weak familiarity across a discourse of multiple utterances, consider the following simplified variant of (5.47).

(5.55) Every cyclist owns a bike. A cyclist rides the bike that he owns.

Since dynamic meanings corresponding to all the words in (5.55) have been defined, a DyCG semantic term modeling the discourse in (5.55) can be given:

$$\vdash (\text{cc} (\text{A CYCLIST})_n. (\text{THE} (\text{BIKE THAT } \lambda_m. \text{HE}_n. \text{OWN } m \ n))_m. \text{RIDE } m \ n) \circ \\ (\text{cc} (\text{EVERY CYCLIST})_n. (\text{A BIKE})_m. \text{OWN } m \ n) : \mathbf{u}$$

To show the entailments it gives rise to, we start by reducing the first utterance of the discourse in (5.55).

$$\vdash (\text{cc} (\text{EVERY CYCLIST})_n. (\text{A BIKE})_m. \text{OWN } m \ n) \\ = (\text{cc FORALL}_n. (\text{CYCLIST } n) \text{ IMPLIES EXISTS}_m. (\text{BIKE } m) \text{ AND } (\text{OWN } m \ n)) \\ \equiv \lambda_{c\mathbf{x}|c|}. (c \ \mathbf{x}) \text{ and not exists}_y. (\text{cyclist } y) \text{ and not exists}_z. (\text{bike } z) \text{ and } (\text{own } z \ y) : \mathbf{u}$$

Importantly, the content representing the first utterance yields a context that, under any suitable input vector, entails that if some entity is a cyclist there is a bike that cyclist owns.

The second utterance then picks up the discourse referent that is made weakly familiar by the first utterance, as demonstrated below.

$$\vdash (\text{A CYCLIST})_n. (\text{THE} (\text{BIKE THAT } \lambda_m. \text{HE}_n. \text{OWN } m \ n))_m. \text{RIDE } m \ n \\ \equiv \lambda_c. \lambda_{\mathbf{x}|c|, y}. (\text{cyclist } y) \text{ and } (\text{male } \mathbf{x}, y_{(\text{PRO MALE } c')}) \text{ and} \\ (\text{ride } \mathbf{x}, y_{(\text{the} (\text{BIKE THAT } \lambda_m. \text{HE}_n. \text{OWN } m \ n) \ c') \ y}) \mid \\ \lambda_{\mathbf{x}|c|, y}. (\text{exists!}_{n:\omega|c'|} c' \text{ k-cons } (\text{MALE } n)) \text{ and} \\ \text{exists!}_{m:\omega|c'|}. c' \text{ k-entails } ((\text{BIKE } m) \text{ AND } (\text{OWN } m \ (\text{pro MALE } c')))) : \mathbf{k}_1$$

This term is passed the context updated by the first utterance, and so the intermediate context updated by A CYCLIST is

$$\begin{aligned} c' \equiv & \lambda_{x|c|,y}.(c \ x) \text{ and} \\ & (\text{not exists}_{z}.\text{(cyclist } z) \text{ and not exists}_{w}.\text{(bike } w) \text{ and (own } w \ z)) \text{ and} \\ & (\text{cyclist } y) : c_{|c|+1} , \end{aligned}$$

where c is the original context passed to the first utterance.

To see that the existential implicature associated with the first utterance of (5.55) licenses the felicitous use of *the bike he owns* in this case, consider the dynamic meaning of (5.55) applied to the empty context τ .

$$\begin{aligned} & \vdash (((cc (A \text{ CYCLIST})_n.(\text{THE (BIKE THAT } \lambda_m.\text{HE}_n.\text{OWN } m \ n))_m.\text{RIDE } m \ n)) \circ \\ & \quad (cc (\text{EVERY CYCLIST})_n.(A \text{ BIKE})_m.\text{OWN } m \ n)) \ \tau) \\ & \equiv \lambda_{x,y}.\text{(not exists}_{z}.\text{(cyclist } z) \text{ and} \\ & \quad \text{not exists}_{w}.\text{(bike } w) \text{ and (own } w \ z)) \text{ and} \\ & \quad (\text{cyclist } x) \text{ and (bike } y) \text{ and (male } x) \text{ and (own } y \ x) \text{ and (ride } y \ x) \text{ and} \\ & \quad (\text{exists!}_{n:\omega_2} c' \text{ k-cons (MALE } n)) \text{ and} \\ & \quad \text{exists!}_{m:\omega_2}.c' \text{ k-entails } ((\text{BIKE } m) \text{ AND OWN } m \text{ (pro MALE } c')) : c_2 \end{aligned}$$

In this case, the intermediate context

$$\begin{aligned} c' \equiv & \lambda_y.\text{(not exists}_{z}.\text{(cyclist } z) \text{ and not exists}_{w}.\text{(bike } w) \text{ and (own } w \ z)) \text{ and} \\ & (\text{cyclist } y) : c_1 . \end{aligned}$$

is exactly the same as the one above, except that it has been instantiated by the application to τ .

With the dynamic meaning of (5.55) expanded out in this way, it is not hard to see why the DyCG analysis of (5.55) is felicitous. The weakly familiar model of *the bike that he owns* is felicitous in a context containing the first utterance of (5.55) because its associated implicature, namely that there is some bike discourse referent that is owned by the cyclist, is contextually entailed. Note, however, that the same would not be true without weak familiarity, because without this mechanism, the input context for THE would not contain a unique bike discourse referent—one is only weakly familiar, entailed to exist by the use of CYCLIST in the context updated by the first utterance.

As implemented here, weak familiarity allows a variant of (5.47) in which *the bike he owns* is replaced by the pronoun *it*, which is much less rich in terms of its descriptive content, requiring only that its antecedent is not entailed to lack the property of nonhumanness.

(5.47') Every cyclist owns a bike. ? One cyclist rides it every day.

Examples like (5.47') are related to the 'marbles' examples in (5.56), and to example (5.57).

- (5.56) a. I dropped ten marbles and found all of them, except for one_{*i*}.
It_{*i*} is probably under the sofa.
b. ? I dropped ten marbles and found only nine of them. It is
probably under the sofa.

(Heim, 1982, example 21, attributed to Barbara Partee)

(5.57) Not every donkey brays. ? It's brown.

(Carl Pollard, personal communication)

I adopt the strategy of claiming that what makes cases like (5.47')–(5.57), different from the 'bathroom' example in (5.46) is the question of maximal salience, one of Roberts's (2005) requirements for (weak) familiarity of pronouns. For (5.56), the variant in (5.56a) is judged to be preferable to the one in (5.56b) in part because the antecedent to the pronoun *It* has been made salient in (5.56a). As for the ability of *it* in the 'bathroom' example (5.46), above, I argue that the task of modeling how the pronoun finds its antecedent is properly part of the pragmatics of salience, and outside the dynamic semantic theory. And so, as it stands, the weak familiarity extensions to DyCG's account of anaphora would allow all of (5.47')–(5.57) using IT.

Importantly, though, discourse referents that are legitimately inaccessible are still treated as such. DyCG predicts infelicity for all of the following examples, which are variants of (5.41), (5.43), and (5.44), respectively.

- (5.58) # If Lance owns every bike_{*i*}, he rides it_{*i*}.
(5.59) # No cyclist_{*i*} arrives. Her_{*i*} name is 'Kim.'
(5.60) # A cyclist does not own a bike_{*i*}. It_{*i*}'s red.

Starting with (5.58), its infelicity can be tested as follows:

$$\vdash \neg (((\text{LANCE}_n \cdot (\text{EVERY BIKE})_m \cdot \text{OWN } m \ n) \text{ IMPLIES HE}_n \cdot \text{IT}_m \cdot \text{RIDE } m \ n) \text{ felicitous-in } \mathbf{T})$$

In this case, the infelicity arises because

$$\begin{aligned} & \vdash \text{LANCE}_n.(\text{EVERY BIKE})_m.\text{OWN } m \ n \\ & \equiv \lambda_c.\lambda_{x|c|}.\text{not exists}_y.(\text{bike } y) \text{ and not } (\text{own } y \ x_{(\text{the NAMED-LANCE } c)}) \mid \\ & \quad \lambda_{x|c|}.\text{exists!}_{n;\omega|c|}.c \text{ k-entails } (\text{NAMED-LANCE } n) : k \end{aligned}$$

does not give rise to a nonhuman discourse referent. And weak familiarity cannot come to the rescue, since there is no nonhuman entity that is even entailed to exist.

For (5.59), we have the following, since (NO CYCLIST ARRIVE) clearly does not introduce any discourse referents, entailed or overt:

$$\vdash \neg ((\text{SHE NAMED-KIM}) \text{ felicitous-in } (\text{cc } (\text{NO CYCLIST ARRIVE}) \text{ T}))$$

Here $\text{NAMED-KIM} =_{\text{def}} (\text{dyn}_1 \text{ named-kim})$. And similarly, for the intended reading of (5.60), in which the negation outscopes the indefinite *a bike*, both the strongly familiar and weakly familiar versions of *it* are infelicitous (here, $\text{RED} =_{\text{def}} (\text{dyn}_1 \text{ red})$).

$$\vdash \neg ((\text{IT RED}) \text{ felicitous-in } (\text{cc } (\text{A CYCLIST})_n.\text{NOT } (\text{A BIKE})_m.\text{OWN } m \ n) \text{ T}))$$

The infelicity in this case occurs for similar reasons as for (5.58) and (5.59): the negation limits the accessibility of the discourse referents introduced in its scope, and

$$\begin{aligned} & \vdash \text{cc } ((\text{A CYCLIST})_n.\text{NOT } (\text{A BIKE})_m.\text{OWN } m \ n) \text{ T} \\ & \equiv \lambda_y.(\text{cyclist } y) \text{ and not exists}_z.(\text{bike } z) \text{ and } (\text{own } z \ y) : c_1 \end{aligned}$$

does not introduce a discourse referent compatible with being nonhuman, nor does it entail the existence of any such entity. Of course, the pronoun *it* could be felicitously used if the first utterance of (5.60) were interpreted with the negation having narrowest scope. This reading would be paraphrasable by *There is a bike that a cyclist does not own, and it is red*.

So implementing weak familiarity in DyCG has the effect of repairing the accessibility relation that is common to all dynamic theories so that it no longer undergenerates, at the price of leaving some examples, such as (5.56), to be ruled in or out based on the pragmatic process of determining maximal salience. I hypothesize that the pragmatic processes governing

salience and weak familiarity work in concert. Another positive aspect of this approach is that the remaining cases of true inaccessibility are maintained, as desired.

5.4 Determiner Strength

I turn now to a proposal for handling the problem of *asymmetric quantification* discussed by Heim (1982), Partee (1984), Rooth (1987), and Kanazawa (1994), among others. Before laying out the DyCG formalization in section 5.4.4, I first discuss the previous proposals due to Kanazawa (1994, in §5.4.2) and Chierchia (1992, 1995, in §5.4.3). I argue below that while each of these authors treat essential elements of the problem correctly, both accounts are flawed in different ways. But first, in §5.4.1, I provide an overview of certain aspects of the asymmetry problem.

5.4.1 Strong and Weak Readings of Determiners

Ignoring generic readings, (5.61b) has an interpretation that is lacking in (5.61a).

- (5.61) a. A cyclist that owns a bike_{*i*} rides it_{*i*}.
 b. Every cyclist that owns a bike_{*i*} rides it_{*i*}.

The ambiguity for (5.61b) centers around the following question: does each cyclist ride *every* bike she owns, or only some of them? Early dynamic theories, such as Kamp's (1981) DRT and Heim's (1982) FCS, only predict the *strong* reading for (5.61b), the one in which each cyclist rides every bike she owns. The *weak* reading, in which cyclists may ride some of their bikes but not all of them, is not available. This is due to the fact that both DRT and FCS adopt the *unselective binding* approach due to Lewis (1975), so that quantificational determiners like *Every* in (5.61b) apply the same quantificational force to all of the free variables in their scope.

This problem is sometimes called the *asymmetric quantification* problem, because the weak reading seems to require that the discourse referent introduced by the head noun *cyclist* in the (5.61) examples should be treated differently from the one introduced by *a bike*. For approaches based on unselective binding, the strong reading is forced because this asymmetry is not respected: unselective binding amounts to quantification over pairs, and so (5.61b) is predicted to be true if and only if, for every

pair $\langle x, y \rangle$ where x is a cyclist and y a bike that x owns, x rides y . When the determiner in question is an indefinite (*a, some, one*) or a negated indefinite (*no*), the pair-quantification reading coincides with the weak reading. But quantification over pairs is insufficiently granular for *every*, and does not work at all for the determiners *most, at least two*, etc.

We might choose to overlook the gaps in empirical coverage associated with unselective binding theories. However, sometimes the weak reading is preferred or the only one available, even for *every*:

(5.62) Every man that had a quarter_{*i*} put it_{*i*} in the parking meter.
(Schubert and Pelletier, 1989)

(5.63) Most cyclists that own a bike_{*i*} ride it_{*i*}.

In (5.62), the strong reading is not available in the usual case because interlocutors generally know that paying for parking does not require each man to empty his pockets into the meter. The usual story, due originally to Rooth (1987, page 254), says about (5.63) that it is false if there are exactly 100 cyclists, 99 of them have only one bike but do not ride it, while the other has 1,000 bikes and rides them all. That is, most cyclists do not ride any of their bikes; it does not matter that out of the 1,099 pairs of a cyclist with a bike she owns, 1,000 involve bike riding. However, unselective binding, which reduces to pair quantification for donkey sentences, predicts that (5.63) is true in this situation.

Explaining this asymmetry is known as the *proportion* or *cardinality* problem, or alternatively, the problem of *farmer/donkey asymmetry*. Two notions that bear on the proportion problem are what I will call the conditions of *distributivity* and *uniqueness*. Distributivity arises in contexts where the asymmetry described above is not present. For example, (5.61b) is considered distributive in any context in which cyclists treat all of their bikes the same way. The strong and weak readings both coincide with intuitively correct truth conditions when distributivity is present, but otherwise only the weak reading is consistent with intuitions, as (5.62) demonstrates. The condition of uniqueness can be seen as a special case of distributivity, because if every cyclist owns only a single bike, then necessarily cyclists must treat all of their bikes the same. However, uniqueness is relatively rare, in fact, it does not apply for the default interpretations of any of (5.61)–(5.63). The notions of distributivity and uniqueness are centrally relevant to two prominent accounts of strong and weak readings of determiners, as I discuss in §5.4.2 and §5.4.3.

5.4.2 Kanazawa's Tonicity-Based Approach

For Kanazawa (1994), the interpretation of a donkey sentence is given by either the weak or the strong reading, with the choice of determiner the main factor that affects which readings are possible. More specifically, the key factor is the *tonicity* of the determiner. Because it is so central to Kanazawa's account, I first recall the notion of tonicity.

Definition 5.9 (Tonicity). If $\langle A, \sqsubseteq \rangle$ and $\langle B, \leq \rangle$ are two preordered sets, then $f : A \rightarrow B$ is called *monotonic* (respectively, *antitonic*) if and only if for all $a, a' \in A$, if $a \sqsubseteq a'$ then $f(a) \leq f(a')$ (resp. $f(a') \leq f(a)$). The function f is called *tonic* if it is either monotonic or antitonic, and *atonic* otherwise.

If $f : A \rightarrow A \rightarrow B$ is a (curried) function, it is called *monotonic* (resp. *antitonic*) in its first (resp. second) argument iff, for each $a \in A$, the function $\lambda x. (f x a)$ (resp. $\lambda x. (f a x)$) is monotonic (resp. antitonic).

The pertinent case for determiners is the one in which B is the set of static propositions, \leq is the entailment relation entails : $p \rightarrow p \rightarrow t$ from definition 3.7, A is the set of static properties, and

$$\sqsubseteq : (e \rightarrow p) \rightarrow (e \rightarrow p) \rightarrow t$$

is defined as the relation between static properties

$$\sqsubseteq =_{\text{def}} \lambda_{PQ} \forall x. (P x) \leq (Q x) .$$

In the linguistic semantics literature, a monotonic determiner is sometimes called *upward monotonic* or *monotone increasing*, while an antitonic determiner is often referred to as *downward monotonic* or *monotone decreasing*. The terms *left* and *right* are sometimes used to refer to the first and second arguments of a two-argument function. For example, in the semantics literature, a determiner that is monotonic in its second argument is often called *right monotone increasing*.

I use the following shorthands to refer to the tonicity properties of determiners:

$\uparrow\uparrow$ monotonic in both arguments (*a, some, several, many, at least n*),

$\uparrow\downarrow$ monotonic in the first argument and antitonic in the second (*not every, not all*),

$\downarrow\uparrow$ antitonic in the first argument and monotonic in the second (*every, all*),

$\downarrow\downarrow$ antitonic in both arguments (*no, few, at most n*), and

$\Downarrow\uparrow$ atonic in the first argument and monotonic in the second (*most*).

The last case ($\Downarrow\uparrow$) can be demonstrated by the entailments present in

- (5.64) a. Most donkeys bray.
 b. Most donkeys bray and snort.
 c. Most brown donkeys bray.

Here neither of (5.64a) or (5.64c) entails the other, and therefore *most* is atonic in its first argument. However, (5.64b) entails (5.64a). This means that *most* is monotonic in its second argument, since we have

$$\vdash (\text{bray that snort}) \sqsubseteq \text{bray} ,$$

where $\text{bray} : e \rightarrow p$ and $\text{snort} : e \rightarrow p$ are the properties of braying and snorting, respectively.

Kanazawa correctly rejects the pair quantification reading for donkey sentences associated with theories that employ a version of the unselective binding approach. He also rejects the E-Type treatment of donkey anaphora advocated by Kadmon (1990) on the grounds that E-Type pronouns require uniqueness, and so they fail to capture the intuitive truth conditions associated with either the strong or weak readings of donkey sentences in almost all situations.

In Kanazawa's account, the interpretation of a donkey sentence is given by either the strong reading or the weak one, with the tonicity of the chosen determiner being the central deciding factor. Under this view, only the weak reading is available for determiners that have the same tonicity in both arguments (either $\uparrow\uparrow$ or $\downarrow\downarrow$). For determiners with mixed tonicity, the strong reading is supposedly preferred for those that are $\downarrow\uparrow$. This preference for the strong reading extends to the case of $\uparrow\downarrow$ determiners, according to Kanazawa, but the preference is diminished relative to that for the $\downarrow\uparrow$ determiners. For $\Downarrow\uparrow$ determiners, both readings are available.

5.4.3 Chierchia's Dynamic/E-Type Account

Chierchia (1992, 1995) proposes a scheme for modifying the dynamic meanings of determiners that is inspired by the *conservativity* property

exhibited by static determiners (Barwise and Cooper, 1981; Keenan and Stavi, 1986).

Definition 5.10 (Conservativity). A generalized determiner $d : (e \rightarrow p) \rightarrow (e \rightarrow p) \rightarrow p$ is *conservative* if and only if, for all properties P and Q , we have

$$\vdash (d P Q) \equiv (d P (P \text{ that } Q)) ,$$

where the property conjunction that is as defined in (3.67).

As an example of the conservativity property of static determiners, note that we have

$$\begin{aligned} &\vdash (\text{a donkey bray}) \\ &= \text{exists}_x.(\text{donkey } x) \text{ and } (\text{bray } x) \\ &\equiv \text{exists}_x.(\text{donkey } x) \text{ and } (\text{donkey } x) \text{ and } (\text{bray } x) \\ &\equiv (\text{a donkey } (\text{donkey that bray})) \end{aligned}$$

and

$$\begin{aligned} &\vdash (\text{every donkey bray}) \\ &= \text{forall}_x.(\text{donkey } x) \text{ implies } (\text{bray } x) \\ &\equiv \text{forall}_x.(\text{donkey } x) \text{ implies } ((\text{donkey } x) \text{ and } (\text{bray } x)) \\ &\equiv (\text{every donkey } (\text{donkey that bray})) , \end{aligned}$$

where *donkey* and *bray* are static properties, and *a* and *every* are as defined in equations (3.35) and (3.49). The determiners *a* and *every* are therefore conservative by definition 5.10.

The essence of the idea is to transform a dynamic determiner into one that gives only the weak readings by forcing the restrictor property to be ‘copied’ into the scope. For example, Chierchia’s approach aims to force the dynamic meaning of (5.61b) to be equivalent to the one for

(5.65) Every cyclist that owns a bike_{*i*} is a cyclist that owns a bike_{*i*} and rides it_{*i*}.

Thus according to Chierchia, the weak reading is the default reading for every determiner.

A problem for Chierchia’s approach is that, if determiners are always dynamically conservative, it is unclear what to say about how the strong

readings arise. To yield the strong readings, Chierchia (1995, page 117) proposes to make pronouns ambiguous between “c-command” binding conditioned by syntax, “dynamic” binding, and E-Type stand-ins for definite descriptions. The E-Type alternative yields the strong reading for (5.61a) and (5.61b) in cases where the uniqueness condition applies, because it requires that the farmer in question owns only one donkey.

Noting the rareness of the uniqueness condition, Chierchia (1992, page 160) posits a fourth ambiguity for pronouns in which the contextually available function corresponding to an E-Type donkey pronoun is actually a choice function, and then stipulates that for (5.61b) to be true every possible choice function must select a bike that the cyclist in question rides.

5.4.4 A Synthesized Dynamic Proposal

I agree with both Kanazawa (1994) and Chierchia (1992, 1995) that the pair-quantification readings yielded by unselective binding approaches are incorrect except in a very limited number of cases. And I also accept Kanazawa’s arguments for ignoring the E-Type readings. I agree with him that for *most*, both weak and strong readings are available. But I reject his claim that the strong reading is preferred for *every*. I would argue that the strong reading for *every* and *most* is favored in cases where there is good reason to assume the distributivity condition is satisfied.

Consider (5.66) as an example.

- (5.66) a. Everyone who had a son_{*i*} was concerned about his_{*i*} grades.
 b. Everyone who had a credit card_{*i*} used it_{*i*} to pay for dinner.

For (5.66a), an interlocutor has no reason to think that someone with more than one son would treat any of them differently, so the strong reading arises, that is, that each person with sons is concerned about each son’s grades. But the pragmatics of situations where (5.66b) might be uttered are different: we know that, in general, dinners are purchased using a single credit card transaction, and so we get the (merely) weak reading. As further evidence that the effect is pragmatic rather than being tied to *every*’s tonicity, it is not too difficult to imagine a scenario where a credit card issuer offered a prize for the person who managed to make transactions on the largest number of distinct credit cards during a given period. In such a scenario, the usual assumptions about the lack of distributivity with

respect to diners and credit cards do not apply, and so, in this scenario, we are left with the strong reading for (5.66b).

And so, I conclude, as do Chierchia (1995) and Roberts (2005), that the only reading generated by the grammar should be the weak reading. But unlike Chierchia, I maintain that apparent strong readings arise via pragmatic inference based on whether the distributivity property is assumed (either explicitly or implicitly). And while I generally agree with the approach of harnessing a dynamicized conservativity to get weak readings, I take issue with the four-way ambiguity for pronouns in Chierchia's proposal. Chierchia argues against a strong/weak ambiguity for determiners on the basis of the fact that there seems to be no language that has separate morphemes corresponding to the two readings. But as far as I can tell, this argument applies equally well against his strategy of positing a four-way ambiguity for pronouns. Chierchia claims that his theory contains no such ambiguity, but Roberts rejoins convincingly that it does: for Chierchia, binding using E-Type pronouns uses variables over entities, while other kinds of binding involve variables over discourse referents.

The proposal for handling the asymmetry problem that I advocate here adopts and rejects a piece of both of Chierchia's and Kanazawa's stories. For Chierchia, I adopt dynamic conservativity but reject his four-way ambiguity that marshals E-Type pronouns in order to force strong readings. In this DyCG semantics, all pronouns are what Chierchia calls the "dynamically bound" case. As for Kanazawa, I adopt the approach of excluding E-Type donkey pronouns from the theory, but reject the idea that a determiner's monotonicity is what decides which of the weak or strong readings applies. In view of these factors, I choose to implement a weak-reading-only analysis of donkey sentences within DyCG, described in detail below.

5.4.4.1 *Weakening Dynamic Determiners*

Although they involve cyclists and bikes rather than farmers and donkeys, the weak reading is available for each of the following 'donkey sentences.'

$$(5.67) \quad \left\{ \begin{array}{l} \text{A/One/Some} \\ \text{Several} \\ \text{At least/most } n \\ \text{Many/Few} \\ \text{(Not) Every} \\ \text{(Not) All} \\ \text{No} \\ \text{Most} \end{array} \right\} \text{cyclist(s) that own(s) a bike}_i \text{ ride(s) it}_i.$$

Observing this, we need a general way to map each of the dynamic generalized determiner definitions given so far for DyCG to its corresponding weakened variant. An initial attempt might be to directly imitate static conservativity as defined in definition 5.10, defining an operator

$$(5.68) \quad \text{weak-determiner} : (d_1 \rightarrow d_1 \rightarrow k) \rightarrow d_1 \rightarrow d_1 \rightarrow k$$

as follows:

$$\text{weak-determiner} =_{\text{def}} \lambda_{dDE}.d D (D \text{ THAT } E)$$

This definition would indeed yield the weak readings, but also gives rise to the problem of ‘donkey doubling,’ since any discourse referents introduced within D are introduced twice. To exemplify this problem, consider weak-determiner, as defined, applied to the dynamic determiner A :

$$\begin{aligned} &\vdash (\text{weak-determiner } A) \\ &= \lambda_{DE}.A D (D \text{ THAT } E) \\ &= \lambda_{DE}.\text{EXISTS } (D \text{ THAT } (D \text{ THAT } E)) \end{aligned}$$

And so using (weak-determiner A) to model the variant of (5.67) with an indefinite, we get

$$\begin{aligned} &\vdash (\text{weak-determiner } A) (\text{CYCLIST THAT } \lambda_n.(\text{A BIKE})_m.\text{OWN } m n) \\ &\quad \lambda_n.(\text{IT}_m.\text{RIDE } m n) \\ &= \text{EXISTS}_n.((\text{CYCLIST } n) \text{ AND EXISTS}_m.(\text{BIKE } m) \text{ AND } (\text{OWN } m n)) \text{ AND} \\ &\quad ((\text{CYCLIST } n) \text{ AND EXISTS}_m.(\text{BIKE } m) \text{ AND } (\text{OWN } m n) \text{ AND} \\ &\quad \text{IT}_m.\text{RIDE } m n) : k. \end{aligned}$$

Notice that two discourse referents are available to serve as the antecedent to *IT*, and therefore its associated implicature means that using *IT* here is infelicitous. And yet, intuitively, this is not an instance of weak familiarity but rather of strong familiarity: the pronoun *it* in (5.67), on the intended reading, is clearly anteceded by the discourse referent introduced by *bike* in the restrictor.

A better attempt at defining weak-determiner makes sure that the situation illustrated above does not arise by limiting the accessibility of discourse referents introduced in the restrictor, but not fundamentally modifying its truth conditions. This can be accomplished by simply applying dynamic double negation to the restrictor, as in the following definition.

$$(5.69) \quad \text{weak-determiner} =_{\text{def}} \lambda_{dDE}.d \text{ (NON (NON } D)) (D \text{ THAT } E)$$

(Here the dynamic property negation **NON** is as defined in (4.25).) This definition is essentially the same as the preliminary one given above, having the same type as in (5.68), with the exception that no discourse referents can be doubly introduced.

To see how the definition of weak-determiner in (5.69) is different from the preliminary one that uses no negation, consider again the result of using it to model the indefinite variant of (5.67), recalling theorem 5.2.

$$\begin{aligned} & \vdash (\text{weak-determiner } A) (\text{CYCLIST THAT } \lambda_n. (A \text{ BIKE})_m. \text{OWN } m \ n) \\ & \quad \lambda_n. (\text{IT}_m. \text{RIDE } m \ n) \\ &= \text{EXISTS}_n. (\text{NOT NOT } ((\text{CYCLIST } n) \text{ AND } (A \text{ BIKE})_m. \text{OWN } m \ n)) \text{ AND} \\ & \quad (\text{CYCLIST } n) \text{ AND EXISTS}_m. (\text{BIKE } m) \text{ AND } (\text{OWN } m \ n) \text{ AND} \\ & \quad \text{IT}_m. \text{RIDE } m \ n \\ &\equiv \lambda_c. \lambda_{x|c|, y, z}. (\text{exists}_w. (\text{cyclist } y) \text{ and } (\text{bike } w) \text{ and } (\text{own } w \ y)) \text{ and} \\ & \quad (\text{cyclist } y) \text{ and } (\text{bike } z) \text{ and } (\text{own } z \ y) \text{ and} \\ & \quad (\text{nonhuman } x, y, z_{(\text{pro NONHUMAN } c')}) \text{ and } (\text{ride } x, y, z_{(\text{pro NONHUMAN } c')}) \mid \\ & \quad \lambda_{x|c|, y, z}. \text{exists!}_{n: \omega|c'|}. c' \text{ k-cons (NONHUMAN } n) : k_2 \end{aligned}$$

Here, the context passed to $(IT_m.RIDE\ m\ n)$ is

$$c' = \lambda_{\mathbf{x}|c|,y,z}.(c\ \mathbf{x}) \text{ and } (\text{exists}_w(\text{cyclist } y) \text{ and } (\text{bike } w) \text{ and } (\text{own } w\ y)) \text{ and } \\ (\text{cyclist } y) \text{ and } (\text{bike } z) \text{ and } (\text{own } z\ y) : c_2,$$

and so the implicature proposition associated with the strongly familiar IT is consistent, because the discourse referents introduced in the restrictor are existentially bound. This means that the only remaining discourse referent that is not entailed to be nonhuman is the one introduced in the scope.

Applying weak-determiner to a version of (5.67) where the determiner is *every* shows how both the strong and weak readings can arise.

$$\begin{aligned} & \vdash (\text{weak-determiner EVERY}) (\text{CYCLIST THAT } \lambda_n.(\text{A BIKE})_m.\text{OWN } m\ n) \\ & \quad \lambda_n.IT_m.RIDE\ m\ n \\ & = \text{FORALL}_n.((\text{NOT } (\text{NOT } ((\text{CYCLIST } n) \text{ AND } \\ & \quad \text{EXISTS}_m.(\text{BIKE } m) \text{ AND } (\text{OWN } m\ n)))) \text{ IMPLIES } \\ & \quad ((\text{CYCLIST } n) \text{ AND } \text{EXISTS}_m.(\text{BIKE } m) \text{ AND } (\text{OWN } m\ n) \text{ AND } \\ & \quad IT_m.RIDE\ m\ n)) \\ & = \text{NOT EXISTS}_n.\text{NOT NOT } ((\text{NOT NOT } ((\text{CYCLIST } n) \text{ AND } \\ & \quad \text{EXISTS}_m.(\text{BIKE } m) \text{ AND } (\text{OWN } m\ n)))) \text{ AND } \\ & \quad \text{NOT } ((\text{CYCLIST } n) \text{ AND } \text{EXISTS}_m.(\text{BIKE } m) \text{ AND } (\text{OWN } m\ n) \text{ AND } \\ & \quad IT_m.RIDE\ m\ n)) \\ & \equiv \lambda_c.\lambda_{\mathbf{x}|c|}.\text{not exists}_y.(\text{exists}_z((\text{cyclist } y) \text{ and } (\text{bike } z) \text{ and } (\text{own } z\ y))) \text{ and } \\ & \quad \text{not exists}_w.(\text{cyclist } y) \text{ and } (\text{bike } w) \text{ and } (\text{own } w\ y) \text{ and } \\ & \quad (\text{nonhuman } \mathbf{x}, y, w_{(\text{pro NONHUMAN } c')}) \text{ and } (\text{ride } \mathbf{x}, y, w_{(\text{pro NONHUMAN } c')} y) \mid \\ & \quad \lambda_{\mathbf{x}|c|}.\text{exists}_{n:\omega|c'|} .c' \text{ k-cons } (\text{NONHUMAN } n) \end{aligned}$$

In this case, the context passed to $IT_m.RIDE\ m\ n$ by the enclosing dynamic existential is

$$c' \equiv \lambda_{\mathbf{x}|c|,y,w}.(c\ \mathbf{x}) \text{ and } (\text{cyclist } y) \text{ and } (\text{bike } w) \text{ and } (\text{own } w\ y),$$

so that IT is able to select the unique discourse referent v that is consistent with being nonhuman. The truth conditions associated with this weakened version of *every* are equivalent to *Every cyclist that owns a bike is a cyclist that*

owns a bike and rides it. In case the distributivity property does not hold, and cyclists may treat some of their bikes differently than others, the bike that the cyclist is said to own by the restrictor may be a different one than the cyclist rides. But when distributivity is present, these truth conditions invite the inference that whichever bike is selected as the one that is owned by the cyclist will also be ridden by her.

The dynamic determiners are then redefined based on their earlier definitions, using weak-determiner, as follows.

$$(5.70) \quad A_{\text{weak}} =_{\text{def}} (\text{weak-determiner } A)$$

$$(5.71) \quad \text{NO}_{\text{weak}} =_{\text{def}} (\text{weak-determiner NO})$$

$$(5.72) \quad \text{EVERY}_{\text{weak}} =_{\text{def}} (\text{weak-determiner EVERY})$$

Notice that for certain determiners, applying weak-determiner has no practical effect. For example:

$$\begin{aligned} & \vdash (A_{\text{weak}} \text{ CYCLIST RIDE}) \\ & = A (\text{NON NON CYCLIST}) (\text{CYCLIST THAT RIDE}) \\ & = \text{EXISTS}_n. (\text{NOT NOT } (\text{CYCLIST } n)) \text{ AND } ((\text{CYCLIST } n) \text{ AND } (\text{RIDE } n)) \\ & \equiv \lambda_c. \lambda_{x|c|, y}. (\text{cyclist } y) \text{ and } (\text{cyclist } y) \text{ and } (\text{ride } y) \mid T_{|c^+|} \\ & \equiv \lambda_c. \lambda_{x|c|, y}. (\text{cyclist } y) \text{ and } (\text{ride } y) \mid T_{|c^+|} \\ & \equiv (A \text{ CYCLIST RIDE}) : k \end{aligned}$$

Since, by theorem 5.2, $\lambda_n. (\text{NOT NOT } (\text{CYCLIST } n))$ is truth-conditionally equivalent to the dynamic property CYCLIST, the weak variant A_{weak} is equivalent to its original definition A. Slightly more complicated cases, in which a discourse referent is introduced in the restrictor, are also equivalent with respect to the weak/strong reading distinction, as the following example shows.

$$\begin{aligned} & \vdash (A_{\text{weak}} (\text{CYCLIST THAT } \lambda_n. (A \text{ BIKE})_m. \text{OWN } m \text{ } n) \text{ RIDE}) \\ & = \text{EXISTS}_n. (\text{NOT NOT } ((\text{CYCLIST } n) \text{ AND } \text{EXISTS}_m. (\text{BIKE } m) \text{ AND } (\text{OWN } m \text{ } n))) \\ & \quad \text{AND } (\text{CYCLIST } n) \text{ AND } \text{EXISTS}_m. (\text{BIKE } m) \text{ AND } (\text{OWN } m \text{ } n) \text{ AND } (\text{RIDE } n) \\ & \equiv \lambda_c. \lambda_{x|c|, y, z}. (\text{exists}_w (\text{cyclist } y) \text{ and } (\text{bike } w) \text{ and } (\text{own } w \text{ } y)) \text{ and} \\ & \quad (\text{cyclist } y) \text{ and } (\text{bike } z) \text{ and } (\text{own } z \text{ } y) \text{ and } (\text{ride } y) \mid T_{|c^{++}|} : k_2 \end{aligned}$$

This pseudo-equivalence holds for weak-determiner applied to any of the $\uparrow\uparrow$ or $\downarrow\downarrow$ determiners, as well as the generalized definitizers **THE** and **PRO** from equations (5.9) and (5.10).

In order to model the dynamic meaning of the determiner *most*, some discussion of its static meaning is in order. I will follow standard practice in making the simplifying assumption that *most* is roughly equivalent to *more than half*. Then, thinking of the extensions of static properties as (characteristic functions of) sets of entities, note that for P and Q static properties, the truth conditions associated with $(\text{most } P \text{ } Q)$ should be that the cardinality of $(P \text{ that } Q)$ is greater than the cardinality of $(P \text{ that non } Q)$.

Accordingly, I define the cardinality function $\text{card}_A : (A \rightarrow \mathbf{t}) \rightarrow \mathbf{n}$ for each set s as the number of entities in s , defined as follows, where s is characterized by an A -predicate for some type A .

$$(5.73) \quad \text{card}_A =_{\text{def}} \lambda_{s:A \rightarrow \mathbf{t}} \lambda_{n:\mathbf{n}} \exists f:A \rightarrow \omega_n. \\ (\forall x:A \forall y:A ((s \ x) \wedge (s \ y)) \Rightarrow ((f \ x) = (f \ y) \Rightarrow x = y)) \wedge \\ \forall i:\omega_n \exists z:A. i = (f \ z)$$

That is, the cardinality of a set s is the natural number equinumerous with s , where equinumerosity is defined as the existence of a bijection between two sets.

I then introduce a new basic meaning type ν of natural number concepts, with

$$\text{Ext}(\nu) = \mathbf{n}.$$

For every meaning type A , the cardinality of properties $A \rightarrow \mathbf{p}$ is then defined in terms of card , in the form of the function

$$\text{kard}_A : (A \rightarrow \mathbf{p}) \rightarrow \nu,$$

axiomatized by

$$(5.74) \quad \vdash \forall P:A \rightarrow \mathbf{p} \forall w:w. (\text{kard}_A P) @ w = \text{card}_A (P @ w).$$

As usual, the type parameter on both card and kard is often dropped.

Then the infix function $\text{exceeds}_{A,B} : (A \rightarrow \mathbf{p}) \rightarrow (B \rightarrow \mathbf{p}) \rightarrow \mathbf{p}$, where A and B are meaning types, simply tests whether a given set is larger than

another, and is subject to the following axiom:

(5.75)

$$\vdash \forall P:A \rightarrow p \forall Q:B \rightarrow p \forall w:w. (P \text{ exceeds } Q) @ w \Leftrightarrow ((\text{kard}_B Q) @ w < (\text{kard}_A P) @ w)$$

The meaning of the static determiner *most* is then defined in terms of *exceeds*, where *non* is as defined in (3.22).

$$(5.76) \quad \text{most} =_{\text{def}} \lambda_{PQ}. (P \text{ that } Q) \text{ exceeds } (P \text{ that non } Q)$$

As an example, the static meaning of *Most donkeys bray* is

$$\begin{aligned} & \vdash \text{most donkey bray} \\ & = (\text{donkey that bray}) \text{ exceeds } (\text{donkey that non bray}) \\ & = (\lambda_x (\text{donkey } x) \text{ and } (\text{bray } x)) \text{ exceeds } \lambda_x. (\text{donkey } x) \text{ and not } (\text{bray } x) : p \end{aligned}$$

As with other dynamic determiners, the dynamic counterpart to *most* is defined by analogy. Toward a preliminary definition of dynamic *most*, we first define the dynamic counterpart to *exceeds*, whose type is

$$\text{EXCEEDS} : d_1 \rightarrow d_1 \rightarrow k,$$

and which is defined as

$$(5.77) \quad \begin{aligned} \text{EXCEEDS} =_{\text{def}} & \lambda_{DEc}. \lambda_{\mathbf{x}[c]}. (\lambda_{\mathbf{y}[\text{EXISTS } D]}. (\text{EXISTS } D \ c)^s \mathbf{x}, \mathbf{y}) \text{ exceeds} \\ & (\lambda_{\mathbf{z}[\text{EXISTS } E]}. (\text{EXISTS } E \ c)^s \mathbf{x}, \mathbf{z}) \mid \\ & \lambda_{\mathbf{x}[c], \mathbf{y}[\text{EXISTS } D], \mathbf{z}[\text{EXISTS } E]}. ((\text{EXISTS } D \ c)^i \mathbf{x}, \mathbf{y}) \text{ and } ((\text{EXISTS } E \ c)^i \mathbf{x}, \mathbf{z}) \end{aligned}$$

Dynamic *EXCEEDS* compares two dynamic properties by λ -binding the discourse referents they introduce, while maintaining their contributions to the implicature. With this definition in place, the dynamic generalized determiner *MOST* could be defined as follows:

$$\text{MOST} =_{\text{def}} \lambda_{DE}. (D \text{ THAT } E) \text{ EXCEEDS } (D \text{ THAT NON } E)$$

Note the similarity between this definition of dynamic *MOST* and its static counterpart in (5.76).

A version of (5.67) with *most* as the determiner could then be accounted for in DyCG. Here, its dynamic meaning is promoted to an update and then

applied to the empty context, in order to resolve the anaphora associated with *IT*. (To clarify the exposition, the anaphoric implication associated with *it* and the pronoun's sortal information are omitted from the following reductions involving *most*.)

$$\begin{aligned}
& \vdash \text{cc} \left(\text{MOST} \left(\text{CYCLIST THAT } \lambda_n.(\text{A BIKE})_m.\text{OWN } m \ n \right) \lambda_n.\text{IT}_m.\text{RIDE } m \ n \right) \text{T} \\
& = \text{cc} \left(\left(\text{CYCLIST THAT } \lambda_n.(\text{A BIKE})_m.\text{OWN } m \ n \right) \text{THAT IT}_m.\text{RIDE } m \right) \\
& \quad \text{EXCEEDS} \left(\left(\text{CYCLIST THAT } \lambda_n.(\text{A BIKE})_m.\text{OWN } m \ n \right) \text{THAT} \right. \\
& \quad \left. \lambda_n.(\text{NOT IT}_m.\text{RIDE } m \ n) \right) \text{T} \\
& \equiv \lambda_{x|c}.\lambda_{y,z}.((\text{cyclist } y) \text{ and } (\text{bike } z) \text{ and } (\text{own } z \ y) \text{ and } (\text{ride } z \ y)) \\
& \quad \text{exceeds } \lambda_{v,w}.(\text{cyclist } v) \text{ and } (\text{bike } w) \text{ and } (\text{own } w \ v) \text{ and not } (\text{ride } w \ v) : c
\end{aligned}$$

One thing to immediately notice about the dynamic meaning given under the current definitions for (5.67) with *most* is that the asymmetry problem is reproduced here, since this is the strong reading: *MOST* is quite literally quantifying over pairs of cyclists and bikes that they own.

To avoid this problem, we redefine the dynamic meaning of *most* as

$$\begin{aligned}
(5.78) \quad \text{MOST} &=_{\text{def}} \lambda_{DE}.(\text{NON NON } (D \text{ THAT } E)) \text{ EXCEEDS} \\
& \quad (\text{NON NON } (D \text{ THAT NON } E)),
\end{aligned}$$

which uses dynamic double negation to ensure that any discourse referents introduced within *D* or *E* are existentially bound without affecting the meaning (theorem 5.2). To see how this redefinition avoids the asymmetry problem, consider how differently it treats a version of (5.67) using *most* as the determiner than the original definition (as before, the dynamic meaning is promoted to an update and applied to the empty context, and

the implications associated with *it* are omitted):

$$\begin{aligned}
& \vdash \text{cc} \left(\text{MOST} \left(\text{CYCLIST THAT } \lambda_n.(\text{A BIKE})_m.\text{OWN } m \ n \right) \lambda_n.\text{IT}_m.\text{RIDE } m \ n \right) \text{T} \\
& = \text{cc} \left(\left(\text{NON NON} \left(\text{CYCLIST THAT } \lambda_n.(\text{A BIKE})_m.\text{OWN } m \ n \text{ THAT } \right. \right. \right. \\
& \quad \left. \left. \left. \lambda_n.\text{IT}_m.\text{RIDE } m \ n \right) \right) \text{EXCEEDS} \left(\text{NON NON} \left(\text{CYCLIST THAT } \right. \right. \right. \\
& \quad \left. \left. \left. \lambda_n.(\text{A BIKE})_m.\text{OWN } m \ n \text{ THAT NON IT}_m.\text{RIDE } m \ n \right) \right) \right) \text{T} \\
& = \text{cc} \left(\left(\lambda_n \left(\text{NOT NOT} \left(\left(\text{CYCLIST } n \right) \text{ AND } \left(\text{A BIKE} \right)_m.\text{OWN } m \ n \right) \text{ AND } \right. \right. \right. \\
& \quad \left. \left. \left. \text{IT}_m.\text{RIDE } m \ n \right) \right) \text{EXCEEDS } \lambda_n. \left(\text{NOT NOT} \left(\left(\text{CYCLIST } n \right) \text{ AND } \right. \right. \right. \\
& \quad \left. \left. \left. \left(\text{A BIKE} \right)_m.\text{OWN } m \ n \right) \text{ AND NOT } \left(\text{IT}_m.\text{RIDE } m \ n \right) \right) \right) \text{T} \\
& \equiv \lambda_{x|c}. \lambda_y.\text{exists}_z. \left(\left(\text{cyclist } y \right) \text{ and } \left(\text{bike } z \right) \text{ and } \left(\text{own } z \ y \right) \text{ and } \left(\text{ride } z \ y \right) \right) \\
& \quad \text{exceeds } \lambda_v.\text{exists}_w. \left(\text{cyclist } v \right) \text{ and } \left(\text{bike } w \right) \text{ and} \\
& \quad \left(\text{own } w \ v \right) \text{ and not } \left(\text{ride } w \ v \right) : c
\end{aligned}$$

This example shows that the definition of **MOST** in (5.78) gives the correct reading, avoiding the asymmetry problem because only the cyclists are being counted, not pairs of cyclists and bikes they own. Thus the DyCG model of (5.67) with *most* yields truth conditions equivalent to *The number of cyclists that own a bike and ride it is larger than the number of cyclists that own a bike but do not ride it*, as desired.

5.5 Summary and Comparison with Other Theories

This chapter demonstrates how DyCG can be extended to a multi-level framework that incorporates insights from both the tradition of dynamic semantic theories that dates to Karttunen (1974, 1976), Lewis (1979), and Heim (1982) with Karttunen and Peters's (1979) approach to separating the sense portion of utterance meaning from its implicatures. The revisions to the dynamic theory presented in chapter 4 are relatively minimal, mostly related to the fact that contents are now considered functions from a context to a pair of contexts rather than to a single context.

The multi-level extensions to DyCG explored in this chapter allow a different take on conventional implicatures than what is usually found in the literature. Potts (2005) advocates a characterization of meaning in which no lexical item contributes both sense and implicature content. But under the view adopted here, in which anaphora is simply an instance of conventional implicature where the implicature must be anchored to the

speaker, we see that Potts's characterization does not apply to the class of conventional implicatures as a whole. This more general view of anaphora also allows a generalization of the usual notion of contextual felicity to one in which infelicity arises because part of the contribution of an utterance is at odds with its context of interpretation.

Note that the novelty condition associated with indefinites, discussed in §2.2.1.1, could be modeled as an implicature in a similar way as the retrievability implication is handled here. But I avoid doing this since the novelty condition is already implicit by the fact that indefinites always introduce an as-yet-unused discourse referent by extending the context's arity. Also, since it is intimately involved with anaphora, I discuss the descriptive content implication in this chapter even though it is actually not obligatorily speaker-anchored like the retrievability implication.

This chapter also lays out an implementation of Roberts's (2003) *weak familiarity*, in the context of providing a more general and more empirically adequate formulation of the accessibility of anaphoric antecedents than is assumed in any other dynamic theory. I show how these ideas about anaphora allow successful accounts not just of the definite anaphora associated with proper names, pronouns, and *the*, but also (both definite and indefinite) possessives and the iterative adverb *too*.

Since the retrievability implication associated with anaphora, under the formal approach given here, is tested only against the input context, the DyCG theory of anaphora qualifies as a *satisfaction* theory, borrowing the terminology of Geurts (1996, 1999). It has the strategy of testing the satisfiability of anaphoric implications against a single context in common with Heim (1982, 1983a,b) and Beaver (2001), among others. Geurts argues stridently that satisfaction theories are hopelessly doomed because there is only ever a single context available to check anaphoric implications against. Instead, he argues for van der Sandt's (1992) approach, in which anaphoric antecedents can be located in (or accommodated in) the *global*, *local*, or any one of several possibly available *intermediate* contexts.

Here, I claim that a single local context suffices. As Beaver (2001, chapter 5) shows, intermediate accommodation sometimes yields strange predictions that are at odds with intuitions. But there is an additional reason DyCG's account of anaphora opts for a single context, namely that the examples given as evidence for the necessity of intermediate accommodation are either instances of weak familiarity or involve lexical

items that are not actually anaphoric: possessive noun phrases, so-called *factives*, and other persistent entailments.

For example, the use of *realize* in the following requires resolution or accommodation according to the [van der Sandt](#) theory:

- (5.79) If Butch is barking, then Mary realizes that Butch is awake.
([Beaver, 2002](#), example 1)

In [van der Sandt](#)'s approach, *realize* presupposes that Butch is awake and therefore a choice is forced as to whether the antecedent should be found in the global context, the intermediate context of the *If ...* clause, or the local context of the *then ...* clause. But no such choice is forced if we instead model *realize* as giving rise to a persistent entailment rather than an anaphoric presupposition.

To take another example, [Bos \(2003\)](#) uses the following to argue that different contexts for accommodation are required.

- (5.80) If Mia_i is married, then her_i husband is out of town.
([Bos, 2003](#), example 7)

Since there is no overt antecedent for *her husband*, a decision must be made about where to accommodate one, according to the [van der Sandt](#) approach. But for DyCG, which implements weak familiarity, no such decision is required because an antecedent for Mia's husband is entailed to exist (at least in legal jurisdictions that allow only opposite-sex marriage).

Lastly, in this chapter I developed a theory of the strength of determiner readings in DyCG that is a blend of two prominent theories on the topic. This hybrid theory, I argue, adopts the desirable characteristics of both while avoiding the pitfalls associated with each. In DyCG, determiners are treated as having the so-called weak reading, and then the strong readings arise only when the condition of *distributivity* holds for the properties in question.

In chapter 6, I show how the two-level extensions to DyCG proposed here impact conventional implicatures other than anaphora, for example, the appositives, nonrestrictive relatives, parentheticals, and expressives discussed by [Potts \(2005\)](#) and others. As I will show, the two-level view of meaning, in which lexical items can contribute sense and implicature content, is the correct generalization for handling both anaphora and the conventional implicatures that are not invariably speaker anchored.

Chapter 6

Variable Conventional Implicatures

In this chapter, I show how the two-level extension to Dynamic Categorical Grammar discussed in chapter 5 applies to conventional implicatures other than just instances of anaphora. Specifically, I give an account of many of the phenomena discussed in Potts 2005: nominal appositives, nonrestrictive relatives, *as*-parentheticals (all *supplements*), and expressives. I show that the strategy of separating sense from implicature yields a theory that has the desirable attributes of both Karttunen and Peters's (1979) theory and the theory of conventional implicatures in Potts 2005, while also capturing anaphora and avoiding many of the considerable problems with Potts's framework.

As a compositional theory that separates sense from implicature content, the approach proposed here has some similarities with recent proposals due to Nouwen (2007), Kubota and Uegaki (2009), Barker, Bernardi, and Shan (2010), AnderBois et al. (2010), and my own work (Kierstead and Martin, 2012). However, the account I offer here departs from both Kubota and Uegaki 2009 and Barker et al. 2010 in that *continuations* are not needed; I also propose looser constraints on the interaction possibilities than do Barker et al., in line with Amaral et al.'s (2007) empirical observations.

The account presented here shares with Nouwen 2007 and AnderBois et al. 2010 an embedding in a dynamic framework, but compares favorably with both of these. In the case of Nouwen's work, this account is more finely grained, for example, allowing supplements to be stacked. AnderBois et al.'s proposal represents a mere sketch of an extension to Dynamic

Montague Grammar (Groenendijk and Stokhof, 1991), but the approach I describe in detail below is far more formally explicit. Perhaps the most attractive aspect of the account explored here is that it shows in detail how anaphora and other conventional implicatures interact, for example, when a definite occurs within a supplement but its antecedent occurs outside it.

After discussing some necessary extensions to DyCG, I show in §6.1 how DyCG can be used to model supplements. The analysis I present in §6.1.1 is finely grained, with supplements in both medial and final positions, as well as stacked supplements (§6.1.1.1). The DyCG account of expressives is laid out in §6.2, and in §6.3, I describe how the DyCG analysis of supplements correctly allows anaphoric links between the sense and implicature levels. Then in §6.4, I show how the generalized notion of felicity from chapter 5 also captures infelicity for both supplements and expressives. I discuss in §6.5 a problem that arises in Karttunen and Peters's theory, the *binding problem*, which prompts Potts to disallow all interaction between sense and implicature, including anaphora. I start in §6.5.1 by exploring how this problem would not manifest itself in DyCG if Karttunen and Peters's analysis were followed. Then §6.5.2 offers an alternative analysis where the binding problem does not arise at all. Unfortunately, the DyCG analysis is not without related problems of its own, which I examine in §6.5.3 and then offer some proposed solutions in §6.5.4. Finally, in §6.6, I give a summary of the DyCG account of variable conventional implicatures.

Although the descriptive content implication associated with definites figures among the variable conventional implicatures (see §2.2.2.1), it is not treated in this chapter for two reasons. The first is that definites are discussed at length in chapter 5, and the second is that giving an account of descriptive content requires a formal model of point of view, which DyCG does not as yet have.

6.1 Supplements

I begin with a very simple example of a nominal appositive in order to clarify the empirical issues associated with supplements as well as the formal approach to modeling them in DyCG. The following example is repeated from chapter 2:

(2.5) Lance, a cyclist, is from Texas.

In (2.5), the proper name *Lance* is called the *anchor* of the appositive *a cyclist*. The correct syntactic and semantic generalizations about the anchor's status are that any syntactic generalized quantifier can serve as the anchor to an appositive, but only certain generalized quantifiers are pragmatically valid, as I discuss below.

Example (2.5) demonstrates why appositives, among other constructions, are called *supplements*. The appositive *a cyclist* is in some sense an additional property that is being attributed to Lance. And as discussed in detail in chapter 2, the appositive's content is different semantically from the rest of the utterance in that it is not targeted by semantic operators. The account of supplements developed here is syntactically similar to the one found in Potts 2005, chapter 4, but is more empirically adequate in terms of its semantics. The reason is that, while sense and implicature content are separated, as for Potts, they are still allowed to interact, which Potts's account strictly disallows.

As for the content of the supplement itself, the distributional facts argue for an analysis in which any predicative construction can be used as a supplement. For example, consider the variants of (2.5) below.

$$(6.1) \quad \text{Lance, } \left\{ \begin{array}{l} \text{a cyclist} \\ \text{no cyclist} \\ \text{pursued by angry fans} \\ \text{fresh from his cheating scandal} \\ \text{tired and discouraged} \\ \text{on a private jet} \\ \text{drinking a sports drink} \end{array} \right\}, \text{ gave an interview.}$$

With these facts in mind, I give a way to turn certain nonpredicative constructions into predicatives, as part of the analysis of supplements below.

6.1.1 Analyzing Supplements

As a preliminary, the DyCG semantics for modeling anaphora in chapter 5 is extended with the following.

Definition 6.1 (Merge). The **merge function** \uparrow , whose type is

$$\uparrow : \Pi_{D:d_{1,i}}.d_{1,i},$$

for some natural number i , is defined as

$$(6.2) \quad \uparrow =_{\text{def}} \lambda_{Dnc} \cdot \mathbf{T}_{|c|+|Dn|} \mid \lambda_{\mathbf{x}^{|c|}, \mathbf{y}^{|Dn|}} \cdot (Dnc)^s \mathbf{x}, \mathbf{y} \text{ and } (Dnc)^i \mathbf{x}, \mathbf{y}.$$

That is, the merge function simply takes a dynamic property and combines its sense and implicature content, yielding a new dynamic property with an empty sense proposition and the combined sense and implicature of the original as its implicature. Note that, as the type of \uparrow shows, any discourse referents introduced in either the sense or implicature contexts of the original property are only introduced into the implicature context of the merged property.

By way of demonstrating how the merge function works, applying it to the dynamic property **CYCLIST** gives

$$\begin{aligned} \vdash (\uparrow \text{CYCLIST}) &= \lambda_{nc} \cdot \mathbf{T}_{|c|} \mid \lambda_{\mathbf{x}^{|c|}} \cdot (\text{cyclist } \mathbf{x}_n) \text{ and true} \\ &\equiv \lambda_{nc} \cdot \mathbf{T}_{|c|} \mid \lambda_{\mathbf{x}^{|c|}} \cdot \text{cyclist } \mathbf{x}_n : d_1 \rightarrow k. \end{aligned}$$

And so the effect of \uparrow is to take a dynamic property and turn it into another dynamic property that only has implicature content.

Based on \uparrow , we can define a dynamic analog of [Potts's \(2005\)](#) account of the comma intonation associated with supplements.

Definition 6.2 (Comma). The **comma function**, representing the comma intonation, has the type

$$\text{COMMA} : d_1 \rightarrow (d_1 \rightarrow k) \rightarrow d_1 \rightarrow k,$$

and is defined as

$$(6.3) \quad \text{COMMA} =_{\text{def}} \lambda_{DQE} \cdot Q((\uparrow D) \text{ THAT } E)$$

As its definition shows, **COMMA** takes a dynamic property D and a dynamic generalized quantifier Q , returning another dynamic generalized quantifier that incorporates D 's content as an implicature. Note that this definition of the dynamic meaning of the comma intonation is somewhat different from the one given in [Kierstead and Martin 2012](#); there, the second property E is passed a context explicitly updated by the first property D , but for the definition in (6.3), the same result is achieved instead by using the dynamic conjunction in **THAT**.

With $\text{CYCLIST} =_{\text{def}} (\text{dyn}_1 \text{cyclist})$ and LANCE as defined in (5.11), for convenience, we define the dynamic property of being from Texas as

$$(6.4) \quad \text{FROM-TEXAS} =_{\text{def}} (\text{dyn}_1 \text{from-texas}),$$

where $\text{from-texas} : e \rightarrow p$ is the static property of being from Texas.

In light of the syntactic evidence in (6.1) that the class of supplements includes predicatives, we need a way to turn a dynamic generalized quantifier into a semantic predicative, that is, a dynamic property. As a preliminary, I define the function $\text{EQUALS} =_{\text{def}} (\text{dyn}_2 \text{equals})$, written infix, where equals is as in definition 5.7.

Definition 6.3 (Predicativizer). The function

$$\text{PRED} : (d_1 \rightarrow k) \rightarrow d_1$$

turns a dynamic generalized quantifier into a dynamic property. It is defined as

$$(6.5) \quad \text{PRED} =_{\text{def}} \lambda_{Qn}.\text{NOT} (\text{NOT} (Q_m.m \text{ EQUALS } n)).$$

As an example, the result of applying the predicativizer function PRED to the dynamic generalized quantifier A CYCLIST is

$$\begin{aligned} & \vdash (\text{PRED A CYCLIST}) \\ &= \lambda_n.\text{NOT} (\text{NOT} ((\text{A CYCLIST})_m.m \text{ EQUALS } n)) \\ &= \lambda_n.\text{NOT} (\text{NOT} (\text{EXISTS}_m.(\text{CYCLIST } m) \text{ AND } (m \text{ EQUALS } n))) : d_1. \end{aligned}$$

As preliminaries for giving lexical entries for modeling (2.5), we first define the nonlogical basic type Pred of predicatives. This type will be used to ensure that the argument to the comma intonation must be a predicative. Generalized quantifiers are converted to predicatives using the rule in figure 6.1. To see how this rule works in practice, note that the sign corresponding to *a cyclist* can be converted from a generalized quantifier to a predicative as follows:

$$\frac{\vdash \lambda_f.f(a \cdot \text{cyclist}) ; (\text{NP} \multimap S) \multimap S ; \text{A CYCLIST} : d_1 \rightarrow k}{\vdash a \cdot \text{cyclist} : s ; \text{Pred} ; (\text{PRED A CYCLIST}) : d_1}$$

(Note that $((\lambda_f.f(a \cdot \text{cyclist})) \lambda_s.s)$ β -reduces to $a \cdot \text{cyclist}$.)

$$\frac{\vdash f : (s \rightarrow s) \rightarrow s ; (NP \multimap S) \multimap S ; Q : d_1 \rightarrow k}{\vdash (f \lambda_s.s) : s ; \text{Pred} ; (\text{PRED } Q) : d_1} \text{ (Pred)}$$

Figure 6.1: DyCG nonlogical rule converting generalized quantifiers to predicates, where PRED is as defined in (6.5).

Since we need a way to say what the comma intonation does in the concrete syntax, I also introduce the function

$$\text{comma} : s \rightarrow s ,$$

but do not define it, leaving its implementation as a ‘hook’ for a future theory of intonation, which is beyond the scope of this thesis. Now a lexicon for modeling (2.5) can be given:

$$\begin{aligned} &\vdash \lambda_{sf}g.g(f \lambda_t.t \cdot (\text{comma } s)) : s \rightarrow ((s \rightarrow s) \rightarrow s) \rightarrow (s \rightarrow s) \rightarrow s ; \\ &\quad \text{Pred} \multimap ((NP \multimap S) \multimap S) \multimap (NP \multimap S) \multimap S ; \text{COMMA} \\ &\vdash \lambda_{s,s}.s \cdot \text{is} \cdot \text{from} \cdot \text{Texas} ; NP \multimap S ; \text{FROM-TEXAS} \end{aligned}$$

The lexical entries corresponding to the generalized quantifier *Lance* and property *cyclist* are as before in chapters 4 and 5. To save space, the semantic types are suppressed.

In proving a DyCG sign modeling (2.5), I adopt the space-saving approach used above, showing the pheno logic proof first, separately from the combined tecto and semantic proof. First the generalized quantifier *a cyclist* is derived and then converted to a predicate.

$$(6.6) \quad \frac{\vdash \lambda_{sf}.f(a \cdot s) : s \rightarrow (s \rightarrow s) \rightarrow s \quad \vdash \text{cyclist} : s}{\frac{\vdash \lambda_f.f(a \cdot \text{cyclist}) : (s \rightarrow s) \rightarrow s}{\vdash a \cdot \text{cyclist} : s}}$$

Then the combined string provides the required first argument to the concrete syntax of the comma intonation.

(6.7)

$$\frac{\vdash \lambda_{sf.g}.g(f \lambda_{t.t}.(\text{comma } s)) \quad \vdash a \cdot \text{cyclist}}{\vdash \lambda_{fg}.g(f \lambda_{t.t}.(\text{comma } (a \cdot \text{cyclist}))) : ((s \rightarrow s) \rightarrow s) \rightarrow (s \rightarrow s) \rightarrow s} \quad (6.6)$$

(In this proof tree, and below, I employ the following shorthands: some of the phenotypes are elided, as are rule labels, and β -reductions for terms are performed whenever possible.) Next, the comma-delineated appositive *a cyclist* derived above combines with its generalized quantifier argument, the anchor *Lance*.

(6.8)

$$\frac{\begin{array}{c} (6.7) \\ \vdots \\ \vdash \lambda_{f_g.g} (f \lambda_t.t \cdot (\text{comma } (a \cdot \text{cyclist}))) \quad \vdash \lambda_f.(f \text{Lance}) : (s \rightarrow s) \rightarrow s \end{array}}{\vdash \lambda_{g.g} (\text{Lance} \cdot (\text{comma } (a \cdot \text{cyclist}))) : (s \rightarrow s) \rightarrow s}$$

And finally, this new generalized quantifier's concrete syntax is applied to the pheno corresponding to the property of being from Texas.

(6.9)

$$\frac{\begin{array}{c} (6.8) \\ \vdots \\ \vdash \lambda_g.g \text{ (Lance} \cdot \text{(comma (a} \cdot \text{cyclist))} \text{))} \quad \vdash \lambda_s.s \cdot \text{is} \cdot \text{from} \cdot \text{Texas} : s \rightarrow s \end{array}}{\vdash \text{Lance} \cdot \text{(comma (a} \cdot \text{cyclist))} \cdot \text{is} \cdot \text{from} \cdot \text{Texas} : s}$$

Thus DyCG generates the correct surface string for (2.5), including the comma intonation, which here is left as an unanalyzed primitive.

Turning to the part of the proof dealing with the abstract syntax and semantics of (2.5), we start again by combining a with *cyclist*. First the predicative is built by combining with the generalized quantifier a *cyclist*, as above.

$$(6.10) \quad \frac{\frac{\vdash (\text{NP} \multimap \text{S}) \multimap (\text{NP} \multimap \text{S}) \multimap \text{S}; \mathbf{A} \quad \vdash \text{NP} \multimap \text{S}; \text{CYCLIST}}{\vdash (\text{NP} \multimap \text{S}) \multimap \text{S}; \mathbf{A} \text{ CYCLIST}}}{\vdash \text{Pred}; (\text{PRED } \mathbf{A} \text{ CYCLIST})}$$

Then COMMA takes the appositive as argument (here, the type of the abstract syntax accompanying COMMA is abbreviated by replacing the generalized quantifier syntactic type $(NP \multimap S) \multimap S$ by QP, as in chapters 3 and 4).

$$(6.11) \quad \frac{\begin{array}{c} (6.10) \\ \vdots \\ \vdots \end{array} \quad \frac{\vdash \text{Pred} \multimap \text{QP} \multimap \text{QP} ; \text{COMMA} \quad \text{Pred} ; (\text{PRED A CYCLIST})}{\vdash \text{QP} \multimap \text{QP} ; \text{COMMA} (\text{PRED A CYCLIST})}}{\vdash \text{QP} \multimap \text{QP} ; \text{COMMA} (\text{PRED A CYCLIST})}$$

Next *Lance* combines with *a cyclist* to form a new dynamic generalized quantifier, finally taking *is from Texas* as its argument.

$$(6.12) \quad \frac{\begin{array}{c} (6.11) \\ \vdots \\ \vdots \end{array} \quad \frac{\vdash \text{QP} \multimap \text{QP} ; \text{COMMA} (\text{PRED A CYCLIST}) \quad \vdash \text{QP} ; \text{LANCE}}{\vdash \text{QP} ; (\text{COMMA} (\text{PRED A CYCLIST})) \text{ LANCE}}}{\vdash \text{QP} ; (\text{COMMA} (\text{PRED A CYCLIST})) \text{ LANCE}}$$

Here again, the abstract syntactic type corresponding to the root label of the proof in (6.11) is abbreviated as before. Finally, the newly composed generalized quantifier takes the verb phrase as argument.

$$(6.13) \quad \frac{\begin{array}{c} (6.12) \\ \vdots \\ \vdots \end{array} \quad \frac{\vdash \text{QP} ; (\text{COMMA} (\text{PRED A CYCLIST})) \text{ LANCE} \quad \vdash \text{NP} \multimap \text{S} ; \text{FROM-TEXAS}}{\vdash \text{S} ; (\text{COMMA} (\text{PRED A CYCLIST})) \text{ LANCE} \text{ FROM-TEXAS}}}{\vdash \text{S} ; (\text{COMMA} (\text{PRED A CYCLIST})) \text{ LANCE} \text{ FROM-TEXAS}}$$

As before, the syntactic type of the root label of the proof in (6.12) is abbreviated.

Reducing the semantic term derived in (6.13) shows how this account of nominal appositives keeps the sense content separate from the associated implicature.

$$\begin{aligned} & \vdash (\text{COMMA} (\text{PRED A CYCLIST}) \text{ LANCE}) \text{ FROM-TEXAS} \\ & = \text{LANCE} ((\uparrow \text{PRED A CYCLIST}) \text{ THAT FROM-TEXAS}) \\ & = \text{THE NAMED-LANCE} ((\uparrow \text{PRED A CYCLIST}) \text{ THAT FROM-TEXAS}) \\ & \equiv \lambda_{c \cdot \lambda_{\mathbf{x}|c|}}.(\text{from-texas } \mathbf{x}_{(\text{the NAMED-LANCE } c)}) \mid \\ & \quad \lambda_{\mathbf{x}|c|}.(\text{exists}_y(\text{cyclist } y) \text{ and } (y \text{ equals } \mathbf{x}_{(\text{the NAMED-LANCE } c)})) \text{ and} \\ & \quad \text{exists!}_{n:\omega|c|}.c \text{ k-entails } (\text{NAMED-LANCE } n) : \mathbf{k} \end{aligned}$$

As this reduction shows, the dynamic meaning of the comma intonation has the effect of setting aside the information that Lance is a cyclist, as an implicature, while leaving the proposition that he is from Texas as part of the sense of the expression.

A negated variant of (2.5) demonstrates that COMMA correctly separates sense from implicature.

(2.5') Lance, a cyclist, is not from Texas.

The DyCG semantics for (2.5') is simply the semantics for (2.5) wrapped by the dynamic negation NOT.

$$\begin{aligned}
 & \vdash \text{NOT} ((\text{COMMA} (\text{PRED A CYCLIST}) \text{LANCE}) \text{FROM-TEXAS}) \\
 & \equiv \lambda_c. \lambda_{x|c|.} . \text{not} (\text{from-texas } x_{(\text{the NAMED-LANCE } c)}) \mid \\
 & \quad \lambda_{x|c|.} . (\text{exists}_y (\text{cyclist } y) \text{ and } (y \text{ equals } x_{(\text{the NAMED-LANCE } c)})) \text{ and} \\
 & \quad \text{exists!}_{n:\omega|c|.} . c \text{ k-entails } (\text{NAMED-LANCE } n) : k
 \end{aligned}$$

Here, no discourse referents become existentially bound in the sense because none are introduced (see lemma 4.11). Note that, in the DyCG semantics for (2.5'), only the proposition that Lance is from Texas is negated, while the proposition that Lance is a cyclist survives as an implicature.

The definition of the comma intonation given in (6.3) is general enough to account for utterance-final appositives, such as the instance of *a cyclist* in

(6.14) Kim met Lance, a cyclist.

Defining the dynamic meaning of *Kim* as

$$(6.15) \quad \text{KIM} =_{\text{def}} \text{THE NAMED-KIM},$$

just as in chapters 4 and 5, the extensions to the lexicon needed for (6.14) are as follows, where

$$\text{MEET} =_{\text{def}} (\text{dyn}_2 \text{meet}),$$

and $\text{meet} : e \rightarrow e \rightarrow p$ is the static relation of meeting.

$$\begin{aligned}
 & \vdash \lambda_f. (f \text{ Kim}) : (s \rightarrow s) \rightarrow s ; (\text{NP} \multimap S) \multimap S ; \text{KIM} : d_1 \rightarrow k \\
 & \vdash \lambda_{st}. t \cdot \text{met} \cdot s : s \rightarrow s \rightarrow s ; \text{NP} \multimap \text{NP} \multimap S ; \text{MEET} : d_2
 \end{aligned}$$

The concrete syntactic proof starts by hypothesizing both arguments to *meet*. Here, and below, I elide some of the types in the concrete syntactic proofs in order to save space, showing only the terms themselves.

$$(6.16) \quad \frac{\frac{\frac{\vdash \lambda_{st}.t \cdot \text{met} \cdot s : s \rightarrow s \rightarrow s \quad s : s \vdash s : s}{s : s \vdash \lambda_t.t \cdot \text{met} \cdot s : s \rightarrow s} \quad t : s \vdash t : s}{s : s, t : s \vdash t \cdot \text{met} \cdot s : s}}{t : s \vdash \lambda_s.t \cdot \text{met} \cdot s : s \rightarrow s}$$

After withdrawing the object trace, *Lance*, a *cyclist* is applied to the resulting term.

$$(6.17) \quad \frac{\frac{\frac{(6.8) \quad \vdots}{\vdash \lambda_g.g (\text{Lance} \cdot (\text{comma} (a \cdot \text{cyclist})))} \quad \frac{(6.16) \quad \vdots}{t : s \vdash \lambda_s.t \cdot \text{met} \cdot s : s \rightarrow s}}{t : s \vdash t \cdot \text{met} \cdot \text{Lance} \cdot (\text{comma} (a \cdot \text{cyclist})) : s}}{\vdash \lambda_t.t \cdot \text{met} \cdot \text{Lance} \cdot (\text{comma} (a \cdot \text{cyclist})) : s \rightarrow s}$$

The last step of the proof in (6.17) is that the subject trace is withdrawn so that *Kim* can take scope.

$$(6.18) \quad \frac{\vdash \lambda_f.(f \text{ Kim}) : (s \rightarrow s) \rightarrow s \quad \frac{(6.17) \quad \vdots}{\vdash \lambda_t.t \cdot \text{met} \cdot \text{Lance} \cdot (\text{comma} (a \cdot \text{cyclist}))}}{\vdash \text{Kim} \cdot \text{met} \cdot \text{Lance} \cdot (\text{comma} (a \cdot \text{cyclist})) : s}$$

And so with no modification to the lexical entry for the comma intonation given to model (2.5), the concrete syntax for a variant in which the appositive appears at the end of the utterance can be captured as well.

As for the semantics for (6.14), the proof also begins by hypothesizing discourse referents for both the subject and object of *meet*.

$$(6.19) \quad \frac{\frac{\vdash \text{NP} \multimap \text{NP} \multimap S ; \text{MEET} \quad \text{NP} ; m \vdash \text{NP} ; m}{\text{NP} ; m \vdash \text{NP} \multimap S ; \text{MEET } m} \quad \text{NP} ; n \vdash \text{NP} ; n}{\frac{\text{NP} ; m, \text{NP} ; n \vdash S ; \text{MEET } m \ n}{\text{NP} ; n \vdash \text{NP} \multimap S ; \lambda_m.\text{MEET } m \ n}}$$

Next the semantics of *Lance, a cyclist* takes as argument the verb phrase with its object abstracted over. I suppress the semantic types here, just like for the pheno proof above, as well as the abstract syntactic type of the root

(6.20)

Finally, *Kim's* semantics takes scope over the verb phrase after the subject discourse referent trace is withdrawn, yielding the final dynamic meaning of (6.14).

(6.20)

Here again, the abstract syntactic type corresponding to κIM , the type of a generalized quantifier, is abbreviated to QP .

To see that DyCG gives the right meaning for this example of an utterance-final appositive, we reduce the term derived in (6.20).

As before, the dynamic meaning is as desired: its sense is that Kim met Lance, and its implicature is that Lance is a cyclist, along with the two requirements associated with the proper names *Kim* and *Lance*. Note that withdrawing the traces in the other order would yield the proof term

$$\vdash (\text{COMMA} (\text{PRED A CYCLIST}) \text{ LANCE})_{m.\text{KIM}_n.\text{MEET } m \text{ } n} : k,$$

also corresponding to the concrete syntax of the surface form in (6.14). However, the truth conditions would not change, since this term β -reduces to the same form as the version with kim taking widest scope.

Although, for simplicity, I have chosen nominal appositives to demonstrate DyCG's handling of variable conventional implicatures, it is also straightforward to extend this account to other supplemental constructions, such as nonrestrictive relatives and *as*-parentheticals. For instance, the following variants of (2.5) are very similar to the original with the nominal appositive:

(6.22) Lance, who's a cyclist, was recently caught doping.

(6.23) Lance, as a cyclist, rides constantly.

Apart from defining the properties corresponding to their respective verb phrases, all that is needed for the variants in (6.22) and (6.23) is to give lexical entries for the nonrestrictive relativizer *who* and the parenthetical *as*. First we define the relativizer WHO_{nrrc} and the parenthetical AS_{paren} , both of which have the type $d_1 \rightarrow d_1$, to have the semantics of the identity function on dynamic properties, and likewise for the predicative copula *is*:

(6.24) $\text{WHO}_{\text{nrrc}} =_{\text{def}} \lambda_D.D$

(6.25) $\text{AS}_{\text{paren}} =_{\text{def}} \lambda_D.D$

(6.26) $\text{IS} =_{\text{def}} \lambda_D.D$

Then the lexicon extensions for modeling (6.22) and (6.23) is as follows.

$$\begin{aligned} &\vdash \lambda_f.\text{who} \cdot (f \mathbf{e}) : (s \rightarrow s) \rightarrow s ; (\text{NP} \multimap S) \multimap \text{Pred} ; \text{WHO}_{\text{nrrc}} \\ &\vdash \lambda_s.\text{as} \cdot s : s \rightarrow s ; \text{Pred} \multimap \text{Pred} ; \text{AS}_{\text{paren}} \\ &\vdash \lambda_{st}.t \cdot \text{is} \cdot s : s \rightarrow s \rightarrow s ; \text{Pred} \multimap \text{NP} \multimap S ; \text{IS} \\ &\vdash \lambda_s.s \cdot \text{was} \cdot \text{recently} \cdot \text{caught} \cdot \text{doping} : s \rightarrow s ; \text{NP} \multimap S ; \text{CAUGHT-DOPING} \\ &\vdash \lambda_s.s \cdot \text{rides} \cdot \text{constantly} : s \rightarrow s ; \text{NP} \multimap S ; \text{RIDES-CONSTANTLY} \end{aligned}$$

Here the dynamic properties CAUGHT-DOPING and RIDES-CONSTANTLY are the simplified dynamic meanings corresponding to the verb phrases in (6.22) and (6.23), respectively.

With these extensions to the lexicon, a (short-form) sign corresponding to (6.22) derived as

$$\vdash \text{Lance} \cdot (\text{comma who} \cdot \mathbf{e} \cdot \text{is} \cdot \text{a} \cdot \text{cyclist}) \cdot \text{was} \cdot \text{recently} \cdot \text{caught} \cdot \text{doping};$$

$$S; (\text{COMMA} (\text{WHO}_{\text{nrrc}} (\text{IS PRED A CYCLIST})) \text{ LANCE}) \text{ CAUGHT-DOPING}$$

and one for (6.23) can be derived as

$$\vdash \text{Lance} \cdot (\text{comma as} \cdot \text{a} \cdot \text{cyclist}) \cdot \text{rides} \cdot \text{constantly};$$

$$S; (\text{COMMA} (\text{AS}_{\text{paren}} (\text{PRED A CYCLIST})) \text{ LANCE}) \text{ RIDES-CONSTANTLY}$$

The semantics of these two signs reduce to

$$\vdash (\text{COMMA} (\text{WHO}_{\text{nrrc}} (\text{IS PRED A CYCLIST})) \text{ LANCE}) \text{ CAUGHT-DOPING}$$

$$= \text{THE NAMED-LANCE} ((\uparrow \text{PRED A CYCLIST}) \text{ THAT CAUGHT-DOPING}) : k$$

and

$$\vdash (\text{COMMA} (\text{AS}_{\text{paren}} (\text{PRED A CYCLIST})) \text{ LANCE}) \text{ RIDES-CONSTANTLY}$$

$$= \text{THE NAMED-LANCE} ((\uparrow \text{PRED A CYCLIST}) \text{ THAT RIDES-CONSTANTLY}) : k,$$

respectively.

6.1.1.1 Stacking

Like Potts's (2005) analysis of supplements, the DyCG analysis presented here can handle *supplement stacking*, in which multiple supplements are attached to a single anchor.

- (6.27) a. Colin Powell's son, Michael, Bush's choice to chair the FCC, is an unabashed free-marketeer convinced that Clinton/Gore's procorporate policies on the media were somehow bad for business.
- b. The reporter interviewed Lance Armstrong, a rider for the US Postal team, a cancer survivor.
- c. I rented *Annie Hall*, which is Woody Allen's finest, a true classic, in order to reminisce about the East Coast US.
- (Potts, 2005, examples 4.20 and 4.23)

Because Potts's theory can handle stacked supplements, it has an empirical edge over a recent dynamic alternative proposed by Nouwen (2007). The reason is that, in Nouwen's account, stacking is expressly forbidden by the formal machinery. Nouwen implements a version of Visser's (2002) extension to Dynamic Predicate Logic (Groenendijk and Stokhof, 1990) that has two *meaning types*: one for what I call sense content, and the other for implicature content. A problem with accounting for supplement stacking arises for Nouwen because adding implicature content is forbidden if the meaning type of the current compositional step is already the implicature type. As a result, Nouwen's incorrectly predicts that at most a single supplement per anchor can ever be present.

But this problem does not arise for the DyCG analysis I present here. I demonstrate DyCG's ability to handle supplement stacking for a simplified version of the stacked appositive in (6.27b). The necessary extensions to the lexicon are only those needed to model the properties of riding for US Postal and being a cancer survivor. For simplicity, these properties are shortened to the properties of being a rider and being a survivor, respectively.

$$\begin{aligned} &\vdash \text{rider} : s ; N ; \text{RIDER} : d_1 \\ &\vdash \text{survivor} : s ; N ; \text{SURVIVOR} : d_1 \end{aligned}$$

Here, $\text{RIDER} =_{\text{def}} (\text{dyn}_1 \text{ rider})$ and $\text{SURVIVOR} =_{\text{def}} (\text{dyn}_1 \text{ survivor})$, where $\text{rider} : e \rightarrow p$ and $\text{survivor} : e \rightarrow p$ are static properties.

The proof involves hypothesizing a generalized quantifier that is the argument to the first supplement, letting the second take the resulting term as argument, then withdrawing the quantifier for combination with the anchor. Starting again with the pheno proof, first the quantifier is hypothesized, then a derivation of the concrete syntax for the supplement *a rider* is derived following the proof in (6.7).

(6.28)

$$\begin{array}{c} \vdots \\ \hline \vdash \lambda_{fg}.g(f \lambda_{t.t} \cdot (\text{comma}(a \cdot \text{rider}))) \quad Q : (s \rightarrow s) \rightarrow s \vdash Q : (s \rightarrow s) \rightarrow s \\ \hline Q : (s \rightarrow s) \rightarrow s \vdash \lambda_g.g(Q \lambda_{t.t} \cdot (\text{comma}(a \cdot \text{rider}))) : (s \rightarrow s) \rightarrow s \end{array}$$

Next the supplement *a survivor* is derived, again via a proof that follows (6.7), and then applied to the term derived in (6.28). Here, the pheno string corresponding to *survivor* is abbreviated, and some of the types are

suppressed, in order to save space. The abbreviated context is $\Gamma = Q : (s \rightarrow s) \rightarrow s$.

(6.29)

$$\frac{\frac{\vdots \quad \vdots}{\vdash \lambda_{fg}.g(f \lambda_t.t \cdot \text{comma}(a \cdot \text{surv})) \quad \Gamma \vdash \lambda_g.g(Q \lambda_t.t \cdot \text{comma}(a \cdot \text{rider}))}}{\Gamma \vdash \lambda_g.g(Q \lambda_t.t \cdot (\text{comma}(a \cdot \text{rider})) \cdot (\text{comma}(a \cdot \text{surv})))} \quad (6.28)$$

As a final step, the hypothesized generalized quantifier phonology Q is withdrawn to make way for taking the proper name *Lance Armstrong*, here modeled as simply the pheno term for *Lance* given above, as argument.

(6.30)

$$\frac{\vdash \lambda_{Q_g.g} (Q \lambda_{t.t} (\text{comma } (a \cdot \text{rider})) \cdot (\text{comma } (a \cdot \text{surv}))) \quad \vdash \lambda_{f.f} (f \text{ Lance})}{\vdash \lambda_{g.g} (\text{Lance} \cdot (\text{comma } (a \cdot \text{rider})) \cdot (\text{comma } (a \cdot \text{surv}))) : (s \rightarrow s) \rightarrow s} \quad (6.29)$$

Applying this term to

$$\vdash \lambda_t.\text{the} \cdot \text{reporter} \cdot \text{interviewed} \cdot t : s \rightarrow s,$$

which models the verb phrase in (6.27b) with the object position abstracted over, yields

$$\vdash \text{the} \cdot \text{reporter} \cdot \text{interviewed} \cdot \text{Lance} \cdot (\text{comma} (\text{a} \cdot \text{rider})) \cdot (\text{comma} (\text{a} \cdot \text{survivor})) : s,$$

the (simplified) surface string corresponding to (6.27b).

The semantics for (6.27b) show how DyCG gets the right meanings for stacked supplements.

$$(6.31) \quad \frac{\vdots \quad \vdash \text{COMMA}(\text{PRED A RIDER}) \quad Q : d_1 \rightarrow k \vdash Q : d_1 \rightarrow k}{Q : d_1 \rightarrow k \vdash (\text{COMMA}(\text{PRED A RIDER}) Q) : d_1 \rightarrow k}$$

Here, similarly as above for the concrete syntax part of the proof, the semantics for the appositive *a rider* is derived by a proof that follows the

(6.32)

The root label of this proof has its type, which is

(6.33)

This term is capable of taking the dynamic property corresponding to being interviewed by someone, not modeled here, as its argument, to give the DyCG semantics for (6.27b). Importantly, the properties of being a rider and

being a survivor, both applied to the discourse referent corresponding to Lance, end up in the implicature part of the dynamic proposition modeling the stacked supplement.

6.2 Expressives

In this section I describe how the DyCG approach to modeling expressives, another type of variable conventional implicature discussed in §2.2.2.2. There I used the epithet, an instance of an expressive, in the following example, as part of a discussion making the empirical case that expressives are conventionally signaled but not necessarily speaker oriented.

(2.58) That socialist Obama got re-elected.

Here, I will give a formal model in DyCG of some instances of a class of related expressives, the class of *expressive adjectives*, exemplified in (6.34) and (6.35).

(6.34) Lance has entered the Tour de France, and the damn dooper will probably win it.

(6.35) The damn Republicans are only against taxes when they affect rich people.

(6.36) At least he isn't one of those damned socialists.

(6.37) Gimme back my damn bike!

(6.38) There's no fucking beer left!

As discussed in §2.2.2.2, the occurrence of the expressive adjectives in the above examples conveys that the anchor, here the speaker, has a heightened or negative attitude toward someone or something. These examples also give good motivation for following Potts (2005, chapter 5) in leaving unspecified the focus of the heightened attitude. In (6.34), the speaker is communicating a negative attitude toward Lance due to the fact that Lance is a dooper; in (6.35), the negative attitude is toward the Republicans because of their position on taxes. But for (6.36), the negative attitude is toward socialists; in (6.37) it is not toward the bike but toward the person who took the bike or to the fact that it was taken; in (6.38), it is toward the *absence* of beer, not toward beer.

The DyCG analysis of expressive adjectives like *damn* is extremely straightforward. By way of demonstrating it, I analyze a simplified variant of (6.34) in

(6.39) Lance entered the Tour de France, and the damn doper won.

For simplicity, I treat the property of entering the Tour de France as

$$\text{ENTER} =_{\text{def}} (\text{dyn}_1 \text{ enter}) ,$$

the dynamic property of entering, based as usual on its counterpart static property $\text{enter} : e \rightarrow p$. Similarly, the property of winning is

$$(6.40) \quad \text{WIN} =_{\text{def}} (\text{dyn}_1 \text{ win}) ,$$

also based on $\text{win} : e \rightarrow p$, its corresponding static property.

As for the DyCG meaning of *damn*, it is treated as a common noun modifier that simply passes through the sense proposition of its argument but adds the implicature that the point of view holder has a negative attitude toward its argument's argument. The type of DAMN is

$$\text{DAMN} : d_1 \rightarrow d_1 ,$$

and its definition is

$$(6.41) \quad \text{DAMN} =_{\text{def}} \lambda_{Dnc} . (D n c)^s \mid \lambda_{x|c|} . (D n c x)^i \text{ and } (\text{neg } D) .$$

This definition of the dynamic meaning of *damn* uses the function $\text{neg} : d_1 \rightarrow p$, which is the property of there being a negative attitude toward some aspect related to its dynamic property argument. Exactly which aspect is left unspecified, as discussed above. And when it comes to the question of which point of view holds the negative attitude toward the entity provided, it is not accounted for in the current account.

With most of the other words in (6.39) already having DyCG lexical entries, it is straightforward to provide the remaining necessary entries.

$$\begin{aligned} &\vdash \lambda_s . s \cdot \text{entered} \cdot \text{the} \cdot \text{Tour} \cdot \text{de} \cdot \text{France} : s \rightarrow s ; \text{NP} \multimap S ; \text{ENTER} : d_1 \\ &\vdash \lambda_{st} . t \cdot \text{and} \cdot s : s \rightarrow s \rightarrow s ; S \multimap S \multimap S ; \lambda_{hk} . k \text{ AND } h : k \rightarrow k \rightarrow k \\ &\vdash \lambda_s . \text{damn} \cdot s : s \rightarrow s ; N \multimap N ; \text{DAMN} : d_1 \rightarrow d_1 \\ &\vdash \text{doper} : s ; N ; \text{DOPER} : d_1 \\ &\vdash \lambda_s . s \cdot \text{won} \cdot \text{it} : s \rightarrow s ; \text{NP} \multimap S ; \text{WIN} : d_1 \end{aligned}$$

The treatment of *and* in this lexicon is just straightforward sentential conjunction, with the argument contents passed to the dynamic conjunction AND.

The concrete syntax of the first conjunct of (6.39) is shown in the proof below, with types suppressed to save space, as before.

$$(6.42) \quad \frac{\vdash \lambda_f.(f \text{ Lance}) \quad \vdash \lambda_s.s \cdot \text{entered} \cdot \text{the} \cdot \text{Tour} \cdot \text{de} \cdot \text{France}}{\vdash \text{Lance} \cdot \text{entered} \cdot \text{the} \cdot \text{Tour} \cdot \text{de} \cdot \text{France}}$$

The second conjunct's concrete syntax has a similarly straightforward proof.

$$(6.43) \quad \frac{\vdash \lambda_{sf}.f(\text{the} \cdot s) \quad \frac{\vdash \lambda_s.\text{damn} \cdot s \quad \vdash \text{doper}}{\vdash \text{damn} \cdot \text{doper}}}{\vdash \lambda_f.f(\text{the} \cdot \text{damn} \cdot \text{doper})} \quad \vdash \lambda_s.s \cdot \text{won} \cdot \text{it}$$

$$\frac{}{\vdash \text{the} \cdot \text{damn} \cdot \text{doper} \cdot \text{won} \cdot \text{it}}$$

And finally the two conjuncts are combined (here the string derived in (6.42) is abbreviated, and the string *Tour · de · France* is abbreviated TdF).

$$(6.44) \quad \frac{\frac{\vdash \lambda_{st}.t \cdot \text{and} \cdot s \quad \vdash \text{the} \cdot \text{damn} \cdot \text{doper} \cdot \text{won} \cdot \text{it}}{\vdash \lambda_t.t \cdot \text{and} \cdot \text{the} \cdot \text{damn} \cdot \text{doper} \cdot \text{won} \cdot \text{it}} \quad \frac{\vdash \text{Lance} \cdot \text{entered} \cdot \dots}{\vdash \text{Lance} \cdot \text{entered} \cdot \text{the} \cdot \text{TdF} \cdot \text{and} \cdot \text{the} \cdot \text{damn} \cdot \text{doper} \cdot \text{won} \cdot \text{it}}}{\vdash \text{Lance} \cdot \text{entered} \cdot \text{the} \cdot \text{TdF} \cdot \text{and} \cdot \text{the} \cdot \text{damn} \cdot \text{doper} \cdot \text{won} \cdot \text{it}}$$

This is clearly the correct surface string for (6.39).

The tecto and semantic proofs are equally straightforward. First, the first conjunct:

$$(6.45) \quad \frac{\vdash (\text{NP} \multimap \text{S}) \multimap \text{S}; \text{LANCE} \quad \vdash \text{NP} \multimap \text{S}; \text{ENTER}}{\vdash \text{S}; (\text{LANCE ENTER})}$$

And next the second. In the proof of the second conjunct, the abstract syntactic type $(\text{NP} \multimap \text{S}) \multimap \text{S}$ is abbreviated to QP for space.

$$(6.46) \quad \frac{\vdash \text{N} \multimap \text{QP}; \text{THE} \quad \frac{\vdash \text{N} \multimap \text{N}; \text{DAMN} \quad \vdash \text{N}; \text{DOPER}}{\vdash \text{N}; (\text{DAMN DOPER})}}{\vdash \text{QP}; \text{THE} (\text{DAMN DOPER})} \quad \vdash \text{NP} \multimap \text{S}; \text{WIN}$$

$$\frac{}{\vdash \text{S}; \text{THE} (\text{DAMN DOPER}) \text{ WIN}}$$

The last step is that the conjuncts are combined, as for the pheno proof above.

$$\begin{array}{c}
 (6.47) \quad \frac{\frac{\frac{\vdash \lambda_{hk}.k \text{ AND } h \quad \vdash S; \text{THE (DAMN DOPER) WIN}}{\vdash S \multimap S; \lambda_k.k \text{ AND THE (DAMN DOPER) WIN}} \quad \frac{\vdash S; \text{(LANCE ENTER)}}{\vdash S; \text{(LANCE ENTER) AND THE (DAMN DOPER) WIN}}}{\vdash S; \text{(LANCE ENTER) AND THE (DAMN DOPER) WIN}}
 \end{array}$$

In this proof, the tecto type assigned to *and* is elided; it is $S \multimap S \multimap S$.

Reducing the semantics for (6.39) derived in (6.47) shows how the DyCG meaning for *damn* functions.

$$\begin{aligned}
 & \vdash \text{(LANCE ENTER) AND THE (DAMN DOPER) WIN} \\
 &= \text{(THE NAMED-LANCE ENTER) AND THE (DAMN DOPER) WIN} \\
 &\equiv \lambda_c. \lambda_{\mathbf{x}|c|.} (\text{enter } \mathbf{x}_{(\text{the NAMED-LANCE } c)}) \text{ and } (\text{win } \mathbf{x}_{(\text{the (DAMN DOPER) } c)}) \mid \\
 & \quad \lambda_{\mathbf{x}|c|.} (\text{exists!}_{n:\omega|c|.} c \text{ k-entails (NAMED-LANCE } n)) \text{ and} \\
 & \quad (\text{neg DOPER) and exists!}_{n:\omega|c'|.} c' \text{ k-entails ((DAMN DOPER) } n) : k
 \end{aligned}$$

Here, the intermediate context passed to the second conjunct is

$$\begin{aligned}
 c' &\equiv \lambda_{\mathbf{x}|c|.} (c \mathbf{x}) \text{ and } (\text{enter } \mathbf{x}_{(\text{the NAMED-LANCE } c)}) \text{ and} \\
 & \quad \text{exists!}_{n:\omega|c|.} c \text{ k-entails (NAMED-LANCE } n)
 \end{aligned}$$

And so the dynamic meaning of (6.39) adds the negative attitude towards the property of being a doper to the implicature portion, whereas its sense portion only contains the information that Lance entered and won, as desired. The anaphoric link between Lance and the doper is possible as long as Lance is unique among the discourse referents entailed to be a doper, and the attitude holder is entailed to have a negative attitude toward dopers in general.

As a demonstration of how the DyCG account of (6.39) correctly captures the implicatures of expressives, consider the variant

(6.39') Lance entered the Tour de France, and the damn doper didn't win.

The only difference between (6.39) and (6.39') is that in (6.39'), the second conjunct is negated. Based on the lexicon given above for (6.39), this variant would receive the following DyCG semantics.

$$\begin{aligned} &\vdash (\text{LANCE ENTER}) \text{ AND NOT } (\text{THE } (\text{DAMN DOPER}) \text{ WIN}) \\ &\equiv \lambda_c \lambda_{\mathbf{x}|c|.} (\text{enter } \mathbf{x}_{(\text{the NAMED-LANCE } c)}) \text{ and not } (\text{win } \mathbf{x}_{(\text{the } (\text{DAMN DOPER}) c)}) \mid \\ &\quad (\text{exists!}_{n:\omega|c|.} c \text{ k-entails } (\text{NAMED-LANCE } n)) \text{ and} \\ &\quad (\text{neg DOPER}) \text{ and exists!}_{n:\omega|c|.} c \text{ k-entails } ((\text{DAMN DOPER}) n) : k \end{aligned}$$

Note that the implicature associated with the use of *damn* in (6.39') is preserved because only the information that Lance entered is negated.

Finally, I note that, similarly to the function possessive in §5.2.1.1, a general function that generates dynamic meanings for expressives can be defined. Potts (2005, pages 167–168) mentions that certain expressives give rise to the implicature of a *positive*, rather than a negative, attitude toward a certain discourse referent, giving the expressive *brilliant* used in British English as an example. So a general function describing the class of expressives should abstract over the implicated attitude. The function describing expressives I define here is called *expressive*, which has the type

$$(d_1 \rightarrow p) \rightarrow d_1 \rightarrow d_1 ,$$

and is defined to take a static property representing the implicated attitude to yield a modifier over dynamic properties. It is defined as

$$(6.48) \quad \text{expressive} =_{\text{def}} \lambda_{PDnc}. (D \text{ } n \text{ } c)^s \mid \lambda_{\mathbf{x}|c|.} (D \text{ } n \text{ } c \text{ } \mathbf{x})^i \text{ and } (P \text{ } D) ,$$

so that, for example, the definition of DAMN in equation (6.41) can instead be given as

$$(6.49) \quad \text{DAMN} =_{\text{def}} (\text{expressive neg}) .$$

A similar definition could be provided for the English expressives *fucking*, *bloody*, *stupid*, and for other expressives that implicate a negative attitude.

As for the positive expressives, such as the British English *brilliant*, they could be defined by invoking *expressive* as defined in (6.48) with the positive counterpart to *neg*. The function *expressive* could also be used to define dynamic meanings for epithets such as *that socialist* in

(2.58) That socialist Obama got re-elected.

by defining an expressive version of *socialist* similar to DAMN in (6.49). Similarly, honorifics in languages like French and Japanese could be defined using expressive by passing it the property of being in a superior or deferential position with respect to another.

6.3 The Interaction between Sense and Implicature

As pointed out by Amaral et al. (2007), Potts's (2005) theory of supplements is deficient in that it imposes the austere restriction on meanings that the sense part can never interact with the implicature part. I discuss Potts's reasons for making this move, along with a discussion of the problem he was attempting to solve, in §6.5. In this section, I show how DyCG correctly allows anaphora between the sense and implicature portions of dynamic meanings.

Examples abound that show why Potts's prohibition on the interaction between sense and implicature is empirically inadequate. Some such examples include those in

(6.50) Stan Bronowski, who took an exam_{*i*}, passed it_{*i*} with flying colors.
(Amaral et al., 2007, page 740, a variant of example 4.24 in Potts 2005)

(6.51) Lance_{*i*}'s riding partner Kim, who's been envying his_{*i*} bike_{*j*} all season, just asked him_{*i*} to borrow it_{*j*}.

In (6.50), a discourse referent representing the exam Stan took is introduced within a supplement (here, a nonrestrictive relative) and antecedes the pronoun *it*, which is outside the supplement. And (6.51) shows that the interaction is even more free than (6.50) demonstrates. In this example, the possessive *his* occurs within a supplement but finds its antecedent outside it, and the pronoun *it* is outside the supplement, but its antecedent *a bike* occurs within it.

To show how DyCG handles the interaction between sense and implicature, I start with the simple example

(6.52) Kim, a cyclist that has a bike_{*i*}, rides it_{*i*}.

The entire DyCG lexicon needed to model (6.52) is already in place, allowing the following dynamic semantics:

$$(6.53) \quad \vdash (\text{COMMA } (\text{PRED A } (\text{CYCLIST THAT } \lambda_n.(\text{A BIKE})_m.\text{HAVE } m \ n)) \text{ KIM}) \\ \lambda_n.\text{IT}_m.\text{RIDE } m \ n : k$$

A good way to clearly show how the pronoun *it* is able to select its antecedent *a bike*, which occurs inside the nominal appositive in (6.52), is to show the effect of *Kim, a cyclist that has a bike* on the input context. This part of the meaning of (6.52) reduces as follows.

$$\begin{aligned} & \vdash \text{COMMA } (\text{PRED A } (\text{CYCLIST THAT } \lambda_n.(\text{A BIKE})_m.\text{HAVE } m \ n)) \text{ KIM} \\ &= \lambda_D.\text{THE NAMED-KIM } ((\uparrow \text{PRED A } (\text{CYCLIST THAT} \\ & \quad \lambda_n.\text{EXISTS}_m.(\text{BIKE } m) \text{ AND } (\text{HAVE } m \ n))) \text{ THAT } D) \\ &\equiv \lambda_{Dc}.(((\uparrow \text{PRED A } (\text{CYCLIST THAT } \lambda_n.\text{EXISTS}_m.(\text{BIKE } m) \text{ AND } (\text{HAVE } m \ n))) \\ & \quad \text{THAT } D) (\text{the NAMED-KIM } c) \ c)^s \mid \\ & \quad \lambda_{\mathbf{x}|c|}.(\text{exists!}_{n:\omega|c|} c \text{ k-entails } (\text{NAMED-KIM } n)) \text{ and} \\ & \quad (\text{exists}_{y,z}.(\text{cyclist } y) \text{ and } (\text{bike } z) \text{ and } (\text{have } z \ y) \text{ and} \\ & \quad (y \text{ equals } \mathbf{x}_{(\text{the NAMED-KIM } c)})) \text{ and } (D (\text{the NAMED-KIM } c) \ c')^i \mathbf{x} : d_1 \rightarrow k \end{aligned}$$

Here, the context passed to *D* by *THAT*, and ultimately by *AND*, is:

$$\begin{aligned} c' &\equiv \lambda_{\mathbf{x}|c|}.(c \ \mathbf{x}) \text{ and } (\text{exists!}_{n:\omega|c|} c \text{ k-entails } (\text{NAMED-KIM } n)) \text{ and} \\ & \quad \text{exists}_{y,z}.(\text{cyclist } y) \text{ and } (\text{bike } z) \text{ and } (\text{have } z \ y) \text{ and} \\ & \quad (y \text{ equals } \mathbf{x}_{(\text{the NAMED-KIM } c)}) \end{aligned}$$

This is the output context of the appositive in (6.53) because it is modeled by the dynamic property

$$\begin{aligned} & \vdash \uparrow \text{PRED A } (\text{CYCLIST THAT } \lambda_n.(\text{A BIKE})_m.\text{HAVE } m \ n) \\ &= \lambda_{nc}.\text{T}_{|c|} \mid \lambda_{\mathbf{x}|c|}.\text{exists}_{y,z}.(\text{cyclist } y) \text{ and } (\text{bike } z) \text{ and } (\text{have } z \ y) \text{ and} \\ & \quad (y \text{ equals } \mathbf{x}_n) : d_1. \end{aligned}$$

Since *THAT* is defined in (4.24) in terms of the dynamic conjunction *AND* (5.4), its second conjunct is evaluated in the context updated by its first.

Here, the AND-internal cc function places the implicature of the appositive into the input context of THAT's second conjunct D .

In this case, the second conjunct of THAT is the dynamic meaning of *rides it*. Expanding the semantics for *rides it* shows how the pronoun can select its antecedent given the input context it is passed, assuming the weak familiarity mechanism described in §5.3.1 provides discourse referents based on the existential entailments present.

$$\begin{aligned}
 & \vdash \lambda_n.IT_m.RIDE\ m\ n \\
 & = \lambda_n.(PRO\ NONHUMAN)_m.RIDE\ m\ n \\
 & \equiv \lambda_{nc}.\lambda_{x|c|}.(nonhuman\ x_{(PRO\ NONHUMAN\ c)})\ and\ (ride\ x_{(PRO\ NONHUMAN\ c)}\ x_n) \mid \\
 & \quad \lambda_{x|c|}.exists!_{n:\omega|c|}.c\ k-cons\ (NONHUMAN\ n) : d_1
 \end{aligned}$$

Since the context c' passed to the semantics for *rides it* contains a discourse referent for the bike, there is an available antecedent to the pronoun *it*: the bike's being nonhuman is consistent with the entailments present in the input context. And so DyCG is capable of handling instances of anaphora in which the anaphor does not occur within a supplement but finds its antecedent inside one.

As for the other type of interaction between senses and implicatures pointed out by Amaral et al., in which the anaphor occurs within a supplement but the antecedent does not, consider

(6.54) Kim_{*i*}'s bike_{*j*}, which has her_{*i*} name on it_{*j*}, is fast.

Providing a DyCG sign corresponding to (6.54) requires a few new definitions and extensions to the lexicon, but these are fairly straightforward.

The semantics of the possessive *Kim's bike* is already available using the possessive function defined in equation (5.18). The definite possessive modeling *Kim's bike* is

$$(6.55) \quad \vdash KIMS_{def} =_{def} possessive\ THE\ KIM ,$$

while its indefinite counterpart is simply

$$(6.56) \quad \vdash KIMS_{indef} =_{def} possessive\ A\ KIM ,$$

where KIM is as defined in (6.15).

Next we need a semantics for the nonrestrictive relativizer *which* in (6.54). This definition is similar to the definitions for $\text{WHOS}_{\text{nrrc}}$ and AS_{paren} given respectively in equations (6.24) and (6.25). As in those definitions, the type of the nonrestrictive *which* is

$$\text{WHICH}_{\text{nrrc}} : d_1 \rightarrow d_1 ,$$

defined as before as

$$(6.57) \quad \text{WHICH}_{\text{nrrc}} =_{\text{def}} \lambda_D.D ,$$

the identity function on dynamic properties.

The ‘semantically vacuous’ preposition *on* is modeled similarly, as the identity function on natural numbers, with the type

$$\text{ON} : \omega \rightarrow \omega ,$$

and defined as

$$(6.58) \quad \text{ON} =_{\text{def}} \lambda_n.n .$$

And lastly, we need a version of the verb *have* that takes three arguments: an *on*-phrase, an object, and a subject. Semantically, this ditransitive is modeled as

$$(6.59) \quad \text{HAVE}_{\text{on}} =_{\text{def}} (\text{dyn}_3 \text{ have}_{\text{on}}) ,$$

where $\text{have}_{\text{on}} : e \rightarrow e \rightarrow e \rightarrow p$ is the corresponding static ditransitive. The dynamic properties $\text{NAME} =_{\text{def}} (\text{dyn}_1 \text{ name})$ and $\text{FAST} =_{\text{def}} (\text{dyn}_1 \text{ fast})$ are defined based on the static properties $\text{name} : e \rightarrow p$ and $\text{fast} : e \rightarrow p$, respectively.

Turning to the task of defining the actual lexicon needed for a DyCG account of (6.54), I begin with the lexical entry corresponding to the possessive *Kim’s*, which is modeled as a dynamic generalized determiner. Here I show the indefinite version of the possessive, but the definite version is exactly the same except that the semantic term is kims_{def} instead of $\text{kims}_{\text{indef}}$. I make the assumption in modeling (6.54), that there is no available discourse referent to antecede *Kim’s bike* in order to more clearly show

how the anaphora between sense and implicature works.

$$\vdash \lambda_{sf}.f(\text{Kim's} \cdot s) : s \rightarrow (s \rightarrow s) \rightarrow s ; N \multimap (NP \multimap S) \multimap S ; \\ \text{KIMS}_{\text{indef}} : d_1 \rightarrow d_1 \rightarrow k$$

As for the nonrestrictive relativizer *which*, its lexical entry is nearly identical to the one for WHO_{nrrc} , above, which also passes the empty string **e** to its argument.

$$\vdash \lambda_f.\text{which} \cdot (f \text{ e}) : (s \rightarrow s) \rightarrow s ; (NP \multimap S) \multimap \text{Pred} ; \text{WHICH}_{\text{nrrc}} : d_1 \rightarrow d_1$$

Then the lexical entry for the ditransitive version of *have* is one that takes an *on*-phrase as its first argument in the abstract syntax, where the abstract syntactic type *On* is a new, basic type.

$$\vdash \lambda_{stu}.u \cdot \text{has} \cdot t \cdot s : s \rightarrow s \rightarrow s \rightarrow s ; \text{On} \multimap NP \multimap NP \multimap S ; \text{HAVE}_{\text{on}} : d_3$$

As for the *on*-phrases themselves, the corresponding lexical entry makes sure the abstract syntactic types are correct, and inserts the phonology of the word *on*.

$$\vdash \lambda_s.\text{on} \cdot s : s \rightarrow s ; NP \multimap \text{On} ; \text{ON} : \omega \rightarrow \omega$$

The lexical entry for the common noun *name* is straightforward, and for simplicity, the one for *is fast* incorporates the predicative *is* syncategorematically.

$$\vdash \text{name} : s ; N ; \text{NAME} : d_1 \\ \vdash \lambda_{s.s}.s \cdot \text{is} \cdot \text{fast} : s \rightarrow s ; NP \multimap S ; \text{FAST} : d_1$$

And finally, the lexical entry corresponding to *her* is defined similarly to the one for *his* in §5.2.1.1.

$$\vdash \lambda_{sf}.f(\text{her} \cdot s) ; N \multimap (NP \multimap S) \multimap S ; \text{HER}_{\text{def}} : d_1 \rightarrow d_1 \rightarrow k$$

Starting with the pheno, the first step is to provide all of the arguments to *has* by hypothesis, so that its arguments, all of which are generalized quantifiers, can take scope. The string *s* is withdrawn first to make way for *it*. (The pheno proof below, and the ones that follow, omit the types to save space, and as usual, β -reductions are performed in terms whenever

possible.)

$$(6.60) \quad \frac{\frac{\frac{\frac{\vdash \lambda_{stu}.u \cdot \text{has} \cdot t \cdot s}{s \vdash \lambda_{tu}.u \cdot \text{has} \cdot t \cdot \text{on} \cdot s} \quad \frac{\frac{\vdash \lambda_s.\text{on} \cdot s \quad s \vdash s}{s \vdash \text{on} \cdot s}}{s, t \vdash \lambda_u.u \cdot \text{has} \cdot t \cdot \text{on} \cdot s} \quad t \vdash t}{s, t, u \vdash u \cdot \text{has} \cdot t \cdot \text{on} \cdot s} \quad u \vdash u}{t, u \vdash \lambda_s.u \cdot \text{has} \cdot t \cdot \text{on} \cdot s}$$

Then *it* is combined with the verb phrase with its *on*-position abstracted over, so that the possessive *her name* can take scope. As a last step, the subject pheno variable is withdrawn to make way for *which*.

(6.61)

$$(6.60) \quad \vdots$$

$$\frac{\frac{\frac{\vdash \lambda_{sf}.f(\text{her} \cdot s) \quad \vdash \text{name}}{\vdash \lambda_f.f(\text{her} \cdot \text{name})} \quad \frac{\frac{\frac{\vdash \lambda_f.(f \text{ it}) \quad t, u \vdash \lambda_s.u \cdot \text{has} \cdot t \cdot \text{on} \cdot s}{t, u \vdash u \cdot \text{has} \cdot t \cdot \text{on} \cdot \text{it}}}{u \vdash \lambda_t.u \cdot \text{has} \cdot t \cdot \text{on} \cdot \text{it}}}{u \vdash u \cdot \text{has} \cdot \text{her} \cdot \text{name} \cdot \text{on} \cdot \text{it}} \quad \vdash \lambda_u.u \cdot \text{has} \cdot \text{her} \cdot \text{name} \cdot \text{on} \cdot \text{it}}$$

Then the relativizer takes the subject-extracted string as its argument:

$$(6.61) \quad \vdots$$

$$(6.62) \quad \frac{\vdash \lambda_f.\text{which} \cdot (f \mathbf{e}) \quad \vdash \lambda_u.u \cdot \text{has} \cdot \text{her} \cdot \text{name} \cdot \text{on} \cdot \text{it}}{\vdash \text{which} \cdot \mathbf{e} \cdot \text{has} \cdot \text{her} \cdot \text{name} \cdot \text{on} \cdot \text{it}}$$

And next the comma intonation surrounds the whole nonrestrictive relative.

(6.63)

$$(6.62) \quad \vdots$$

$$\frac{\vdash \lambda_{sfg}.g(f \lambda_t.t \cdot (\text{comma } s)) \quad \vdash \text{which} \cdot \mathbf{e} \cdot \text{has} \cdot \text{her} \cdot \text{name} \cdot \text{on} \cdot \text{it}}{\vdash \lambda_{fg}.g(f \lambda_t.t \cdot (\text{comma which} \cdot \mathbf{e} \cdot \text{has} \cdot \text{her} \cdot \text{name} \cdot \text{on} \cdot \text{it}))}$$

Now the generalized quantifier pheno for *Kim's bike* is built.

$$(6.64) \quad \frac{\vdash \lambda_{sf}.f(\text{Kim's} \cdot s) \quad \vdash \text{bike}}{\vdash \lambda_f.f(\text{Kim's} \cdot \text{bike})}$$

$$\begin{array}{c}
 \text{(6.63)} \qquad \qquad \qquad \text{(6.64)} \\
 \vdots \qquad \qquad \qquad \vdots \\
 \text{(6.65)} \quad \frac{\vdash \lambda_{f.g.g} (f \lambda_{t.t} \cdot (\text{comma which} \cdot \dots)) \quad \vdash \lambda_f.f (\text{Kim's} \cdot \text{bike})}{\vdash \lambda_{g.g} (\text{Kim's} \cdot \text{bike} \cdot (\text{comma which} \cdot \mathbf{e} \cdot \text{has} \cdot \text{her} \cdot \text{name} \cdot \text{on} \cdot \text{it}))}
 \end{array}$$
$$(6.66) \quad \frac{\vdots \quad \vdash \lambda_g.g \text{ (Kim's \cdot bike \cdot (comma which \cdot \cdot \cdot))} \quad \vdash \lambda_s.s \text{ is \cdot fast}}{\vdash \text{Kim's \cdot bike \cdot (comma which \cdot e \cdot has \cdot her \cdot name \cdot on \cdot it) \cdot is \cdot fast}}$$
$$(6.67) \quad \frac{\frac{\vdash \text{On} \multimap \text{NP} \multimap \text{NP} \multimap S; \text{HAVE}_{\text{on}}}{\text{NP}; j \vdash \text{NP} \multimap \text{NP} \multimap S; \text{HAVE}_{\text{on}} (\text{ON } j)} \quad \frac{\vdash \text{NP} \multimap \text{On}; \text{ON} \quad \text{NP}; j \vdash \text{NP}; j}{\text{NP}; j \vdash \text{On}; (\text{ON } j)}}{\text{NP}; j \vdash \text{NP} \multimap \text{NP} \multimap S; \text{HAVE}_{\text{on}} (\text{ON } j)}$$
$$(6.68) \quad \frac{\begin{array}{c} (6.67) \\ \vdots \\ \text{NP}; j \vdash \text{NP} \multimap \text{NP} \multimap \text{S}; \text{HAVE}_{\text{on}}(\text{ON } j) \end{array} \quad \text{NP}; m \vdash \text{NP}; m}{\text{NP}; j, \text{NP}; m \vdash \text{NP} \multimap \text{S}; \text{HAVE}_{\text{on}}(\text{ON } j) m}$$

Next the subject trace is provided, and the *on*-phrase's argument is withdrawn.

(6.69)

$$\begin{array}{c}
 (6.68) \\
 \vdots \\
 \frac{\text{NP} ; j, \text{NP} ; m \vdash \text{NP} \multimap \text{S} ; \text{HAVE}_{\text{on}} (\text{ON } j) m \quad \text{NP} ; n \vdash \text{NP} ; n}{\text{NP} ; j, \text{NP} ; m, \text{NP} ; n \vdash \text{S} ; \text{HAVE}_{\text{on}} (\text{ON } j) m n} \\
 \hline
 \text{NP} ; m, \text{NP} ; n \vdash \text{NP} \multimap \text{S} ; \lambda_j. \text{HAVE}_{\text{on}} (\text{ON } j) m n
 \end{array}$$

Then *it* is combined with the term derived in (6.68), and then the object position trace is withdrawn to make ready for *her name*.

(6.70)

$$\begin{array}{c}
 (6.69) \\
 \vdots \\
 \frac{\vdash (\text{NP} \multimap \text{S}) \multimap \text{S} ; \text{IT} \quad \text{NP} ; m, \text{NP} ; n \vdash \text{NP} \multimap \text{S} ; \lambda_j. \text{HAVE}_{\text{on}} (\text{ON } j) m n}{\text{NP} ; m, \text{NP} ; n \vdash \text{S} ; \text{IT}_j. \text{HAVE}_{\text{on}} (\text{ON } j) m n} \\
 \hline
 \text{NP} ; n \vdash \text{NP} \multimap \text{S} ; \lambda_m. \text{IT}_j. \text{HAVE}_{\text{on}} (\text{ON } j) m n
 \end{array}$$

Then *her name* is built so that it can take the term derived in (6.70) as its argument.

(6.71)

$$\frac{\vdash (\text{NP} \multimap \text{S}) \multimap (\text{NP} \multimap \text{S}) \multimap \text{S} ; \text{HER}_{\text{def}} \quad \vdash (\text{NP} \multimap \text{S}) ; \text{NAME}}{\vdash (\text{NP} \multimap \text{S}) \multimap \text{S} ; (\text{HER}_{\text{def}} \text{NAME})}$$

The syntax and semantics for *her name* derived in (6.71) is then combined with its argument (here the abstract syntactic type $(\text{NP} \multimap \text{S}) \multimap \text{S}$ corresponding to *her name* has been abbreviated to QP in order to save space). And next the subject trace is withdrawn.

(6.72)

$$\begin{array}{c}
 (6.71) \qquad \qquad \qquad (6.70) \\
 \vdots \qquad \qquad \qquad \vdots \\
 \frac{\vdash \text{QP} ; (\text{HER}_{\text{def}} \text{NAME}) \quad \text{NP} ; n \vdash \text{NP} \multimap \text{S} ; \lambda_m. \text{IT}_j. \text{HAVE}_{\text{on}} (\text{ON } j) m n}{\text{NP} ; n \vdash \text{S} ; (\text{HER}_{\text{def}} \text{NAME})_m. \text{IT}_j. \text{HAVE}_{\text{on}} (\text{ON } j) m n} \\
 \hline
 \vdash \text{NP} \multimap \text{S} ; \lambda_n. (\text{HER}_{\text{def}} \text{NAME})_m. \text{IT}_j. \text{HAVE}_{\text{on}} (\text{ON } j) m n
 \end{array}$$

The nonrestrictive relativizer *which*, whose abstract syntactic type has been omitted, now takes this term as argument.

(6.73)

$$\frac{\begin{array}{c} \vdots \\ \vdash \text{WHICH}_{\text{nrrc}} \quad \vdash \text{NP} \multimap \text{S} ; \lambda_n.(\text{HER}_{\text{def}} \text{NAME})_m.\text{IT}_j.\text{HAVE}_{\text{on}}(\text{ON } j) \text{ } m \text{ } n \end{array}}{\vdash \text{Pred} ; \text{WHICH}_{\text{nrrc}} \lambda_n.(\text{HER}_{\text{def}} \text{NAME})_m.\text{IT}_j.\text{HAVE}_{\text{on}}(\text{ON } j) \text{ } m \text{ } n}$$

The nonrestrictive relative is completed by surrounding the term derived in (6.73) with the comma intonation (in the following proof, none of the abstract syntactic types are shown to save space).

(6.74)

$$\frac{\begin{array}{c} \vdots \\ \vdash \text{COMMA} \quad \vdash \text{WHICH}_{\text{nrrc}} \lambda_n.(\text{HER}_{\text{def}} \text{NAME})_m.\text{IT}_j.\text{HAVE}_{\text{on}}(\text{ON } j) \text{ } m \text{ } n \end{array}}{\vdash \text{COMMA} (\text{WHICH}_{\text{nrrc}} \lambda_n.(\text{HER}_{\text{def}} \text{NAME})_m.\text{IT}_j.\text{HAVE}_{\text{on}}(\text{ON } j) \text{ } m \text{ } n)}$$

The abstract syntactic type of the root label of this proof is

$$((\text{NP} \multimap \text{S}) \multimap \text{S}) \multimap (\text{NP} \multimap \text{S}) \multimap \text{S}.$$

Now the nonrestrictive relative's anchor *Kim's bike* is constructed.

(6.75)

$$\frac{\vdash (\text{NP} \multimap \text{S}) \multimap (\text{NP} \multimap \text{S}) \multimap \text{S} ; \text{KIMS}_{\text{indef}} \quad \vdash \text{NP} \multimap \text{S} ; \text{BIKE}}{\vdash (\text{NP} \multimap \text{S}) \multimap \text{S} ; (\text{KIMS}_{\text{indef}} \text{BIKE})}$$

Then the nonrestrictive relative takes its anchor and finally, the verb phrase, as argument (with some types omitted, the type of syntactic generalized quantifiers abbreviated to QP, as before, and part of the relative clause abbreviated, for space).

(6.76)

$$\frac{\begin{array}{c} \vdots \\ \vdash \text{COMMA} (\text{WHICH}_{\text{nrrc}} \dots) \end{array} \quad \begin{array}{c} \vdots \\ \vdash (\text{KIMS}_{\text{indef}} \text{BIKE}) \end{array}}{\vdash \text{QP} ; \text{COMMA} (\text{WHICH}_{\text{nrrc}} \dots) (\text{KIMS}_{\text{indef}} \text{BIKE})} \quad \vdash \text{NP} \multimap \text{S} ; \text{FAST}$$

$$\vdash \text{S} ; \text{COMMA} (\text{WHICH}_{\text{nrrc}} \dots) (\text{KIMS}_{\text{indef}} \text{BIKE}) \text{FAST}$$

So the DyCG sign modeling (6.54) is

(6.77)

$$\begin{aligned} &\vdash \text{Kim's} \cdot \text{bike} \cdot (\text{comma which} \cdot \text{has} \cdot \text{her} \cdot \text{name} \cdot \text{on} \cdot \text{it}) \cdot \text{is} \cdot \text{fast} : s ; S ; \\ &\quad \text{COMMA} (\text{WHICH}_{\text{nrrc}} \lambda_n. (\text{HER}_{\text{def}} \text{NAME})_m. \text{IT}_j. \text{HAVE}_{\text{on}} (\text{ON } j) m n) \\ &\quad (\text{KIMS}_{\text{indef}} \text{BIKE}) \text{FAST} : k \end{aligned}$$

Reducing the semantics of (6.77) shows how the anaphora between sense and implicature is modeled for (6.54) by DyCG.

$$\begin{aligned} &\vdash \text{COMMA} (\text{WHICH}_{\text{nrrc}} \lambda_n. (\text{HER}_{\text{def}} \text{NAME})_m. \text{IT}_j. \text{HAVE}_{\text{on}} (\text{ON } j) m n) \\ &\quad (\text{KIMS}_{\text{indef}} \text{BIKE}) \text{FAST} \\ &= \text{KIMS}_{\text{indef}} \text{BIKE} (\\ &\quad (\uparrow \text{WHICH}_{\text{nrrc}} \lambda_n. (\text{HER}_{\text{def}} \text{NAME})_m. \text{IT}_j. \text{HAVE}_{\text{on}} (\text{ON } j) m n) \text{ THAT FAST}) \\ &= \text{A} (\text{BIKE THAT } \lambda_j. \text{KIM}_i. \text{HAVE } j i) \\ &\quad ((\uparrow \text{WHICH}_{\text{nrrc}} \lambda_n. (\text{HER}_{\text{def}} \text{NAME})_m. \text{IT}_j. \text{HAVE}_{\text{on}} (\text{ON } j) m n) \text{ THAT FAST}) \\ &= \text{EXISTS}_n. ((\text{BIKE } n) \text{ AND } \text{KIM}_i. \text{HAVE } n i) \text{ AND} \\ &\quad (\uparrow (\text{HER}_{\text{def}} \text{NAME})_m. \text{IT}_j. \text{HAVE}_{\text{on}} (\text{ON } j) m n) \text{ AND } (\text{FAST } n)) \\ &\equiv \lambda_c \lambda_{\mathbf{x}^{[c]}, y}. (\text{bike } y) \text{ and } (\text{have } y \mathbf{x}_{(\text{the NAMED-KIM } c)}) \text{ and} \\ &\quad (\text{nonhuman } \mathbf{x}, y_{(\text{pro NONHUMAN } c'')}) \text{ and } (\text{fast } y) \mid \\ &\quad \lambda_{\mathbf{x}^{[c]}, y}. (\text{exists!}_{n: \omega_{[c]}} c \text{ k-entails } (\text{NAMED-KIM } n)) \text{ and} \\ &\quad (\text{exists!}_{n: \omega_{[c']}} c' \text{ k-entails } ((\text{NAME } n) \text{ AND } \text{SHE}_m. \text{HAVE } n m)) \text{ and} \\ &\quad (\text{exists!}_{n: \omega_{[c'']}} c'' \text{ k-cons } (\text{FEMALE } n)) \text{ and } (\text{female } \mathbf{x}, y_{(\text{pro FEMALE } c'')}) \text{ and} \\ &\quad (\text{exists!}_{n: \omega_{[c''']}} c''' \text{ k-cons } (\text{NONHUMAN } n)) \text{ and} \\ &\quad (\text{have}_{\text{on}} \mathbf{x}, y_{(\text{pro NONHUMAN } c''')} \mathbf{x}, y_{(\text{the } (\text{NAME THAT } \lambda_n. \text{SHE}_m. \text{HAVE } n m) c''')} y) : k \end{aligned}$$

The intermediate contexts c' , c'' and c''' serve to illuminate how the anaphora between the sense and implicature parts is possible. The context passed to the nonrestrictive relative is

$$\begin{aligned} c' &= \lambda_{\mathbf{x}^{[c]}, y}. (c \mathbf{x}) \text{ and } (\text{bike } y) \text{ and } (\text{have } y \mathbf{x}_{(\text{the NAMED-KIM } c)}) \text{ and} \\ &\quad \text{exists!}_{n: \omega_{[c]}}. c \text{ k-entails } (\text{NAMED-KIM } n), \end{aligned}$$

and so, assuming felicity to this point, the the anaphora associated with HER_{def} NAME and IT in the nonrestrictive relative can resolve to Kim and to the bike she has, respectively. The context passed to IT is

$$\begin{aligned} c'' = & \lambda_{\mathbf{x}|c|,y}.(c' \mathbf{x}, y) \text{ and } (\text{female } \mathbf{x}, y_{(\text{pro FEMALE } c')}) \text{ and} \\ & (\text{exists!}_{n:\omega|c'|} c' \text{ k-entails } ((\text{NAME } n) \text{ AND } \text{SHE}_m.\text{HAVE } n \ m)) \text{ and} \\ & \text{exists!}_{n:\omega|c'|} c' \text{ k-cons } (\text{FEMALE } n) , \end{aligned}$$

and similarly, the context passed to HAVE_{on} is

$$\begin{aligned} c''' = & \lambda_{\mathbf{x}|c|,y}.(c'' \mathbf{x}, y) \text{ and } (\text{nonhuman } \mathbf{x}, y_{(\text{pro NONHUMAN } c'')}) \text{ and} \\ & \text{exists!}_{n:\omega|c''|} c'' \text{ k-cons } (\text{NONHUMAN } n) , \end{aligned}$$

and so the two anaphoric antecedents required by the occurrence of *have* in the nonrestrictive relative are available, namely the nonhuman discourse referent and the discourse referent that is the name of the most salient referent whose entailments do not conflict with being female. (Here I make the assumption that the world knowledge is incorporated into the context that the antecedent of *her* has a unique name.) The nonhuman discourse referent can resolve to the bike introduced in context c' , while the antecedents for *her* and *her name* are available as long as an antecedent for *Kim* is available to start with.

Note that, in the DyCG account of (6.54), not only is the anaphora handled correctly, but the sense content is kept separate from the implicature content: the sense proposition of (6.77) contains the information that the newly introduced discourse referent is a fast bike that Kim has; the implicature proposition, on the other hand, contains the information from the nonrestrictive relative that Kim's fast bike additionally has her name on it. As a result, in a parallel with the DyCG accounts for (2.5) and (2.5') given above, the DyCG account of a negated variant of (6.54) like (6.54') would correctly yield a negated sense proposition, but the implicature proposition derived for (6.54) would survive intact.

(6.54') Kim_i's bike_j, which has her_i name on it_j, is not fast.

I also note briefly that the interaction between sense and implicature also functions the same way when an expressive is involved, using the

DyCG mechanism for linking anaphors to antecedents. For example, DyCG captures the anaphora in utterances like

(6.78) Every damn dooper_i rode his_i bike.

With all of the required lexical entries for modeling (6.78) already defined, the following DyCG semantics is available.

$$\begin{aligned}
 & \vdash (\text{EVERY DAMN DOPER})_n.(\text{HIS}_{\text{def}} \text{BIKE})_m.\text{RIDE } m \ n \\
 & = \text{NOT EXISTS } ((\text{NON NON } (\text{DAMN DOPER})) \text{ THAT} \\
 & \quad \text{NON } ((\text{DAMN DOPER}) \text{ THAT } \lambda_n.(\text{HIS}_{\text{def}} \text{BIKE})_m.\text{RIDE } m \ n)) \\
 & \equiv \lambda_c.\lambda_{x|c|}.\text{not exists}_y.(\text{doper } y) \text{ and} \\
 & \quad \text{not } ((\text{doper } y) \text{ and } (\text{ride } x_{(\text{the } (\text{BIKE THAT } \lambda_m.\text{HE}_n.\text{HAVE } m \ n) \ c')} \ y))) \mid \\
 & \quad \lambda_{x|c|}.\text{(neg DOPER) and} \\
 & \quad (\text{exists!}_{n:\omega|c'|} \ c' \text{ k-entails } (\text{BIKE THAT } \lambda_m.\text{HE}_n.\text{HAVE } m \ n)) \text{ and} \\
 & \quad \text{exists!}_{n:\omega|c''|} \ c'' \text{ k-cons } (\text{MALE } n) : k
 \end{aligned}$$

Here *Every* is given the semantics of the weak reading, defined in (5.72), and the definite possessive HIS_{def} is as defined in (5.20). The intermediate contexts are

$$c' = \lambda_{x|c|}.(c \ x) \text{ and } (\text{doper } x_n) \text{ and } (\text{neg DOPER})$$

and

$$c'' = \lambda_{x|c'|}.(c' \ x) \text{ and } (\text{bike } x_{(\text{the } (\text{BIKE THAT } \lambda_m.\text{HE}_n.\text{HAVE } m \ n) \ c')}) .$$

And so the possessive *his bike* can select the antecedent to *his*, the dooper, from the previous part of the utterance, without the expressive *damn* interfering. A different scoping is available in which the *Every* takes scope before *his bike*:

$$\vdash (\text{HIS}_{\text{def}} \text{BIKE})_m.(\text{EVERY DAMN DOPER})_n.\text{RIDE } m \ n : k$$

However, this alternative scoping does not alter either the truth conditions or the concrete syntax. Likewise, the treatment of anaphora is unchanged if the indefinite possessive $\text{HIS}_{\text{indef}}$ from (5.23) is used instead of HIS_{def} , although the truth conditions differ slightly.

In conjunction with the account given above for (6.52) and (6.54), this shows that DyCG has quite robust machinery allowing anaphoric links between sense and implicature, both when the antecedent is in the sense proposition but the anaphor is not, and vice versa. As such, it represents a great advance in empirical coverage over Potts's (2005) theory, which allows neither.

6.4 Variable Conventional Implicatures and Contextual Felicity

Supplements and expressives fit well into the generalized notion of contextual felicity developed in §5.2.2. Just as for anaphoric conventional implicatures, the notion of felicity is essentially a test for whether the content of a given utterance is in conflict with its context of interpretation. For anaphora, infelicity can arise when the associated implicature of the existence of a discourse referent bearing suitable entailments is in conflict with the context. For supplements and expressives, as I detail below, infelicity also results when the context conflicts with the implicature in question. The common thread connecting anaphora and the variable conventional implicatures, then, is that infelicity arises whenever they conflict with the context.

I first demonstrate the notion of felicity applied to a variable conventional implicature by examining the DyCG account of the discourse in

(6.79) Lance is not a cyclist. # Lance, a cyclist, is from Texas.

This discourse is a variant of the one in (2.108). The first utterance of the discourse in (6.79) is straightforwardly modeled as

$$\begin{aligned} & \vdash \text{NOT} (\text{LANCE CYCLIST}) \\ & \equiv \lambda_c. \lambda_{x|c}. \text{not} (\text{cyclist } x_{(\text{the NAMED-LANCE } c)}) \mid \\ & \quad \lambda_{x|c}. \text{exists!}_{n:\omega|c}. c \text{ k-entails } (\text{NAMED-LANCE } n) : k. \end{aligned}$$

Given that the second utterance of (6.79) is interpreted in the context that is first updated by its first utterance, it is not hard to see why we have

$$\begin{aligned} & \vdash \neg (((\text{COMMA} (\text{PRED A CYCLIST}) \text{ LANCE}) \text{ FROM-TEXAS}) \text{ felicitous-in} \\ & \quad (\text{cc NOT} (\text{LANCE CYCLIST}) \text{ T})). \end{aligned}$$

(Here the term (COMMA (PRED A CYCLIST) LANCE) FROM-TEXAS : k is the dynamic meaning derived for (2.5), the first utterance of (6.79), in the proof in (6.13).) The reason for the infelicity observed in (6.79) is simple: as shown above, the sense of the first utterance is the proposition

$$\text{not } (\text{cyclist } x_{(\text{the NAMED-LANCE } c)}) ,$$

which is in direct conflict with one of the implicatures of the second, namely the proposition

$$(\text{cyclist } x_{(\text{the NAMED-LANCE } c)}) ,$$

as the proof in (6.13) shows. Clearly the dynamic conjunction of any two contents that incorporate these conflicting propositions yields an inconsistent context, and therefore (6.79) is judged infelicitous according to the notion of contextual felicity given in definition 5.6.

The situation is similar when expressives are used infelicitously. By way of demonstrating the similarity, consider the following example.

Context (6.80): The speaker is Lance's doctor, who has administered his doping program for years and approves of Lance's doping.

(6.81) # Lance entered the Tour de France, and the damn doper won.

The reason (6.81) is infelicitous in the context (6.80) is the same as the reason for the supplement above. Assuming, based on context (6.80), that the context of interpretation for (6.81) contains at least the entailment that the speaker does not have a negative attitude towards the practice of doping, the input context to (6.81) contains the proposition

$$\text{not } (\text{neg DOPER}) .$$

However, just as for the infelicitous supplement in (6.79), the utterance of (6.81) has as part of its implicature the proposition

$$(\text{neg DOPER}) ,$$

as shown by the proof in (6.47). Thus the dynamic meaning of (6.81) is infelicitous in the input context modeling (6.80), and so definition 5.6 captures the infelicity in this case as well.

And so it seems that the claim that contextual felicity is properly defined for conventional implicatures generally is borne out. The infelicity that arises when an anaphor lacks a suitable antecedent, as discussed at length in chapter 5, is explained by the exact same formal mechanism used to explain the infelicity that can arise when a variable conventional implicature conflicts with its context of interpretation. Both are simply instances in which an associated implicature is inconsistent with some aspect of the input context, in the sense of contextual felicity formally defined in 5.6.

6.5 The Binding Problem Revisited

For Potts's (2005) account of supplements, the reason for prohibiting anaphora between sense and implicature has to do with the so-called "binding problem" discussed by Karttunen and Peters (1979, page 53) for sentences like

(6.82) Someone managed to succeed George V on the throne of England.

The problem Karttunen and Peters's account has with (6.82), as Potts (2005, §3.10) discusses, is related to the fact that it is treated as having their analog of

$$\text{exists}_x.(\text{succeed } x)$$

as its sense (Karttunen and Peters call this its *extension expression*), and

$$\text{exists}_y.(\text{hard}(\text{succeed } y) y)$$

as its implicature (Karttunen and Peters's *implicature expression*). However, there is no way, in Karttunen and Peters's theory, to state that the existentially bound variables x and y in these translations have to represent the same individual. And so (6.82) could be true in a situation in which George V's successor claimed the throne with ease, while some other person tried but failed to succeed him, which are incorrect truth conditions.

Potts (2005) adopts a strategy of making a "virtue" of this necessary problem with Karttunen and Peters's theory, by treating sense and implicature content as living on completely separate "dimensions" that are forbidden to interact. As I discuss in §6.3, Amaral et al. (2007) show why this strategy of complete separation is empirically inadequate, because in fact sense and implicature can interact quite freely. But Potts is motivated

by examples like the following, in which a quantificational noun phrase is purportedly the anchor of an infelicitous appositive:

(6.83) # No reporter_{*i*} believes that Ames, often a subject of his_{*i*} columns, is a spy.

(Potts, 2005, example 3.70)

(6.84) $\left\{ \begin{array}{l} \text{A} \\ \# \text{ Every} \end{array} \right\}$ Dutch boxer, a famous one, won the tournament.

(Nouwen, 2007, examples 15a and 16a)

DyCG, however, does not resort to the rather extreme move of completely ruling out any interaction between sense and implicature. In this section I explore the binding problem, the issues surrounding quantified supplements and expressives, and some possible solutions.

6.5.1 A Dynamic Analog of the Binding Problem?

I start by re-examining in DyCG Karttunen and Peters's original example (6.82) associated with the binding problem in order to describe what the problem is, then give an alternative analysis. By way of demonstrating how the binding problem does not manifest itself in the theory I present here, using Karttunen and Peters's analysis of *manage*, first define the dynamic properties PERSON : d₁ and SUCCEED : d₁ in the usual way:

$$\begin{aligned} \text{PERSON} &=_{\text{def}} (\text{dyn}_1 \text{ person}) \\ \text{SUCCEED} &=_{\text{def}} (\text{dyn}_1 \text{ succeed}) \end{aligned}$$

Here, person : e → p and succeed : e → p are the static properties of being a person and succeeding, respectively. Then the dynamic meaning of *managed* is MANAGE : d₁, defined as follows.

(6.85)

$$\text{MANAGE} =_{\text{def}} \lambda_{Dnc}. (Dnc)^s \mid \lambda_{x[c]}. ((Dnc)^i x) \text{ and } (\text{hard} ((Dnc)^s x) x_n)$$

As the definition shows, the sense proposition of MANAGE is just that of its complement property, while the implicature proposition inherits the implicature of its complement but also adds the information that the accomplishment required effort. Here hard : (e → p) → e → p is extensionally the relation between an entity *x* and a property *P* that says that attaining *P* is hard for *x*.

Then the dynamic meaning of *Someone* is a generalized quantifier, with the type $d_1 \rightarrow k$, defined as

$$(6.86) \quad \text{SOMEONE} =_{\text{def}} (\text{A PERSON})$$

based on the weak dynamic indefinite from equation (5.70), and the dynamic property PERSON as described above.

The following is a DyCG lexicon for modeling (6.82) based on these definitions, where the type Bse and Inf are the basic abstract syntactic types of verbs in base form and infinitive verbs, respectively.

$$\begin{aligned} &\vdash \lambda_f.f(\text{someone} \cdot s) : (s \rightarrow s) \rightarrow s ; (\text{NP} \multimap S) \multimap S ; \text{SOMEONE} \\ &\vdash \lambda_{fs}.s \cdot \text{managed} \cdot (f \mathbf{e}) : (s \rightarrow s) \rightarrow s \rightarrow s ; \text{Inf} \multimap \text{NP} \multimap S ; \text{MANAGE} \\ &\vdash \lambda_{fs}.s \cdot \text{to} \cdot (f \mathbf{e}) : (s \rightarrow s) \rightarrow s \rightarrow s ; \text{Bse} \multimap \text{Inf} ; \lambda_D.D \\ &\vdash \lambda_s.s \cdot \text{succeed} : s \rightarrow s ; \text{Bse} ; \text{SUCCEED} \end{aligned}$$

In this lexicon, the semantic types are suppressed to save space. The proof of (6.82) based on this lexicon proceeds in two parts: first the concrete syntax and then the abstract syntax and semantics.

The concrete syntactic proof starts by combining *to* with *succeed*. Here, and in the following proofs, I follow the practice observed for previous proofs by performing β - and η -reductions whenever possible.

$$(6.87) \quad \frac{\vdash \lambda_{fs}.s \cdot \text{to} \cdot (f \mathbf{e}) : (s \rightarrow s) \rightarrow s \rightarrow s \quad \vdash \lambda_s.s \cdot \text{succeed} : s \rightarrow s}{\vdash \lambda_s.s \cdot \text{to} \cdot \mathbf{e} \cdot \text{succeed} : s \rightarrow s}$$

Next, *managed* takes the derived concrete syntactic term in (6.87) as its argument. In the proof in (6.88), the type of the pheno term corresponding to *managed* is elided to save space.

$$(6.88) \quad \frac{\vdash \lambda_{fs}.s \cdot \text{managed} \cdot (f \mathbf{e}) \quad \begin{array}{c} (6.87) \\ \vdots \\ \vdash \lambda_s.s \cdot \text{to} \cdot \mathbf{e} \cdot \text{succeed} : s \rightarrow s \end{array}}{\vdash \lambda_s.s \cdot \text{managed} \cdot \mathbf{e} \cdot \text{to} \cdot \mathbf{e} \cdot \text{succeed} : s \rightarrow s}$$

Lastly, the noun phrase *someone* takes the concrete syntax from the proof in (6.88) as its argument. For space, the type of the term derived in (6.88) is

omitted.

(6.89)

$$\frac{\vdash \lambda_f.f(\text{someone} \cdot s) : (s \rightarrow s) \rightarrow s \quad \vdash \lambda_s.s \cdot \text{managed} \cdot \mathbf{e} \cdot \text{to} \cdot \mathbf{e} \cdot \text{succeed}}{\vdash \text{someone} \cdot \text{managed} \cdot \mathbf{e} \cdot \text{to} \cdot \mathbf{e} \cdot \text{succeed} : s} \quad \begin{array}{c} (6.88) \\ \vdots \end{array}$$

And so the lexicon given for (6.82) produces the correct surface form.

Turning to the abstract syntactic and semantic parts of the proof, we start again by combining *manage* with its argument *succeed*.

(6.90)

$$\frac{\vdash \text{Inf} \multimap \text{NP} \multimap \text{S} ; \text{MANAGE} \quad \frac{\vdash \text{Bse} \multimap \text{Inf} ; \lambda_D.D \quad \vdash \text{Bse} ; \text{SUCCEED}}{\vdash \text{Inf} ; \text{SUCCEED}}}{\vdash \text{NP} \multimap \text{S} ; (\text{MANAGE SUCCEED})}$$

Then the dynamic generalized quantifier *SOMEONE* takes scope.

(6.91)

$$\frac{\vdash (\text{NP} \multimap \text{S}) \multimap \text{S} ; \text{SOMEONE} \quad \vdash \text{NP} \multimap \text{S} ; (\text{MANAGE SUCCEED})}{\vdash \text{S} ; \text{SOMEONE} (\text{MANAGE SUCCEED})} \quad \begin{array}{c} (6.90) \\ \vdots \end{array}$$

Reducing the semantic term derived in (6.91) shows how this analysis of (6.82) does not replicate the binding problem.

$$\begin{aligned} & \vdash \text{SOMEONE} (\text{MANAGE SUCCEED}) \\ &= \text{A PERSON} (\text{MANAGE SUCCEED}) \\ &= \text{EXISTS} ((\text{NON NON PERSON}) \text{ THAT } (\text{PERSON THAT MANAGE SUCCEED})) \\ &\equiv \lambda_c.\lambda_{\mathbf{x}^{[c]}.y}.(\text{exists person}) \text{ and } (\text{person } y) \text{ and } (\text{succeed } y) \mid \\ &\quad \lambda_{\mathbf{x}^{[c]}.y}.(\text{hard} (\text{succeed } y) y) : k_1 \end{aligned}$$

To see why this DyCG account of (6.82) avoids recreating a dynamicized version of the binding problem, at least using an analog of Karttunen and Peters's analysis, we promote the content derived in (6.91) to an update

and then apply it to the empty context.

$$\begin{aligned} &\vdash \text{cc}(\text{SOMEONE MANAGE SUCCEED}) \text{ T} \\ &\equiv \lambda_y.(\text{exists person}) \text{ and } (\text{person } y) \text{ and } (\text{succeed } y) \text{ and} \\ &\quad (\text{hard}(\text{succeed } y) y) : c_1 \end{aligned}$$

So here we avoid giving problematic truth conditions, namely that some person y succeeded and some potentially different entity had a hard time doing so, because of the definition of the *cc* function in (5.3): promoting a content to an update ‘collapses together’ both the sense and implicature content, passing the same discourse referents to the conjoined result.

6.5.2 An Alternative Analysis

However, there are empirical reasons to doubt that Karttunen and Peters’s (1979) analysis of *manage* is the correct one. In the terminology established in chapter 2, their analysis treats *manage* as a conventional implicature: it has, as part of its conventional meaning, the implicature that whoever succeeded had a hard time doing so.

But consider the following variants of (6.82):

$$(6.82') \quad \left\{ \begin{array}{l} \# \text{ Someone} \\ \text{No one} \end{array} \right\} \text{ managed to succeed George V. No one even tried.}$$

The *Someone* variant of (6.82') gives rise to infelicity because someone’s managing to succeed George V entails that someone actually did succeed him, and this directly conflicts with entailments in the second utterance. But for the variant with *No one*, no infelicity arises. The fact that no one tried to succeed George V., and therefore that no one had a hard time, is completely compatible with no one having succeeded him.

Seen from this angle, *manage* has more in common with nonconventional implicatures like *stop* or *win* than it does with conventional implicatures. If the implication associated with *manage* were really conventional in nature, we would not expect to observe the divergent behavior in the following examples.

- (6.92) # Lance, a cyclist, didn’t enter the Tour de France because Lance isn’t even a cyclist.
- (6.93) It’s not true that someone managed to succeed George V because no one even tried.

In (6.92), the supplement *a cyclist* is a conventional implicature. Part of its conventional meaning is to contribute the proposition that Lance is a cyclist to the meaning of the entire utterance. Its conventional nature is evidenced by the infelicity that arises when the utterance it occurs within is followed up by one that directly contradicts its content. Similar evidence is given by the infelicitous example (6.79) in §6.4, above. However, for (6.93), the situation is different. For this example, no infelicity arises: the entailment that someone had a hard time is in conflict with the information that no one tried at all, and therefore the entailment does not persist.

The behavior of *manage* seems parallel to the following examples with *stop* and *win*, repeated from chapter 2, in which the trigger is embedded within an attitude predicate.

- (2.74) Kim has never smoked in her life, but Robin thinks Kim stopped smoking.
- (2.85) Lance didn't participate in the Tour de France, but Sandy believes Lance won the Tour de France.
- (6.94) No one has yet tried to take George V_i's place on the throne, but Robin thinks someone managed to succeed him_i.

For (6.94), just as for *stop* in (2.74) and *win* in (2.85), the entailment associated with *manage* does not persist because the speaker makes clear that it does not. On the basis of this similarity, and the dissimilarity with conventional implicatures demonstrated in (6.92) and (6.93), I claim that *manage* is properly classified as a nonconventional implicature that may or may not give rise to a persistent entailment.

In §7.1, I discuss how persistent entailments should be accounted for in DyCG. Unfortunately, DyCG is not without problems analogous to Karttunen and Peters's binding problem. I turn to these problems in the next section.

6.5.3 The Problem of Quantified Supplements

The DyCG treatment of supplements makes some unwanted predictions when the anchor is a quantificational noun phrase. To simplify the analysis and exposition of this problem, I first demonstrate the differing DyCG analyses for the following:

- (6.95) Lance, a dooper, raced in the Tour de France.

(6.96) # No cyclist, a doper, raced in the Tour de France.

Most of the dynamic meanings of the words in (6.95) and (6.96) are already defined. I again adopt the simplifying measure of defining the dynamic property of racing in the Tour de France as simply $\text{RACE} =_{\text{def}} (\text{dyn}_1 \text{ race})$, where $\text{race} : e \rightarrow p$ is its static counterpart.

With these definitions in place, we have

(6.97) $\vdash \text{COMMA} (\text{PRED A DOPER}) \text{ LANCE RACE} : k$

for (6.95), and for (6.96),

(6.98) $\vdash \text{COMMA} (\text{PRED A DOPER}) (\text{NO CYCLIST}) \text{ RACE} : k.$

DyCG yields a perfectly plausible treatment of (6.95), but a somewhat odd one for (6.96). To see why this is, note that (6.97) reduces as follows.

$$\begin{aligned}
 & \vdash \text{COMMA} (\text{PRED A DOPER}) \text{ LANCE RACE} \\
 &= \text{LANCE} ((\uparrow \text{PRED A DOPER}) \text{ THAT RACE}) \\
 &\equiv \lambda_c. \lambda_{x|c|.} (\text{race } x_{(\text{the NAMED-LANCE } c)}) \mid \\
 &\quad \lambda_{x|c|.} (\text{exists!}_{n:\omega|c|} c \text{ k-entails } (\text{NAMED-LANCE } n)) \text{ and} \\
 &\quad \text{exists}_y. (\text{doper } y) \text{ and } (y \text{ equals } x_{(\text{the NAMED-LANCE } c)}) : k
 \end{aligned}$$

And so for (6.95), DyCG generates the intuitively correct dynamic reading: the sense proposition is that Lance raced, while its implicatures are that Lance is a doper and that there is a unique discourse referent with the property of being named *Lance*.

The difference for (6.96) can be seen by reducing the term derived in (6.98). Here I use the version of the determiner *no* defined in (5.52), which yields the strong reading, as a simplification, since it is equivalent to the weak reading in (5.71). See §5.4.4.1 for more discussion.

$$\begin{aligned}
 & \vdash \text{COMMA} (\text{PRED A DOPER}) (\text{NO CYCLIST}) \text{ RACE} \\
 &= \text{NO CYCLIST} ((\uparrow \text{PRED A DOPER}) \text{ THAT RACE}) \\
 &= \text{NOT EXISTS}_n. (\text{CYCLIST } n) \text{ AND } ((\uparrow \text{PRED A DOPER}) n) \text{ AND } (\text{RACE } n) \\
 &\equiv \lambda_c. \lambda_{x|c|.} \text{not exists}_y. (\text{cyclist } y) \text{ and } (\text{race } y) \mid \\
 &\quad \lambda_{x|c|.} \text{exists}_{y,z}. (\text{doper } z) \text{ and } (z \text{ equals } y) : k
 \end{aligned}$$

The problem that arises for this negated variant has to do with the way dynamic negation is defined in equation (5.5). Since dynamic negation targets only discourse referents introduced in the sense, there is a new existential entailment introduced into the implicature that there is some entity that has the property of being a dooper. This analysis of (6.96) would be more in line with a hypothetical utterance with a sense corresponding to *No cyclist raced in the Tour de France* and an implicature corresponding to *Someone is a dooper*.

Note that this problem would arise for any dynamic generalized determiner that is defined based on the dynamic negation NOT that scopes over both its arguments. And so not only are quantified appositives that use the determiner *No* problematic, as in examples (6.83) and (6.96), but so are, for example, quantified appositives where the determiner is *Every*, such as the relevant variant of (6.84). So it seems that the approach of allowing interaction between sense and implicature is not without associated complications.

6.5.4 Potential Solutions

A naïve attempt to rectify DyCG's problem with quantified supplements-discussed above would simply be to redefine the dynamic generalized quantifiers that employ negation to use a function like the merge function defined in equation (6.2) that instead places all of the content in the sense proposition. We might call this the **flatten function**, and write it \Downarrow , mnemonically with its cousin, the merge function \Uparrow used in modeling supplements (see §6.1). This function would be typed as

$$\Downarrow : \Pi_{D:d_{1,i}}.d_{1,i} ,$$

for a natural number i , and it would be defined as

$$(6.99) \quad \Downarrow =_{\text{def}} \lambda_{Dnc}.\lambda_{\mathbf{x}|c|,\mathbf{y}|Dn|}.(Dnc)^s \mathbf{x}, \mathbf{y} \text{ and } (Dnc)^i \mathbf{x}, \mathbf{y} \mid \tau_{|c|+|Dn|} .$$

Clearly, the effect of flattening a dynamic property is just to place both its associated sense and implicature into the sense of the resulting dynamic property.

We could then redefine the negation-based dynamic generalized quantifiers as follows:

(6.100)

$$\text{NO} =_{\text{def}} \lambda_{DE}.\text{NOT } A \ D \ (\Downarrow E)$$

(6.101)

$$\text{EVERY} =_{\text{def}} \lambda_{DE}.\text{FORALL}_n.(\text{NOT NOT } (D \ n)) \text{ IMPLIES } ((D \ \text{THAT } (\Downarrow E)) \ n)$$

(As for the analysis of (6.96) in (6.98), this definition of NO is based on the *strong* version from (5.52) for simplicity.) Note that the definition of A is left alone, so that for A, any implicature contributed is maintained as part of the implicature, rather than as part of the sense. So already we can see that redefining the negation-based quantifiers NO and EVERY as in equations (6.100) and (6.101), respectively, represent a philosophical departure from the DyCG approach: for these quantifiers, the separation of sense and implicature is effectively neutralized, with all of the content washed into the sense proposition.

Assuming we were prepared to ignore this inconsistency in the DyCG approach to meanings, there are much worse consequences to redefining the negation-based quantifiers this way: supplements with quantificational anchors now give incorrect truth conditions. To demonstrate this, consider a reanalysis of (6.96) using a version of NO redefined as in (6.100), a reanalysis of (6.96) is derivable in DyCG, which reduces as follows.

$$\begin{aligned} & \vdash \text{COMMA } (\text{PRED } A \ \text{DOPER}) \ (\text{NO } \text{CYCLIST}) \ \text{RACE} \\ & = \text{NO } \text{CYCLIST} \ ((\Uparrow \text{PRED } A \ \text{DOPER}) \ \text{THAT } \text{RACE}) \\ & = \text{NOT EXISTS}_n.(\text{CYCLIST } n) \ \text{AND} \ (\Downarrow ((\Uparrow \text{PRED } A \ \text{DOPER}) \ \text{THAT } \text{RACE}) \ n) \\ & \equiv \lambda_c.\lambda_{x|c|}.\text{not exists}_y.(\text{cyclist } y) \ \text{and} \ (\text{race } y) \ \text{and} \\ & \quad \text{exists}_z.(\text{doper } z) \ \text{and} \ (z \ \text{equals } y) \mid T|c| : k \end{aligned}$$

With this redefinition of NO, the implicature content now gets caught up in the quantification, with the result being that (6.96) now has as its meaning the proposition that there is no doping cyclist that races.

And so the tack of reanalyzing DyCG to yield quantifiers that are based on the flatten function \Downarrow seems less than promising. It makes possible analyses of supplements with quantificational anchors, which should be ruled out, and worse, it does so at the cost of losing the distinction between sense and implicature and yielding odd truth conditions.

However, notice that there is a straightforward argument about *why* supplements with quantificational anchors are odd. In all of the following examples, repeated from above, the infelicity is related to pragmatic concerns associated with the use of the quantificational anchor.

- (6.83) # No reporter_i believes that Ames, often a subject of his_i columns, is a spy.
- (6.84) # Every Dutch boxer, a famous one, won the tournament.
- (6.96) # No cyclist, a doper, raced in the Tour de France.

For (6.83), the oddity is centered around Ames's supposedly being the subject of no reporter's columns. Given what speakers know about tournaments, (6.84) is strange because it states that every boxer one won. And the infelicity in (6.96) is similar to (6.83) in that adding the information that no cyclist is a doper is puzzling.

In view of the pragmatic oddity of these examples, we might make an appeal to a pragmatic process that rules out supplements with quantificational anchors on the basis of implausibility. Nouwen's (2007) approach to ruling out quantified supplements offers a promising way to formalize this intuition. Nouwen notes that (6.102) and (6.103) are both perfectly felicitous:

- (6.102) Every Dutch boxer took part in the event. They are all famous.
- (6.103) Every climber, all experienced adventurers, made it to the summit.
(Nouwen, 2007, examples 10 and 20)

However, (6.102) becomes infelicitous if the second utterance is replaced by # *He is famous*, and similarly, (6.103) is no longer acceptable if the appositive is replaced by # *an experienced adventurer*. On the basis of such evidence, he constructs an account in which quantified supplements are ruled out because "strong quantifiers" (such as *every* and *no*) introduce *plural* antecedents into the discourse, but any supplements in their scope expect a *singular* antecedent, as in (6.84).

An approach like Nouwen's could be implemented in DyCG, and it would render the theory incapable of deriving signs for all of (6.83), (6.84), (6.96), and similar examples in which a supplement has a quantificational anchor, as desired. Unfortunately, there are still other examples that would be insusceptible to a remedy like Nouwen's:

- (6.104) No Tibetan Buddhist_i believes that the Dalai Lama, his_i spiritual mentor, would ever bow to Chinese pressure tactics.
(Carl Pollard, personal communication)
- (6.105) Every professional man_i I polled said that while his_i wife_j, who had earned a bachelor's degree_k, nevertheless had no work experience, he_i thought she_j could use it_k to get a good job if she_j needed one.
(Amaral et al., 2007, example 35, intended reading subscripts mine)

The singular pronoun *his_i* in (6.104) seems to have no problem accessing its antecedent, the Tibetan Buddhist in the restrictor of *No*, as its antecedent. In (6.105), the nonrestrictive relative *who had earned a bachelor's degree* applies to every professional man's wife, yet its anchor is the singular *his_i wife_j*. These examples run counter to Nouwen's approach of ruling out quantified supplements when the anaphor is singular.

But notice that there is a crucial difference between the acceptable (6.104) and (6.105) and the unacceptable (6.84) and (6.96). In the unacceptable examples, the anchor of the supplement is a generalized quantifier, but for the acceptable ones, the supplement's anchor is instead a definite that is itself anteceded by a quantifier. This difference hints toward an analysis in which supplements are treated as being anaphoric to an antecedent, so that in cases like (6.84) and (6.96), the reason for the infelicity is that *Every* and *No* do not give rise to discourse referents that can later serve as the antecedents for definite anaphora. Such an analysis would also assume that examples like (6.83), which would otherwise be felicitous, are ruled out on the basis of pragmatic factors.

In the case of (6.105), part of the challenge seems to be to allow each professional man's wife's bachelor's degree to be represented in both the sense and implicature. Similarly, an account of (6.104) needs to give a representation whose sense is that no Tibetan Buddhist believes the Dalai Lama would bow to Chinese pressure, and whose implicature is that he is every Tibetan Buddhist's spiritual mentor. To be fully successful, an account would need to accomplish both of these while still disallowing the quantificational anchor in examples like (6.84) and (6.96). I leave the implementation of a solution to the problem of quantified supplements in DyCG as a question for further research.

6.6 Summary

In this chapter, I showed how the two-level extensions to the DyCG machinery for modeling the implicatures associated with anaphora from chapter 5 can also be extended to handle nonanaphoric conventional implicatures such as supplements and expressives. For supplements, the approach I use involves writing lexical entries for the comma intonation that surrounds nominal appositives, nonrestrictive relatives and parentheticals that essentially treat it as turning sense content into implicature. The lexical entries for expressives like *damn* are similar: they are treated as common noun modifiers that leave their argument's sense proposition untouched but contribute the additional implicature of a negative attitude toward the associated property.

I would argue that the approach pursued in this chapter compares favorably with other accounts of variable conventional implicatures. The work of Potts (2005), the first major formal attempt to focus on the phenomena examined here, has some very serious flaws. Probably the most serious was originally pointed out by Amaral et al. (2007): Potts's meaning "dimensions" are simply prohibited from interacting with respect to anaphora, which is very much at odds with the empirical facts, as I discuss in §6.3. The account I propose here not only allows anaphora between sense and implicature, but does so without requiring any further extensions than the mechanisms for modeling anaphora explored in chapter 5. In addition to this positive characteristic, my account of supplements also allows them to be stacked (§6.1.1.1), which a recent alternative to Potts's theory due to Nouwen (2007) does not.

One of the other attractive attributes of the account given here is that it is relatively straightforward compared with alternatives. Its two-level architecture can be seen as essentially an adaptation of Karttunen and Peters's (1979) two-level semantics to the dynamic setting, with one level for sense content and a second for implicature content. Unlike the accounts of variable conventional implicatures due to Kubota and Uegaki (2009) or Barker et al. (2010), the technique of *continuations* is not pursued, thus avoiding considerable technical complications. And unlike Nouwen's (2007) two-level dynamic semantics, no mechanism is required for managing the switch between the sense and implicature content types. For DyCG, there are a few adjustments necessary to move to a two-level semantics, but beyond those, nearly everything is handled in the lexical entries themselves.

The DyCG account of variable conventional implicatures also compares favorably with Potts’s on a technical level in addition to being more empirically adequate. The theory in Potts 2005 requires the additional complexity of a separate type hierarchy for meanings that contribute implicatures in addition to senses, then a rule of “parsetree interpretation” that traverses a meaning language representation to aggregate all of the implicature content. For DyCG, sense and implicature are computed compositionally in parallel, with each combinatoric step resulting in a pair of contexts (each parameterized by an input entity vector) capturing both types of meaning as computed up to that point.

I demonstrated in §6.4 that the generalized notion of felicity developed in chapter 5 really is general enough to capture cases of infelicity involving supplements and expressives. The cases when these variable conventional implicature meanings give rise to infelicity is the exact reason that infelicity can arise for definites: a variable conventional implicature is infelicitous when its dynamic meaning is in conflict with the context of interpretation in the sense of yielding an inconsistent context.

Lastly, §6.5 reprises Karttunen and Peters’s “binding problem,” specifically, discussing how an analog of the binding problem does not arise in DyCG even if the original analysis of *manage* were followed. A simpler analysis in which no problem arises is also provided, based on a reanalysis of *manage* as a nonconventional implicature. I then show how a related problem arises in DyCG for supplements that have a quantificational anchor. I propose some potential solutions to this problem, entertaining the adoption a variant of Nouwen’s (2007) analysis which casts “strong” quantifiers as yielding plural antecedents. An improved analysis would likely cast supplements as being anaphoric to an antecedent, although there are some unresolved issues, as I show.

Chapter 7 presents a conclusion of this thesis, evaluating its approach to compositionally and dynamically modeling meanings that incorporate both sense and implicature content. It discusses some remaining loose ends, and speculates about some possibilities for computationally implementing DyCG as presented so far.

Chapter 7

Conclusions and Future Directions

In this thesis, I have developed and formally implemented a characterization of what [Simons et al. \(2010\)](#) call *projective meaning*: aspects of the meanings of utterances that persist even when they occur within the scope of a semantic operator. This characterization, which expands on [Grice's \(1975\)](#) characterization of implicatures, allows phenomena such as anaphora, nominal appositives, and expressives, which are often treated as disparate in the semantics literature, to be classified and accounted for under a single, general mechanism.

In this way of classifying meanings, presupposition satisfaction and presupposition failure are not thought of as the primary source of infelicity. In fact, the felicity constraints usually called *presuppositions* are simply the implicatures associated with anaphora, which are merely an instance of a more general phenomenon in which an implicature gives rise to an inconsistent discourse context. The notion of felicity explored in this thesis is general enough to encompass not only anaphora, but also the variable conventional implicatures associated with appositives, nonrestrictive relatives, parentheticals, and expressives.

The formal encoding of these ideas is designed as a discourse semantics that captures the insights of [Kamp \(1981\)](#), [Heim \(1982\)](#), [Groenendijk and Stokhof \(1990, 1991\)](#), [Muskens \(1994, 1996\)](#), [Beaver \(2001\)](#), and [de Groote \(2006\)](#) while requiring the least amount of technical machinery. The approach I propose starts with a basic static semantics that has only the essentials needed to model a logic of propositions: necessarily true and

false propositions, negation, the binary operations of conjunction, disjunction, and implication, and the existential and universal quantifiers. It makes no assumptions about how (or even whether) a notion of possible worlds underlies the propositions themselves, nor about the models needed to interpret the logic of propositions.

Then a dynamic semantics is built on top of this static logic of propositions by modeling the meanings of declarative utterances as functions from contexts to contexts, where a context is simply a function from a vector of entities to a proposition, and a discourse referent is just an index into the context's domain. The upshot is that any static semantics which has a logic of propositions as a subsystem can be turned into a dynamic semantics by following this method.

The resulting dynamic semantics itself is based on three central notions from the dynamic tradition. The first has to do with the asymmetric nature of conjunction in discourse, with the second conjunct interpreted in the context obtained by integrating the first conjunct's content into the input context. The second is that the existential quantifier, used in the dynamic meaning of indefinites, has the effect of introducing a new discourse referent. And the third is that negation limits the scope of discourse referents, here by existentially binding them. By merely changing the type of the meanings of declaratives and redefining these three operators, a dynamic semantics results that can model anaphoric accessibility across arbitrary stretches of discourse, including so-called 'donkey anaphora.'

In order to formally encode the empirical characterization of senses and implicatures I propose, the dynamic semantics is next extended so that declaratives are treated as functions from a context into a pair of contexts, rather than a single context. One of these propositions bears the sense content of an expression, and the other bears any implicatures it gives rise to. On a conceptual level, this approach can be thought of as a blending of the two-level semantics proposed by [Karttunen and Peters \(1979\)](#) with the branch of dynamic semantics most closely associated with [Heim \(1982, 1983a, 1992\)](#) and [Beaver \(2001\)](#).

There is also a sense in which the two-level dynamic semantics I investigate here is a generalization of [Potts's \(2005\)](#) logic of conventional implicatures. However, rather than keeping sense and implicature content completely separate and then aggregating the two levels together after an utterance has been fully derived, my formal theory allows sense and implicature to interact at every compositional step. As [Amaral et al. \(2007\)](#) point

out, this is more in line with the empirical facts related to implicatures. The framework I propose also allows a formal account of both anaphora and the nonanaphoric conventional implicatures using the same basic mechanisms for managing sense and implicature content.

I argue that the two-level dynamic semantics explored in this thesis compares favorably with other attempts to model sense and implicature in discourse, including not only the classic dynamic theories due to [Kamp \(1981\)](#), [Heim \(1982\)](#), and [Groenendijk and Stokhof \(1990, 1991\)](#), but also more recent accounts such as those due to [Potts \(2005\)](#), [Nouwen \(2007\)](#), [Kubota and Uegaki \(2009\)](#), and [Barker et al. \(2010\)](#). In this chapter, I offer some potential future directions in which the theory presented in this thesis might be elaborated and implemented.

Specifically, §7.1 points toward a DyCG account of nonconventional implicatures like those associated with factives, achievements, and aspectuals, which can give rise to persistent entailments. I briefly sketch a way that the dynamic semantics developed in this thesis might be extended to model the anchoring of implicatures to different points of view in §7.2, and in §7.3, I speculate on some issues that would be involved in any computational implementation of my formal theory of discourse. Finally, §7.4 sums up.

7.1 Persistent Entailments in DyCG

In chapters 5 and 6, I discuss at length a detailed DyCG account of conventional implicatures. The reason there is no chapter dealing with the nonconventional implicatures discussed in §2.3 is that a DyCG account of these implicatures is simply not very interesting. The nonconventional implicatures triggered by the use of factives, achievements, and aspectuals are not part of their conventional meanings, but rather result from entailments that may persist beyond the scope of enclosing operators.

As a very simple example, I show how a DyCG account of the achievement *win* might go, for

(7.1) Lance won the Tour de France.

Since the dynamic meanings of *Lance* and *win* have already been given in equations (5.7) and (6.40), respectively, we only need a DyCG semantics for the noun phrase *the Tour de France*. For simplicity, I give a lexical entry

corresponding to this noun phrase as follows:

$$\vdash \lambda_f.f(\text{the} \cdot \text{Tour} \cdot \text{de} \cdot \text{France}) ; (\text{NP} \multimap \text{S}) \multimap \text{S} ; \text{THE TOUR-DE-FRANCE}$$

Here, $\text{TOUR-DE-FRANCE} =_{\text{def}} (\text{dyn}_1 \text{tour-de-france})$ is the dynamic property of being the Tour de France, with $\text{tour-de-france} : e \rightarrow p$ its corresponding static version.

With this extension to the lexicon, DyCG derives the correct surface string corresponding to (7.1), namely

$$\vdash \text{Lance} \cdot \text{won} \cdot \text{the} \cdot \text{Tour} \cdot \text{de} \cdot \text{France} : s,$$

and the derived sign's abstract syntactic category is S, as expected. A DyCG semantics for (7.1) can be given as shown below.

$$\begin{aligned} & \vdash \text{LANCE}_n \cdot (\text{THE TOUR-DE-FRANCE})_m \cdot \text{WIN } m \ n \\ & = (\text{THE NAMED-LANCE})_n \cdot (\text{THE TOUR-DE-FRANCE})_m \cdot \text{WIN } m \ n \\ & \equiv \lambda_c \lambda_{\mathbf{x}|c}. (\text{win } \mathbf{x}_{(\text{the TOUR-DE-FRANCE } c)} \ \mathbf{x}_{(\text{the NAMED-LANCE } c)}) \mid \\ & \quad \lambda_{\mathbf{x}|c}. (\text{exists!}_{n:\omega|c} c \text{ k-entails } (\text{NAMED-LANCE } n)) \text{ and} \\ & \quad \text{exists!}_{n:\omega|c} c \text{ k-entails } (\text{TOUR-DE-FRANCE } n) : k \end{aligned}$$

(A different semantic scoping can also be given for (7.1) in which *the Tour de France* outscopes *Lance*, but the results are equivalent.) This semantics is unremarkable: ignoring the implicatures associated with the definite anaphora, the sense proposition simply states that Lance won the Tour de France, as desired for (7.1). The preparatory phase associated with *win* that Lance entered the Tour is just an entailment.

If this DyCG semantics for (7.1) were adapted for a variant in which *win* were outscoped by an operator, the entailment of *win*'s preparatory phase would not necessarily persist. But the preparatory phase could still be inferred, for example, in a context where the interlocutors were discussing which Tours de France Lance participated in, illustrated below.

Context (7.2): The interlocutors are discussing which of the Tours de France Lance is known to have entered.

(7.1') Lance didn't win the Tour de France in 1996.

For (7.1') uttered in a context like (7.2), the addressee might infer that the speaker intended for the 1996 Tour to count as one of the ones Lance entered.

But it seems to me that accounting for the likelihood of an addressee making such an inference is heavily influenced by the pragmatics of the situation and the discourse context, and as such, I leave the task of accounting for it outside the formal semantic theory.

7.2 Towards an Account of Anchoring

Ideally, a semantic theory for implicatures should have a way to represent anchoring to a point of view since, as chapter 2 discusses, point of view anchoring is among the criteria that distinguish implicatures. To recall an implicature that can take on a point of view different from the speaker's, consider

(2.74) Kim has never smoked in her life, but Robin thinks Kim stopped smoking.

In (2.74), it cannot be inferred that the speaker believes Kim used to smoke, the pre-state implication associated with *stopped*, because the speaker explicitly denies it. Instead, the addressee must infer that Kim believes Kim used to smoke. Compare this with

(2.10) Kim stopped smoking.

Depending on the context in which (2.10) is uttered, the addressee may infer that the speaker believes the pre-state implication of *stopped* that Kim used to smoke. This contrasts with (2.74), where such an inference is only possible if attributed to Kim. For the following example, the addressee may infer that the speaker believes Kim used to smoke, depending on the prior context:

(7.3) Robin thinks Kim stopped smoking.

However, note that the addressee for (7.3) *must* infer that Robin thinks Kim used to smoke—the pre-state is an entailment of *stopped* when it is not embedded beneath an operator.

Clearly, what determines which inference the addressee makes in the less constrained (7.3) are sensitive to context and other pragmatic factors, some of which may even be nonlinguistic. But the importance of these factors do not necessarily preclude the semantic theory from providing a representation of anchoring to a specific point of view, even if the theory is not able to say exactly when and how point of view gets shifted for every

case. In this section, I sketch a way that certain aspects of point of view anchoring could be implemented in the DyCG semantic theory.

To start with, the type c of discourse contexts would need to be elaborated so that point of view information could be tracked for each proposition contributed. Both the sense and implicature propositions would need to be more finely grained, as (characteristic functions of) sets of pairs of a discourse referent and a proposition. That is, the type p of propositions from the definition of c in equation (4.2) would be replaced by the type

$$(7.4) \quad P =_{\text{def}} (\omega \times p) \rightarrow t.$$

Then the type of contexts would be redefined as

$$(7.5) \quad c_n =_{\text{def}} e^n \rightarrow P,$$

so that contexts are functions from a vector of entities, as before, but the single proposition is replaced by a set of pairs of a discourse referent and a proposition. The discourse referent component represents the associated point of view, and the proposition the meaning contribution, as before.

The contributions made by dynamic meanings would need to change to pairs of the form $\langle n, p \rangle$, where n is a natural number among the discourse referents in the input vector and p is the proposition anchored to the point of view associated with n . In view of this, contexts would need to maintain a distinguished discourse referent as the currently represented point of view. To access this information, a suitably axiomatized function pov , with the type

$$\text{pov} : \Pi_{c:c}.\omega_{|c|},$$

could be defined to provide the current point of view for a given context. A ‘default’ point of view, associated with the speaker, would then need to be stipulated, for example, the discourse referent 0, although this would have the implication that no context could ever have arity 0. But with pov defined, each contribution to the sense or implicature part of a dynamic meaning would be of the form $\langle (\text{pov } c), p \rangle$, where c is the current context and p is the proposition being contributed.

For variable conventional implicatures like appositives, parentheticals, nonrestrictive relatives, and expressives, for which point of view anchoring can be altered by contextual factors, the function pov would serve as a ‘hook’ for discourse pragmatics to influence how dynamic meanings are

represented. But for the nonconventional implicatures (factives, aspectuals, achievements), things could be made more concrete. For example, the meanings of attitude predicates such as *think*, *believe*, etc., could be defined to give them the ability to modify the value returned by *pov* for the current context. This way, for (2.74), the dynamic meaning of *thinks* could inspect the context to find that the pre-state implication of *stopped smoking* is in conflict with the speaker's stated beliefs, and as a result change the point of view to the subject of *thinks*, namely Robin. This approach would also require a redefinition of the notions of consistency and felicity from definitions 5.5 and 5.6 to take into account the point of view information.

I did not fully pursue a model of anchoring in this thesis because there is already so much to say about the conventional implicatures. The topic of point of view opens the door to so much of indexicality and the pragmatics of beliefs and belief reports as to put anchoring beyond the scope of my analysis. But I think the sketch I offer above holds promise for implementing in a framework like DyCG the parts of point of view anchoring that belong in a compositional semantics.

7.3 Computational Considerations

The choice to encode DyCG in type theory has many desirable consequences. From a purely formal standpoint, it is reassuring that DyCG is grounded in a mainstream area of mathematics that has been the object of a great deal of study for more than 75 years, and about which much is understood. From the point of view of linguistic methodology, type theory provides a straightforward way to translate from a syntax, here tensor-implication logic, to a compositional semantic representation. Moreover, as I discuss in chapter 4, the additional expressive power of dependent type theory provides a built-in mechanism for tracking information about contexts such as the number of available discourse referents, and thus is well suited to implementing a compositional dynamic semantics.

This section is concerned with the positive attributes of DyCG from still another vantage point: its prospects for computational implementation. An implementation of a compositional dynamic theory could have impacts on natural language processing tasks as diverse as dialog systems, text mining, document summarization, and machine translation. Such an implementation already exists in the form of the implementation of [van der](#)

Sandt's (1992) and Geurts's (1999) anaphoric theory of presupposition due to Blackburn and Bos (1999) and Bos (2003, 2005) in ProLog, which is built upon a computational rendering of Kamp's (1981) Discourse Representation Theory.

However, this approach is in some ways incompatible with the theory I have explored in this thesis. One incompatibility is that van der Sandt's approach is too simplistic, theorizing that all implicature is presuppositional, and moreover, that all presupposition behaves exactly as does anaphora. Although Venhuizen et al. 2013 shows some promise in terms of generalizing an implementation of the approach advocated by van der Sandt and Geurts, it is unclear how to reconcile their approach with the empirical taxonomy of implicatures I give in chapter 2, in which only the obligatorily speaker-anchored conventional implicatures behave like anaphora, but other kinds of implicatures show different behavior.

Another incompatibility between DyCG and the approach of van der Sandt and Geurts is the central role that accommodation plays in their theory. In their scheme, any unsatisfied presupposition is accommodated by repairing the context to contain the required information. DyCG has no such high-powered accommodation mechanism, which, as I argue in chapter 2, is too unconstrained. Lastly, the van der Sandt approach is incompatible with DyCG in that it allows for multiple resolution sites, whereas in DyCG there is only ever a single input context. As I discuss in §5.5, the criticism of van der Sandt's theory due to Beaver (2001, 2002) is compelling, and moreover, the characterization of aspectuals, achievements, and factives as nonconventional that I develop in chapter 2 implies that resolution and accommodation are simply unnecessary for these classes of implicatures.

Van Eijck and Unger (2010, chapter 12) provide a Haskell implementation of a dynamic semantics that very closely resembles the single-level version of DyCG presented in chapter 4. Due to its many similarities with DyCG, the success of their implementation implies that an implementation of DyCG would not be too difficult. The following list highlights some of the ways van Eijck and Unger's implementation resembles DyCG:

- Discourse referents are modeled as natural numbers using the Haskell type `Int`.

- The dynamic meanings of declaratives are modeled as functions from lists of entities to propositions, with discourse referents indices into the domain.
- The dynamic existential quantifier is implemented based on a very close analog to the extension function $(\cdot)^+$ from definition 4.7, via the function `extends`.
- Dynamic conjunction and negation, as well as the dynamic existential, are basically straightforward Haskell implementations of the definitions of `AND`, `NOT`, and `EXISTS` from chapter 4.
- The dynamicizer functions `dynn` from definition 4.4 are implemented for intransitives, di- and tritransitives via the functions `blowUpPred`, `blowUpPred2` and `blowUpPred3`, respectively.

In §12.7, [van Eijck and Unger](#) even provide elaborations to their core implementation for resolving anaphora, with routines for checking the descriptive content of definites that are not incompatible with the theory of anaphora I lay out in chapter 5.

There are some prominent differences between DyCG’s semantics and [van Eijck and Unger](#)’s implementation, such as the type system and the ways the context is updated. But these differences do not seem insurmountable, and I speculate that their software could be extended to the two-level model of semantics explored for DyCG in chapters 5 and 6 of this thesis without too much difficulty. It also seems entirely possible to port [van Eijck and Unger](#)’s software to another programming language or even to another programming paradigm, for example, to the object-oriented language Java, or to a blended language like Scala.

More work would be required to implement the part of a DyCG system that allowed signs to be derived. Both the DRT-based implementation of [Blackburn and Bos \(1999\)](#) and [Bos \(2003, 2005\)](#) and the Haskell dynamic semantics of [van Eijck and Unger \(2010\)](#) use a basic context-free grammar as their syntax, and so their approach to syntax is not readily adaptable to handling syntax in the DyCG setting.

The work of [Hepple \(1996, 1999\)](#) on parsing natural deduction proofs in (multiplicative) linear logic may prove insightful for the task of parsing text to yield DyCG signs, at least at the level of abstract syntax. [Hepple](#)’s method seems adaptable to DyCG parsing because most of the moves he

makes to simplify the task of parsing linear deductions are quite acceptable in the DyCG setting. In fact, one of [Hepple's \(1996\)](#) simplifications, that a variable name can be used at most once, is already required by the DyCG grammar rules (see definition 3.2).

Some subtleties would be involved in implementing [Hepple's](#) ideas, as his chart parsing method works on labeled spans in the input string, and the corresponding DyCG pheno terms are unreduced lambda terms that may have undischarged hypotheses accompanying them. One approach may be to perform β -reduction on the pheno terms when a derived term has the type *s*, then compare it against the corresponding span in the string. This approach could potentially integrate some insights from [König's \(1994\)](#) chart-parsing algorithm for Lambek grammars with hypothetical proof. As for the task of reducing the lambda terms derived for the concrete syntax and the semantics in parallel with the abstract syntactic parse, a well-known algorithm for lambda calculus reduction could be used, such as the one described in [Lamping 1990](#).

7.4 In Sum

The empirical characterization of implicatures I have offered in this thesis, which extends [Grice's \(1975\)](#) taxonomy of implicatures based in part on insights from [Simons et al. \(2010\)](#), represents a different take on persistent meanings that allows anaphora, what are sometimes called *presuppositions*, and the phenomena discussed by [Potts \(2005\)](#) to be unified under the category of conventional implicature. In this way, it fills in a gap in [Grice's](#) taxonomy: [Grice](#) mentioned conventional implicature as a category, but said very little else about the members of this category, as his primary purpose was to characterize conversational implicatures, a category I have left alone.

The taxonomy I discuss in chapter 2 adds a criterion for distinguishing implicatures in addition to [Grice's](#) conventional/nonconventional distinction: the criterion of anchoring to a speaker or nonspeaker point of view. With this cross-cutting distinction, anaphora can be seen as the class of conventional implicatures that must be anchored to the speaker, and [Potts's \(2005\)](#) "CIs" as the conventional implicatures that can possibly have a nonspeaker anchoring. Certain other lexical items, such as factives, aspectuals, and achievements, which are sometimes characterized as presupposition

triggers, are revealed to belong to a class different from anaphora, namely the nonconventional implicatures. These lexical items may sometimes give rise to persistent entailments, but their associated implicatures are highly sensitive to context because they are not part of the conventional meanings of the words themselves. As a result, the term *presupposition*, used to apply to triggers other than anaphora, has no distinguished place in this taxonomy—*anaphora* and *presupposition* are synonyms.

Starting in chapter 3, I then define a new theory of grammar with a two-part syntax which divides the work between a component that models combinatorics (*abstract syntax* or *tecto*) and one that models word order (*concrete syntax* or *pheno*). I then show, in chapter 4, how to take a generic static propositional semantics and dynamicize it into a theory of discourse in the tradition of Karttunen (1974), Heim (1982), Muskens (1994, 1996), Beaver (2001) and de Groote (2006). The resulting semantics is both compositional in the sense of Montague 1973 and dynamic, with utterances modeled as functions from discourse contexts to discourse contexts. It is also quite flexible, as I show in chapters 5 and 6, in that with some simple extensions, it can be turned into a formally explicit encoding of the taxonomy of conventional implicatures from chapter 2. The formal account of conventional implicatures presented here is robust and detailed, and compares favorably with its competitors. Along the way I revise and critique some notions central to Potts's (2005) theory of conventional implicatures, casting anaphora, supplements, and expressives as members of a more general class, and calling into question Potts's idea that no lexical item can contribute to both the sense and the implicature of an expression.

It is my hope that the groundwork I have laid in this thesis will prove useful to future research. As I discuss in §7.3, the promise of computational implementation for the formal theory given here is strong. But it also leaves open several avenues of further research on a purely theoretical level. For one, as I discuss in §7.2, the semantics discussed here could also be extended to an account of point of view anchoring, as part of a larger treatment of indexicality and belief attribution. Another loose end is that an in-depth investigation of the proper treatment of supplements with quantificational anchors is needed (see §6.5.3). Finally, there are questions that have not even been raised in this thesis that the formal theory I develop could be brought to bear on, for example, what predictions a formal implementation of Roberts's (1996) *question under discussion* would make about which entailments persist and what is or is not targeted by

semantic operators. I aim to explore these and many other questions in the future, and I project that the semantics given here will prove a very useful starting point for taking them on.

Appendix A

Tensor-Implication Logic

The tensor-implication fragment of intuitionistic linear logic (Girard, 1987; Troelstra, 1992; Girard, 1995) is a subsystem that uses only the multiplicative connectives \otimes (tensor product) and \multimap (linear implication), along with the unit 1 . It does not make use of the quantifiers, the exponentials, any of the additive connectives, or the multiplicative disjunction.

This fragment is discussed by Girard, Scedrov, and Scott (1992) as *rudimentary linear logic* and by Hyland and de Paiva (1993) as *tensor-implication logic*, a term I adopt. In this appendix, I give the syntax and proof theory for tensor-implication logic (§A.1 and §A.1.1), as well as an algebraic semantics based on de Paiva’s (2002) *lineales* (§A.2).

A.1 Syntax

Definition A.1 (Formulas). There is a set of atomic formulas, and the nullary connective 1 is a formula. If A and B are formulas, then so are $(A \otimes B)$ and $(A \multimap B)$. Nothing else is a formula.

Outer parentheses surrounding formulas are dropped, so that, for example, $(A \otimes B)$ is written simply $A \otimes B$. The binary connective \multimap is written right-associatively, with $A \multimap B \multimap C$ abbreviating $A \multimap (B \multimap C)$, but \otimes associates to the left, with $A \otimes B \otimes C$ shorthand for $(A \otimes B) \otimes C$.

Definition A.2 (Contexts). A *context* is a finite multiset of formulas. For contexts Γ and Δ , the context $\Gamma \uplus \Delta$ is the multiset union of Γ and Δ . The empty multiset is written \emptyset .

$$\begin{array}{c}
A \vdash A \quad (\text{Id}) \\
\\
\frac{\Gamma, A \vdash B}{\Gamma \vdash A \multimap B} (\multimap\text{I}) \qquad \frac{\Gamma \vdash A \multimap B \quad \Delta \vdash A}{\Gamma, \Delta \vdash B} (\multimap\text{E}) \\
\\
\frac{\Gamma \vdash A \quad \Delta \vdash B}{\Gamma, \Delta \vdash A \otimes B} (\otimes\text{I}) \qquad \frac{\Gamma \vdash A \otimes B \quad \Delta, A, B \vdash C}{\Gamma, \Delta \vdash C} (\otimes\text{E}) \\
\\
\vdash \mathbf{1} \quad (\mathbf{1}\text{I}) \qquad \frac{\Gamma \vdash \mathbf{1} \quad \Delta \vdash A}{\Gamma, \Delta \vdash A} (\mathbf{1}\text{E})
\end{array}$$

Figure A.1: Inference rule schemas for tensor-implication logic, where A and B are arbitrary formulas and Γ and Δ multisets of formulas.

Provability is expressed by sequents of the form $\Gamma \vdash A$, where Γ is a context and A is a single formula. Sequents of the form $\emptyset \vdash A$ are abbreviated as simply $\vdash A$. The shorthand Γ, Δ abbreviates the multiset union $\Gamma \uplus \Delta$. Lastly, Γ, A is further shorthand for $\Gamma, \{A\}$.

A.1.1 Proof Theory

Figure A.1 diagrams the rules for tensor-implication logic, which are based on the natural deduction rules found in [Troelstra 1992](#), chapter 6. [Girard et al. \(1992\)](#) and [Hyland and de Paiva \(1993\)](#) give an equivalent presentation using Gentzen sequents.

Natural deduction for tensor-implication logic is simple, with rule schemas for identity (Id), for introducing and eliminating implications ($\multimap\text{I}$ and $\multimap\text{E}$) and products ($\otimes\text{I}$ and $\otimes\text{E}$). The rule $\multimap\text{E}$ is a straightforward statement of *modus ponens*: that A and $A \multimap B$ together imply B . The rule $\multimap\text{I}$ captures its inverse. The introduction rule $\otimes\text{I}$ conjoins two formulas in a resource-sensitive way, since eliminating a product of the form $A \otimes B$ requires a proof of both A and B , as $\otimes\text{E}$ makes clear. The schemas pertaining to the connective $\mathbf{1}$ essentially state that it is a unit with respect to \otimes . Using $\mathbf{1}\text{I}$, an instance of the unit $\mathbf{1}$ can be introduced without its introduction being ‘remembered,’ and it can be eliminated via $\mathbf{1}\text{E}$ with no effect on the rest of the proof’s structure.

Definition A.3 (Proof). A *proof* of the sequent $\Gamma \vdash A$ consists of a natural deduction tree labeled with sequents, where

1. The root node's label is $\Gamma \vdash A$,
2. Every leaf is labeled with an instantiation of the Id rule, and
3. Every mother node's label is licensed by an instantiation of one of the rules in figure A.1 with its daughters.

Each instantiation of a deduction rule can optionally be labeled with the name of the corresponding rule for greater clarity.

Example A.4. An example tensor-implication proof is given in below, for A , B , and C metavariables over formulas and Γ abbreviating $A \multimap (B \multimap C)$ for readability.

$$\begin{array}{c}
 \frac{A \otimes B \vdash A \otimes B}{\frac{A \otimes B, A \multimap (B \multimap C) \vdash C}{A \multimap (B \multimap C) \vdash (A \otimes B) \multimap C} (\multimap I)} \\
 \frac{\frac{\frac{\Gamma \vdash A \multimap (B \multimap C) \quad A \vdash A}{\Gamma, A \vdash B \multimap C} (\multimap E) \quad B \vdash B}{\Gamma, A, B \vdash C} (\multimap E)}{A \otimes B, A \multimap (B \multimap C) \vdash C} (\otimes E)
 \end{array}$$

Importantly, although the structural rule of *permutation* (also known as *exchange*) is implicit in the choice to model contexts as multisets, the structural rules of *weakening* and *contraction* are not available. In a proof of a sequent of the form $\vdash A$, this implies that every formula introduced into the context by an instance of Id must be ‘used’ exactly once.

A.2 Algebraic Semantics

Several different options for modeling linear logic are available, and an interpretation of tensor-implication logic can be embedded into any of them. Girard (1987) originally proposed *coherent spaces* as models for linear logic (called *phase spaces* in Girard 1995), and Troelstra (1992) gives both generalized algebraic models and alternative categorical models for it. Yetter (1990) provides an interpretation of linear logic using *quantales*.

Here, I use a class of models called *lineales*, due to de Paiva (2002). A lineale is a specialization of a symmetric monoidal closed category to the case of ordered algebras. Recall that an ordered monoid is a set M together

with an order \sqsubseteq on M , an associative binary operation \circ on M that is monotonic in both arguments, and a two-sided identity 1 for \circ . An ordered monoid is written as a quadruple $\langle M, \sqsubseteq, \circ, 1 \rangle$.

Let $\langle M, \sqsubseteq, \circ, 1 \rangle$ be an ordered monoid. If $a \circ b = b \circ a$ for every $a, b \in M$, then $\langle M, \sqsubseteq, \circ, 1 \rangle$ is called a *commutative monoid*. If the set

$$\{m \in M \mid a \circ m \sqsubseteq b\}$$

has a greatest member, it is called the *relative pseudocomplement of a with respect to b* , and is written $a \multimap b$.

Definition A.5 (Lineale). A *lineale* is a commutative monoid such that the relative pseudocomplement exists for all $a, b \in M$. We denote a lineale by a quintuple $\langle M, \sqsubseteq, \circ, 1, \multimap \rangle$, where $\langle M, \sqsubseteq, \circ, 1 \rangle$ is a commutative monoid.

Residuation is an important aspect of the relationship between \circ and \multimap in a lineale.

Lemma A.6 (Residuation). *In a lineale, for all $a, b, c \in M$ we have $a \circ b \sqsubseteq c$ if and only if $a \sqsubseteq b \multimap c$.*

Proof. Let $\langle M, \sqsubseteq, \circ, 1, \multimap \rangle$ be a lineale with $a, b, c \in M$. Suppose that $a \circ b \sqsubseteq c$. Then we have $b \circ a \sqsubseteq c$ by commutativity, and since $b \multimap c$ is greatest in $\{m \in M \mid b \circ m \sqsubseteq c\}$, it immediately follows that $a \sqsubseteq b \multimap c$.

Conversely, suppose that $a \sqsubseteq b \multimap c$. Note that we have $b \circ (b \multimap c) \sqsubseteq c$ by definition. Then, by monotonicity, we have $b \circ a \sqsubseteq b \circ (b \multimap c)$, and therefore by transitivity $b \circ a \sqsubseteq c$. Then $a \circ b \sqsubseteq c$ by commutativity. \square

Because of the property in lemma A.6, the relative pseudocomplement operation \multimap is called the *residual* of \circ .

Models for tensor-implication logic are defined following Troelstra's (1992) *IL-models*.

Definition A.7 (TIL Model). A model of tensor-implication logic, or *TIL model*, is a lineale $\langle M, \sqsubseteq, \circ, 1, \multimap \rangle$ together with a mapping I that interprets formulas, so that $I(A) \in M$ for every atomic A . The interpretation mapping is extended to the connectives as follows:

$$I(1) = 1,$$

and for all formulas A, B ,

$$\begin{aligned} I(A \multimap B) &= I(A) \multimap I(B), \text{ and} \\ I(A \otimes B) &= I(A) \circ I(B). \end{aligned}$$

Multisets of formulas are interpreted as being concatenated by \circ , with the empty multiset's interpretation defined to be

$$I(\emptyset) = 1,$$

and for Γ, Δ a multiset of formulas,

$$I(\Gamma, \Delta) = I(\Gamma) \circ I(\Delta).$$

For a given TIL model, the sequent $\Gamma \vdash A$ is *valid* if and only if

$$I(\Gamma) \sqsubseteq I(A).$$

If Γ is empty, we say simply that A is valid. We can now show that every provable sequent is valid.

Theorem A.8 (Soundness). *If the sequent $\Gamma \vdash A$ is provable, then $\Gamma \vdash A$ is valid in every TIL model.*

Proof. The proof is straightforward but a bit tedious. By induction on the structure of TIL proofs, we assume $\Gamma \vdash A$ and show that this implies $I(\Gamma) \sqsubseteq I(A)$ for a given TIL model. We consider each deduction rule by cases based on which rule was last used in the proof. First, note that the Ax schema $A \vdash A$ is valid for any formula A since $I(A) \sqsubseteq I(A)$ by reflexivity.

In the case of the \multimap I rule, validity of $I(\Gamma) \sqsubseteq I(A \multimap B)$ follows immediately from $I(\Gamma, A) \sqsubseteq I(B)$ because \multimap is a residual of \circ by the [Residuation](#) lemma [A.6](#). For \multimap E, we need to show that $I(\Gamma, \Delta) \sqsubseteq I(B)$ follows from assuming $I(\Gamma) \sqsubseteq I(A) \multimap I(B)$ and $I(\Delta) \sqsubseteq I(A)$. We start by showing that the inductive hypothesis corresponding to the major premise

of \multimap E implies that $I(A) \sqsubseteq I(\Gamma) \multimap I(B)$:

$$\begin{aligned}
 I(\Gamma) &\sqsubseteq I(A) \multimap I(B) \\
 I(\Gamma) \circ I(A) &\sqsubseteq I(B) && \text{(by residuation)} \\
 I(A) \circ I(\Gamma) &\sqsubseteq I(B) && \text{(by commutativity)} \\
 I(A) &\sqsubseteq I(\Gamma) \multimap I(B) && \text{(by residuation)}
 \end{aligned}$$

Then by the assumption that $I(\Delta) \sqsubseteq I(A)$, we have:

$$\begin{aligned}
 I(\Delta) &\sqsubseteq I(\Gamma) \multimap I(B) && \text{(by transitivity)} \\
 I(\Delta) \circ I(\Gamma) &\sqsubseteq I(B) && \text{(by residuation)} \\
 I(\Gamma) \circ I(\Delta) &\sqsubseteq I(B) && \text{(by commutativity)} \\
 I(\Gamma, \Delta) &\sqsubseteq I(B)
 \end{aligned}$$

So $\Gamma, \Delta \vdash B$ is valid whenever $\Gamma \vdash A \multimap B$ and $\Delta \vdash A$ are valid.

The case of the \otimes I rule immediately follows from the monotonicity of \circ and the transitivity of \sqsubseteq . For the \otimes E rule, we need to show that $I(\Gamma) \sqsubseteq I(A \otimes B)$ and $I(\Delta, A, B) \sqsubseteq I(C)$ together imply that $I(\Gamma, \Delta) \sqsubseteq I(C)$. Note that we have

$$\begin{aligned}
 I(\Delta) \circ I(A) \circ I(B) &\sqsubseteq I(C) \\
 I(A) \circ I(B) \circ I(\Delta) &\sqsubseteq I(C) && \text{(by commutativity)} \\
 I(A) \circ I(B) &\sqsubseteq I(\Delta) \multimap I(C) && \text{(by residuation)}
 \end{aligned}$$

and since $I(\Gamma) \sqsubseteq I(A) \circ I(B)$, we have $I(\Gamma) \sqsubseteq I(\Delta) \multimap I(C)$ by transitivity, and $I(\Gamma) \circ I(\Delta) \sqsubseteq I(C)$ immediately follows.

The two cases of rules involving **1** are completely straightforward by the monotonicity of \circ , the properties of the ordering relation \sqsubseteq , and the identity property of the monoidal unit **1**. \square

To show the converse of theorem A.8, that every valid sequent is provable, we first need to construct a lineale. The method used here uses a *Lindenbaum algebra* based on equivalence classes of formulas.

Define the equivalence relation \sim on the set \mathcal{F} of all formulas in terms of mutual implication, so that for all $A, B \in \mathcal{F}$,

$$A \sim B \quad \text{iff} \quad \vdash A \multimap B \quad \text{and} \quad \vdash B \multimap A.$$

It is easy to see that \sim is an equivalence relation on sets of formulas. For any formula A , the corresponding equivalence class is

$$[A]_{\sim} = \{B \mid A \sim B\},$$

and is written simply $[A]$ by convention, suppressing the subscript.

The relation \sqsubseteq is then defined so that $[A] \sqsubseteq [B]$ if and only if $\vdash A \multimap B$. This relation induces the following equivalence on \mathcal{F} / \sim , the set of equivalence classes of formulas: $[A] = [B]$ if and only if $[A] \sqsubseteq [B]$ and $[B] \sqsubseteq [A]$. That is, \sqsubseteq is an order on \mathcal{F} / \sim .

Proposition A.9. *Let \mathcal{F} / \sim and \sqsubseteq be defined as above, and for all formulas A and B , define*

$$\begin{aligned} [A] \circ [B] &= [A \otimes B], \text{ and} \\ [A] \multimap [B] &= [A \multimap B]. \end{aligned}$$

Then $\langle \mathcal{F} / \sim, \sqsubseteq, \circ, \{\mathbf{1}\}, \multimap \rangle$ is a lineale.

Proof. Clearly \circ and \multimap are well defined. To see that $\{\mathbf{1}\}$ is an identity for \circ , note that $[\mathbf{1}] = \{\mathbf{1}\}$ and for all $A \in \mathcal{F}$ it is easy to check that

$$[\mathbf{1} \otimes A] = [A] = [A \otimes \mathbf{1}].$$

One part of this equivalence follows from proofs of $\vdash (A \otimes \mathbf{1}) \multimap A$ and $\vdash A \multimap (A \otimes \mathbf{1})$; the other part is demonstrated in a similar way. The associativity and commutativity of \circ are also readily verified.

To show that \circ is monotonic in both arguments, we need to prove that, for all $[A], [B], [C] \in \mathcal{F} / \sim$, if $[A] \sqsubseteq [B]$ then both $[A] \circ [C] \sqsubseteq [B] \circ [C]$ and $[C] \circ [A] \sqsubseteq [C] \circ [B]$. For the first, note that the following proof is available from assuming $\vdash A \multimap B$:

$$\frac{\frac{A \otimes C \vdash A \otimes C}{A \otimes C \vdash B \otimes C} \quad \frac{\frac{\frac{\vdash A \multimap B \quad A \vdash A}{A \vdash B} (\multimap E) \quad C \vdash C}{A, C \vdash B \otimes C} (\otimes I)}{A \otimes C \vdash B \otimes C} (\otimes E) \quad \vdash (A \otimes C) \multimap (B \otimes C) (\multimap I)$$

From this it follows that the assumption $[A] \sqsubseteq [B]$ leads to $[A \otimes C] \sqsubseteq [B \otimes C]$, and therefore we have $[A] \circ [C] \sqsubseteq [B] \circ [C]$, as desired. The proof that \circ is monotonic in its second argument is similar.

It remains to show that the relative pseudocomplement exists for all $[A], [B] \in \mathcal{F} / \sim$. That is, we need to prove that the set

$$R = \{X \in \mathcal{F} / \sim \mid [A] \circ X \sqsubseteq [B]\}$$

has a greatest member. Note first that $[A] \multimap [B] \in R$, since the sequent $\vdash (A \otimes (A \multimap B)) \multimap B$ is provable. To see that $[A] \multimap [B]$ is greatest in R , let $[C] \in R$. Since we have $[A] \circ [C] \sqsubseteq [B]$, and therefore $\vdash (A \otimes C) \multimap B$, we can prove the following:

$$\frac{\vdash (A \otimes C) \multimap B \quad \frac{A \vdash A \quad C \vdash C}{A, C \vdash A \otimes C} (\otimes I)}{A, C \vdash B} (\multimap E) \\ \frac{A, C \vdash B}{C \vdash A \multimap B} (\multimap I) \\ \vdash C \multimap A \multimap B \quad (\multimap I)$$

Then $[C] \sqsubseteq [A] \multimap [B]$, and we have shown that $\langle \mathcal{F} / \sim, \sqsubseteq, \circ, \{1\}, \multimap \rangle$ is a lineale. \square

Since $\langle \mathcal{F} / \sim, \sqsubseteq, \circ, \{1\}, \multimap \rangle$ is a lineale, we are in a position to prove completeness. The proof strategy follows Troelstra (1992, chapter 8), but with respect to a different class of models.

Theorem (Completeness). *If $I(\Gamma) \sqsubseteq I(A)$ holds in a TIL model, then $\Gamma \vdash A$ is provable.*

Proof. Let $\langle \mathcal{F} / \sim, \sqsubseteq, \circ, 1, \multimap, I \rangle$ be a TIL model constructed from a lineale as described above in A.7 and A.9 based on the set \mathcal{F} / \sim of equivalence classes of formulas, where $I(A) = [A]$ for every atomic A . It is not hard to see that I interprets every formula into its corresponding equivalence class. Now suppose that A is a valid formula, that is, $I(\emptyset) \sqsubseteq I(A)$, assuming that $\Gamma = \emptyset$ without loss of generality. Then we have $[1] \sqsubseteq [A]$, and so $\vdash 1 \multimap A$, from which $\vdash A$ follows immediately by **1I** and $\multimap E$. \square

Appendix B

Type Theory with Cartesian Products

Type theory, also known as *higher-order logic*, is a formal system that allows quantification over variables of all types. It has much in common with the *(simply) typed lambda calculus*, with the most important difference being that the notion of *reduction* is expressed in the object language rather than in the metalanguage.

This formalization of type theory essentially follows Church’s (1940) *simple theory of types*, with the semantics given by Henkin (1950), extended with cartesian product types and the associated term constructors and identities for surjective pairing following Barendregt, Dekkers, and Statman 2013. I adopt many of the elaborations and refinements due to Henkin (1963) and Andrews (2002) with the exception that I use a simplified axiom corresponding to β -conversion after Carpenter 1997 (see axiom schema (B.5) in definition B.12, below).

B.1 Syntax

Definition B.1 (Types). There is a set of basic types that includes at least the type t of truth values and the unit type 1 . Then if A and B are types, so are $(A \rightarrow B)$ and $(A \times B)$, and nothing else is a type.

The types t and 1 are the sole *logical types*; other members of the set of basic types are called *nonlogical types*. Unlike Church’s original 1940 system, which uses a logical type ι of individuals (sometimes written e), there is

no logical type of individuals. Types of the form $A \rightarrow B$ are interpreted as functions from type A to type B , and types of the form $A \times B$ as cartesian products of the types A and B , with 1 the type of the nullary cartesian product. Outer parentheses surrounding types built using the binary type constructors \rightarrow and \times are almost always dropped. Function types are written right-associatively, with $A \rightarrow B \rightarrow C$ abbreviating $A \rightarrow (B \rightarrow C)$, while product types associate to the left, so that $A \times B \times C$ is shorthand for $(A \times B) \times C$.

Definition B.2 (Variables). There is a countably infinite set of variables of each type A , written x_n^A for n a natural number.

Variable names are usually abbreviated by dropping the subscripted number and superscripted type when they are irrelevant or clear from context. I also usually write, for example, x, y, z for x_0^A, x_1^A, x_2^A , respectively, observing the usual convention that different stand-ins are used for distinct x_i^A .

Definition B.3 (Logical Constants). There is a *logical constant* $*$ of type 1 . For each type A , there is a corresponding logical constant $=_A$, of type $A \rightarrow A \rightarrow t$.

The constant $*$, the nullary cartesian product term, is the only term with the unit type 1 , enforced by equation (B.4), below. The family of constants $=_A$ expresses identity between terms of type A (the rules for term formation are given below in definition B.4; definition B.12 axiomatizes the $=_A$ constants). The identity constant is usually written infix, with $(a =_A b)$ abbreviating $((=_A a) b)$. As a further shorthand, the subscript is usually dropped in practice when the type is clear from context, so that just $=$ is often written. In addition to the logical constants, there may also be some *nonlogical constants* that are stated in order to model aspects of the empirical domain in question. Different instances of type theories are obtained by specifying different sets of nonlogical types and nonlogical constants.

Definition B.4 (Term Formation). Every term is associated with a type. If a is a term and A is a type, the (typing) *declaration* $a : A$ says that the type of a is A .

The set of terms is formed recursively according to the following rules:

1. A variable or constant is a term.
2. If $x : A$ is a variable and $b : B$, then $(\lambda_{x:A} b) : A \rightarrow B$.

3. If $f : A \rightarrow B$ and $a : A$, then $(f a) : B$.
4. If $a : A$ and $b : B$, then $\langle a, b \rangle : A \times B$.
5. If $c : A \times B$, then $(\pi_1 c) : A$ and $(\pi_2 c) : B$.

In a declaration of the form $a : A$, the term a is said to be an *inhabitant* of the type A . A term can be thought of as a proof that its corresponding type is inhabited (see chapter 4 of [Sørensen and Urzyczyn 2006](#) for a lucid overview of the Curry-Howard perspective on terms, types, and proofs). Terms of the form $(\lambda_{x:A} a)$ are called *(λ -)abstracts*; terms of the form $(f a)$ are called *applications*. The term $\langle a, b \rangle$ is the *(ordered) pair* of a and b , while terms of the form $(\pi_1 c)$ and $(\pi_2 c)$ are the *first* and *second projections* of c , respectively.

Outer parentheses around both abstracts, applications, and projections are often dropped, and the typing declaration is often elided from variables occurring as subscripts on a λ symbol when the typing information can be inferred from context, so that $(\lambda_{x:A} a)$ becomes simply $(\lambda_x a)$. The symbol $.$ abbreviates parentheses surrounding an abstract, allowing $(\lambda_x a)$ to be shortened to $\lambda_x.a$. Nested abstracts are sometimes abbreviated with a single $.$ symbol, for example, $\lambda_x \lambda_y.a$. Abstracts are sometimes further abbreviated by subscripting several variables to a single λ symbol, as in $\lambda_{xy}.a$. Parentheses around applications and projections are written left-associatively, with $((f a) b)$ abbreviated by $(f a b)$. The angled brackets surrounding pairs also associate to the left, with $\langle a, b, c \rangle$ shorthand for $\langle \langle a, b \rangle, c \rangle$.

Definition B.5 (Variable Contexts). A *(variable) context* is a finite, unordered set of declarations of the form $\{x_1 : A_1, \dots, x_n : A_n\}$ in which all of the variables x_i are distinct (but the types A_i need not be). For contexts Γ and Δ , the context $\Gamma \cup \Delta$ is the union of Γ and Δ .

Similar conventions are observed for variable contexts as for the contexts of tensor-implication logic (see definition [A.2](#)). The context $\Gamma \cup \Delta$ is abbreviated Γ, Δ , and the context $\Gamma, a : A$ is shorthand for $\Gamma, \{a : A\}$. The curly set-formation brackets surrounding contexts are nearly always elided in practice.

Definition B.6 (Typing Judgments). A *(typing) judgment* of the form

$$\Gamma \vdash a : A$$

is a metalanguage statement that the declaration $a : A$ is derivable in the context Γ .

In case Γ is empty, then $a : A$ is a (*formal*) *theorem* of type theory and the corresponding judgment is written simply $\vdash a : A$. Judgments with the truth-value type t are often abbreviated, with $\Gamma \vdash a : t$ shortened to $\Gamma \vdash a$. The notation $\Gamma \nvdash a : A$ is used to state that the judgment $a : A$ cannot be derived in Γ . A term identity is sometimes stated simultaneously with a typing judgment in the form $\Gamma \vdash a =_A b : A$, which is interpreted as saying that in the context Γ , a and b are proved identical and have the type A .

The following defines some commonly used term shorthands in the usual way.

Definition B.7 (Term Shorthands). In these definitions, the symbol $=_{\text{def}}$ denotes definitional equality, which is distinct from the term identity constants $=_A$.

$$\begin{aligned}
\top &=_{\text{def}} (=_{\text{t}} =_{\text{t} \rightarrow \text{t} \rightarrow \text{t}} =_{\text{t}}) \\
(\forall_{x:A} a) &=_{\text{def}} (\lambda_{x:A}. a =_{A \rightarrow \text{t}} \lambda_{x:A}. \top) \\
\text{F} &=_{\text{def}} (\forall_{x:t} x) \\
\wedge &=_{\text{def}} \lambda_{x:t} \lambda_{y:t}. (\lambda_{f:t \rightarrow \text{t} \rightarrow \text{t}} (f \top \top)) =_{(\text{t} \rightarrow \text{t} \rightarrow \text{t}) \rightarrow \text{t}} (\lambda_{f:t \rightarrow \text{t} \rightarrow \text{t}} (f x y)) \\
\Rightarrow &=_{\text{def}} \lambda_{x:t} \lambda_{y:t}. x =_{\text{t}} (x \wedge y) \\
\Leftrightarrow &=_{\text{def}} \lambda_{x:t} \lambda_{y:t}. (x \Rightarrow y) \wedge (y \Rightarrow x) \\
\neg &=_{\text{def}} (=_{\text{t}} \text{F}) \\
(a \neq b) &=_{\text{def}} (\neg (a =_A b)) \\
\vee &=_{\text{def}} \lambda_{x:t} \lambda_{y:t}. (\neg ((\neg x) \wedge (\neg y))) \\
(\exists_{x:A} a) &=_{\text{def}} (\neg (\forall_{x:A} (\neg a))) \\
(\exists!_{x:A} a) &=_{\text{def}} \exists_{y:A}. ((\lambda_{x:A} a) =_{A \rightarrow \text{t}} (=_{\text{t}} y))
\end{aligned}$$

Note that the shorthands in definition B.7 are all defined in terms of λ -abstraction, application, and the logical constant $=$. The following abbreviations are often used. The binary logical connectives are usually written infix, and outer parentheses are almost always dropped for \neg and for the binary functions $=$, \wedge , \Rightarrow and \vee when they are written infix. The quantifiers \forall and \exists observe the same notational shorthand as for λ , so that, for $x : A$ a variable, $\forall_x. a$ and $\exists_x. a$ are often written instead of $(\forall_{x:A} a)$ and $(\exists_{x:A} a)$.

A similar shorthand to the one used for λ is followed for subscripting multiple variables onto a single quantifier, as in $\forall_{xy}.a$.

Definition B.8 (Truth-Value Description Operator). The *truth-value description operator* $\iota_t : (t \rightarrow t) \rightarrow t$ is defined as

$$\iota_t =_{\text{def}} \lambda_{f:t \rightarrow t}.f = (\lambda_{x:t}x).$$

That is, ι_t tests whether a function $f : t \rightarrow t$ is the identity function on t .

Theorem B.9 (Generalized Description Operators). *If A is a type, then for each $f : A \rightarrow t$, the description operator $\iota_A : (A \rightarrow t) \rightarrow A$ yields the unique inhabitant $a : A$ such that $(f a) = \top$, provided one exists. Otherwise, ι_A yields a distinguished fixed inhabitant of A .*

Proof. See [Henkin 1963](#), §4.9. □

The description operators use the following shorthand, by analogy to the quantifiers \forall and \exists :

$$(1_{x:A}a) =_{\text{def}} (\iota_A \lambda_{x:A}.a)$$

Here, $a : t$ and A is any type.

B.1.1 Term Identification and Reduction

Definition B.10 (Binding and Substitutability). An occurrence of the variable x is (λ -)bound in the term a if it is in a part of a of the form $(\lambda_x b)$. An occurrence of x is otherwise free in a . The term b is substitutable for x in a if x does not occur free in a part of a of the form $(\lambda_y c)$, where y occurs free in b .

Note that in the judgment $\Gamma \vdash a : A$, the free variables of a are contained in Γ . If a term a contains no free occurrence of a variable, then a is said to be *closed*.

Terms are derived based on previously-derived term identities by invoking the substitution rule in definition [B.11](#).

Definition B.11 (Substitution Rule). If $a : A$ and $b = c$, then a term of type A is obtainable by replacing a single free instance of b with c in a .

The notation $a[b/x]$ is used to denote the term that results from repeatedly applying the substitution rule to replace each free occurrence of the variable

x in a with b . The term $a[b/x]$ is called the *capture-avoiding substitution of b for x in a* , and can be performed even when b is not substitutable for x in a because a term equivalent to a with different bound variable names can always be obtained (see the α -conversion theorem B.16, below). In view of this, I observe the usual practice of renaming bound variables without mention in invocations of the substitution rule where the substitutability provision fails.

The term identities in definition B.12 are the (logical) axiom schemas of type theory. They provide a starting point for applying the substitution rule by stipulating the behavior of the logical constants. Nonlogical axioms are also definable by stating judgments with empty contexts of the form $\vdash c : A$, where c is a nonlogical constant.

Definition B.12 (Axioms for Term Identity).

- | | | |
|-------|--|--|
| (B.1) | $((f \top) \wedge (f \text{F})) =_{\text{t}} \forall_{x:\text{t}}.(f x)$ | for $f : \text{t} \rightarrow \text{t}$ |
| (B.2) | $(x =_A y) \Rightarrow ((f x) =_{\text{t}} (f y))$ | for $x : A, y : A, f : A \rightarrow \text{t}$ |
| (B.3) | $(f =_{A \rightarrow B} g) =_{\text{t}} \forall_{x:A}.(f x) =_B (g x)$ | for $f : A \rightarrow B, g : A \rightarrow B$ |
| (B.4) | $u =_1 *$ | for $u : 1$ |
| (B.5) | $((\lambda_{x:A} b) a) =_B b[a/x]$ | for $a : A, b : B$ |
| (B.6) | $(\pi_1 \langle a, b \rangle) =_A a$ | for $a : A, b : B$ |
| (B.7) | $(\pi_2 \langle a, b \rangle) =_B b$ | for $a : A, b : B$ |
| (B.8) | $\langle (\pi_1 c), (\pi_2 c) \rangle =_{A \times B} c$ | for $c : A \times B$ |

The first three of the axioms in definition B.12 ensure that functions behave as expected. Equation (B.1) essentially states that there are exactly two truth values: if a function of type $\text{t} \rightarrow \text{t}$ (a property of truth values) holds for both \top and F , then it holds for every truth value. The fact that identical terms are indiscernible is expressed by (B.2), which states that $x : A$ and $y : A$ being identical implies that every property with domain A yields the same value for both. (A more general version of the axiom in (B.2) is straightforwardly derivable with $f : A \rightarrow B$ for any type B .) Equation (B.3) encodes extensionality for functional terms by requiring that two functions yielding the same value on every input be identical, and (B.4) says that the term of the unit type 1 is unique.

Substitution of suitable arguments into the body of abstracts is axiomatized in (B.5), and the projection of pair terms to their respective

components is defined in (B.6) and (B.7). Taken together, (B.5), (B.6), and (B.7) encode the notion of β -conversion from the lambda calculus (Barendregt, 1980) as term identities. Equation (B.8) encodes *surjective pairing*, which can be thought of as a form of η -conversion for products terms (see theorem B.15, below). Based on the substitution rule and the axioms in definition B.12, analogs of *modus ponens*, the deduction theorem, universal generalization and instantiation, etc., are available for the term logic of type theory. See Andrews 2002 for details.

Theorem B.13 (Term Equivalence). *The term identity constant $=_A$ induces an equivalence relation on the set of terms of type A , since for all terms $a : A$, $b : A$ and $c : A$, we have:*

$$\begin{array}{ll} a = a & \text{(reflexivity)} \\ \text{If } a = b \text{ and } b = c, \text{ then } a = c & \text{(transitivity)} \\ \text{If } a = b, \text{ then } b = a & \text{(symmetry)} \end{array}$$

Proof. For reflexivity, invoke the β -conversion axiom in (B.5) using the identity function $\lambda_x.x : A \rightarrow A$ applied to a . The rest of the proof is straightforward based on reflexivity and the substitution rule in definition B.11. \square

Theorem B.14 (Identity of Biimplication). *For every $x : t$ and $y : t$, we have $(x \Leftrightarrow y) = (x = y)$.*

Proof. Let $x : t$ and $y : t$. First, note that the \wedge operation, as defined in B.7, is commutative: $(x \wedge y) = (y \wedge x)$. Suppose that both $(x \Rightarrow y)$ and $(y \Rightarrow x)$. Then by the definition of \Rightarrow , we have $x = (x \wedge y)$ and $y = (y \wedge x)$, and therefore $x = y$ by the substitution rule, the commutativity of \wedge , and the properties of $=$ established in the Term Equivalence theorem (B.13).

Conversely, assume that $x = y$. It follows that $x = (x \wedge x) = (y \wedge y) = y$, and so $(x \Leftrightarrow y)$ by substitution and Term Equivalence. \square

In light of theorem B.14, I often observe the convention of writing \Leftrightarrow rather than $=_t$.

Two important term identities corresponding to η -conversion and α -conversion in the lambda calculus are derivable in type theory.

Theorem B.15 (η -Conversion). *If $f : A \rightarrow B$ and $x : A$ is a variable not occurring free in f , then $\lambda_x.(f x) = f$.*

Proof. Invoke the substitution rule along with the functional extensionality (B.3) and β -conversion (B.5) axiom schemas. \square

The intuition behind η -conversion is that functional terms do not change when their λ -bindings are stripped away as long as bound variables do not become free.

Theorem B.16 (α -Conversion). *Let $a : A$ and $x : B, y : B$ be variables such that y is substitutable for x in a but y does not occur free in a . Then $\lambda_x.a = \lambda_y.a[y/x]$.*

Proof. Use the η -conversion theorem and the β -conversion axiom schema in (B.5). \square

The identity in theorem B.16 shows that the choice of variable names is irrelevant, since abstracts are equivalent up to alphabetic change of their bound variables. Chapter 5 of Andrews 2002 contains proofs for both of these identities that use a slightly different, but equivalent, axiomatization of the term identities in definition B.12.

In an instance of η -conversion of the form $\lambda_x.(f\ x) = f$, the term to the left of the $=$ symbol is called an η -redex, and the term to the right is called an η -contractum. The substitution of a η -redex for its corresponding contractum is called η -expansion. An η -reduction occurs when a η -contractum replaces its corresponding redex. Similar terminology is used for β -conversions that instantiate the axiom schemas in (B.5), (B.6), and (B.7). The term *reduction* subsumes both β - and η -reduction, and similarly for *expansion*. The cover term (λ -)conversion is used for α -, β -, and η -conversion.

Lastly, we have the following, reflecting the fact that the term logic is classical.

Theorem B.17 (Double Negation Elimination). *For every $a : t$, we have*

$$(\neg(\neg a)) = a.$$

Proof. Let $a : t$ and note that, by definition B.7, we have $(\neg a) = (a = F)$. By axiom (B.1), there are two cases, depending on whether $a = T$ or $a = F$. Supposing $a = T$, we have $((T = F) = F) = T$ by substitution and “Rule T” (Andrews, 2002). On the other hand, supposing $a = F$, we have $((F = F) = F) = F$. See Andrews 2002, chapter 5 for more details. \square

B.1.2 Term Normalization

Definition B.18 (Normal Form). A term a is in $(\beta\eta)$ -normal form if there is no term that can be obtained from a by reduction. A term rendered in normal form is said to be *normalized*.

Example B.19. Consider the functional term

$$\mathbf{S} =_{\text{def}} \lambda_f \lambda_g \lambda_x. f \ x \ (g \ x) : (A \rightarrow B \rightarrow A) \rightarrow (A \rightarrow B) \rightarrow A \rightarrow A$$

applied to two variants of constant functions, defined schematically as

$$\mathbf{K}_{A,B} =_{\text{def}} \lambda_a \lambda_b. a : A \rightarrow B \rightarrow A,$$

so that they always return the value of their first argument regardless of what the value of the second is. Assuming $y : B$, the term $(\mathbf{S} \ \mathbf{K}_{A,B} \ (\mathbf{K}_{B,A} \ y))$ can be reduced to a normal form as follows, invoking the substitution rule in conjunction with the indicated identity at each step:

$$\begin{aligned} \mathbf{S} \ \mathbf{K}_{A,B} \ (\mathbf{K}_{B,A} \ y) &= (\lambda_f \lambda_g \lambda_x. (f \ x \ (g \ x)) \ \mathbf{K}_{A,B} \ (\mathbf{K}_{B,A} \ y)) && \text{(definition)} \\ &= (\lambda_g \lambda_x. (\mathbf{K}_{A,B} \ x \ (g \ x)) \ (\mathbf{K}_{B,A} \ y)) && (\beta\text{-reduction (B.5)}) \\ &= \lambda_x. \mathbf{K}_{A,B} \ x \ (\mathbf{K}_{B,A} \ y \ x) && (\beta\text{-reduction}) \\ &= \lambda_x. (\lambda_a \lambda_b. a \ x) \ (\mathbf{K}_{B,A} \ y \ x) && \text{(definition)} \\ &= \lambda_x. (\lambda_a \lambda_b. a \ x) \ (\lambda_b. y \ x) && \text{(definition)} \\ &= \lambda_x. (\lambda_a \lambda_b. a \ x) \ y && (\beta\text{-reduction}) \\ &= \lambda_x. (\lambda_b \ x) \ y && (\beta\text{-reduction}) \\ &= \lambda_x. x && (\beta\text{-reduction}) \end{aligned}$$

Since no further reduction is possible, $\lambda_x. x : A \rightarrow A$ (the identity function on A), is a normal form for $\mathbf{S} \ \mathbf{K}_{A,B} \ (\mathbf{K}_{B,A} \ y)$. This is a type-theoretic reconstruction of a well-known identity in combinatory logic (Curry and Feys, 1958).

Term reduction in type theory shares some notable properties of the typed lambda calculus. Like the typed lambda calculus, but in contrast with the *untyped* lambda calculus, we have the following:

Theorem B.20 (Normalizability). *Every term has a normal form.*

Moreover, every reduction strategy for a given term terminates, that is, every term is *strongly normalizable*.

Theorem B.21 (Strong Normalization). *If a is a term, then there is no infinite sequence of reductions starting from a .*

An important implication of the normalizability and strong normalization properties is that a normal form for any term can always be obtained after finitely many reduction steps. Thus every term is normalizable regardless of the chosen reduction strategy.

Finally, we have the *Church-Rosser property*, sometimes referred to as the *diamond property* or the property of *confluence*:

Theorem B.22 (Church-Rosser). *If $a = b$ and $a = c$, then there is a term d such that $b = d$ and $c = d$.*

This theorem states, intuitively, that in terms where multiple reductions are possible, the order in which the reductions are carried out is irrelevant because every term has a (unique) normal form. As a consequence of the Church-Rosser property and the fact that all terms are strongly normalizable, any two terms can be compared for identity by normalizing them and then comparing their normal forms.

See [Sørensen and Urzyczyn 2006](#), chapter 3 for proofs of theorems [B.20](#), [B.21](#), and [B.22](#) in the context of an in-depth discussion of reduction in the simply typed lambda calculus. [Barendregt \(1980\)](#) gives two proofs of the Church-Rosser theorem for the untyped lambda calculus, and [Lambek and Scott \(1986\)](#) provide a proof for type theory in a categorical setting. [Barendregt et al. \(2013\)](#) extend the proofs of normalizability, strong normalization, and the Church-Rosser property for the typed lambda calculus extended with surjective pairing, results that do not hold for the untyped lambda calculus.

B.1.3 Proof Theory

The inference rules for type theory are given in figure [B.1](#) in a natural deduction presentation. Note that the structural rules of permutation, weakening, and contraction are implicit since contexts are modeled as sets of typing declarations.

The rules for introducing and eliminating functional types (\rightarrow I and \rightarrow E) capture clauses [2](#) and [3](#) for term formation in definition [B.4](#). Rules \times I

$$\begin{array}{c}
\vdash c : A \quad (\text{Const}) \qquad x : A \vdash x : A \quad (\text{Var}) \\
\\
\frac{\Gamma, x : A \vdash b : B}{\Gamma \vdash (\lambda_x b) : A \rightarrow B} (\rightarrow I) \qquad \frac{\Gamma \vdash f : A \rightarrow B \quad \Delta \vdash a : A}{\Gamma, \Delta \vdash (f a) : B} (\rightarrow E) \\
\\
\frac{\Gamma \vdash a : A \quad \Delta \vdash b : B}{\Gamma, \Delta \vdash \langle a, b \rangle : A \times B} (\times I) \qquad \frac{\Gamma \vdash a : A_1 \times A_2}{\Gamma \vdash (\pi_i a) : A_i} (\times E_i) \quad i \in \{1, 2\}
\end{array}$$

Figure B.1: Inference rule schemas for type theory. The symbols c in the Const rule and x in the Var rule are metavariables over (both logical and nonlogical) constants and variables, respectively. The symbols A , B , A_1 , and A_2 are metavariables over types, and both Γ and Δ are metavariables over contexts.

and $\times E_i$ provide the counterpart of $\rightarrow I$ and $\rightarrow E$ for pairing and projection in clauses 4 and 5. The rule $\times E_i$ is expressed as a shorthand for two distinct rules, one for the first projection of a pair ($\times E_1$), and one for the second projection ($\times E_2$).

When a variable is introduced into a derivation via the Var rule, the introduction is ‘remembered’ by the variable context. This contrasts with the Const rules for introducing constants, which does not add the introduced constant to the context because it is not a variable and therefore cannot be used to create an abstract.

The structural rules of *permutation* (also known as *exchange*), *weakening*, and *contraction* are available, and are diagrammed in natural deduction presentation in figure B.2. Instances of the structural rules are usually elided without mention.

The definition of proof is similar to the one given in definition A.3 for tensor-implication logic, except that sequents are replaced by typing judgments.

Definition B.23 (Proofs). A *proof* of the typing judgment $\Gamma \vdash a : A$ is a natural deduction tree labeled with typing judgments, where

1. The label of the root is $\Gamma \vdash a : A$,
2. The label of each leaf is an instantiation of either Const or Var, and

$$\frac{\Gamma, x : A, y : B, \Delta \vdash c : C}{\Gamma, y : B, x : A, \Delta \vdash c : C} \text{ (Permutation)}$$

$$\frac{\Gamma, x : A \vdash c : C}{\Gamma, x : A, y : B \vdash c : C} \text{ (Weakening)}$$

$$\frac{\Gamma, x : A, x : A \vdash c : C}{\Gamma, x : A \vdash c : C} \text{ (Contraction)}$$

Figure B.2: Structural rule schemas for type theory, with x and y ranging over variables, c a metavariable over terms, and A , B , and C metavariables over types.

3. The label of each mother node is licensed by an instantiation of one of the rules in figure B.1 or B.2 by its daughters.

As for tensor-implication proofs, rule labels are sometimes given to clarify a proof's structure. The following example demonstrates a basic syntactic result that can be obtained using the inference rules for type theory.

Example B.24. The following is a proof that suitably typed functions can be composed, with A , B , and C metavariables over types and Γ shorthand for the context $\{f : A \rightarrow B, x : A\}$.

(B.9)

$$\frac{\frac{g : B \rightarrow C \vdash g : B \rightarrow C \quad \frac{f : A \rightarrow B \vdash f : A \rightarrow B \quad x : A \vdash x : A}{\Gamma \vdash (f x) : B} (\rightarrow E)}{g : B \rightarrow C, \Gamma \vdash (g (f x)) : C} (\rightarrow E)}{g : B \rightarrow C, f : A \rightarrow B \vdash \lambda_x. g (f x) : A \rightarrow C} (\rightarrow I)$$

This proof shows that, assuming functions $f : A \rightarrow B$ and $g : B \rightarrow C$, the function $\lambda_x. g (f x) : A \rightarrow C$ can be obtained that composes the effects of f and g .

Importantly, the judgment labeling the root node of the tree contains all of the information needed to reconstruct a proof of the resulting term. Each application of the form (ab) corresponds to an instantiation of the $\rightarrow E$ schema, and the occurrence of the λ -bound variable corresponds to an instantiation of $\rightarrow I$. The bound variable x itself corresponds to an instance

of the Var rule, as do the undischarged functional variables $f : A \rightarrow B$ and $g : B \rightarrow C$, which remain as hypotheses. The \rightarrow I rule can then be invoked twice to give a term that composes any two suitably typed functions:

$$\begin{array}{c}
 \text{(B.9)} \\
 \vdots \\
 \frac{g : B \rightarrow C, f : A \rightarrow B \vdash \lambda_x. g(f x) : A \rightarrow C}{f : A \rightarrow B \vdash \lambda_{gx}. g(f x) : (B \rightarrow C) \rightarrow A \rightarrow C} (\rightarrow\text{I}) \\
 \frac{\quad}{\vdash \lambda_{f gx}. g(f x) : (A \rightarrow B) \rightarrow (B \rightarrow C) \rightarrow A \rightarrow C} (\rightarrow\text{I})
 \end{array}$$

Thus function composition is a theorem of type theory, provided the functions in question are suitably typed.

B.2 Semantics

The model-theoretic semantics for type theory with cartesian products is a slight extension of the usual semantics due to [Henkin \(1950\)](#) and [Andrews \(2002\)](#). Recall that for two sets A and B , B^A is the set of all functions from A to B and $A \times B$ is the cartesian product of A and B .

Definition B.25 (Domains and Frames). For every type A , there is a corresponding set D_A , called the *domain* of A . A *frame* \mathcal{D} is some collection of nonempty domains D_A for each type A , defined so that

1. The domain of truth values is $D_t =_{\text{def}} \{0, 1\}$,
2. The domain of a functional type $A \rightarrow B$ is $D_{A \rightarrow B} \subseteq D_B^{D_A}$,
3. The domain of the nullary product type D_1 is a singleton, and
4. The domain of a nonnull product type $A \times B$ is $D_{A \times B} \subseteq D_A \times D_B$.

To handle description operators, define $U_A \subseteq D_{A \rightarrow t}$ for each A so that each member of U_A is the characteristic function of the singleton set $\{x\}$ for some $x \in D_A$. Then define the function $\text{uniq}_A : D_{A \rightarrow t} \rightarrow D_A$ so that, for every $P \in D_{A \rightarrow t}$ and $x \in D_A$,

$$(B.10) \quad \text{uniq}_A(u) = x \quad \text{if } P \in U_A, \text{ and}$$

$$(B.11) \quad \text{uniq}_A(u) = a \quad \text{otherwise, where } a \text{ is some fixed element of } D_A.$$

For every domain D_A , define the function $\text{eq}_A : D_A \rightarrow D_A \rightarrow D_t$ for all $x, y \in D_A$ as follows:

$$(B.12) \quad \text{eq}_A(x)(y) = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{otherwise} \end{cases}$$

Let \mathcal{D} be a frame and V the set of variables of any type. An *assignment* is a function $\varphi : V \rightarrow \bigcup \mathcal{D}$ that maps each variable x of type A to a member of D_A . For $x : A$ a variable and $y \in D_A$, the assignment φ_y^x is derived from the assignment φ as follows, for any variable z :

$$(B.13) \quad \varphi_y^x = \begin{cases} y & \text{if } z = x \\ \varphi(z) & \text{otherwise} \end{cases}$$

That is, φ_y^x coincides with φ on all variables except possibly for x , which is mapped to y .

Definition B.26 (Interpretations). An *interpretation* $\langle \mathcal{D}, I \rangle$ is a frame together with a function I that maps every logical constant to its corresponding domain, so that

$$\begin{aligned} I(*) & \text{ is the member of the singleton } D_1, \text{ and for each type } A, \\ I(=_A) &= \text{eq}_A \quad \text{and} \\ I(\iota_A) &= \text{uniq}_A. \end{aligned}$$

Definition B.27 (General Models). An interpretation $\langle \mathcal{D}, I \rangle$ is a *general model* if and only if there is an extension I_φ of I relative to every assignment φ such that $I_\varphi(a) \in D_A$ for every term $a : A$ and the following conditions are satisfied:

$$\begin{aligned} I_\varphi(x) &= \varphi(x) & \text{if } x \text{ is a variable,} \\ I_\varphi(c) &= I(c) & \text{if } c \text{ is a constant,} \\ I_\varphi((f a)) &= I_\varphi(f)(I_\varphi(a)), \\ I_\varphi(\lambda_{x:A}.b) & & \text{is the function mapping} \\ & & \text{every } a \in D_A \text{ to } I_{\varphi_a^x}(b), \end{aligned}$$

and for $c = \langle a, b \rangle$ a product term,

$$\begin{aligned} I_\varphi(\pi_1 c) &= I_\varphi(a), \\ I_\varphi(\pi_2 c) &= I_\varphi(b), \quad \text{and} \\ I_\varphi(c) &= \langle I_\varphi(\pi_1 c), I_\varphi(\pi_2 c) \rangle. \end{aligned}$$

A term $a : t$ is *true* in a general model $\mathcal{M} = \langle \mathcal{D}, I \rangle$ if and only if, for every extension I_φ of I , we have $I_\varphi(a) = 1$. Conversely, just in case $I_\varphi(a) = 0$ for every I_φ , a is said to be *false* in \mathcal{M} . If a is true in every general model, then a is *valid*, written $\models a$.

Some important results for the semantics of type theory are based on the formalization of general models in definition B.27. One is that type theory is well-behaved in the sense of being *consistent*, since no provable theorem can be false.

Theorem B.28 (Consistency). $\not\models F$.

Another important result, first due to [Henkin \(1950\)](#), states that every provable theorem is valid and every valid theorem is provable.

Theorem B.29 (Soundness and Completeness). *For every term $a : t$, we have $\vdash a$ if and only if $\models a$.*

Proof. The soundness and completeness proofs in [Andrews 2002](#), chapter 5 must be extended to account for cartesian product terms. For soundness, this requires proving the soundness of the substitution rule in definition B.11 for the product case, and the validity of the projection axioms in (B.6) and (B.7) and the surjective pairing axiom in (B.8). For completeness, it needs to be shown that general models still exist for type theory extended with cartesian products. To accomplish these, first let $\mathcal{M} = \langle \mathcal{D}, I \rangle$ be a general model.

Taking the case of the substitution rule, note that the case of the nullary product $*$ is trivial since $I(u)$ is the same for every $u : 1$ by definition of \mathcal{M} . Now let a and b be terms and suppose that $I_\varphi(a) = I_\varphi(b)$ for every extension I_φ of I . Suppose also that the term $\langle c', d' \rangle$ is obtained from $\langle c, d \rangle$ by invoking the substitution rule at most once to replace an occurrence of a with b . Proceeding by induction on the construction of $\langle a, b \rangle$, we have that $I_\varphi(c) = I_\varphi(c')$ and $I_\varphi(d) = I_\varphi(d')$ by the inductive hypothesis. Therefore, $I_\varphi(\langle c, d \rangle) = \langle I_\varphi(c), I_\varphi(d) \rangle = \langle I_\varphi(c'), I_\varphi(d') \rangle = I_\varphi(\langle c', d' \rangle)$, and the substitution rule is sound for the case of product terms.

To see that the axioms for projection are valid, first note that for any $a : A$ and $b : B$ we can instantiate axiom (B.6) as $(\pi_1 \langle a, b \rangle) =_A a$. Similarly, axiom (B.7) can be instantiated as $(\pi_2 \langle a, b \rangle) =_B b$. Since for every I_φ extending I , we have $I_\varphi(\pi_1 \langle a, b \rangle) = I_\varphi(a)$ and $I_\varphi(\pi_2 \langle a, b \rangle) = I_\varphi(b)$ by the definition of general models in definition B.27, the axioms for projection are valid. Similarly, instantiate (B.8) for some $c : A \times B$, and note that $\langle I_\varphi(\pi_1 c), I_\varphi(\pi_2 c) \rangle = I_\varphi(c)$ for every I_φ since \mathcal{M} is a general model.

Finally, the proof tactic for showing completeness of grouping terms together into equivalence classes, used by both Henkin and Andrews, is easily extended to the case of products. For the nullary cartesian product, take $I(*)$ as a singleton that is the only member of D_1 . Now suppose that the domains D_A and D_B have already been defined as sets of equivalence classes for the respective inhabitants of A and B . Also suppose $a : A$ and $b : B$ are terms, and let I_φ be the extension of some interpretation function I . Define $I_\varphi(\pi_1 \langle a, b \rangle) \in D_A$ and $I_\varphi(\pi_2 \langle a, b \rangle) \in D_B$ as the respective equivalence classes corresponding to a and b . Then define $I_\varphi(\langle a, b \rangle) \in D_{A \times B}$ as the set $\{c \mid c = \langle I_\varphi(a), I_\varphi(b) \rangle\}$. Clearly $I_\varphi(\langle a, b \rangle)$ is an equivalence class, and the requirements for product terms in a general model in definition B.27 are satisfied. \square

Andrews (2002) also gives a detailed proof of the consistency theorem B.28, and provides an elaborate discussion of the behavior of type theory with respect to both general models and the important special case of *standard models*, which are general models in which $D_{A \rightarrow B} = D_B^{D_A}$ for all types A and B . For standard models that additionally account for product types, the stipulation that $D_{A \times B} = D_A \times D_B$ for all types A and B would also be required. But a discussion of product types with respect to standard models is beyond the scope of this thesis.

Appendix C

Dependent Typing with Sums

As discussed in appendix B, terms in type theory may depend on other terms. For example, the functional term $f : A \rightarrow B$ depends in some sense on a term of type A : given $a : A$, the application $(f\ a) : B$ can be formed, and if a is a variable, the abstract $\lambda_a.f\ a : A \rightarrow B$ can also be formed. This dependency does not extend to types in the simple type theory of appendix B, however. Aside from the set of basic types, the only ways to form types are via the type constructors \rightarrow and \times . But to form the types $A \rightarrow B$ or $A \times B$, all that is required are two types A and B , and neither depends on the other.

With *dependent types*, functional types can be constructed in which one of the component types may depend on the other. Similarly, it is possible to construct product types where the pair's second projection can depend on the first. The system of dependent types discussed here is a generalization of type theory that additionally allows types that depend on terms. It is similar to the system λP of Barendregt (1991, 1992) and to the (Edinburgh) Logical Framework (LF) of Harper, Honsell, and Plotkin (1993).

In addition to the usual *dependent product* types, this system is extended with *dependent sums* that generalize the cartesian products in appendix B, following Aspinall and Hofmann (2005). Martin-Löf (1984, 1998) also discusses both dependent products and sums in the context of an intuitionistic type theory. The simplified formulation given here resembles the *compact* λP of Sørensen and Urzyczyn (2006, chapter 14), and is based in part on the exposition of Church-style dependent types in Hindley and Seldin 2008, chapter 13. Some example applications of dependent typing are discussed

in §C.3, and in §C.4, I discuss a method of extending Barendregt's *pure type systems* to include dependent sum types, after Barthe 1995.

Dependent types are mainly interesting because they give a type system more expressive power while at the same time imposing a certain amount of discipline. Because of this, most authors, including the ones mentioned above, limit their discussion of dependent types to the syntactic and proof-theoretic levels. However, Hofmann (1997) and Jacobs (1999) both give a categorical semantics for dependent types.

C.1 An Enriched Typing Ontology

The central idea in dependent typing is to make a system more expressive by moving the task of defining types from the metalanguage, as it is in the type theory in appendix B, to the object language. So the elaborated type system must account not only for types but also for *(type) constructors*, type symbols that may contain free term variables. Type constructors are interpreted as functions that yield types given certain arguments.

Accordingly, the type system is enriched to include the notions of *sort* and *kind* in addition to the notion of *type*. The constants \star and \square , both *sorts*, differentiate kinds, of sort \square , from the types of sort \star . *Kinds* classify *constructors*: functions from some number of terms to a type. For example, a constructor that requires two terms to produce a type is of a different kind than one requires only one. The sort \star is sometimes referred to as the sort of *proper types*, reflecting the fact that a type can be thought of as a zero-argument function that yields a type. That is, the sort of types has a dual status because it is also the kind of nullary constructors.

In dependent type theory, the set of terms is bifurcated into *objects*, which are associated with a type, and *kinds*, associated with a sort. The type-inhabiting objects of dependent type theory are exactly the terms of type theory. *(Dependent) product types*, notated with Π , are a generalization of type-theoretic functions that subsumes functions from objects to objects and functions from objects to types. The cartesian products of type theory are similarly generalized as *(dependent) sum types* at both the level of objects and of kinds, notated with Σ . Dependent products differ from ordinary type-theoretic functions in that the result type may depend on the value provided as the argument. Dependent sums differ from the cartesian

products of simple type theory because the type of the second component may depend on the first.

C.2 Syntax

The syntax of the dependent type theory $\lambda\mathbf{P}_\Sigma$ is an extension of the syntax of the simple type theory in appendix B. The metalanguage definition of types in definition B.1 is instead handled by the inference rules in figure C.1.

As before, there is a countably infinite set of variables for each type A , and there is in addition a similar set of variables for each constructor of kind K . The same notational conventions are observed for variables of any kind. There is also a set of basic type constants, some of which may require one or more objects to form a type.

Since type symbols can now contain variables, the definitions of object and types cannot be kept separate. Both objects and types are instances of *pseudoterms*, which generalizes the notion of terms for simple type theory in definition B.4.

Definition C.1 (Pseudoterms). The set of pseudoterms is defined as follows.

1. A variable or constant is a pseudoterm.
2. If f and a are pseudoterms, then so is $(f a)$.
3. If x is a variable and A, f are pseudoterms, then so is $(\lambda_{x:A} f)$.
4. If A and f are pseudoterms and x is a variable not occurring free in A , then both $(\Pi_{x:A} f)$ and $(\Sigma_{x:A} f)$ are pseudoterms.

Similar notational shorthands to those for simple type theory are used for applications, pairing, and for abstracts formed via Π and Σ . Also, analogous notions of substitutability and variable binding apply to Π and Σ , and the notation $a[b/x]$ is used in the same way (see definition B.10).

Variable contexts are also more finely grained in $\lambda\mathbf{P}_\Sigma$ because a variable declaration may contain free variables that occur elsewhere in the context.

Definition C.2 (Variable Contexts). A (*variable*) *context* Γ is a list of variable declarations of the form $x_1 : \alpha_1, \dots, x_n : \alpha_n$, where α ranges over both types and kinds, and the following conditions hold:

1. All of the variables x_i are distinct, and
2. Γ is *sequentially valid*: the free variables of each α_i are found in $\{x_1, \dots, x_{i-1}\}$.

The *domain* of a context Γ , written $\text{dom}(\Gamma)$ is defined as the set

$$\{x \mid x : \alpha \text{ occurs in } \Gamma, \text{ for some } \alpha\}$$

of variables in Γ .

This definition differs from the definition of type-theoretic variable contexts (definition B.5) by requiring that contexts are lists rather than sets, and that free variables occurring in kinds are mentioned ‘earlier’ (further to the left) in the list.

The notation for typing declarations is overloaded, with the syntax $a : A$ extended to apply to declarations of both types and kinds. As before, when a is an object and A is a type, the notation $a : A$ is a metalanguage statement that a is a object of type A . But the *kinding declaration* $A : \star$ says that A is a type, while $\star : \square$ says that \star is the kind of nullary constructors, and $\Pi_{x:A}.\star : \square$ says that $\Pi_{x:A}.\star$ is the kind of constructors that take an inhabitant of A to a type. If a declaration of the form $\Gamma \vdash A : B$ is derivable, where Γ is a context according to definition C.2, then A and B are called *terms*.

Judgments are written just as for type theory (definition B.6), so that $\Gamma \vdash a : A$ states that the declaration $a : A$ is derivable in the context Γ . Suppose $\Gamma \vdash a : A$ is a judgment, where Γ is a variable context according to definition C.2. Then the a is a term that is either an object or a constructor, and A is a term that is either a type or a kind, depending on whether $a : A$ is a typing or kinding declaration.

For a given formulation of dependent type theory, axioms corresponding to the basic types are stated as judgments. For instance, rather than stating in the metalanguage that t is a basic type, as in type theory, it is instead axiomatized as a basic type constant via the judgment $\vdash t : \star$ (and similarly for any nonlogical basic types).

The notion of β -conversion in λP_Σ is defined for pseudoterms, not terms as in simple type theory (see axiom (B.5)), and η -conversion holds for terms of a fixed type (Geuvers, 1992, 1993). Results analogous to the strong normalization (theorem B.21) and the Church-Rosser (theorem

B.22) properties for type theory are available for λP , see [Barendregt \(1992\)](#), [Geuvers \(1992, 1993\)](#), and references therein.

C.2.1 Proof Theory

A natural deduction presentation of the inference rules of λP_Σ is given in figure C.1 on page 272. These rules are essentially the ones used by [Barendregt \(1991, 1992\)](#) extended with the rules for dependent sums found in [Barthe 1995](#). The Conv rule follows [Geuvers \(1992, 1993\)](#) in allowing β - or η -equivalent types to be substituted for one another, whereas [Barendregt](#)'s variant allows only β -equivalence. No structural rules are specified because contexts must meet definition C.2, however, contexts can be harmlessly permuted as long as the property of sequential validity is retained. As for the inference rules for tensor-implication logic and type theory, the rule labels are optional. Proof trees are defined in a similar way as those in type theory (definition B.23), except that a leaf may either be an axiom corresponding to a basic type constant or an instance of the Ax rule.

The rule Ax is an axiom stating that \star is a kind. This rule is instrumental in proving kinds of higher arities, as (C.1) shows.

$$(C.1) \quad \frac{\frac{\vdash \star : \square \quad \vdash A : \star}{x : A \vdash \star : \square} \text{ (Weak)}}{\vdash \Pi_{x:A} \star : \square} \text{ (Prod)}$$

Assuming A is a type, this proof demonstrates that one kind of constructor is $\Pi_{x:A} \star : \square$, the kind that map a variable $x : A$ to a type. As its name implies, the Weak rule is an explicit statement of its analog in type theory, the structural rule of weakening, which is not otherwise available in dependent type theory due to the higher complexity of variable contexts.

The Var rule simply states that variables are available both for objects and for constructors. The Prod rule not only allows kinds of constructors to be derived (as in (C.1)), but also generalizes the type-theoretic constructor \rightarrow when $s = \star$:

$$(C.2) \quad \frac{\frac{\vdash B : \star \quad \vdash A : \star}{x : A \vdash B : \star} \text{ (Weak)}}{\vdash \Pi_{x:A} B : \star} \text{ (Prod)}$$

The proof in (C.2) is an object-language restatement of the rule of type formation via \rightarrow in definition B.1. It says that, supposing A and B are

$$\begin{array}{c}
\vdash \star : \square \quad (\text{Ax}) \\
\\
\frac{\Gamma \vdash A : s}{\Gamma, x : A \vdash x : A} (\text{Var}) \quad x \notin \text{dom}(\Gamma) \\
\\
\frac{\Gamma \vdash A : B \quad \Gamma \vdash C : s}{\Gamma, x : C \vdash A : B} (\text{Weak}) \quad x \notin \text{dom}(\Gamma) \\
\\
\frac{\Gamma \vdash A : B \quad \Gamma \vdash C : s}{\Gamma \vdash A : C} (\text{Conv}) \quad B =_{\beta\eta} C \\
\\
\frac{\Gamma \vdash A : \star \quad \Gamma, x : A \vdash B : s}{\Gamma \vdash \Pi_{x:A}.B : s} (\text{Prod}) \\
\\
\frac{\Gamma, x : A \vdash C : B \quad \Gamma \vdash \Pi_{x:A}.B : s}{\Gamma \vdash \lambda_{x:A}.C : \Pi_{x:A}.B} (\text{Abs}) \\
\\
\frac{\Gamma \vdash C : \Pi_{x:A}.B \quad \Gamma \vdash D : A}{\Gamma \vdash (CD) : B[D/x]} (\text{App}) \\
\\
\frac{\Gamma \vdash A : \star \quad \Gamma, x : A \vdash B : s}{\Gamma \vdash \Sigma_{x:A}.B : s} (\text{Sum}) \\
\\
\frac{\Gamma \vdash C : A \quad \Gamma \vdash D : B[A/x] \quad \Gamma \vdash \Sigma_{x:A}.B : s}{\Gamma \vdash \langle C, D \rangle : \Sigma_{x:A}.B} (\text{Pair}) \\
\\
\frac{\Gamma \vdash c : \Sigma_{x:A}.B}{\Gamma \vdash (\pi_1 c) : A} (\text{Proj}_1) \qquad \frac{\Gamma \vdash c : \Sigma_{x:A}.B}{\Gamma \vdash (\pi_2 c) : B[(\pi_1 c)/x]} (\text{Proj}_2)
\end{array}$$

Figure C.1: Inference rule schemas for $\lambda\mathbf{P}_\Sigma$. The metavariable s ranges over the set $\{\star, \square\}$ of sorts, and $=_{\beta\eta}$ denotes the relation of sharing a $\beta\eta$ -normal form. These rules make the implicit assumption that all contexts meet the conditions given in definition C.2.

types, the type $\Pi_{x:A}.B$ of functions from A to B is derivable. In view of this, the type $A \rightarrow B$ is defined for types A and B as the special case of a product type in which B does not depend on an object of type A :

$$(C.3) \quad A \rightarrow B =_{\text{def}} \Pi_{x:A}.B \quad \text{where } x \text{ does not occur free in } B.$$

To simplify the notation, I use the abbreviation $A \rightarrow B$ wherever possible.

Analogously to the Prod rule, the rule Abs generalizes the inference rule $\rightarrow I$ of type theory (see figure B.1). As a prerequisite, (C.4) shows that a variable of type B is derivable in a context with a variable of type A .

$$(C.4) \quad \frac{\frac{}{b : B \vdash b : B} (\text{Var}) \quad \frac{}{b : B \vdash A : \star} (\text{Weak})}{b : B, x : A \vdash b : B} (\text{Weak})$$

The proof in (C.5) then shows that, for the special case when $s = \star$, Abs simply restates the exact content of $\rightarrow I$, with an extra leaf to ensure that the type $\Pi_{x:A}.B$ is derivable.

$$(C.5) \quad \frac{\begin{array}{c} (C.4) \\ \vdots \\ b : B, x : A \vdash b : B \end{array} \quad \frac{\begin{array}{c} (C.2) \\ \vdots \\ \vdots \end{array} \quad \frac{}{b : B \vdash \Pi_{x:A}.B : \star} (\text{Weak})}{b : B \vdash \lambda_{x:A}.b : \Pi_{x:A}.B} (\text{Abs})$$

With the definition in (C.3) in place, this is simply a proof of the judgment $b : B \vdash \lambda_{x:A}.b : A \rightarrow B$, since x is not free in B .

For the other case $s = \square$, where Abs is used to form a constructor, assume that B_x is a basic type constant depending on a variable of type A (see (C.1) for proof that $\Pi_{x:A}.\star : \square$ is derivable). To derive the corresponding constructor, a proof similar to the one in (C.4) is used to get a version of B_x with the variable $x : A$ represented in the context.

$$(C.6) \quad \frac{\frac{}{x : A \vdash B_x : \Pi_{x:A}.\star} (\text{Weak}) \quad \frac{}{x : A \vdash x : A} (\text{Var})}{x : A \vdash (B_x x) : \star[x/x]} (\text{App})$$

The declaration in the root label of (C.6) reduces to $B_x : \star$ because x is vacuously substituted for x in B_x . Then the constructor is derivable as follows, starting by converting $\star[x/x]$ to simply \star , which is a licensed use

of Conv since x does not occur in \star .

(C.7)

$$\begin{array}{c}
 \text{(C.6)} \\
 \vdots \\
 \frac{x : A \vdash B_x : \star[x/x]}{x : A \vdash B_x : \star} \quad \frac{\frac{\vdash \star : \square \quad \vdash A : \star}{x : A \vdash \star : \square} \text{(Var)}}{x : A \vdash B_x : \star} \text{(Conv)} \quad \frac{\text{(C.1)} \quad \vdots}{\vdash \Pi_{x:A} \star : \square} \text{(Abs)} \\
 \hline
 \vdash \lambda_{x:A}. B_x : \Pi_{x:A} \star
 \end{array}$$

The different instantiations of Abs in (C.5) and (C.7) also show the difference between instantiations of the App rule for constructors as opposed to object-level functions. First, the abstract labeling the root of the proof in (C.5) is applied to a suitable argument of type A :

$$\begin{array}{c}
 \text{(C.5)} \\
 \vdots \\
 \frac{b : B, y : A \vdash \lambda_{x:A}. b : \Pi_{x:A}. B \quad \frac{\frac{\vdash A : \star \quad \vdash B : \star}{b : B \vdash A : \star} \text{(Weak)}}{b : B, y : A \vdash y : A} \text{(Var)}}{b : B, y : A \vdash (\lambda_{x:A}. b y) : B[y/x]} \text{(App)}
 \end{array}$$

(In the proof in (C.8), there is a suppressed instance of Weak that augments the context of (C.5) to contain the variable $y : A$.) Since x does not occur free in either b or B , the root label of this proof reduces to $b : B, y : A \vdash b : B$, and so the instance of App in (C.8) is exactly like the type-theoretic inference rule $\rightarrow E$. The situation is different for the constructor derived in (C.7), however.

$$\begin{array}{c}
 \text{(C.7)} \\
 \vdots \\
 \text{(C.9)} \quad \frac{\frac{\vdash \lambda_{x:A}. B_x : \Pi_{x:A} \star \quad \vdash A : \star}{y : A \vdash \lambda_{x:A}. B_x : \Pi_{x:A} \star} \text{(Weak)} \quad \frac{\vdash A : \star}{y : A \vdash y : A} \text{(Var)}}{y : A \vdash (\lambda_{x:A}. B_x y) : \star[y/x]} \text{(App)}
 \end{array}$$

Note that the declaration in the root label of the proof in (C.9) reduces to $B_y : \star$ because B_x contains an occurrence of x but \star does not.

The proof in (C.9) allows objects of the type B_a to be derived for any derivation of some $a : A$ in a context Γ (in the following proof, some

(C.10)

$$\begin{array}{c}
(C.9) \\
\vdots \\
\frac{\Gamma \vdash A : \star \quad \frac{\Gamma, y : A \vdash B_y : \star[y/x] \quad \Gamma, y : A \vdash \star : \square}{\Gamma, y : A \vdash B_y : \star} (\text{Conv})}{\Gamma \vdash \Pi_{y:A}. B_y : \star} (\text{Prod}) \qquad \vdots \\
\frac{\Gamma \vdash \Pi_{y:A}. B_y : \star \quad \Gamma \vdash a : A}{\Gamma \vdash (B_y a) : \star[a/y]} (\text{App})
\end{array}$$

Noting that the root label of (C.10) reduces to $\Gamma \vdash B_a : \star$, the proof of an object of type B_a just involves using the Conv and Var rules (with some instances of Weak suppressed):

$$(C.11) \quad \frac{\frac{\frac{\vdots}{\Gamma \vdash B_a : \star[a/y]} \quad \frac{\vdots}{\Gamma \vdash \star : \Box}}{\Gamma \vdash B_a : \star} \text{ (Var)}}{\Gamma, b : B_a \vdash b : B_a} \text{ (Conv)}$$

The Sum rule derives dependent sum types in a way parallel to the Prod rule: when the variable s is instantiated as the sort \star of proper types, it is exactly equivalent to the rule $\times I$ from type theory.

$$(C.12) \quad \frac{\frac{}{\vdash A : \star} \quad \frac{\vdash B : \star \quad \vdash A : \star}{x : A \vdash B : \star} (\text{Weak})}{\vdash \Sigma_{x:A}. B : \star} (\text{Sum})$$

(Compare this proof with the one in (C.2) for the special case of products with $s = \star$.) Since the type B does not depend on the variable x , the dependent sum type $\Sigma_{x:A}.B$ is the type-theoretic cartesian product type of pairs $\langle a, b \rangle$ where $a : A$ and $b : B$. Analogously to \rightarrow , the type constructor \times is then defined, for A and B types, as the special case when the dependency is not present:

$$(C.13) \quad A \times B =_{\text{def}} \Sigma_{x:A}. B \quad \text{where } x \text{ does not occur free in } B.$$

As for \rightarrow , I use \times where possible as a simplification. For the generalized case, the proof in (C.1) of the kind of product constructors can be modified so that the last inference rule invoked is Sum instead of Prod, yielding

$\vdash \Sigma_{x:A}.\star : \square$, the kind of constructors that form pair types where the first component is of type A .

The Pair rule forms objects and constructors with a dependent sum type the same way the Abs rule does for the dependent product type. First, the special case when there is no dependency between the components ($s = \star$) requires that an object of the type $B[a/x]$ be proved.

$$(C.14) \quad \frac{\begin{array}{c} (C.5) \\ \vdots \\ b : B \vdash \lambda_{x:A}.b : \Pi_{x:A}.B \end{array} \quad \frac{\begin{array}{c} \vdots \\ \vdash a : A \end{array} \quad \vdash B : \star \text{ (Weak)}}{b : B \vdash a : A} \text{ (App)}}{b : B \vdash (\lambda_{x:A}.b a) : B[a/x]} \text{ (App)}$$

And since a is not free in b , we have $\vdash b : B[a/x]$ by the β -conversion axiom in (B.5). Then the pair $\langle a, b \rangle$ can be formed as follows:

$$(C.15) \quad \frac{\begin{array}{c} \vdots \\ b : B \vdash a : A \end{array} \quad \begin{array}{c} (C.14) \\ \vdots \\ b : B \vdash b : B[a/x] \end{array} \quad \begin{array}{c} (C.12) \\ \vdots \\ b : B \vdash \Sigma_{x:A}.B : \star \end{array}}{b : B \vdash \langle a, b \rangle : \Sigma_{x:A}.B} \text{ (Pair)}$$

In (C.15), the leftmost and rightmost subproofs use contexts that are derived by omitted applications of the Weak rule. Similarly to (C.5), the proof in (C.15) combined with the definition of nondependent pairing in (C.13) is equivalent to a proof of $\vdash \langle a, b \rangle : A \times B$. So just as for the rule Abs for the case $s = \star$, the instantiation of the Pair rule in (C.15) is again just an object-language restatement of the type-theoretic pair formation rule $\times I$.

For an instance of pairing where a dependency between the components is present, invoke the proof in (C.6) in an instance of the Sum rule (the following proof omits instances of both Weak and Conv).

$$(C.16) \quad \frac{\begin{array}{c} (C.6) \\ \vdots \\ \vdash A : \star \quad x : A \vdash B_x : \star \end{array}}{\vdash \Sigma_{x:A}.B_x : \star} \text{ (Sum)}$$

Invoking the proof in (C.10) in conjunction with (C.16), a pair can then be formed in which the second component depends on the first. To start, we

(C.17)

$$\begin{array}{c}
\vdots \\
\frac{x : A \vdash b : B_x}{\vdash \lambda_{x:A}.b : \Pi_{x:A}.B_x} \text{ (Abs)} \\
\vdots \\
\frac{\vdash \lambda_{x:A}.b : \Pi_{x:A}.B_x \quad \vdash a : A}{\vdash (ba) : B_x[a/x]} \text{ (App)}
\end{array}
\quad \text{(C.6)}$$

$$(C.18) \quad \frac{\begin{array}{c} \vdots \\ \vdash a : A \end{array} \quad \begin{array}{c} (C.17) \\ \vdots \\ \vdash b : B_x[a/x] \end{array} \quad \begin{array}{c} (C.16) \\ \vdots \\ \vdash \Sigma_{x:A}. B_x : \star \end{array}}{\vdash \langle a, b \rangle : \Sigma_{x:A}. B_x} \text{ (Pair)}$$

The rules for projection show how the dependency information is maintained for such pairs. The Proj_1 rule simply retrieves the first component of a pair in a way exactly parallel to the type-theoretic rule $\times E_1$, as shown in the following proof:

$$(C.19) \quad \frac{\begin{array}{c} (C.18) \\ \vdots \\ \vdash \langle a, b \rangle : \Sigma_{x:A}. B_x \end{array}}{\vdash (\pi_1 \langle a, b \rangle) : A} (\text{Proj}_1)$$

$$(C.20) \quad \frac{\begin{array}{c} (C.18) \\ \vdots \\ \vdash \langle a, b \rangle : \Sigma_{x:A}. B_x \end{array}}{\vdash (\pi_2 \langle a, b \rangle) : B_x[(\pi_1 \langle a, b \rangle)/x]} \text{ (Proj}_2)$$

Since $(\pi_1 \langle a, b \rangle) = a$ and $(\pi_2 \langle a, b \rangle) = b$, the judgment in the conclusion of (C.20) reduces to $\vdash b : B_a$ by an application of β -reduction and the Conv

(C.21)

In this proof, a β -reduction is performed in the instance of App to substitute a for the bound variable x in the abstract in the root label of (C.7).

C.3 Applications

C.3.1 Von Neumann-style Natural Numbers

The dynamic semantics discussed in this thesis starting in chapter 4 makes use of a type ω_n , which, for each n is intuitively the type of natural numbers less than n . To define the types ω_n in dependent type theory, we start with the type n of natural numbers, which we assume is defined in the usual way with $\vdash 0 : n$, a function mapping each inhabitant of n to its successor, the linear order $<$, and the addition and subtraction operations $+$ and $-$. This type n may simply be assumed as a basic type and interpreted in the obvious way in every model, or defined, for example, following [Andrews \(2002\)](#).

We then add the axiom

$$(C.22) \quad \vdash \omega_n : \Pi_{n:n} \star$$

to our dependent type theory, which states simply that the type ω_n depends on a natural number n . Then for each $n : \mathbb{n}$, the function $\text{nat}_n : \Pi_{\omega_n} \mathbb{n}$ ensures that the type ω_n has the desired inhabitants, due to the following axioms:

$$(C.23) \quad \vdash \forall_{m:n} \forall_{n:n} . (m < n) \Leftrightarrow (\exists !_{i:\omega_n} (\text{nat}_n i) = m)$$

$$(C.24) \quad \vdash \forall_{n:n} \forall_{i:\omega_n} \forall_{j:\omega_n} . ((\text{nat}_n i) = (\text{nat}_n j)) \Rightarrow (i = j)$$

The axiom in (C.23) states that an inhabitant of ω_n exists for each $m < n$; the one in (C.24) ensures that nat_n is an injection. Then for each pair of natural numbers with $m < n$, we define

$$m_n =_{\text{def}} !_{i:\omega_n} . (\text{nat}_n i) = m ,$$

where $! : (\omega_n \rightarrow \mathbb{t}) \rightarrow \omega_n$ is defined based on theorem B.9. That is, m_n is shorthand for the unique member of ω_n which is mapped to m by nat_n . These axioms guarantee, for example, that ω_3 has the inhabitants 0_3 , 1_3 , and 2_3 .

However, this axiomatization of ω_n and the nat_n function also implies that there is an inhabitant 0_n of *every* ω_n except for ω_0 . This is somewhat inconvenient, as we often need to use the inhabitant $0 : \omega_n$ for some n , without being concerned about the actual value of n . To connect all of these inhabitants, which intuitively represent the same natural number across different ω_n , we define the coercion function $\text{coerce}_{m,n} : \omega_m \rightarrow \omega_n$ for each pair m, n of natural numbers with $m < n$:

$$\text{coerce}_{m,n} =_{\text{def}} \lambda_{i:\omega_m} . !_{j:\omega_n} . (\text{nat}_m i) = (\text{nat}_n j)$$

This function allows us to drop the subscripts on inhabitants of some type ω_n when they are irrelevant. For example, assuming that $0_m : \omega_m$, we can adopt the notational convention of writing simply $0 : \omega_n$ instead of $(\text{coerce}_{m,n} 0_m) : \omega_n$ in a context where it is understood that $m < n$. It also allows us to engage in the mild but convenient notational abuse of writing inhabitants of ω_n for some n interchangeably with their natural number counterparts.

Defining the type ω as

$$\omega =_{\text{def}} \Sigma_{n:\mathbb{N}} \omega_n ,$$

we can equip ω with analogs of the $<$ order and $+$ and $-$ operations on the natural numbers. This is done by defining the linear order $<_{\omega}: \omega \rightarrow \omega \rightarrow \mathbf{t}$ and the addition and subtraction operations $+_{\omega}$ and $-_{\omega}$, both of type $\omega \rightarrow \omega \rightarrow \omega$. These functions are axiomatized based on their natural number counterparts, as follows:

$$\begin{aligned} &\vdash \forall m:\omega \forall n:\omega. (m <_{\omega} n) = ((\text{nat}(\pi_2 m)) < (\text{nat}(\pi_2 n))) \\ &\vdash \forall m:\omega \forall n:\omega. (\text{nat}(\pi_2(m +_{\omega} n))) = ((\text{nat}(\pi_2 m)) + (\text{nat}(\pi_2 n))) \\ &\vdash \forall m:\omega \forall n:\omega. (\text{nat}(\pi_2(m -_{\omega} n))) = ((\text{nat}(\pi_2 m)) - (\text{nat}(\pi_2 n))) \end{aligned}$$

In practice, the subscript on these functions is omitted.

C.3.2 Vectors as n -ary Cartesian Products

DyCG contexts must be able to keep track of an arbitrary number of discourse referents, and the n -ary vectors for every type, defined in this section, fit the bill. The idea is to use the notion of n -ary cartesian products to model vectors, so that the type A^n is informally the type

$$\overbrace{A \times \cdots \times A}^{n \text{ times}},$$

for n a natural number.

Definition C.3 (Vectors). For each type A , there is a basic type constant $A^n : \Pi_{n:\mathbb{N}} \star$, which for each $n : \mathbb{N}$ gives the type of n -ary cartesian products whose factors are each of type A , and is defined recursively as follows:

$$\begin{aligned} A^0 &=_{\text{def}} 1 \\ A^{n+1} &=_{\text{def}} A^n \times A \end{aligned}$$

For example, the only inhabitant of type A^0 is $*$, the constant of the unit type 1; inhabitants of A^1 are of the form $\langle *, a \rangle$ for some $a : A$; A^2 has inhabitants $\langle \langle *, a \rangle, b \rangle$ for a and b inhabitants of A , etc. To distinguish variables of vector types from variables of other types, write, for example, \mathbf{a} for a variable of a type A^n .

Definition C.4 (Vector Length). The length of a vector is obtained by the function

$$|\cdot|_n : \Pi_{n:n} \Pi_{\mathbf{a}:A^n} . n,$$

which is defined¹ for each natural number n as simply

$$|\cdot|_n =_{\text{def}} \lambda_{\mathbf{a}.A^n} . n$$

The natural number subscript is usually dropped from the vector length function when it is irrelevant or contextually determinable.

Definition C.5 (Accessing Coordinates of a Vector). Recalling that, for example, the triple $\langle a, b, c \rangle$ is defined as the pair $\langle \langle a, b \rangle, c \rangle$, vector data is accessed via the functions $\text{head}_n : \Pi_{n:n} \Pi_{\mathbf{a}:A^{n+1}} . A$, which gets the last element, and $\text{tail}_n : \Pi_{n:n} \Pi_{\mathbf{a}:A^{n+1}} . A^n$, which gets the embedded tuple. The axioms governing these functions are the following:

$$(C.25) \quad \vdash \forall_{n:n} \forall_{\mathbf{a}:A^{n+1}} . (\text{head}_n \mathbf{a}) = (\pi_2 \mathbf{a})$$

$$(C.26) \quad \vdash \forall_{n:n} \forall_{\mathbf{a}:A^{n+1}} . (\text{tail}_n \mathbf{a}) = (\pi_1 \mathbf{a})$$

Note that neither of the functions head_n or tail_n can be called on an empty vector due to their types.

For an n -ary vector \mathbf{a} on A , the function

$$\text{coord}_n : \Pi_{n:n} \Pi_{\mathbf{a}:A^n} \Pi_{\omega_{|\mathbf{a}|}} . A$$

retrieves the coordinate of \mathbf{a} at a specified index, which is type-constrained to be among the indices in \mathbf{a} . This function is abbreviated using suffix notation, so that for every vector $\mathbf{a} : A^n$ and natural number $i < |\mathbf{a}|$, \mathbf{a}_i abbreviates $(\text{coord}_n \mathbf{a} i)$.

The following axioms recursively spell out how coordinates of a vector are selected.

$$(C.27) \quad \vdash \forall_{n:n} \forall_{\mathbf{a}:A^n} . \mathbf{a}_{|\mathbf{a}|-1} = (\text{head}_{|\mathbf{a}|-1} \mathbf{a})$$

$$(C.28) \quad \vdash \forall_{n:n} \forall_{\mathbf{a}:A^n} \forall_{i:\omega_{|\mathbf{a}|-1}} . \mathbf{a}_i = (\text{coord}_{|\mathbf{a}|-1} (\text{tail}_{|\mathbf{a}|-1} \mathbf{a}) i)$$

¹Functions like the arity function for vectors $|\cdot|$ might be said to be written in **outfix** notation, the opposite of *infix* notation, because they surround their argument rather than being surrounded by their arguments.

As an example, letting $\mathbf{x} = \langle \langle \langle *, x \rangle, y \rangle, z \rangle$ be a vector of type X^3 for some type X , we have

$$\begin{aligned}
 \mathbf{x}_2 &= (\text{coord}_3 \mathbf{x} \ 2) \\
 &= (\text{head}_2 \mathbf{x}) && \text{(by (C.27))} \\
 &= z && \text{(C.25),} \\
 \mathbf{x}_1 &= (\text{coord}_3 \mathbf{x} \ 1) \\
 &= (\text{coord}_2 (\text{tail}_2 \mathbf{x}) \ 1) && \text{(C.28)} \\
 &= (\text{head}_1 (\text{tail}_2 \mathbf{x})) && \text{(C.27)} \\
 &= y && \text{(C.25) and (C.26),}
 \end{aligned}$$

and

$$\begin{aligned}
 \mathbf{x}_0 &= (\text{coord}_3 \mathbf{x} \ 0) \\
 &= (\text{coord}_2 (\text{tail}_2 \mathbf{x}) \ 0) \\
 &= (\text{coord}_1 (\text{tail}_1 (\text{tail}_2 \mathbf{x})) \ 0) && \text{(C.28)} \\
 &= (\text{head}_0 (\text{tail}_1 (\text{tail}_2 \mathbf{x}))) && \text{(C.27)} \\
 &= x && \text{(C.25) and (C.26).}
 \end{aligned}$$

Similarly to the other functions involving vectors, the subscript n on the function coord_n is often dropped in practice.

For a one-coordinate vector \mathbf{x} , since there is only a single coordinate accessible via coord , we might like to identify \mathbf{x} with its sole coordinate. But this is impossible because the type A^1 is not the same as the type A of \mathbf{x} 's coordinate. However, we do have the following.

Proposition C.6. *Every $f : A \rightarrow B$ is interderivable with*

$$\lambda_{\mathbf{x}:A^1}.(f \ \mathbf{x}_0) : A^1 \rightarrow B.$$

Proof. If we suppose that $f : A \rightarrow B$, then the proof below is available, suppressing some instances of *Weak*, omitting rule labels, and performing

β -reductions when possible.

$$\frac{\frac{\frac{\vdash \text{coord}_1 : \Pi_{\mathbf{a}:A^1} \Pi_{\omega_{|\mathbf{a}|}}.A \quad \mathbf{x} : A^1 \vdash \mathbf{x} : A^1}{\mathbf{x} : A^1 \vdash (\text{coord}_1 \mathbf{x}) : \omega_{|\mathbf{x}|} \rightarrow A} \quad \vdash 0 : \omega_1}{\vdash f : A \rightarrow B} \quad \frac{\mathbf{x} : A^1 \vdash \mathbf{x}_0 : A}{\mathbf{x} : A^1 \vdash (f \mathbf{x}_0) : B} \\ \vdash \lambda_{\mathbf{x}:A^1}.(f \mathbf{x}_0) : A^1 \rightarrow B$$

Conversely, if we suppose $\lambda_{\mathbf{x}:A^1}.(f \mathbf{x}_0) : A^1 \rightarrow B$, then we can prove the following.

$$\frac{\frac{\vdash * : 1 \quad x : A \vdash x : A}{x : A \vdash \langle *, x \rangle : A^1} \quad \vdash \lambda_{\mathbf{x}:A^1}.(f \mathbf{x}_0) : A^1 \rightarrow B}{x : A \vdash (f x) : B} \\ \vdash \lambda_x.(f x) : A \rightarrow B$$

The term of the root label $\lambda_x.(f x)$ is then identical to f by η -reduction (theorem B.15). \square

Because of the interderivability in proposition C.6, I often substitute a function with its counterpart whose domain is a one-coordinate vector without comment in proofs.

Definition C.7 (Vector Concatenation). The concatenation operation on vectors

$$\bullet_{m,n} : \Pi_{m:n} \Pi_{n:n}.A^m \rightarrow A^n \rightarrow A^{m+n}$$

is axiomatized by the following:

$$(C.29) \quad \vdash \forall_{m:n} \forall_{n:n} \forall_{\mathbf{a}:A^m} \forall_{\mathbf{b}:A^n} \forall_{i:\omega_m}.(\text{coord}_{m+n}(\mathbf{a} \bullet_{m,n} \mathbf{b}) i) = \mathbf{a}_i$$

$$(C.30) \quad \vdash \forall_{m:n} \forall_{n:n} \forall_{\mathbf{a}:A^m} \forall_{\mathbf{b}:A^n} \forall_{i:\omega_n}.(\text{coord}_{m+n}(\mathbf{a} \bullet_{m,n} \mathbf{b}) m + i) = \mathbf{b}_i$$

The subscripts on the vector concatenation symbol \bullet are often dropped, and concatenations are usually further abbreviated as

$$\mathbf{a}, \mathbf{b} =_{\text{def}} \mathbf{a} \bullet \mathbf{b}$$

for vectors \mathbf{a} and \mathbf{b} .

For every type A , define $\epsilon_A : A^0$ as the empty vector on A . Then the following holds:

Proposition C.8 (Identity Under Concatenation). *For every vector $\mathbf{a} : A^n$ on A , we have $\mathbf{a} \bullet \epsilon_A = \mathbf{a}$ and $\epsilon_A \bullet \mathbf{a} = \mathbf{a}$.*

Proof. Let $\mathbf{a} : A^n$ be an n -ary vector on A and $\epsilon : A^0$ the empty vector. First note that $(\mathbf{a} \bullet \epsilon) : A^n$ and $(\epsilon \bullet \mathbf{a}) : A^n$ by definition C.7. Then by simple induction and axiom (C.29), for every $i < n$, $(\text{coord}(\mathbf{a} \bullet \epsilon) i) = \mathbf{a}_i$. A similar induction shows that $(\text{coord}(\epsilon \bullet \mathbf{a}) 0 + i) = \mathbf{a}_i$, invoking axiom (C.30) for each $i < n$. Since both $(\mathbf{a} \bullet \epsilon)$ and $(\epsilon \bullet \mathbf{a})$ agree with \mathbf{a} at every coordinate, we have $(\mathbf{a} \bullet \epsilon) = \mathbf{a} = (\epsilon \bullet \mathbf{a})$. \square

The vector concatenation function is also associative, implying that for every type A , the set of all vectors A^n where $n : \mathbb{N}$ has a monoidal structure. However, the proof of associativity is somewhat tedious, and so I spare the reader the details here since they are not directly relevant to the current purpose.

Definition C.9 (Vector Prefixes). The n -ary prefix of a vector $\mathbf{a} : A^m$ is available for all $n \leq m$ via the function

$$\text{prefix}_m : \prod_{m:\mathbb{N}} \prod_{\mathbf{a}:A^m} \prod_{n:\omega_{|\mathbf{a}|+1}} A^n,$$

axiomatized by the following:

$$\begin{aligned} &\vdash \forall_{m:\omega} \forall_{\mathbf{a}:A^m}. (\text{prefix}_m \mathbf{a} 0) = * \\ &\vdash \forall_{m:\omega} \forall_{\mathbf{a}:A^m} \forall_{n:\omega_{|\mathbf{a}|+1}}. (0 < n) \Rightarrow \forall_{i:\omega_{(\text{nat } n)}}. (\text{coord}_n (\text{prefix}_m \mathbf{a} n) i) = \mathbf{a}_i \end{aligned}$$

In words, the nullary prefix of any vector is the unit $*$, and for some vector $\mathbf{a} : A^m$ and natural number n with $0 < n \leq |\mathbf{a}|$, the vector $(\text{prefix}_m \mathbf{a} n)$ is the vector of arity n whose coordinates coincide with the first n coordinates of \mathbf{a} .

The subscript on the prefix function is usually suppressed. As shorthand, for $\mathbf{a} : A^m$, write

$$_n \mathbf{a} \stackrel{\text{def}}{=} (\text{prefix}_m \mathbf{a} n)$$

to denote the n -ary prefix of \mathbf{a} .

C.3.3 Bit Vector Selection

A more concrete demonstration of the utility of dependent types is their application to *bit vectors*, n -ary sequences of boolean values. Using dependent types, the type of a function picking out an index in a given bit vector can be defined in a way that makes it impossible for an out-of-range index to be specified.

First note that the truth value type t can be thought of as a boolean type because it has only two possible values (see equation (B.1)). Define the type of n -ary arrays of bits (truth values) using the cartesian product type t^n from definition C.3, above. Here we adopt a different strategy than the one used for vectors in definition C.3, storing the length of a bit vector rather than providing a length function for it. Since the axiom $\vdash n : \star$ is available, the dependent sum type $\Sigma_{n:n}.t^n$ can be derived from the axiom schema in definition C.3 as follows. First we derive the type t^n for some $n : n$:

(C.31)

$$\frac{\begin{array}{c} \vdots \\ n : n \vdash \lambda_i.t^i : \Pi_{i:n}.\star \end{array} \quad \frac{\vdash n : \star}{n : n \vdash n : n} \text{ (Var)} \quad \text{ (App)} \quad \frac{\vdash \star : \square \quad \vdash n : \star}{n : n \vdash \star : \square} \text{ (Weak)}}{n : n \vdash t^n : \star[n/n]} \text{ (Conv)}$$

(The proof in (C.31) has a suppressed instanced of Weak.) And next the sum type is formed:

$$\text{(C.32)} \quad \frac{\begin{array}{c} \vdots \\ \vdash n : \star \end{array} \quad \frac{n : n \vdash t^n : \star}{\vdash \Sigma_{n:n}.t^n : \star} \text{ (Sum)}}{\vdash \Sigma_{n:n}.t^n : \star} \text{ (C.31)}$$

This type provides the desired functionality: it is the type of pairs whose first component is a natural number n and whose second is a bit array of length n , and so the length of the bit array is stored along with its data.

To actually retrieve an element of a bit vector of type $\Sigma_{n:n}.t^n$, a selection function must be defined to take a bit vector v and an index i and return the element of the vector at i . The caveat immediately arises that the index may be greater than the vector's length. Such issues are sometimes handled by simply saying that the value of a length function is undefined for inputs that are out of range. But dependent type theory offers a more elegant solution.

To this end, note that with a proof closely resembling (C.1), the product $\Pi_{n:n}.\star : \square$ can now be derived, which is the kind of constructors that take a natural number to a type. It is then possible to form a constructor based on this type and the axiom in (C.22) as follows.

$$(C.33) \quad \frac{\begin{array}{c} (C.22) \\ \vdots \\ n : n \vdash \omega_i : \Pi_{i:n}.\star \end{array} \quad \frac{\frac{}{n : n \vdash n : n} \text{(Var)} \quad \vdots}{n : n \vdash \omega_n : \star} \text{(App)} \quad \frac{}{\vdash \Pi_{n:n}.\star : \square} \text{(Abs)}}{\vdash \lambda_{n:n}.\omega_n : \Pi_{n:n}.\star} \text{(Abs)}$$

(In this proof, an instance of both Weak and Conv has been suppressed, and a reduction step is not shown.) The bit vector type $\Sigma_{n:n}.t^n$ makes the following proof available:

$$(C.34) \quad \frac{\begin{array}{c} (C.32) \\ \vdots \\ \vdash \Sigma_{n:n}.t^n : \star \end{array} \text{(Var)} \quad \frac{v : \Sigma_{n:n}.t^n \vdash v : \Sigma_{n:n}.t^n}{v : \Sigma_{n:n}.t^n \vdash (\pi_1 v) : n} \text{(Proj}_1\text{)}}{\vdash \Sigma_{n:n}.t^n : \star} \text{(Proj}_1\text{)}$$

Then an instance of App gives the type of natural numbers smaller than the length of a given bit vector. The proof is as follows, with Γ abbreviating $v : \Sigma_{n:n}.t^n$:

$$(C.35) \quad \frac{\begin{array}{c} (C.33) \\ \vdots \\ \Gamma \vdash \lambda_{n:n}.\omega_n : \Pi_{n:n}.\star \end{array} \quad \begin{array}{c} (C.34) \\ \vdots \\ \Gamma \vdash (\pi_1 v) : n \end{array}}{\Gamma \vdash (\lambda_{n:n}.\omega_n (\pi_1 v)) : \star[(\pi_1 v)/n]} \text{(App)}$$

(Here again, an instance of the Weak rule has been suppressed.) This proof, whose root label reduces to $\Gamma \vdash \omega_{(\pi_1 v)} : \star$ following an instance of Conv, demonstrates that the type ω_n can be formed where n is the length of some bit vector v .

Next, the Weak rule is invoked in conjunction with the axiom that t is a type, to produce an appropriately-situated variable of the type derived in

(C.35).

$$(C.36) \quad \frac{\frac{\frac{\vdash t : \star}{v : \Sigma_{n:n}.t^n \vdash t : \star} \text{ (Weak)} \quad \frac{\frac{\frac{\vdash \Sigma_{n:n}.t^n : \star}{v : \Sigma_{n:n}.t^n \vdash \omega_{(\pi_1 v)} : \star} \text{ (C.35)}}{v : \Sigma_{n:n}.t^n, i : \omega_{(\pi_1 v)} \vdash t : \star} \text{ (Weak)}}{\vdash \Sigma_{n:n}.t^n, i : \omega_{(\pi_1 v)} \vdash t : \star} \text{ (Weak)}$$

Finally, the Prod rule is applied twice to create the type of the selection function for bit vectors.

$$(C.37) \quad \frac{\frac{\frac{\frac{\vdash \Sigma_{n:n}.t^n : \star}{v : \Sigma_{n:n}.t^n \vdash \omega_{(\pi_1 v)} : \star} \text{ (C.35)}}{\vdash \Sigma_{n:n}.t^n : \star} \text{ (C.32)} \quad \frac{\frac{\frac{\frac{\vdash \Sigma_{n:n}.t^n : \star}{v : \Sigma_{n:n}.t^n \vdash \Pi_{i:\omega_{(\pi_1 v)}}.t : \star} \text{ (C.36)}}{v : \Sigma_{n:n}.t^n \vdash \Pi_{i:\omega_{(\pi_1 v)}}.t : \star} \text{ (Prod)}}{\vdash \Pi_{v:\Sigma_{n:n}.t^n} \Pi_{i:\omega_{(\pi_1 v)}}.t : \star} \text{ (Prod)}$$

The judgment derived in (C.37) is the type of functions that take two arguments. The first argument v is a bit vector, a pair of a natural number that is the length of the vector and the data itself. Importantly, the second argument, an index into the bit vector v , depends on the length of the first argument because its type is $\omega_{(\pi_1 v)}$, the type of natural numbers less than the length of v . A function retrieving the lengths of bit vectors with this typing cannot be called with an index that is out of bounds for the vector in question, and no object-level axioms are required to enforce the boundedness constraint.

C.4 Extending Pure Type Systems with Dependent Sums

A *pure type system* (Barendregt, 1991, 1992) is a further generalization of systems like λP that allows all of the systems of Barendregt's λ -cube, all with differing expressive power, to be specified in a single framework. Barendregt's original pure type systems generalized the Ax and Prod rules of λP , but did not include dependent sum types. Fortunately, as Barthe (1995, appendix) discusses, extending pure type systems to account for

$$\begin{array}{c}
\vdash s_1 : s_2 \quad (\text{Ax}) \quad \langle s_1, s_2 \rangle \in \mathcal{A} \\
\\
\frac{\Gamma \vdash A : s_1 \quad \Gamma, x : A \vdash B : s_2}{\Gamma \vdash \Pi_{x:A}.B : s_3} \text{ (Prod)} \quad \langle s_1, s_2, s_3 \rangle \in \mathcal{R}_\Pi \\
\\
\frac{\Gamma \vdash A : s_1 \quad \Gamma, x : A \vdash B : s_2}{\Gamma \vdash \Sigma_{x:A}.B : s_3} \text{ (Sum)} \quad \langle s_1, s_2, s_3 \rangle \in \mathcal{R}_\Sigma
\end{array}$$

Figure C.2: Inference rules for pure type systems, with \mathcal{A} the set of axioms and \mathcal{R}_Π and \mathcal{R}_Σ the sets of rules.

dependent sums is straightforward and follows the original pattern used for the Prod rule.

Definition C.10 (Pure Type Systems with Sum Types). A pure type system is a set \mathcal{S} of sorts, a set $\mathcal{A} \subseteq \mathcal{S} \times \mathcal{S}$ called the *axioms*, and two sets $\mathcal{R}_\Pi \subseteq \mathcal{S} \times \mathcal{S} \times \mathcal{S}$ and $\mathcal{R}_\Sigma \subseteq \mathcal{S} \times \mathcal{S} \times \mathcal{S}$, called the Π - and Σ -rules, respectively. The typing rules are the same as those for $\lambda\mathbf{P}_\Sigma$ except with Ax, Prod, and Sum replaced by the generalized rules in figure C.2.

Everything else about the syntax of pure type systems is the same as for $\lambda\mathbf{P}_\Sigma$: the notions of variables, contexts, and conversion are all unchanged. What is new is that a single framework can specify systems with as little power as the typed lambda calculus and with as much power as the calculus of constructions.

For example, the pure type system defined based on definition C.10 with following axioms and rules is the type theory extended with cartesian products from appendix B (letting the set \mathcal{S} of sorts be $\{\star, \square\}$, as before):

$$\begin{aligned}
\mathcal{A} &=_{\text{def}} \{ \langle \star, \square \rangle \} \\
\mathcal{R}_\Pi &=_{\text{def}} \{ \langle \star, \star, \star \rangle \} \\
\mathcal{R}_\Sigma &=_{\text{def}} \{ \langle \star, \star, \star \rangle \}
\end{aligned}$$

The reason is that, with the set of rules so instantiated, Prod and Sum can only be used to derive product types of the form $\Pi_{x:A}.B : \star$ and sum types of the form $\Sigma_{x:A}.B : \star$. Nothing of the kind \square of a type constructor can be formed using this system, only nondependent functions $A \rightarrow B$ and cartesian products $A \times B$ are available.

But consider the slight elaboration of the set of rules to

$$\mathcal{R}_\Pi = \mathcal{R}_\Sigma =_{\text{def}} \{ \langle \star, \star, \star \rangle, \langle \star, \square, \square \rangle \} .$$

Then the resulting system is $\lambda\mathbf{P}_\Sigma$, because the kinds $\Pi_{x:A}.\star : \square$ and $\Sigma_{x:A}.\star : \square$ can additionally be derived. This elaboration could be continued to provide dependent sum types for all of the systems of the λ -cube. [Aspinall and Hofmann \(2005\)](#) sketch an extended proof of strong normalization for dependent sum types in an extension of LF that is essentially an instance of a pure type system as presented here with $\mathcal{R}_\Pi = \{ \langle \star, \star, \star \rangle, \langle \star, \square, \square \rangle \}$ and $\mathcal{R}_\Sigma = \{ \langle \star, \star, \star \rangle \}$.

Bibliography

- Abbott, Barbara. Presuppositions as nonassertions. *Journal of Pragmatics*, 32(10):1419–1437, 2000. doi:[10.1016/S0378-2166\(99\)00108-3](https://doi.org/10.1016/S0378-2166(99)00108-3).
- Abusch, Dorit. Presupposition triggering from alternatives. *Journal of Semantics*, 27(1):37–80, 2010. doi:[10.1093/jos/ffp009](https://doi.org/10.1093/jos/ffp009).
- Amaral, Patricia, Craige Roberts, and E. Allyn Smith. Review of *The Logic of Conventional Implicatures* by Chris Potts. *Linguistics and Philosophy*, 30(6):707–749, 2007. doi:[10.1007/s10988-008-9025-2](https://doi.org/10.1007/s10988-008-9025-2).
- AnderBois, Scott, Adrian Brasoveanu, and Robert Henderson. Crossing the appositive/at-issue meaning boundary. In *Proceedings of the 20th Conference on Semantics and Linguistic Theory*, 2010.
- Andrews, Peter B. *An Introduction to Mathematical Logic and Type Theory: To Truth Through Proof*, volume 27 of *Applied Logic Series*. Kluwer Academic Publishers, 2002.
- Aspinall, David and Martin Hofmann. Dependent types. In B. C. Pierce, editor, *Advanced Topics in Types and Programming Languages*, pages 45–86. MIT Press, Cambridge, Massachusetts, 2005.
- Barendregt, Henk. *The Lambda Calculus: Its Syntax and Semantics*, volume 103 of *Studies in Logic and the Foundations of Mathematics*. North-Holland, 1980.
- Barendregt, Henk. Introduction to generalized type systems. *Journal of Functional Programming*, 1(2):125–154, 1991.
- Barendregt, Henk. Lambda calculi with types. In S. Abramsky, D. M. Gabbay, and T. S. E. Maibaum, editors, *Handbook of Logic in Computer Science*, pages 117–309. Oxford University Press, 1992.

- Barendregt, Henk, Wil Dekkers, and Richard Statman. *Lambda Calculus with Types*. Perspectives in Logic. Cambridge University Press, 2013.
- Barker, Chris. *Possessive Descriptions*. Ph.D. thesis, University of California, Santa Cruz, 1991.
- Barker, Chris. Definite possessives and discourse novelty. *Theoretical Linguistics*, 26(3):211–228, 2000. doi:[10.1515/thli.2000.26.3.211](https://doi.org/10.1515/thli.2000.26.3.211).
- Barker, Chris, Raffaella Bernardi, and Chung-chieh Shan. Principles of interdimensional meaning interaction. In *Proceedings of the 20th Conference on Semantics and Linguistic Theory*, 2010.
- Barthe, Gilles. Extensions of pure type systems. In M. Dezani-Ciancaglini and G. Plotkin, editors, *Typed Lambda Calculi and Applications*, number 902 in Lecture Notes in Computer Science, 1995. doi:[10.1007/BFb0014042](https://doi.org/10.1007/BFb0014042).
- Barwise, Jon and Robin Cooper. Generalized quantifiers and natural language. *Linguistics and Philosophy*, 4(2):159–220, 1981. doi:[10.1007/BF00350139](https://doi.org/10.1007/BF00350139).
- Beaver, David. Have you noticed that your belly button lint colour is related to the colour of your clothing? In R. Bäuerle, U. Reyle, and T. E. Zimmermann, editors, *Presuppositions and Discourse: Essays offered to Hans Kamp*. Elsevier, 2010. doi:[10.1163/9789004253162_004](https://doi.org/10.1163/9789004253162_004).
- Beaver, David and Henk Zeevat. Accommodation. In G. Ramchand and C. Reiss, editors, *Oxford Handbook of Linguistic Interfaces*. Oxford University Press, 2007. doi:[10.1093/oxfordhb/9780199247455.013.0017](https://doi.org/10.1093/oxfordhb/9780199247455.013.0017).
- Beaver, David I. *Presupposition and Assertion in Dynamic Semantics*. CSLI Publications, 2001.
- Beaver, David I. Presupposition projection in DRT: A critical assessment. In D. Beaver, L. D. C. Martínez, B. Z. Clark, and S. Kaufmann, editors, *The Construction of Meaning*. CSLI Publications, 2002.
- Blackburn, Patrick and Johan Bos. *Representation and Inference for Natural Language: A First Course in Computational Semantics*. Volume 2 of [Blackburn and Bos 2005](#), 1999. Unpublished volume subtitled *Working with Discourse Representation Structures*.

- Blackburn, Patrick and Johan Bos. *Representation and Inference for Natural Language: A First Course in Computational Semantics*, volume 1. CSLI Publications, 2005.
- Boër, Steven E. and William G. Lycan. The myth of semantic presupposition. In A. M. Zwicky, editor, *Papers in Nonphonology*, number 21 in Working Papers in Linguistics, pages 1–90. Ohio State University Department of Linguistics, 1976.
- Bos, Johan. Implementing the binding and accommodation theory for anaphora resolution and presupposition projection. *Computational Linguistics*, 29(2):179–210, 2003. doi:[10.1162/089120103322145306](https://doi.org/10.1162/089120103322145306).
- Bos, Johan. Towards wide-coverage semantic interpretation. In *Proceedings of the Sixth International Workshop on Computational Semantics*, 2005.
- Carpenter, Bob. *Type Logical Semantics*. MIT Press, Cambridge, Massachusetts, 1997.
- Chierchia, Gennaro. Anaphora and dynamic binding. *Linguistics and Philosophy*, 15(2):111–183, 1992. doi:[10.1007/BF00635805](https://doi.org/10.1007/BF00635805).
- Chierchia, Gennaro. *The Dynamics of Meaning: Anaphora, Presupposition, and the Theory of Grammar*. University of Chicago Press, 1995.
- Chierchia, Gennaro and Sally McConnell-Ginet. *Meaning and Grammar: An Introduction to Semantics*. MIT Press, Cambridge, Massachusetts, 1990.
- Chierchia, Gennaro and Mats Rooth. Configurational notions in DR theory. In C. Jones and P. Sells, editors, *Proceedings of the 14th Meeting of the North Eastern Linguistics Society*, 1984.
- Church, Alonzo. A formulation of the simple theory of types. *Journal of Symbolic Logic*, 5(2):56–68, 1940. doi:[10.2307/2266170](https://doi.org/10.2307/2266170).
- Cooper, Robin. The interpretation of pronouns. In F. Heny and H. Schnelle, editors, *Selections from the Third Groningen Round Table*, volume 10 of *Syntax and Semantics*, pages 61–92. Academic Press, New York, 1979.
- Curry, Haskell. Some logical aspects of grammatical structure. In R. Jakobson, editor, *Structure of Language and its Mathematical Aspects*, number 12 in *Proceedings of Symposia in Applied Mathemat-*

- ics. American Mathematical Society, Providence, Rhode Island, 1961. doi:[10.1090/psapm/012/9981](https://doi.org/10.1090/psapm/012/9981).
- Curry, Haskell B. and Roert Feys. *Combinatory Logic, volume I*. Studies in Logic and the Foundations of Mathematics. North-Holland, Amsterdam, 1958.
- de Groote, Philippe. Towards abstract categorial grammars. In *Association for Computational Linguistics, 39th Annual Meeting and 10th Conference of the European Chapter, Proceedings of the Conference*, 2001. doi:[10.3115/1073012.1073045](https://doi.org/10.3115/1073012.1073045).
- de Groote, Philippe. Towards a Montagovian account of dynamics. In *Proceedings of the 16th Conference on Semantics and Linguistic Theory*, 2006.
- de Groote, Philippe and Ekaterina Lebedeva. Presupposition accommodation as exception handling. In *Proceedings of SIGDIAL 2010: The 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, 2010.
- de Groote, Philippe and Mark-Jan Nederhof, editors. *Formal Grammar*, number 7395 in Lecture Notes in Computer Science. 2012. doi:[10.1007/978-3-642-32024-8](https://doi.org/10.1007/978-3-642-32024-8).
- de Paiva, Valeria. Lineales: Algebras and categories in the semantics of linear logic. In D. Barker-Plummer, D. I. Beaver, J. van Benthem, and P. S. di Luzio, editors, *Words, Proofs, and Diagrams*. CSLI Publications, 2002.
- Dekker, Paul. Grounding dynamic semantics. In [Reimer and Bezuidenhout 2005](#), pages 484–502.
- Evans, Gareth. Pronouns, quantifiers and relative clauses. *Canadian Journal of Philosophy*, 7:467–536, 1977.
- Evans, Gareth. Pronouns. *Linguistic Inquiry*, 11(2):337–362, 1980.
- Frege, Gottlob. Über Sinn und Bedeutung. *Zeitschrift für Philosophie und philosophische Kritik*, 100:25–50, 1892. English translation titled *On Sense and Reference* in [Geach and Black 1952](#), pages 56–78.
- Gallin, Daniel. *Intensional and Higher Order Modal Logic*, volume 19 of *Mathematics Studies*. North-Holland, Amsterdam, 1975.

- Gauker, Christopher. Against accommodation: Heim, van der Sandt and the presupposition projection problem. *Philosophical Perspectives*, 22(1):171–205, 2008. doi:[10.1111/j.1520-8583.2008.00145.x](https://doi.org/10.1111/j.1520-8583.2008.00145.x).
- Geach, Peter. *Reference and Generality: An Examination of Some Medieval and Modern Theories*. Cornell University Press, Ithaca and London, 1962.
- Geach, Peter and Max Black. *Translations from the Philosophical Writings of Gottlob Frege*. Blackwell, Oxford, 1952.
- Geurts, Bart. Local satisfaction guaranteed: a presupposition theory and its problems. *Linguistics and Philosophy*, 19(3):259–294, 1996. doi:[10.1007/BF00628201](https://doi.org/10.1007/BF00628201).
- Geurts, Bart. *Presuppositions and Pronouns*, volume 3 of *Current Research in the Semantics/Pragmatics Interface*. Elsevier, 1999.
- Geuvers, Herman. The Church-Rosser property for $\beta\eta$ -reduction in typed λ -calculi. In *Proceedings of the Seventh Annual IEEE Symposium on Logic in Computer Science*, 1992. doi:[10.1109/LICS.1992.185556](https://doi.org/10.1109/LICS.1992.185556).
- Geuvers, Jan Herman. *Logics and Type Systems*. Ph.D. thesis, Universiteit Nijmegen, 1993.
- Girard, Jean-Yves. Linear logic. *Theoretical Computer Science*, 50(1):1–101, 1987. doi:[10.1016/0304-3975\(87\)90045-4](https://doi.org/10.1016/0304-3975(87)90045-4).
- Girard, Jean-Yves. Linear logic: Its syntax and semantics. In J.-Y. Girard, Y. Lafont, and L. Regnier, editors, *Advances in Linear Logic*, number 222 in London Mathematical Society Lecture Note Series, pages 1–42. Cambridge University Press, 1995.
- Girard, Jean-Yves, Andre Scedrov, and Philip J. Scott. Bounded linear logic: A modular approach to polynomial time computability. *Theoretical Computer Science*, 97(1):1–66, 1992. doi:[10.1016/0304-3975\(92\)90386-T](https://doi.org/10.1016/0304-3975(92)90386-T).
- Grice, H. P. Meaning. *The Philosophical Review*, 66(3):377–388, 1957. doi:[10.2307/2182440](https://doi.org/10.2307/2182440).
- Grice, H. Paul. Logic and conversation. In P. Cole and J. Morgan, editors, *Speech Acts*, volume 3 of *Syntax and Semantics*, pages 43–58. Academic Press, New York, 1975. Reprinted in [Martinich 2001](#), pages 165–175.

- Groenendijk, Jeroen, Theo M. V. Janssen, and Martin Stokhof, editors. *Truth, Interpretation, and Information: Selected Papers from the Third Amsterdam Colloquium*. Number 2 in Groningen-Amsterdam Studies in Semantics. Foris, Dordrecht, 1984. doi:[10.1515/9783110867602.1](https://doi.org/10.1515/9783110867602.1).
- Groenendijk, Jeroen and Martin Stokhof. Dynamic Montague grammar. In L. Kálmán and L. Pólos, editors, *Papers from the Second Symposium on Logic and Language*. Akadémiai Kiadó, 1990.
- Groenendijk, Jeroen and Martin Stokhof. Dynamic predicate logic. *Linguistics and Philosophy*, 14(1):39–100, 1991. doi:[10.1007/BF00628304](https://doi.org/10.1007/BF00628304).
- Harper, Robert, Furio Honsell, and Gordon Plotkin. A framework for defining logics. *Journal of the Association for Computing Machinery*, 40(1):143–184, 1993. doi:[10.1007/BFb0014042](https://doi.org/10.1007/BFb0014042).
- Harris, Jesse A. and Christopher Potts. Perspective-shifting with appositives and expressives. *Linguistics and Philosophy*, 32(6):532–552, 2009. doi:[10.1007/s10988-010-9070-5](https://doi.org/10.1007/s10988-010-9070-5).
- Heim, Irene. *The Semantics of Definite and Indefinite Noun Phrases*. Ph.D. thesis, University of Massachusetts, Amherst, 1982.
- Heim, Irene. File change semantics and the familiarity theory of definiteness. In R. Bäuerle, C. Schwarze, and A. von Stechow, editors, *Meaning, Use and the Interpretation of Language*. De Gruyter, Berlin, 1983a. doi:[10.1515/9783110852820.164](https://doi.org/10.1515/9783110852820.164).
- Heim, Irene. On the projection problem for presuppositions. In M. Barlow, D. Flickinger, and M. Westcoat, editors, *WCCFL2: Second Annual West Coast Conference on Formal Linguistics*. Stanford University, 1983b.
- Heim, Irene. E-type pronouns and donkey anaphora. *Linguistics and Philosophy*, 13(2):137–178, 1990a. doi:[10.1007/BF00630732](https://doi.org/10.1007/BF00630732).
- Heim, Irene. Presupposition projection. In R. van der Sandt, editor, *Presupposition, Lexical Meaning, and Discourse Processes: Workshop Reader*. University of Nijmegen, 1990b.
- Heim, Irene. Presupposition projection and the semantics of attitude verbs. *Journal of Semantics*, 9(3):183–221, 1992. doi:[10.1093/jos/9.3.183](https://doi.org/10.1093/jos/9.3.183).

- Henkin, Leon. Completeness in the theory of types. *Journal of Symbolic Logic*, 15(2):81–91, 1950. doi:[10.2307/2266967](https://doi.org/10.2307/2266967).
- Henkin, Leon. A theory of propositional types. *Fundamenta Mathematicae*, 52:323–344, 1963.
- Hepple, Mark. A compilation-chart method for linear categorial deduction. In *Proceedings of the 16th International Conference on Computational Linguistics*, 1996. doi:[10.3115/992628.992721](https://doi.org/10.3115/992628.992721).
- Hepple, Mark. An Earley-style predictive chart parsing method for Lambek grammars. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, 1999. doi:[10.3115/1034678.1034749](https://doi.org/10.3115/1034678.1034749).
- Hindley, J. Roger and Jonathan P. Seldin. *Lambda Calculus and Combinators, An Introduction*. Cambridge University Press, 2008.
- Hofmann, Martin. Syntax and semantics of dependent types. In A. M. Pitts and P. Dybjer, editors, *Semantics and Logics of Computation*, Publications of the Newton Institute, pages 79–130. Cambridge University Press, 1997. doi:[10.1017/CBO9780511526619.004](https://doi.org/10.1017/CBO9780511526619.004).
- Hyland, Martin and Valeria de Paiva. Full intuitionistic linear logic (extended abstract). *Annals of Pure and Applied Logic*, 64(3):273–291, 1993.
- Jacobs, Bart. *Categorical Logic and Type Theory*, volume 141 of *Studies in Logic and the Foundations of Mathematics*. Elsevier, 1999.
- Kadmon, Nirit. Uniqueness. *Linguistics and Philosophy*, 13(3):273–324, 1990. doi:[10.1007/BF00627710](https://doi.org/10.1007/BF00627710).
- Kamp, Hans. A theory of truth and semantic representation. In J. Groenendijk, T. Janssen, and M. Stokhof, editors, *Formal Methods in the Study of Language*. Mathematical Center, Amsterdam, 1981. Reprinted in [Groenendijk, Janssen, and Stokhof 1984](#), pages 1–42.
- Kamp, Hans and Uwe Reyle. *From Discourse to Logic: Introduction to Model-theoretic Semantics of Natural Language, Formal Logic, and Discourse Representation Theory*. Number 42 in *Studies in Linguistics and Philosophy*. Kluwer Academic Publishers, Dordrecht, 1993.

- Kanazawa, Makoto. Weak vs. strong readings of donkey sentences and monotonicity inference in a dynamic setting. *Linguistics and Philosophy*, 17(2):109–158, 1994. doi:[10.1007/BF00984775](https://doi.org/10.1007/BF00984775).
- Karttunen, Lauri. Some observations on factivity. *Papers in Linguistics*, 4(1):55–69, 1971. doi:[10.1080/08351817109370248](https://doi.org/10.1080/08351817109370248).
- Karttunen, Lauri. Presuppositions of compound sentences. *Linguistic Inquiry*, 4(2):169–193, 1973.
- Karttunen, Lauri. Presupposition and linguistic context. *Theoretical Linguistics*, 1:181–194, 1974. doi:[10.1515/thli.1974.1.1-3.181](https://doi.org/10.1515/thli.1974.1.1-3.181).
- Karttunen, Lauri. Discourse referents. In J. McCawley, editor, *Notes from the Linguistic Underground*, volume 7 of *Syntax and Semantics*, pages 363–385. Academic Press, New York, 1976. doi:[10.3115/990403.990490](https://doi.org/10.3115/990403.990490).
- Karttunen, Lauri and Stanley Peters. Conventional implicature. In C.-K. Oh and D. A. Dineen, editors, *Presupposition*, volume 11 of *Syntax and Semantics*, pages 1–56. Academic Press, New York, 1979.
- Keenan, Edward L. and Jonathan Stavi. A semantic characterization of natural language determiners. *Linguistics and Philosophy*, 9(3):253–326, 1986. doi:[10.1007/BF00630273](https://doi.org/10.1007/BF00630273).
- Kierstead, Gregory and Scott Martin. A multistratal account of the projective Tagalog evidential ‘daw’. In *Proceedings of the 22nd Conference on Semantics and Linguistic Theory*. CLC Publications, 2012.
- König, Esther. A hypothetical reasoning algorithm for linguistic analysis. *Journal of Logic and Computation*, 4(1):1–19, 1994. doi:[10.1093/logcom/4.1.1](https://doi.org/10.1093/logcom/4.1.1).
- Kripke, Saul A. Presupposition and anaphora: Remarks on the formulation of the projection problem. *Linguistic Inquiry*, 40(3):367–386, 2009. doi:[10.1162/ling.2009.40.3.367](https://doi.org/10.1162/ling.2009.40.3.367).
- Kubota, Yusuke and Wataru Uegaki. Continuation-based semantics for conventional implicatures: The case of Japanese benefactives. In *Proceedings of the 19th Conference on Semantics and Linguistic Theory*, 2009.
- Lambek, Joachim. The mathematics of sentence structure. *American Mathematical Monthly*, 65(3):154–170, 1958. doi:[10.2307/2310058](https://doi.org/10.2307/2310058).

- Lambek, Joachim and Phil J. Scott. *Introduction to Higher-Order Categorical Logic*, volume 7 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, 1986.
- Lamping, John. An algorithm for optimal lambda calculus reduction. In *Proceedings of the 17th Association for Computing Machinery Symposium on Principles of Programming Languages*, 1990. doi:[10.1145/96709.96711](https://doi.org/10.1145/96709.96711).
- Langendoen, D. Terence and Harris Savin. The projection problem for presuppositions. In C. Fillmore and D. T. Langendoen, editors, *Studies in Linguistic Semantics*. Holt, Reinhart, and Winston, New York, 1971.
- Lebedeva, Ekaterina. *Expression de la dynamique du discours à l'aide de continuations*. Ph.D. thesis, Université de Lorraine, 2012.
- Lewis, David. Adverbs of quantification. In E. Keenan, editor, *Formal Semantics of Natural Language*, pages 3–15. Oxford University Press, 1975. doi:[10.1017/CBO9780511897696.003](https://doi.org/10.1017/CBO9780511897696.003).
- Lewis, David. Scorekeeping in a language game. *Journal of Philosophical Logic*, 8:339–359, 1979. doi:[10.1007/BF00258436](https://doi.org/10.1007/BF00258436).
- Martin, Scott. Dynamic semantics in direct style, 2010. Unpublished manuscript, Ohio State University.
- Martin, Scott. Weak familiarity and anaphoric accessibility in dynamic semantics. In [de Groote and Nederhof 2012](#), pages 287–306. doi:[10.1007/978-3-642-32024-8_19](https://doi.org/10.1007/978-3-642-32024-8_19).
- Martin, Scott and Carl Pollard. A higher-order theory of presupposition. *Studia Logica*, 100(4):729–754, 2012a. doi:[10.1007/s11225-012-9427-6](https://doi.org/10.1007/s11225-012-9427-6). Special issue on logic and natural language.
- Martin, Scott and Carl Pollard. Hyperintensional dynamic semantics: Analyzing definiteness with enriched contexts. In [de Groote and Nederhof 2012](#), pages 114–129. doi:[10.1007/978-3-642-32024-8_8](https://doi.org/10.1007/978-3-642-32024-8_8).
- Martin-Löf, Per. *Intuitionistic Type Theory*. Studies in Proof Theory. Bibliopolis, 1984.
- Martin-Löf, Per. An intuitionistic theory of types. In G. Sambin and J. M. Smith, editors, *Twenty-Five Years of Constructive Type Theory*, volume 36 of *Oxford Logic Guides*. Clarendon Press, 1998.

- Martinich, A. P., editor. *The Philosophy of Language*. Oxford University Press, 2001.
- Mihaliček, Vedrana. *Serbo-Croatian Word Order: A Logical Approach*. Ph.D. thesis, Ohio State University, 2012.
- Mihaliček, Vedrana and Carl Pollard. Distinguishing phenogrammar from tectogrammar simplifies the analysis of interrogatives. In [de Groote and Nederhof 2012](#), pages 130–145. doi:[10.1007/978-3-642-32024-8_9](#).
- Montague, Richard. The proper treatment of quantification in ordinary English. In K. Hintikka, J. Moravcsik, and P. Suppes, editors, *Approaches to Natural Language*. D. Reidel, Dordrecht, 1973. doi:[10.1007/978-94-010-2506-5_10](#). Reprinted in [Thomason 1974](#), pages 247–270.
- Moortgat, Michael. Categorical type logics. In J. van Benthem and A. ter Meulen, editors, *Handbook of Logic and Language*, pages 93–177. Elsevier, Amsterdam, 1997. doi:[10.1016/B978-044481714-3/50005-9](#).
- Morrill, Glyn. *Type Logical Grammar: Categorical Logic of Signs*. Kluwer, Dordrecht, 1994.
- Muskens, Reinhard. Categorical grammar and discourse representation theory. In *Proceedings of the 15th Conference on Computational Linguistics*, 1994. doi:[10.3115/991886.991974](#).
- Muskens, Reinhard. Combining Montague semantics and discourse representation theory. *Linguistics and Philosophy*, 19(2):143–186, 1996. doi:[10.1007/BF00635836](#).
- Muskens, Reinhard. λ -grammars and the syntax-semantics interface. In R. van Rooy and M. Stokhof, editors, *Proceedings of the 13th Amsterdam Colloquium*, 2001.
- Muskens, Reinhard. Sense and the computation of reference. *Linguistics and Philosophy*, 28(4):473–504, 2005. doi:[10.1007/s10988-004-7684-1](#).
- Muskens, Reinhard. Separating syntax and combinatorics in categorical grammar. *Research on Language and Computation*, 5(3):267–285, 2007. doi:[10.1007/s11168-007-9035-1](#).
- Neale, Stephen. *Descriptions*. MIT Press, Cambridge, Massachusetts, 1990.

- Nouwen, Rick. On appositives and dynamic binding. *Research on Language and Computation*, 5(1):87–102, 2007. doi:[10.1007/s11168-006-9019-6](https://doi.org/10.1007/s11168-006-9019-6).
- Oehrle, Richard T. Term-labeled categorial type systems. *Linguistics and Philosophy*, 17(6):633–678, 1994. doi:[10.1007/BF00985321](https://doi.org/10.1007/BF00985321).
- Parigot, Michel. $\lambda\mu$ -calculus: An algorithmic interpretation of classical natural deduction. In *Proceedings of the International Conference on Logic Programming and Automated Reasoning*, number 624 in Lecture Notes in Computer Science, pages 190–201, 1992. doi:[10.1007/BFb0013061](https://doi.org/10.1007/BFb0013061).
- Parigot, Michel. On the computational interpretation of negation. In *Computer Science Logic*, number 1862 in Lecture Notes in Computer Science, pages 472–484, 2000. doi:[10.1007/3-540-44622-2_32](https://doi.org/10.1007/3-540-44622-2_32).
- Partee, Barbara H. Nominal and temporal anaphora. *Linguistics and Philosophy*, 7(3):243–286, 1984.
- Plummer, Andrew and Carl Pollard. Agnostic possible worlds semantics. In *Logical Aspects of Computational Linguistics*, number 7351 in Lecture Notes in Computer Science, pages 201–212. Springer, 2012. doi:[10.1007/978-3-642-31262-5_14](https://doi.org/10.1007/978-3-642-31262-5_14).
- Pollard, Carl. Hyperintensional questions. In *Logic, Language, Information, and Computation*, number 5110 in Lecture Notes in Computer Science, pages 272–285, 2008a. doi:[10.1007/978-3-540-69937-8_24](https://doi.org/10.1007/978-3-540-69937-8_24).
- Pollard, Carl. Hyperintensions. *Journal of Logic and Computation*, 18(2):257–282, 2008b. doi:[10.1093/logcom/exm003](https://doi.org/10.1093/logcom/exm003).
- Pollard, Carl and E. Allyn Smith. An alternative explanation for presupposition projection variability. ESSLLI Workshop on Projective Meaning, Ljubljana, Slovenia, August, 2011.
- Potts, Christopher. *The Logic of Conventional Implicatures*. Oxford University Press, 2005.
- Pullum, Geoffrey K. Formal linguistics meets the boojum. *Natural Language and Linguistic Theory*, 7:137–143, 1989.
- Reimer, Marga and Anne Bezuidenhout, editors. *Descriptions and Beyond*. Oxford University Press, 2005.

- Roberts, Craige. Modal subordination and pronominal anaphora in discourse. *Linguistics and Philosophy*, 12(6):683–721, 1989. doi:[10.1007/BF00632602](https://doi.org/10.1007/BF00632602).
- Roberts, Craige. Information structure in discourse: Towards an integrated formal theory of pragmatics. In J.-H. Yoon and A. Kathol, editors, *Papers in Semantics*, number 49 in Working Papers in Linguistics, pages 91–136. Ohio State University Department of Linguistics, 1996. Reprinted as [Roberts 2012c](#).
- Roberts, Craige. Uniqueness in definite noun phrases. *Linguistics and Philosophy*, 26(3):287–350, 2003.
- Roberts, Craige. Pronouns as definites. In [Reimer and Bezuidenhout 2005](#), pages 503–543.
- Roberts, Craige. Retrievability and incomplete descriptions, 2010. Unpublished manuscript, Ohio State University.
- Roberts, Craige. *Only*, a case study in projective meaning. In B. H. Partee, M. Glanzberg, and J. Škilters, editors, *Formal Semantics and Pragmatics: Discourse, Context, and Models*, volume 6 of *The Baltic International Yearbook of Cognition, Logic and Communication*, pages 1–56. New Prairie Press, Manhattan, Kansas, 2011. doi:[10.4148/biyclc.v6i0.1581](https://doi.org/10.4148/biyclc.v6i0.1581).
- Roberts, Craige. Accommodation in a language game, 2012a. To appear in a volume on David Lewis edited by Barry Loewer and Jonathan Schaffer for Blackwell.
- Roberts, Craige. Information structure: Afterword. *Semantics and Pragmatics*, 5(7):1–19, 2012b. doi:[10.3765/sp.5.7](https://doi.org/10.3765/sp.5.7).
- Roberts, Craige. Information structure in discourse: Towards an integrated formal theory of pragmatics. *Semantics and Pragmatics*, 5(6):1–69, 2012c. doi:[10.3765/sp.5.6](https://doi.org/10.3765/sp.5.6). Accompanying afterword in [Roberts 2012b](#).
- Roberts, Craige. Solving for interpretation, 2012d. Unpublished manuscript, Ohio State University.
- Roberts, Craige. Some types of suppositions, 2012e. Unpublished manuscript, Ohio State University.

- Roberts, Craige. *Only*, presupposition and implicature. *Journal of Semantics*, to appear.
- Roberts, Craige, Mandy Simons, David Beaver, and Judith Tonhauser. Presupposition, conventional implicature, and beyond: A unified account of projection. In N. Klinedinst and D. Rothschild, editors, *Proceedings of New Directions in the Theory of Presupposition*, 2009.
- Rooth, Mats. Noun phrase interpretation in Montague grammar, file change semantics, and situation semantics. In P. Gärdenfors, editor, *Generalized Quantifiers*. Reidel, Dordrecht, 1987. doi:[10.1007/978-94-009-3381-1_9](https://doi.org/10.1007/978-94-009-3381-1_9).
- Rothschild, Daniel. Explaining presupposition projection with dynamic semantics. *Semantics and Pragmatics*, 4(3):1–43, 2011. doi:[10.3765/sp.4.3](https://doi.org/10.3765/sp.4.3).
- Russell, Bertrand. On denoting. *Mind*, 66:479–493, 1905. doi:[10.1093/mind/XIV.4.479](https://doi.org/10.1093/mind/XIV.4.479).
- Schlenker, Philippe. Anti-dynamics: presupposition projection without dynamic semantics. *Journal of Logic, Language and Information*, 16(3):325–356, 2007. doi:[10.1007/s10849-006-9034-x](https://doi.org/10.1007/s10849-006-9034-x).
- Schubert, Lenhart K. and Francis Jeffrey Pelletier. Generically speaking, or, using discourse representation theory to interpret generics. In G. Chierchia, B. H. Partee, and R. Turner, editors, *Properties, Types and Meaning volume 2: Semantic Issues*, number 39 in *Studies in Linguistics and Philosophy*, pages 193–268. Kluwer, Dordrecht, 1989. doi:[10.1007/978-94-009-2723-0_6](https://doi.org/10.1007/978-94-009-2723-0_6).
- Simons, Mandy. On the conversational basis of some presuppositions. In *Proceedings of the 11th Conference on Semantics and Linguistic Theory*, 2001.
- Simons, Mandy, Craige Roberts, David Beaver, and Judith Tonhauser. What projects and why. In *Proceedings of the 20th Conference on Semantics and Linguistic Theory*, 2010.
- Smith, E. Allyn. *Correlational Comparison in English*. Ph.D. thesis, Ohio State University, 2010.
- Sørensen, Morton Heine and Paweł Urzyczyn. *Lectures on the Curry-Howard Isomorphism*, volume 149 of *Studies in Logic and the Foundations of Mathematics*. Elsevier, Amsterdam, 2006.

- Stalnaker, Robert. Presuppositions. *Journal of Philosophical Logic*, 2(4):447–457, 1973. doi:[10.1007/BF00262951](https://doi.org/10.1007/BF00262951).
- Stalnaker, Robert. Assertion. In P. Cole, editor, *Pragmatics*, volume 9 of *Syntax and Semantics*, pages 315–332. Academic Press, New York, 1978.
- Stalnaker, Robert. On the representation of context. *Journal of Logic, Language, and Information*, 7(1):3–19, 1998. doi:[10.1093/0198237073.003.0006](https://doi.org/10.1093/0198237073.003.0006).
- Steedman, Mark. *The Syntactic Process*. MIT Press, Cambridge, Massachusetts, 2000.
- Stone, Matthew and Bonnie Webber. Textual economy through close coupling of syntax and semantics. In *Proceedings of the Ninth International Workshop on Natural Language Generation*, 1998.
- Thomason, Richmond. A model theory for propositional attitudes. *Linguistics and Philosophy*, 4(1):47–70, 1980. doi:[10.1007/BF00351813](https://doi.org/10.1007/BF00351813).
- Thomason, Richmond H., editor. *Formal Philosophy: Selected Papers of Richard Montague*. Yale University Press, 1974.
- Thomason, Richmond H. Accommodation, meaning, and implicature: Interdisciplinary foundations for pragmatics. In P. Cohen, J. Morgan, and M. Pollack, editors, *Intentions in Communication*, pages 325–363. MIT Press, Cambridge, Massachusetts, 1990.
- Thomason, Richmond H., Matthew Stone, and David DeVault. Enlightened update: A computational architecture for presupposition and other pragmatic phenomena, 2006. Draft for the Ohio State University workshop on Presupposition and Accommodation.
- Tonhauser, Judith, David Beaver, Craige Roberts, and Mandy Simons. Towards a taxonomy of projective content. *Language*, 89(1):66–109, 2013. doi:[10.1353/lan.2013.0001](https://doi.org/10.1353/lan.2013.0001).
- Troelstra, Anne S. *Lectures on Linear Logic*. Number 29 in CSLI Lecture Notes. CSLI Publications, 1992.
- van der Sandt, Rob A. Presupposition projection as anaphora resolution. *Journal of Semantics*, 9(4):333–377, 1992. doi:[10.1093/jos/9.4.333](https://doi.org/10.1093/jos/9.4.333).

- van Eijck, Jan and Christina Unger. *Computational Semantics with Functional Programming*. Cambridge University Press, 2010.
- Venhuizen, Noortje J., Johan Bos, and Harm Brouwer. Parsimonious semantic representations with projection pointers. In *Proceedings of the Tenth International Workshop on Computational Semantics*, 2013.
- Visser, Albert. The donkey and the monoid: Dynamic semantics with control elements. *Journal of Logic, Language and Information*, 11(1):107–131, 2002. doi:[10.1023/A:1013026830239](https://doi.org/10.1023/A:1013026830239).
- von Fintel, Kai. What is presupposition accommodation, again? *Philosophical Perspectives*, 22(1):137–170, 2008. doi:[10.1111/j.1520-8583.2008.00144.x](https://doi.org/10.1111/j.1520-8583.2008.00144.x).
- Yetter, David N. Quantales and (noncommutative) linear logic. *Journal of Symbolic Logic*, 55(1):41–64, 1990. doi:[10.2307/2274953](https://doi.org/10.2307/2274953).
- Zeevat, Henk. Presuppositions and accommodation in update semantics. *Journal of Semantics*, 9(4):379–412, 1992. doi:[10.1093/jos/9.4.379](https://doi.org/10.1093/jos/9.4.379).

Index of Citations

- Abbott (2000), [39](#), [52](#)
 Abusch (2010), [17](#), [39](#), [52](#)
 Amaral et al. (2007), [2](#), [18](#), [19](#), [31](#), [37](#),
[183](#), [204](#), [206](#), [218](#), [228](#), [229](#),
[232](#)
 AnderBois et al. (2010), [154](#), [183](#)
 Andrews (2002), [7](#), [103](#), [251](#), [257](#), [258](#),
[263](#), [265](#), [266](#), [278](#)
 Aspinall and Hofmann (2005), [8](#), [267](#),
[289](#)
 Barendregt et al. (2013), [251](#), [260](#)
 Barendregt (1980), [257](#), [260](#)
 Barendregt (1991), [8](#), [267](#), [271](#), [287](#)
 Barendregt (1992), [8](#), [267](#), [268](#), [271](#),
[287](#)
 Barker et al. (2010), [183](#), [229](#), [233](#)
 Barker (1991), [29](#)
 Barker (2000), [30](#), [144](#)
 Barthe (1995), [268](#), [271](#), [287](#)
 Barwise and Cooper (1981), [71](#), [110](#),
[170](#)
 Beaver and Zeevat (2007), [53](#)
 Beaver (2001), [3](#), [26](#), [49](#), [88](#), [90](#), [91](#),
[127](#), [130](#), [138](#), [150](#), [154](#), [181](#),
[231](#), [232](#), [238](#), [241](#)
 Beaver (2002), [182](#), [238](#)
 Beaver (2010), [39](#), [52](#)
 Blackburn and Bos (1999), [3](#), [7](#), [154](#),
[238](#), [239](#)
 Blackburn and Bos (2005), [154](#)
 Boër and Lycan (1976), [39](#), [44](#)
 Bos (2003), [3](#), [7](#), [154](#), [182](#), [238](#), [239](#)
 Bos (2005), [154](#), [238](#), [239](#)
 Carpenter (1997), [251](#)
 Chierchia and McConnell-Ginet (1990),
[13](#)
 Chierchia and Rooth (1984), [154](#)
 Chierchia (1992), [88](#), [154](#), [166](#), [169](#),
[171](#)
 Chierchia (1995), [3](#), [6](#), [31](#), [49](#), [88](#), [154](#),
[155](#), [166](#), [169–172](#)
 Church (1940), [7](#), [251](#)
 Cooper (1979), [90](#), [155](#)
 Curry and Feys (1958), [259](#)
 Curry (1961), [4](#), [5](#), [55](#), [86](#)
 Dekker (2005), [90](#)
 Evans (1977), [155](#)
 Evans (1980), [90](#), [155](#)
 Frege (1892), [12](#), [13](#), [19](#)
 Gallin (1975), [68](#)
 Gauker (2008), [39](#), [52](#)
 Geach (1962), [89](#), [118](#)
 Geurts (1996), [181](#)
 Geurts (1999), [49](#), [51](#), [90](#), [181](#), [238](#)
 Geuvers (1992), [270](#), [271](#)
 Geuvers (1993), [270](#), [271](#)
 Girard et al. (1992), [243](#), [244](#)
 Girard (1987), [7](#), [243](#), [245](#)
 Girard (1995), [243](#), [245](#)
 Grice (1957), [12](#), [25](#)

- Grice (1975), 1, 2, 4, 11, 25, 39, 48, 49, 51, 134, 145, 146, 231, 240
- Groenendijk and Stokhof (1990), 3, 31, 88, 127, 130, 153, 196, 231, 233
- Groenendijk and Stokhof (1991), 3, 88, 127, 130, 153, 155, 156, 184, 231, 233
- Harper et al. (1993), 267
- Harris and Potts (2009), 2, 18, 31, 37
- Heim (1982), 2, 3, 6, 21–25, 31, 88–92, 95, 103, 127, 130, 146, 153, 156, 161, 164, 166, 180, 181, 231–233, 241
- Heim (1983a), 31, 88, 153, 181, 232
- Heim (1983b), 14, 39, 49, 51, 90, 181
- Heim (1990a), 155
- Heim (1990b), 26, 149, 150, 152
- Heim (1992), 88, 90, 153, 232
- Henkin (1950), 7, 8, 251, 263, 265, 266
- Henkin (1963), 7, 251, 255
- Hepple (1996), 239, 240
- Hepple (1999), 239, 240
- Hindley and Seldin (2008), 267
- Hofmann (1997), 268
- Hyland and de Paiva (1993), 243, 244
- Jacobs (1999), 268
- König (1994), 240
- Kadmon (1990), 169
- Kamp and Reyle (1993), 88, 130, 153
- Kamp (1981), 3, 6, 22, 31, 88, 89, 91, 92, 127, 130, 146, 153, 166, 231, 233, 238
- Kanazawa (1994), 6, 31, 166, 168, 169, 171, 172
- Karttunen and Peters (1979), 4, 6, 7, 11, 49, 129, 131, 137, 180, 183, 184, 218, 219, 221–223, 229, 230, 232
- Karttunen (1971), 17, 43
- Karttunen (1973), 17, 49
- Karttunen (1974), 3, 14, 39, 49, 51, 52, 89, 180, 241
- Karttunen (1976), 21, 22, 31, 89, 93, 180
- Keenan and Stavi (1986), 170
- Kierstead and Martin (2012), 91, 128, 183, 186
- Kripke (2009), 26, 27, 152
- Kubota and Uegaki (2009), 183, 229, 233
- Lambek and Scott (1986), 69, 260
- Lambek (1958), 55, 57, 87
- Lamping (1990), 240
- Langendoen and Savin (1971), 13, 49
- Lebedeva (2012), 154
- Lewis (1975), 166
- Lewis (1979), 2, 3, 14, 28, 39, 49, 51–53, 89, 180
- Martin and Pollard (2012a), 25, 43, 49, 91, 128, 135, 154
- Martin and Pollard (2012b), 49, 91, 128, 135, 154
- Martin-Löf (1984), 267
- Martin-Löf (1998), 267
- Martinich (2001), 2, 145, 146
- Martin (2010), 91
- Martin (2012), 49, 91, 128, 130, 143, 153, 154, 156, 157
- Mihaliček and Pollard (2012), 55, 87
- Mihaliček (2012), 55
- Montague (1973), 62, 68, 71, 76, 87, 88, 110, 131, 241
- Moortgat (1997), 55, 87
- Morrill (1994), 55, 87

- Muskens (1994), 3, 88, 91, 154, 231, 241
 Muskens (1996), 3, 88, 91, 127, 154, 231, 241
 Muskens (2001), 4, 55, 60
 Muskens (2005), 62
 Muskens (2007), 4, 55, 60
 Neale (1990), 155
 Nouwen (2007), 6, 129, 154, 183, 196, 219, 227–230, 233
 Oehrle (1994), 56, 71
 Parigot (1992), 91
 Parigot (2000), 91
 Partee (1984), 166
 Plummer and Pollard (2012), 5, 62, 65, 67–69
 Pollard and Smith (2011), 2, 11, 18, 20, 24, 29, 30, 34, 39, 43, 52, 144
 Pollard (2008a), 62, 68, 69, 87
 Pollard (2008b), 62, 68
 Potts (2005), 1–6, 11, 14, 17, 18, 20, 21, 31, 35–37, 48–50, 54, 131, 134, 180–186, 195, 196, 199, 203, 204, 216, 218, 219, 229, 230, 232, 233, 240, 241
 Roberts et al. (2009), 2, 11, 13, 16, 21, 27, 39, 52
 Roberts (1989), 19, 20, 46, 154
 Roberts (1996), 11, 147, 241
 Roberts (2003), 6, 24, 25, 31, 122, 130, 156, 160, 181
 Roberts (2005), 25, 31, 137, 155, 156, 160, 161, 164, 172
 Roberts (2010), 25, 27
 Roberts (2011), 53
 Roberts (2012a), 52, 53
 Roberts (2012c), 11
 Roberts (2012d), 24, 25
 Roberts (2012e), 11
 Roberts (to appear), 53
 Rooth (1987), 31, 166, 167
 Rothschild (2011), 90
 Russell (1905), 25, 155
 Schlenker (2007), 90
 Schubert and Pelletier (1989), 167
 Simons et al. (2010), 2, 4, 11, 12, 39, 52, 231, 240
 Simons (2001), 39, 44, 52
 Smith (2010), 55
 Stalnaker (1973), 49
 Stalnaker (1978), 2, 14, 39, 49, 51
 Stalnaker (1998), 28
 Steedman (2000), 55, 87
 Stone and Webber (1998), 25, 137
 Sørensen and Urzyczyn (2006), 253, 260, 267
 Thomason et al. (2006), 53
 Thomason (1980), 62
 Thomason (1990), 52
 Tonhauser et al. (2013), 21, 23, 24, 27, 33, 34, 39
 Troelstra (1992), 7, 243–246, 250
 Venhuizen et al. (2013), 154, 238
 Visser (2002), 196
 Yetter (1990), 245
 Zeevat (1992), 33, 34
 de Groote and Lebedeva (2010), 51, 130, 154
 de Groote (2001), 4, 55
 de Groote (2006), 3, 88, 91, 127, 128, 130, 154, 231, 241
 de Paiva (2002), 7, 243, 245
 van Eijck and Unger (2010), 3, 7, 127, 154, 238, 239

van der Sandt (1992), [49](#), [51](#), [131](#), [181](#),
[182](#), [237](#), [238](#)
von Fintel (2008), [28](#), [52](#)