Learning and the language of thought

by

Steven Thomas Piantadosi

Submitted to the Department of Brain and Cognitive Sciences
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Cognitive Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2011

Author . . . .
Department of Brain and Cognitive Sciences
September 1, 2011

Certified by . . . .
Edward Gibson
Professor of Cognitive Sciences
Thesis Supervisor

Accepted by. . . . .
Earl Miller
Chairman, Department Committee on Graduate Theses

Learning and the language of thought

by

Steven Thomas Piantadosi

Submitted to the Department of Brain and Cognitive Sciences
on September 1, 2011, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Cognitive Science

Abstract

This thesis develops the hypothesis that key aspects of learning and development can be understood as rational statistical inferences over a compositionally structured representation system, a *language of thought* (LOT) (Fodor, 1975). In this setup, learners have access to a set of primitive functions and learning consists of composing these functions in order to created structured representations of complex concepts. We present an inductive statistical model over these representations that formalizes an optimal Bayesian trade-off between representational complexity and fit to the observed data. This approach is first applied to the case of number-word acquisition, for which statistical learning with a LOT can explain key developmental patterns and resolve philosophically troublesome aspects of previous developmental theories. Second, we show how these same formal tools can be applied to children's acquisition of quantifiers. The model explains how children may achieve adult competence with quantifiers' literal meanings and presuppositions, and predicts several of the most-studied errors children make while learning these words. Finally, we model adult patterns of generalization in a massive concept-learning experiment. These results provide evidence for LOT models over other approaches and provide quantitative evaluation of different particular LOTs.

Thesis Supervisor: Edward Gibson
Title: Professor of Cognitive Sciences

Acknowledgments

This work would not have been possible without the support, enthusiasm, teaching, and guidance of Ted Gibson and Josh Tenenbaum. I am greatly indebted to the time and effort they have put into developing both this work and my life as a scientist.

Noah Goodman has been a key collaborator on all of these projects and has importantly and crucially shaped my thinking on these topics. I am also grateful to Irene Heim for providing feedback and crucial discussions during the course of my thesis. Many others have contributed greatly to this work through discussions and feedback, including Lance Rips, Sue Carey, Ken Wexler, Justin Halberda, Liz Spelke, Celeste Kidd, Rebecca Saxe, Avril Kenney, Ev Fedorenko, Leon Bergen, Dave Barner, Eyal Dechter, Jesse Snedeker, Ed Vul, Brenden Lake, Andreas Stuhlmüller and members of CoCoSci and TedLab at MIT. While at MIT, I have been extremely grateful for the opportunity to collaborate on other projects with Hal Tily, Ev Fedorenko, Rebecca Saxe, Dick Aslin, and Celeste Kidd. I am extraordinarily grateful to Celeste and Avril for providing detailed comments on drafts of this work.

This work would not have been possible without the technological generosity of several people and projects. From top to bottom, the research projects in this thesis use *free software*, for which I owe extreme gratitude to the Free Software Foundation. The implementations in this thesis use *Ikarus Scheme* and a variant, *Vicare Scheme*, for which I am indebted to Abdulaziz Ghuloum and Marco Maggi. The Scheme typsetting is done with SLaTeX by Dorai Sitaram. I am indebted to David Wingate for providing support to the CoCoSci cluster.

Many have contributed to making MIT the best of all possible worlds to work in: Tim Brady, Ed Vul, David Gray, Talia Konkle, John Kraemer, Todd Thompson, Michael Frank, Melissa Troyer, Melissa Kline, Lauren Schmidt, Tim O'Donnell, Dan Roy, Chris Baker, Tomer Ullman, Barbara Hidalgo-Sotelo, Danny Dilks, John McCoy, Brenden Lake, Liz Bonawitz, Mara Breen, and Amy Perfors. Denise Heintze deserves special thanks for the work and care she puts into the graduate program. I thank Will Macfarlane and Katie Gradowski for keeping me sane, occasionally distracted, and full of ice cream. Marlene

Jillespie has been an excellent companion for exploring R and Davis Square, and also prevented Galileo from dying of loneliness during this thesis. Benjamin Ellis deserves most of the credit for getting me interested in semantics and has been the best source for debates and answers to technical questions about the history and philosophy of mathematics, logic, and human anatomy.

Many great teachers who I have never sufficiently thanked have significantly influenced my own cognitive development: Ron Miller, Marty Stranathan, Steve Umstead, John Wagner, Kathleen Jones, and Robert Kirkpatrick, among others. Several people deserve special thanks for their generosity in taking time to promote the undergraduate and high school research that I was involved in: Karl Petersen, Jim Crutchfield, Jen Smith, and Bruce Bochner.

My most important thanks go to my family. My grandfather Claude brought the family into the research lab, and my other grandparents have provided us with innumerable opportunities. My grandmother Irene and Uncle Sean fostered a love of language. My parents have been incredibly supportive and enthusiastic, and great models themselves. Annie, Pat, and Cecelia, and Celeste's family have been kind and supportive throught graduate school and this thesis. Celeste has been the best companion in life and work I could ask for.

This thesis is dedicated to the two best things in this world: my family and Wikipedia.

blue headed monkey / goddess of the curious / stay with me through this
irene roach

Contents

List of Figures

List of Tables

Chapter 1

Foreword

Language learning involves integrating numerous cognitive capacities—our ability for structured representations, statistical learning, compositional thoughts, and abstract concepts. However, our most basic ideas about language acquisition and development are fragmented across subdisciplines. Many theories that work with detailed linguistic structures lack a workable notion of learning (Wexler & Culicover, 1983; J. Dresher & Elan, 1990; Gibson & Wexler, 1994; Niyogi & Berwick, 1996; Fodor, 1998b; B. Dresher, 1999; Kohl, 1999; Sakas & Fodor, 2001; Yang, 2002), instead falling back on very simple parameter-setting accounts, which are only capable of using simple environmental triggers to change representations. In this way they do not capture the powerful inductive and statistical capacities of even infant learners (e.g., Xu & Garcia, 2008; Xu & Denison, 2009; Téglás et al., 2011; Kidd, Piantadosi, & Aslin, under review). Many computational models that have tackled language learning more directly—either connectionist (e.g., Rumelhart & McClelland, 1987; Elman, 1990, 1993; Gasser & Smith, 1998) or otherwise (e.g Frank, Goodman, & Tenenbaum, 2007a; Xu & Tenenbaum, 2007; Yu & Ballard, 2007)—lack sufficiently structured representations for even elementary linguistic compositionality[1]. Indeed, many of these models only associate words with events, actions, or properties, and thus fail to address the complex compositional structures of language that make it distinctive in animal cognition. Similarly, empirical studies of statistical learning (Saffran, Aslin,

[1] Exceptions might be Siskind (1996), and more recently Bod (2009),Zettlemoyer and Collins (2005), Liang, Jordan, and Klein (2009), Liang, Jordan, and Klein (2011), Kwiatkowski, Goldwater, and Steedman (2009), and Piantadosi, Goodman, Ellis, and Tenenbaum (2008).

& Newport, 1996; Aslin, Saffran, & Newport, 1998; Gomez & Gerken, 1999; Marcus, Vijayan, Bandi Rao, & Vishton, 1999; Saffran, Johnson, Aslin, & Newport, 1999; Kirkham, Slemmer, & Johnson, 2002; Fiser & Aslin, 2002) have yet to tackle the complex types of concepts and representations necessary for all of language learning. Finally, theories of the innate conceptual *core* that human learners bring to development (Spelke, 2003, 2004; Spelke & Kinzler, 2007; Carey, 2009) are not yet integrated with computational theories of learning. If infants' early knowledge is the foundation for adults' later representations, there is not yet a computational account of how early abilities might be elaborated into rich cognitive systems.

The goal of this thesis is to show how all of these aspects of cognition can be combined into a unified computational framework—in short, to show how learners can induce the types of structures that are necessary for complex cognitive processes like those in language, using only a simple set of formalized core abilities. This work elaborates a middle ground between classically *nativist* and *empiricist* approaches, attempting to draw on the strengths of each theoretical viewpoint. Whatever representational system children begin with, it must be powerful enough to support the eventual creation of the rich types of structures and computational processes of adult cognition. At the same time, much of adult knowledge is not known to young children; natural numbers are one case where children appear to actively construct a novel representation system (Carey, 2009). Theories of cognition must account for both of these types of facts—the initial state that allows for eventual richness, and the learning that is clearly at work in many areas of development.

The starting point for our approach is Fodor (1975)'s *language of thought* (LOT) hypothesis. This theory posits that a structured, compositional representation system provides a substrate for thinking (see also Boole, 1854), and has been argued to explain key properties of cognitive processes (Fodor & Pylyshyn, 1988). For our purposes, the LOT provides a set of innately specified core concepts, and their means of combination. Following a standard approach in semantics (Heim & Kratzer, 1998; Steedman, 2000), we treat core primitives as *functions* which can be composed to form representations of more complex concepts (see also Siskind, 1996). This setup formalizes the components of a learning theory: learners are innately given a set of core primitive functions—typically ones that perform simple logical or set-theoretic operations, such as conjunction, disjunction, set-union, and object individuation. Learning consists of composing these functions in novel ways to create representations that explain observed data. For instance, if the meaning of a function word can be characterized using set-theoretic operations, then the task of the learner is to work backwards from cross-situational usages of the word to infer what composition of elementary set operations it must denote.

The key assumption of this framework is that learning consists of manipulating explicit representations in a syntactically constrained language, much like that in mathematics and logic (e.g., Boole, 1854), semantic theories (Heim & Kratzer, 1998; Steedman, 2000), or functional programming languages like Scheme (Abelson & Sussman, 1996). This distinguishes the current approach from "triggering" approaches to language acquisition (e.g., Gibson & Wexler, 1994) in that for our models, the specific structures being learned need not be innately specified and most potential hypotheses do not need to be actively represented by learners. Indeed, we show that strongly constraining the space of possible meanings does not substantially aid learning. Our approach is also distinguished from classically empiricist models that address learning using arguably plausible representational and implementational assumptions (e.g., Rumelhart & McClelland, 1986; Elman, 1997): we focus on computational-level accounts (Marr, 1982) of inductive phenomena without addressing issues of neural implementation. Connectionist approaches have, however, attempted to understand how hierarchical structures like those in our LOT models might be encoded with neurally plausible representations (Smolensky & Legendre, 2006). Our work builds on previous computational work in language learning that has attempted to learn rich linguistic structures by "building in" relatively minimal structural components (Bod, 2009; O'Donnell, Tenenbaum, & Goodman, 2009; Perfors, Tenenbaum, & Regier, 2011). Our approach corresponds to empiricism about complex conceptual knowledge and representations, combined with nativism about scaffolding that allows for structured representations—the syntax and primitives of the language of thought.

The assumptions of this setup are simultaneously trivial and consequential. Trivially, any theory that posits that development is driven by combining and reusing early abilities must specify a means of combination. Here, this is chosen to be perhaps the simplest means of combination that is also powerful: function composition. Our formalism that captures this compositionality, *lambda calculus* (Church, 1936), builds in only a few simple—almost vacuous—rules for composition (see Hindley & Seldin, 1986). But including functional composition is tremendously consequential for a cognitive or developmental theory since it allows any computable function to be expressed. This approach gives a powerful tool for developmental theories, as it allows learners to consider hypotheses of arbitrary computational complexity. The right learning theory in such a system has the potential to be far-reaching, providing a computational framework for explaining people's remarkably productive cognitive capacities. The idea of learning in computationally powerful systems builds off of recent work examining the learnability of language from the perspective of algorithmic information theory (Chater & Vitányi, 2007). The present work takes their theoretical idea—based on inferring bit strings that describe Turing machines—and implements it in a real *cognitive* theory that expresses computation using developmentally plausible primitives.

The first paper in this thesis studies the case of number-word acquisition in detail. Number words are interesting because their acquisition seems to involve a dramatic conceptual shift (Carey, 2009), in which children's number-word understanding changes qualitatively and fundamentally (e.g., Wynn, 1992). In this conceptual change, children go from successively understanding only the first few number words to being able to use counting to determine cardinality. Carey (2009) proposed these developmental patterns result from children inferring that "one more" on their counting list corresponds to "one more" cardinality. However, Carey's account has been argued to be philosophically incoherent, incapable of explaining in what sense numerical representations may be created without actually presupposing them (Rips, Asmuth, & Bloomfield, 2006, 2008; Rips, Bloomfield, & Asmuth, 2008). We provide an implemented LOT learning model that discovers number-word meanings by composing set-theoretic primitives along the lines of Carey's proposal. This model shows developmental patterns much like those that children exhibit, eventually arriving at a recursive system of numerical meaning. The model is also capable of learning other types of systems, such as those required for learning singular/plural distinctions, or more interestingly structured modular systems, like those discussed by Rips et al. (2006).

This work shows how a LOT learning model can explain developmental patterns, and also how Carey's general approach can be made philosophically and computationally sound.

The second paper in this thesis addresses natural language semantics more directly by presenting a learning model for quantifiers. Quantifiers are often taken to denote relations between sets (see also Montague, 1973; Barwise & Cooper, 1981; Keenan & Stavi, 1986; Keenan & Westerståhl, 1997; Heim & Kratzer, 1998), and their meaning is expressed by semanticists in a logical representation language. Quantifiers are especially interesting for statistical learning theories because their meanings are abstract, and often involve subtle presuppositional and pragmatic content. We implement a learning model that is capable of learning these types of representations—including presuppositions—from positive evidence alone. We present a simple proof that the model is theoretically always capable of recovering the correct meanings, and also show that the implemented model does so with a developmentally plausible amount of data. This implementation allows us to test various restrictions on the space of quantifier meanings, including "maximally nativist" theories in which only the correct set of quantifier meanings is innately specified, to learning in a space of only conservative quantifiers (Keenan & Stavi, 1986; Barwise & Cooper, 1981), and finally to a full, unrestricted hypothesis space. We show that learning in the full, unrestricted space is not substantially harder than the maximally nativist space, nor is it substantially harder than learning in the space of conservative quantifiers. We compare errors made by the model to patterns observed developmentally in the learning of "the" (Wexler, 2011), and "every" (Roeper, Strauss, & Pearson, 2004; Philip, 1995), and show that the learning model makes similar patterns of mistakes. This provides a domain-general account of these errors based on idealized statistical learnability. We also contrast our learning model to previous theories of quantifier learning based on finite-state automata (e.g., Clark, 1996), which require either positive and negative evidence, or cannot provably learn all quantifier meanings (Tiede, 1999).

One advantage of studying development as inductive learning in a LOT is that doing so allows for some flexibility in theorizing. One can write down any hypothetical LOT and see its consequences for learning. This has recently allowed LOT-learning theories to be applied to explain a wide range of developmental phenomena, including those outside of natural language, such as learning family-tree relations and theories like magnetism (Katz, Goodman, Kersting, Kemp, & Tenenbaum, 2008; Kemp, Goodman, & Tenenbaum, 2008a; Goodman, Ullman, & Tenenbaum, 2009; Ullman, Goodman, & Tenenbaum, 2010). Most of this work shows how learning could proceed when assuming a particular set of primitive components. In the third paper, we extend this approach to quantitatively compare different LOTs in the same domain. In a massive concept-learning experiment, we taught subjects rule-based concepts on sets of objects, ranging from simple Boolean predicates (*circle or red*) to predicates involving quantification (*at least one other object in a set is the same color*). This work extends previous research on Boolean concept learning (Shepard, Hovland, & Jenkins, 1961; Feldman, 2000; Goodman, Tenenbaum, Feldman, & Griffiths, 2008) to the types of concepts that are likely necessary for natural language semantics—in particular function words which manipulate and quantify over sets. Any concept can be computed or represented in a huge number of ways; for instance, all Boolean concepts can be written using standard logical connectives (*and*, *or*, *not*), or using only a single universal logical connective (*nand*, or not-and). Or, one might imagine a system full of a rich set of logical connectives including, perhaps, logical implication (*implies*) or biconditional (*iff*). Similarly, expressions with quantifiers may employ variously rich or simplified systems, ranging from a single existential or universal quantifier, to rich types of first- or higher-order quantification. In each case, a set of logical operations represents a specific *representational theory* that the study of rule-based concept representation and learning should aim to discover. By implementing a learning model and a Bayesian data analysis model, we are able to take the learning experiment and produce a "score" for any hypothesized language, corresponding to its ability to predict human learning curves. Through this, we are able to provide evidence against intuitively implausible bases such as the *nand*-basis, and can test the cognitive plausibility of several interesting representation systems that have been suggested in cognitive science and AI. In general, we find that representational systems with non-restricted syntactic forms, a rich set of primitive connectives, and quantification can best explain human learning. This work moves the LOT from philosophical ground to a firm empirical basis, and the experiments provide a compelling data set for comparing different paradigmatic approaches in cognitive science.

Chapter 2

Bootstrapping in a language of thought: a formal model of conceptual change in number word learning[1]

2.1 Introduction

> *"We used to think that if we knew one, we knew two, because one and one are two. We are finding that we must learn a great deal more about 'and'."* [Sir Arthur Eddington]

Cognitive development is most remarkable where children appear to acquire genuinely novel concepts. One particularly interesting example of this is the acquisition of number words. Children initially learn the count list *"one"*, *"two"*, *"three"*, up to *"six"* or higher, without knowing the exact numerical meaning of these words (Fuson, 1988). They then progress through several *subset-knower* levels, successively learning the meaning of *"one"*, *"two"*, *"three"* and sometimes *"four"* (Wynn, 1990, 1992; Sarnecka & Lee, 2009; Lee & Sarnecka, 2010b, 2010a). Two-knowers, for example, can successfully give one or two objects when asked, but when asked for three or more will simply give a handful of objects, even though they can recite much more of the count list.

[1] This work is joint with Noah D. Goodman and Joshua B. Tenenbaum.

After spending roughly a year learning the meanings of the first three or four words, children make an extraordinary conceptual leap. Rather than successively learning the remaining number words on the count list—up to infinity—children at about age 3;6 suddenly infer all of their meanings at once. In doing so, they become cardinal-principal (CP) knowers, and their numerical understanding changes fundamentally (Wynn, 1990, 1992). This development is remarkable because CP-knowers discover the abstract relationship between their counting routine and number-word meanings: they know how to count and how their list of counting words relates to numerical meaning. This learning pattern cannot be captured by simple statistical or associationist learning models which only track co-occurrences between number words and sets of objects. Under these models, one would expect that number words would continue to be acquired gradually, not suddenly as a coherent conceptual system. Rapid change seems to require a learning mechanism which comes to some knowledge that is more than just associations between words and cardinalities.

We present a formal learning model which shows that statistical inference over a sufficiently powerful representational space can explain why children follow this developmental trajectory. The model uses several pieces of machinery, each of which has been independently proposed to explain cognitive phenomena in other domains. The representational system we use is lambda calculus, a formal language for compositional semantics (e.g., Heim & Kratzer, 1998; Steedman, 2000), computation more generally (Church, 1936), and other natural-language learning tasks (Zettlemoyer & Collins, 2005, 2007; Piantadosi et al., 2008). The core inductive part of the model uses Bayesian statistics to formalize what inferences learners should make from data. This involves two key parts: a likelihood function which measures how well hypotheses fit observed data, and a prior which measures the complexity of individual hypotheses. We use simple and previously proposed forms of both. The model uses a likelihood function that uses the *size principle* (Tenenbaum, 1999) to penalize hypotheses which make overly broad predictions. Frank, Goodman, and Tenenbaum (2007b) proposed that this type of likelihood function is important in cross-situational word learning and Piantadosi et al. (2008) showed that it could solve the *subset problem* in learning compositional semantics. The prior is from the *rational rules* model of Goodman et al. (2008), which first linked probabilistic inference with formal, compositional, representations. The prior assumes that learners prefer simplicity and re-use in compositional hypotheses and has been shown to be important in accounting for human rule-based concept learning.

Our formal modeling is inspired by the bootstrapping theory of Carey (2009), who proposes that children observe a relationship between numerical quantity and the counting routine in early number words, and use this relationship to inductively define the meanings of later number words. The present work offers several contributions beyond Carey's formulation. Bootstrapping has been criticized for being too vague (Gallistel, 2007), and we show that it can be made mathematically precise and implemented by straightforward means[2]. Second, bootstrapping has been criticized for being incoherent or logically circular, fundamentally unable to solve the critical problem of inferring a discrete infinity of novel numerical concepts (Rips et al., 2006; Rips, Asmuth, & Bloomfield, 2008; Rips, Bloomfield, & Asmuth, 2008). We show that this critique is unfounded: given the assumptions of the model, the correct numerical system can be learned while still considering conceptual systems much like those suggested as possible alternatives by Rips, Asmuth and Bloomfield. The model is capable of learning conceptual systems like those they discuss, as well as others that are likely important for natural language. We also show that the model robustly gives rise to several qualitative phenomena in the literature which have been taken to support bootstrapping: the model progresses through three or four distinct subset knower-levels (Wynn, 1990, 1992; Sarnecka & Lee, 2009; Lee & Sarnecka, 2010b, 2010a), does not assign specific numerical meaning to higher number words at each subset-knower level (Condry & Spelke, 2008), and suddenly infers the meaning of the remaining words on the count list after learning *"three"* or *"four"* (Wynn, 1990, 1992; Sarnecka & Lee, 2009; Lee & Sarnecka, 2010b, 2010a).

This modeling work demonstrates how children might combine statistical learning and rich representations to create a novel conceptual system. Because we provide a fully implemented model which takes naturalistic data and induces representations of numerosity, this work requires making a number of assumptions about facts which are under-determined by the experimental data. This means that the model provides *at minimum* an existence proof

[2] Running code is available from the first author.

for how children might come to numerical representations. However, one advantage of this approach is that it provides a computational platform for testing multiple theories within this same framework—varying the parameters, representational system, and probabilistic model. We argue that all assumptions made are computationally and developmentally plausible, meaning that the particular version of the model presented here provides a justifiable working hypothesis for how numerical acquisition might progress.

2.2 The bootstrapping debate

Carey (2009) argues that the development of number meanings can be explained by *(Quinian) bootstrapping*. Bootstrapping contrasts with both associationist accounts and theories that posit an innate *successor function* that can map a representation of a number N onto a representation of its successor $N+1$ (Gallistel & Gelman, 1992; R. Gelman & Gallistel, 1978; Leslie, Gelman, & Gallistel, 2008). In Carey's formulation, early number-word meanings are represented using mental models of small sets. For instance two-knowers might have a mental model of "one" as $\{X\}$ and "two" as $\{X,X\}$. These representations rely on children's ability for *enriched parallel individuation*, a representational capacity that Le Corre and Carey (2007) argue can individuate objects, manipulate sets, and compare sets using one-to-one correspondence. Subset-knowers can, for instance, check if "two" applies to a set S by seeing if S can be put in 1-1 correspondence with their mental model of two, $\{X,X\}$.

In bootstrapping, the transition to CP-knower occurs when children notice the simple relationship between their first few mental models and their memorized count list of number words: by moving one element on the count list, one more element is added to the set represented in the mental model. Children then use this abstract rule to *bootstrap* the meanings of other number words on their count list, recursively defining each number in terms of its predecessor. Importantly, when children have learned the first few number-word meanings they are able to recite many more elements of the count list. Carey argues that this linguistic system provides a *placeholder structure* which provides the framework for the critical inductive inference. Subset knowers have only stored a few set-based representations; CP-knowers have discovered the generative rule that relates mental representations to position in the counting sequence.

Bootstrapping explains why children's understanding of number seems to change so drastically in the CP-transition and what exactly children acquire that's "new": they discover the simple recursive relationship between their memorized list of words and the infinite system of numerical concepts. However, the theory has been criticized for its lack of formalization (Gallistel, 2007) and the fact that it does not explain how the abstraction involved in number-word meanings is learned (R. Gelman & Butterworth, 2005). Perhaps the most philosophically interesting critique is put forth by Rips et al. (2006), who argue that the bootstrapping hypothesis actually *presupposes* the equivalent of a successor function, and therefore cannot explain where the numerical system comes from (see also Margolis & Laurence, 2008; Rips, Asmuth, & Bloomfield, 2008; Rips, Bloomfield, & Asmuth, 2008). Rips, Asmuth, & Bloomfield argue that in transitioning to CP-knowers, children critically infer,

> If k is a number word that refers to the property of collections containing n objects, then the next number word in the counting sequence, next(k), refers to the property of collections containing one more than n objects.

Rips, Asmuth & Bloomfield note that to even consider this as a possible inference, children must know how to construct a representation of the property of collections containing $n+1$ objects, for any n. They imagine that totally naive learners might entertain, say, a Mod-10 system in which numerosities start over at ten, with *"eleven"* meaning one and *"twelve"* meaning two. This system would be consistent with the earliest-learned number meanings and thus bootstrapping number meanings to a Mod-10 system would seem to be a logically consistent inference. Since children avoid making this and infinitely many other possible inferences, they must already bring to the learning problem a conceptual system isomorphic to natural numbers.

The formal model we present shows how children could arrive at the correct inference and learn a recursively bootstrapped system of numerical meanings. Importantly, the model can entertain other types of numerical systems like such Mod-*N* systems, and, as we demonstrate, will learn them when they are supported by the data. These systems are not ruled out by any hard constraints and therefore the model demonstrates one way bootstrapping need not assume specific knowledge of natural numbers.

2.3 Rebooting the bootstrap: a computational model

The computational model we present focuses on only one slice of what is undoubtedly a complex learning problem. Number learning is likely influenced by social, pragmatic, syntactic, and pedagogical cues. However, we simplify the problem by assuming that the learner hears words in contexts containing sets of objects and attempts to learn structured representations of meaning. The most basic assumption of this work is that meanings are formalized using a "language of thought (LOT)" (Fodor, 1975), which, roughly, defines a set of primitive cognitive operations and composition laws. These meanings can be interpreted analogously to short computer programs which "compute" numerical quantities. The task of the learner is to determine which compositions of primitives are likely to be correct, given the observed data. Our proposed language of thought is a serious proposal in the sense that it contains primitives which are likely available to children by the age they start learning number, if not earlier. However, like all cognitive theories, the particular language we use is a simplification of the computational abilities of even infant cognition.

We begin by discussing the representational system and then describe basic assumptions of the modeling framework. We then present the primitives in the representational system and the probabilistic model.

2.3.1 A formalism for the LOT: lambda calculus

We formalize representations of numerical meaning using *lambda calculus*, a formalism which allows complex functions to be defined as compositions of simpler primitive functions. Lambda calculus is computationally and mathematically convenient to work with, yet is rich enough to express a wide range of conceptual systems[3]. Lambda calculus is also

[3] In fact, untyped lambda calculus could represent any *computable* function from sets to number words. While we use a typed version of lambda calculus, our numerical meanings still have the potential to "loop

a standard formalism in semantics (Heim & Kratzer, 1998; Steedman, 2000), meaning that, unlike models that lack structured representations, our representational system can interface easily with existing theories of linguistic compositionality. Additionally, lambda calculus representations have been used in previous computational models of learning words with abstract or functional properties (Zettlemoyer & Collins, 2005, 2007; Piantadosi et al., 2008).

The main work done by lambda calculus is in specifying how to compose primitive functions. An example lambda expression is

$$\lambda\, x\ .\ (\textit{not}\ (\textit{singleton?}\ x)). \tag{2.1}$$

Each lambda calculus expression represents a function and has two parts. To the left of a period, there is a "λx". This denotes that the argument to the function is the variable named x. On the right hand side of the period, the lambda expression specifies how the expression evaluates its arguments. Expression (2.1) returns the value of *not* applied to (*singleton? x*). In turn, (*singleton? x*) is the function *singleton?* applied to the argument x[4]. Since this lambda expression represents a function, it can be applied to arguments—in this case, sets—to yield return values. For instance, (2.1) applied to $\{\text{Bob}, \text{Joan}\}$ would yield *true*, but $\{\text{Carolyn}\}$ yields $false$ since only the former is not a singleton set.

2.3.2 Basic assumptions

We next must decide on the appropriate interface conditions for a system of numerical meaning—what types of questions can be asked of it and what types of answers can it provide. There are several possible ways of setting up a numerical representation: (i) The system of numerical meaning might map each number word to a predicate on sets. One would ask such a system for the meaning of *"three"*, and be given a function which is true of sets containing exactly three elements. Such a system would fundamentally repre-

infinitely," requiring us to cut off their evaluation after a fixed amount of time.

[4] As in the programming language *scheme*, function names often include "?" when they return a truth value. In addition, we use *prefix notation* on functions, meaning that the function f applied to x is written as (fx).

sent a function which could answer “Are there *n*?” for each possible *n*. (ii) The system of numerical meaning might work in the *opposite* direction, mapping any given set to a corresponding number word. In this setup, the numerical system would take a set, perhaps $\{\text{duck}_A, \text{duck}_B, \text{duck}_C\}$, and return a number *word* corresponding to the size of the set—in this case, *“three”*. Such a system can be thought of as answering the question “How many are there?” (iii) It is also possible that the underlying representation for numerical meaning is one which relies on *constructing* a set. For instance, the “meaning” of *“three”* might be a function which takes three elements from the local context and binds them together into a new set. Such a function could be cached out in terms of motor primitives rather than conceptual primitives, and could be viewed as responding to the command “Give me *n*.”

It is known that children are capable of all of these numerical tasks (Wynn, 1992). This is not surprising because each type of numerical system can potentially be used to answer other questions. For instance, to answer “How many are there?” with a type-(i) system, one could test whether are are *n* in the set, for $n = 1, 2, 3, \ldots$. Similarly, to answer “Are there *n*” with a type-(ii) system, one could compute which number word represents the size of the set, and compare it to *n*.

Unfortunately, the available empirical data does not provide clear evidence for any of these types of numerical representations over the others. We will assume a type-(ii) system because we think that this is the most natural formulation, given children’s counting behavior. Counting appears to be a procedure which takes a set and returns the number word corresponding to its cardinality, not a procedure which takes a number and returns a truth value.

Importantly, assuming a type-(ii) system (or any other type) only determines the form of the inputs and outputs[5]—the inputs are sets and the outputs are number words. Assuming a type-(ii) system does not mean that we have assumed the *correct* input and output pairings. Other conceptual systems can map sets to words, but do it in the “wrong” way: a Mod-10 system would take a set containing *n* elements and return the $n \mod 10$’th number word.

[5] This is analogous to the type signature of a function in computer programming.

2.3.3 Primitive operations in the LOT

In order to define a space of possible lambda expressions, we must specify a set of primitive functional elements which can be composed together to create lambda expressions. These primitives are the basic cognitive components that learners must figure out how to compose in order to arrive at the correct system of numerical meanings. The specific primitives we choose represent only one particular set of choices, but this modeling framework allows others to be explored to see how well they explain learning patterns. The primitives we include can be viewed as partial implementation of the *core knowledge* hypothesis (Spelke, 2003)—they form a core set of computations that learners bring to later development. Unlike core knowledge, however, the primitives we assume are not *necessarily* innate—they must only be available to children by the time they start learning number. These primitives—especially the set-based and logical operations—are likely useful much more broadly in cognition and indeed have been argued to be necessary in other domains. Similar language-like representations using overlapping sets of logical primitives have previously been proposed in learning kinship relations and taxonomy (Katz et al., 2008), a theory of causality (Goodman et al., 2009), magnetism (Ullman et al., 2010), boolean concepts (Goodman et al., 2008), and functions on sets much like those needed for natural language semantics. We therefore do not take this choice of primitives as specific to number learning, although these primitives may be the only ones which are most relevant. The primitive operations we assume are listed in Table 2.1.

First, we include a number of primitives for testing small set size cardinalities, *singleton?*, *doubleton?*, *tripleton?*. These respectively test whether a set contains exactly 1, 2, and 3 elements. We include these because the ability of humans to subitize and compare small set cardinalities (Wynn, 1992) suggests that these cognitive operations are especially "easy," especially by the time children start learning number words. In addition, we include a number of functions which manipulate sets. This is motivated in part by children's ability to manipulate sets, and in part by the primitives hypothesized in formalizations of natural language semantics (e.g., Steedman, 2000; Heim & Kratzer, 1998). Semantics often expresses word meanings—especially quantifiers and other function words—as compositions

Functions mapping sets to truth values	
(*singleton? X*)	Returns true iff the set *X* has exactly one element.
(*doubleton? X*)	Returns true iff the set *X* has exactly two elements.
(*tripleton? X*)	Returns true iff the set *X* has exactly three elements.
Functions on sets	
(*set-difference X Y*)	Returns the set that results from removing *Y* from *X*.
(*union X Y*)	Returns the union of sets *X* and *Y*.
(*intersection X Y*)	Returns the intersect of sets *X* and *Y*.
(*select X*)	Returns a set containing a single element from *X*.
Logical functions	
(*and P Q*)	Returns *true* if *P and Q* are both true.
(*or P Q*)	Returns *true* if either *P or Q* is true.
(*not P*)	Returns *true* iff *P* is false.
(*if P X Y*)	Returns *X* iff *P* is true, *Y* otherwise.
Functions on the counting routine	
(*next W*)	Returns the word after *W* in the counting routine.
(*prev W*)	Returns the word before *W* in the counting routine.
(*equal-word? W V*)	Returns *true* if *W* and *V* are the same word.
Recursion	
(*L S*)	Returns the result of evaluating the entire current lambda expression *S*.

Table 2.1: Primitive operations allowed in the LOT. All possible compositions of these primitives are valid hypotheses for the model.

of set-theoretic operations. Such functions are likely used in adult representations and are so simple that it is difficult to see from what basis they could be learned, or why—if they are learned—they should not be learned relatively early. We therefore assume that they are available for learners by the time they start acquiring number word meanings. The functions *select* and *set-difference* play an especially important role in the model: the recursive procedure the model learns for counting the number of objects in a set first selects an element from the set of objects-to-be-counted, removes it via *set-difference*, and recurses. We additionally include logical operations. The function *if* is directly analogous to a conditional expression in a programming language, allowing a function to return one of two values depending on the truth value of a third. This is necessary for most interesting systems of numerical meaning, and is such a basic computation that it is reasonable to assume children have it as an early conceptual resource.

The sequence of number words *"one"*, *"two"*, *"three"*, etc. is known to children before they start to learn the words' numerical meanings (Fuson, 1988). In this formal model, this means that the sequential structure of the count list of number words should be available to the learner via some primitive operations. We therefore assume three primitive operations for words in the counting routine: *next*, *prev*, and *equal-word?*. These operate on the domain of *words*, not on the domain of sets or numerical representations. They simply provide functions for moving forwards and backwards on the count list, and checking if two words are equal[6].

Finally, we allow for recursion via the primitive function *L* permitting the learner to potentially construct a recursive system of word meanings. Recursion has been argued to be a key human ability (Hauser, Chomsky, & Fitch, 2002) and is a core component of many computational systems (e.g., Church, 1936). *L* is the name of the function the learner is trying to infer and this can be used in the definition of *L* itself. That is, *L* is a special primitive in that it maps a set to the word for that set in the *current* hypothesis (i.e. the hypothesis where *L* is being used). By including *L* also as a primitive, we allow the learner to potentially use their currently hypothesized meaning for *L* in the definition of *L* itself. One simple example of a recursive definition is,

$$\begin{aligned}\lambda\, S\, .\, (&\textit{if}\ (\textit{singleton?}\ S)\\ &\quad \textit{``one''}\\ &\quad (\textit{next}\ (L\ (\textit{select}\ S)))).\end{aligned}$$

This returns *"one"* for sets of size one. If given a set *S* of size greater than one, it evaluates (*next* (*L* (*select S*))). Here, (*select S*) always is a set of size one since *select* selects a single element. *L* is therefore evaluated the singleton set returned by (*select S*). Because *L* returns the value of the lambda expression it is used in, it returns *"one"* on singleton sets in this example. This means that (*next* (*L* (*select S*))) evaluates to (*next "one"*), or *"two"*. Thus, this recursive function returns the same value as, for instance, λ *S* . (*if* (*singleton? S*) *"one"* *"two"*).

Note that *L* is crucially *not* a successor function. It does not map a number to its

[6] It is not clear that children are capable of easily moving *backwards* on the counting list (Fuson, 1984; Baroody, 1984). This may mean that it is better not to include "prev" as a cognitive operation; however, for our purposes, "prev" is relatively unimportant and not used in most of the interesting hypotheses considered by the model. We therefore leave it in and note that it does not affect the performance of the model substantially.

successor: it simply evaluates the current hypothesis on some set. Naive use of L can give rise to lambda expressions which do not halt, looping infinitely. However, L can also be used to construct hypotheses which implement useful computations, including the correct successor function and many other functions. In this sense, L is much more basic than a successor function[7].

It is worthwhile discussing what types of primitives are *not* included in this LOT. Most notably, we do not include a Mod-N operation as a primitive. A Mod-N primitive might, for instance, take a set and a number word, and return true if the set's cardinality mod N is equal to the number word[8]. The reason for not including Mod-N is that there is no independent reason for thinking that computing Mod-N is a basic ability of young children, unlike logical and set operations. As may be clear, the fact that Mod-N is not included as a primitive will be key for explaining why children make the correct CP inference rather than the generalization suggested by Rips, Asmuth, & Bloomfield[9]. Importantly, though, we also do not include a successor function, meaning a function which maps the representation of N to the representation of $N+1$. While neither a successor function or a Mod-N function is assumed, both can be constructed in this representational system, and the model explains why children learn the successor function and not the Mod-N system—or any others—in response to the data they observe.

2.3.4 Hypothesis space for the model

The hypothesis space for the learning model consists of *all* ways these primitives can be combined to form lambda expressions—*lexicons*—which map sets to number words. This therefore provides a space of exact numerical meanings. In a certain sense, the learning model is therefore quite restricted in the set of possible meanings it will consider. It will

[7] Interestingly, the computational power to use recursion comes for *free* if lambda calculus is the representational system: recursion can be constructed via the Y-combinator out of nothing more than the composition laws of lambda calculus. Writing L this way, however, is considerably more complex than treating it as a primitive, suggesting recursion may be especially difficult or unlikely in the prior.

[8] That is, if S is the set and $|S| = k \cdot N + w$ for some integer k, then this function would return true when applied to S and the word for w.

[9] This means that if very young children could be shown to compute Mod-N easily, it would need to be included as a cognitive primitive, and would substantially change the predictions of the model. Thus, the model with its current set of primitives could be argued against by showing that computing Mod-N is as easy for children as manipulating small sets.

One-knower

```
λ S . (if (singleton? S)
          "one"
          undef)
```

Two-knower

```
λ S . (if (singleton? S)
          "one"
          (if (doubleton? S)
              "two"
              undef))
```

Three-knower

```
λ S . (if (singleton? S)
          "one"
          (if (doubleton? S)
              "two"
              (if (tripleton? S)
              "three"
              undef))
```

CP-knower

```
λ S . (if (singleton? S)
          "one"
          (next (L (set-difference S
                                   (select S)))))
```

Singular-Plural

```
λ S . (if (singleton? S)
          "one"
          "two")
```

Mod-5

```
λ S . (if (or (singleton? S)
              (equal-word? (L (set-difference S)
                                (select S))
                           "five"))
          "one"
          (next (L (set-difference S
                                   (select S)))))
```

2-not-1-knower

```
λ S . (if (doubleton? S)
          "two"
          undef)
```

2N-knower

```
λ S . (if (singleton? S)
          "one"
          (next (next (L (set-difference S (select S))))))
```

Figure 2-1: Example hypotheses in the LOT. These include subset-knower, CP-knower, and Mod-*N* hypotheses. The actual hypothesis space for this model is infinite, including all expressions which can be constructed in the LOT.

not ever, for instance, map a set to a different concept or a word not on the count list. This restriction is computationally convenient and developmentally plausible. Wynn (1992) provided evidence that children know number words refer to some kind of numerosity before they know their exact meanings. For example, even children who did not know the exact meaning of *"four"* pointed to a display with several objects over a display with few when asked "Can you show me four balloons?" They did not show this patten for nonsense word such as "Can you show me blicket balloons?" Similarly, children map number words to some type of cardinality, even if they do not know which cardinalities (Sarnecka & Gelman, 2004; Lipton & Spelke, 2006). Bloom and Wynn (1997) suggest that perhaps this can be accounted for by a learning mechanism that uses syntactic cues to determine that number words are a class with a certain semantics.

However, within the domain of functions which map sets to words, this hypothesis space is relatively unrestricted. Example hypotheses are shown in Figure 2-1. The hypothesis space contains functions with partial numerical knowledge—for instance, hypotheses that have the correct meaning for *"one"* and *"two"*, but not *"three"* or above. For instance, the 2-knower hypothesis takes an argument *S*, and first checks if (*singleton? S*) is true—if *S* has one element. If it does, the function returns *"one"*. If not, this hypothesis returns the value of (*if* (*doubleton? S*) *"two" undef*). This expression is another *if*-statement, one which returns *"two"* if *S* has two elements, and *undef* otherwise. Thus, this hypothesis represent a 2-knower who has the correct meanings for *"one"* and *"two"*, but not for any higher numbers. Intuitively, one could build much more complex and interesting hypotheses in this format—for instance, ones that check more complex properties of *S* and return other word values.

Figure 2-1 also shows an example of a CP-knower lexicon. This function makes use of the counting routine and recursion. First, this function checks if *S* contains a single element, returning *"one"* if it does. If not, this function calls *set-difference* on *S* and (*select S*). This has the effect of choosing an element from *S* and removing it, yielding a new set with one fewer element. The function calls *L* on this set with one fewer element, and returns the *next* number after the value returned by *L*. Thus, the CP-knower lexicon represents a function which recurses down through the set *S* until it contains a singleton element, and up the counting routine, arriving at the correct word. This is a version of bootstrapping in which children would discover that they move "one more" element on the counting list for every additional element in the set-to-be-counted.

Importantly, this framework can learn a number of other types of conceptual systems. For example, the Mod-5 system is similar to the CP-knower, except that it returns *"one"* if

S is a singleton, or the word before for the set *S* minus an element is *"four"*. Intuitively, this lexicon works similarly to the CP-knower lexicon for set sizes 1 through 4. However, on a set of size 5, the lexicon will find that (*L* (*set-difference S*) (*select S*)) is equal to *"four"*, meaning that the first *if* statement returns *"one"*: sets of size 5 map to *"one"*. Because of the recursive nature of this lexicon, sets of size 6 will map to *"two"*, etc..

Figure 2-1 also contains a few of the other hypotheses expressible in this LOT. For instance, there is a singular/plural hypothesis, which maps sets of size 1 to *"one"* and everything else to *"two"*. There is also a 2N lexicon which maps a set of size *N* to the $2 \cdot N$'th number word, and one which has the correct meaning for *"two"* but not *"one"*.

It is important to emphasize that Figure 2-1 does not contain the complete list of hypotheses for this learning model. The complete hypothesis space is infinite and corresponds to all possible ways to compose the above primitive operations. These examples are meant only to illustrate the types of hypotheses which could be expressed in this LOT, and the fact that many are not like natural number.

2.3.5 The probabilistic model

So far, we have defined a space of functions from sets to number words. This space was general enough to include many different types of potential representational systems. However, we have not yet specified how a learner is to choose between the available hypotheses, given some set of observed data. For this, we use a probabilistic model built on the intuition that the learner should attempt to trade-off two desiderata. On the one hand, the learner should prefer "simple" hypotheses. Roughly, this means that the lexicon should have a short description in the language of thought. On the other hand, the learner should find a lexicon which can explain the patterns of usage seen in the world. Bayes' rule provides a principled and optimal way to balance between these desiderata.

We suppose that the learner hears a sequence of number words $W = \{w_1, w_2, \ldots\}$. Each number word is paired with an object type $T = \{t_1, t_2, \ldots\}$ and a context set of objects $C = \{c_1, c_2, \ldots\}$. For instance, a learner might hear the expression "two cats" ($w_i =$"two" and $t_i =$"cats") in a context containing a number of objects,

$$c_i = \{\text{cat}_A, \text{horse}_A, \text{dog}_A, \text{dog}_B, \text{cat}_B, \text{dog}_C\}. \tag{2.2}$$

If L is an expression in the LOT—for instance, one in Figure 2-1—then by Bayes rule we have

$$P(L \mid W,T,C) \propto P(W \mid T,C,L)P(L) = \left[\prod_i P(w_i \mid t_i, c_i, L)\right] P(L) \tag{2.3}$$

under the assumption that the w_i are independent given L. This equation says that the probability of any lexicon L given W, T, C is proportional to the prior $P(L)$ times the likelihood $P(W \mid T,C,L)$. The prior gives the learner's *a priori* expectations that a particular hypothesis L is correct. The likelihood gives the probability of the observed number words W occurring given that the hypothesis L is correct, providing a measure of how well L explains or predicts the observed number words. We discuss each of these terms in turn.

The prior $P(L)$ has two key assumptions. First, hypotheses which are more complex are assumed to be less likely a priori. We use the *rational rules* prior (Goodman et al., 2008), which was originally proposed as a model of rule-based concept learning. This prior favors lexicons which re-use primitive components, and penalizes complex, long expressions. To use this prior, we construct a probabilistic context free grammar using all expansions consistent with the argument and return types of the primitive functions in Table 2.1. The probability of a lambda expression is determined by the probability it was generated from this grammar, integrating over rule production probabilities[10].

The second key assumption is that recursive lexicons are less likely a priori. We introduce this extra penalty for recursion because it seems natural that recursion is an additionally complex operation. Unlike the other primitive operations, recursion requires a potentially unbounded memory space—a *stack*—for keeping track of which call to L is currently being evaluated. Every call to L also costs more time and computational resources than other primitives since using L requires evaluating a whole lambda expression—potentially even with its own calls to L. We therefore introduce a free parameter, γ, which penalizes

[10] Similar results are found using simply the PCFG production probability as a prior.

lexicons which use of recursion:

$$P(L) \propto \begin{cases} \gamma \cdot P_{RR}(L) \text{ if } L \text{ uses recursion} \\ (1-\gamma) \cdot P_{RR}(L) \text{ otherwise} \end{cases} \tag{2.4}$$

where $P_{RR}(L)$ is the prior of L according to the rational rules model.

We use a simple form of the likelihood, $P(w_i \mid t_i, c_i, L)$, that is most easily formulated as a generative model. We first evaluate L on the set of all objects in c_i of type t_i. For instance suppose that c_i is the set in (2.2) and $t_i =$"cat", we would first take only objects of type "cat", $\{\text{cat}_A, \text{cat}_B\}$. We then evaluate L on this set, resulting in either a number word or *undef*. If the result is *undef*, we generate a number word uniformly at random, although we note that this is a simplification, as children appear to choose words from a non-uniform baseline distribution when they do not know the correct word (see Sarnecka & Lee, 2009; Lee & Sarnecka, 2010b, 2010a). If the result is not *undef*, with high probability α, we produce the computed number word; with low probability $1-\alpha$ we produce the another word, choosing uniformly at random from the count list. Thus,

$$P(w_i \mid t_i, c_i, L) = \begin{cases} \frac{1}{N} \text{ if } L \text{ evaluates to } \textit{undef} \\ \alpha + (1-\alpha)\frac{1}{N} \text{ if } L \text{ evaluates to } w_i \\ (1-\alpha)\frac{1}{N} \text{ if } L \text{ does not evaluate to } w_i \end{cases} \tag{2.5}$$

where N is the length of the count routine[11]. This likelihood reflects the fact that speakers will typically use the correct number word for a set. But occasionally, the listener will misinterpret what is being referred to and will hear an incorrectly paired number word and set. This likelihood therefore penalizes lexicons which generate words for each set which do not closely follow the observed usage. It also penalizes hypotheses which make incorrect predictions over those which return *undef*, meaning that it is better for a learner to remain uncommitted than to make a strong incorrect predictions[12]. The likelihood uses

[11] The second line is "$\alpha + (1-\alpha)\frac{1}{N}$" instead of just "$\alpha$" since the correct word w_i can be generated either by producing the correct word with probability α or by generating uniformly with probability $1-\alpha$.

[12] It would be interesting to study the relationship of number learning to acquisition of other quantifiers, since they would likely be other alternatives that could be considered in the likelihood.

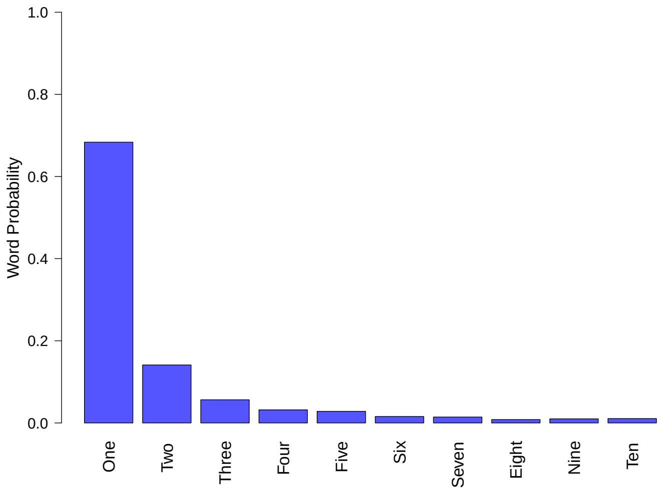


Figure 2-2: Number word frequencies from CHILDES (MacWhinney 2000) used to simulate learning data for the model.

a second free parameter, α, which controls the degree to which the learner is penalized for data which does not agree with their hypothesis.

To create data for the learning model, we simulated noisy pairing of words and sets of objects, where the word frequencies approximate the naturalistic word probabilities in child-directed speech from CHILDES (MacWhinney, 2000). We used all English transcripts with children aged between 20 and 40 months to compute these probabilities. This distribution is shown in Figure 2-2. Note that all occurrences of number words were used to compute these probabilities, regardless of their annotated syntactic type. This was because examination of the data revealed many instances in which it is not clear if labeled pronoun usages actually have numerical content—e.g., “give me one” and “do you want one?” We therefore simply used the raw counts of number words. This provides a distribution of number words much like that observed cross-linguistically by Dehaene and Mehler (1992), but likely overestimates the probability of *“one”*. Noisy data that fits the generative assumptions of the model was created for the learner by pairing each set size with the correct word with probability α, and with a uniformly chosen word with probability $1-\alpha$.

2.3.6 Inference & methods

The previous section established a formal probabilistic model which assigns any potential hypothesized numerical system L a probability, conditioning on some observed data consisting of sets and word-types. This probabilistic model defines the probability of a lambda expression, but does not say how one might find high-probability hypotheses or compute predicted behavioral patterns. To solve these problems, we use a general inference algorithm similar to the tree-substitution Markov-chain monte-carlo (MCMC) sampling used in the rational rules model.

This algorithm essentially performs a stochastic search through the space of hypotheses L. For each hypothesized lexicon L, a change is proposed to L by resampling one piece of a lambda expression in L according to a PCFG. The change is accepted with a certain probability such that in the limit, this process can be shown to generate samples from the posterior distribution $P(L \mid W,T,C)$. This process builds up hypotheses by making changes to small pieces of the hypothesis: the entire hypothesis space need not be explicitly enumerated and tested. Although the hypothesis space is in principle infinite, the "good" hypotheses can be found by this technique since they will be high-probability, and this sampling procedure finds regions of high probability.

This process is not necessarily intended as an algorithmic theory for how children actually discover the correct lexicon (though see Ullman et al., 2010). Children's actual discovery of the correct lexicon probably relies on numerous other cues and cognitive processes and likely does not progress through such a simple random search. Our model is intended as a computational level model (Marr, 1982), which aims to explain children's behavior in terms of how an idealized statistical learner would behave. Our evaluation of the model will rely on seeing if our idealized model's degree of belief in each lexicon is predictive of the correct behavioral pattern as data accumulates during development.

To ensure that we found the highest probability lexicons for each amount of data, we ran this process for one million MCMC steps, for varying γ and amounts of data from 1 to 1000 pairs of sets, words, and types. This number of MCMC steps was much more than was strictly necessary to find the high probability lexicons and children could search a much smaller effective space. Running MCMC for longer than necessary ensures that no unexpectedly good lexicons were missed during the search, allowing us to fully evaluate predictions of the model. In the MCMC run we analytically computed the expected log likelihood of a data point for each lexicon, rather than using simulated data sets. This allowed each lexicon to be efficiently evaluated on multiple amounts of data.

Ideally, we would be able to compute the exact posterior probability of $P(L \mid W,T,C)$ for any lexicon L. However, Equation 2.3 only specifies something *proportional* to this probability. This is sufficient for the MCMC algorithm, and thus would be enough for any child engaging in a stochastic search through the space of hypotheses. However, to compute the model's predicted distribution of responses, we used a form of selective model averaging (Madigan & Raftery, 1994; Hoeting, Madigan, Raftery, & Volinsky, 1999), looking at all hypotheses which had a posterior probability in the top 1000 for any amount of data during the MCMC runs. This resulted in approximately $11,000$ hypotheses. Solely for the purpose of computing $P(L \mid W,T,C)$, these hypotheses were treated as a fixed, finite hypothesis space. This finite hypothesis space was also used to compute model predictions for various γ and α. Because most hypotheses outside of the top 1000 are extremely low probability, this provides a close approximation to the true distribution $P(L \mid W,T,C)$.

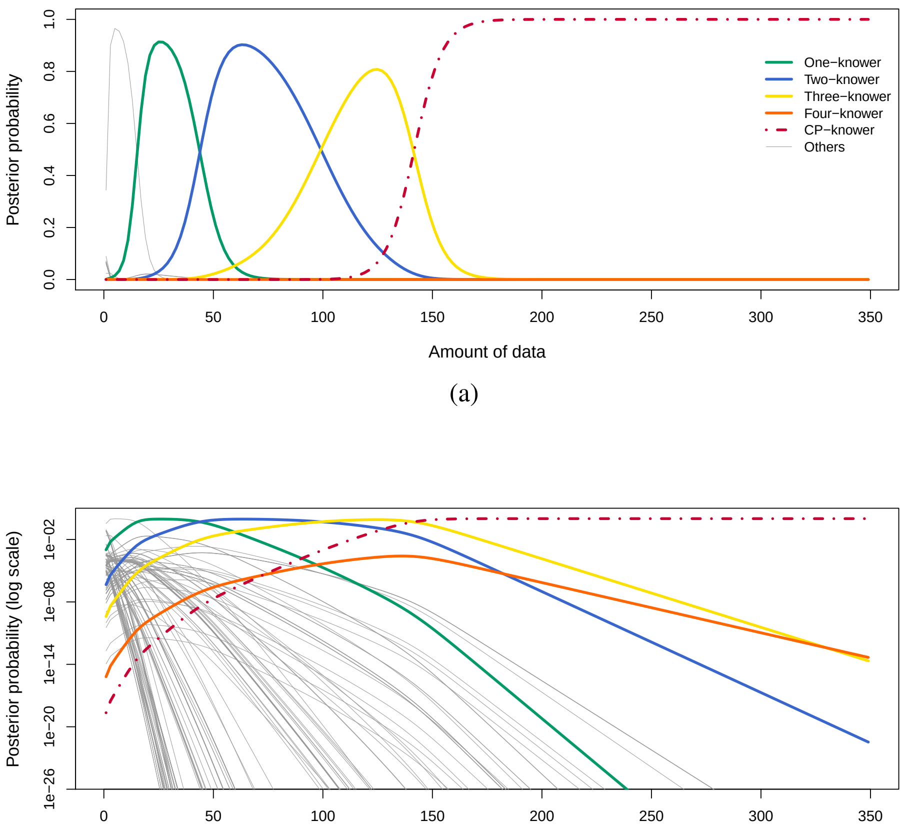
2.4 Results

We first show results for learning natural numbers from naturalistic data. After that, we apply the same model to other data sets.

2.4.1 Learning natural number

The precise learning pattern for the model depends somewhat on the parameter values α and γ. We first look at typical parameter values that give the empirically demonstrated learning pattern, and then examine how robust the model is to changing these parameters.

Figure 2-3 shows learning curves for the behavioral pattern exhibited by the model for $\alpha = 0.75$ and $\log\gamma = -25$. This plot shows the marginal probability of each type of behavior, meaning that each line represents the sum of the posterior probability all hypotheses



(a)

(b)

Figure 2-3: Figure 2-3(a) shows marginal posteriors probability of exhibiting each type of behavior, as a function of amount of data. Figure 2-3(b) shows the same plot on a log y-axis demonstrating the large number of other numerical systems which are considered, but found to be unlikely given the data.

that show a given type of behavior. For instance, the 2-knower line shows the sum of the posterior probability of all LOT expressions which map sets of size 1 to *"one"*, sets of size 2 to *"two"*, and everything else to *undef*. Intuitively, this marginal probability corresponds to the proportion of children who should look like subset- or CP-knowers at each point in time. This figure shows that the model exhibits the correct developmental pattern. The first gray line on the left represents many different hypotheses which are high probability in the prior—such as all sets map to the same word, or are undefined—and are quickly dispreferred. The model successively learns the meaning of *"one"*, then *"two"*, *"three"*, finally transitioning to a CP-knower who knows the correct meaning of all number words. That is, with very little data the "best" hypothesis is one which looks like a 1-knower, and as more and more data is accumulated, the model transitions through subset-knowers. Eventually, the model accumulates enough evidence to justify the CP-knower lexicon that recursively defines all number words on the count list. At that point, the model exhibits a conceptual re-organization, changing to a hypothesis in which all number word meanings are defined recursively as in the CP-knower lexicon in Figure 2-1.

The reason for the model's developmental pattern is the fact that Bayes' theorem implements a simple trade-off between complexity and fit to data: with little data, hypotheses are preferred which are simple, even if they do not explain all of the data. The numerical systems which are learned earlier are simple, or higher prior probability in the LOT. In addition, the data the model receives follows word frequency distributions in CHILDES, in which the earlier number words are more frequent. This means that it is "better" for the model to explain the more frequent number words. Number word frequency falls off with the number word's magnitude, meaning that, for instance, just knowing *"one"* is a better approximation than just knowing *"two"*: children become 1-knowers before they become 2-knowers. As the amount of data increases, increasingly complex hypotheses become justified. The CP-knower lexicon is most "complex," but also optimally explains the data since it best predicts when each number word is most likely to be uttered.

The model prefers hypotheses which leave later number words as *undef* because it is better to predict *undef* than the wrong answer: in the model, each word has likelihood $1/N$ when the model predicts *undef*, but $(1-\alpha)/N$ when the model predicts incorrectly. This means that a hypothesis like $\lambda\, S$. (*if* (*singleton? S*) *"one" undef*) is preferred in the likelihood to $\lambda\, S$. *"one"*. If the model did not employ this preference for non-commitment (*undef*) over incorrect guesses, the 1-knower stage of the model would predict children say *"one"* to sets of all sizes. Thus, this assumption of the likelihood drives learners to avoid incorrectly guessing meanings for higher number words, preferring to not assign them any specific numerical meaning—a pattern observed in children.

Figure 2-3(b) shows the *same* results with a log y-axis, making clear that many other types of hypotheses are considered by the model and found to have low probability. Each gray line represents a different kind of knower-level behavior—for instance, there is a line for correctly learning *"two"* and not *"one"*, a line for thinking *"two"* is true of sets of size 1, and dozens of other behavioral patterns representing thousands of other LOT expressions. These are all given very low probability, showing the data that children plausibly receive is sufficient to rule out many other kinds of behavior. This is desirable behavior for the model because it shows that the model needs not have strong a priori knowledge of how the numerical system works. Many different kinds of functions could be built, considered by children, and ruled-out based on the observed data.

Table 2.2 shows a number of example hypotheses chosen by hand. This table lists each hypothesis' behavioral pattern and log probability after 200 data points. The behavioral patterns show what sets of each size are mapped to[13]: for instance, "(1 2 U U U U U U U U)" means that sets of size 1 are mapped to *"one"* ("1"), sets of size 2 are mapped to *"two"* ("2"), and all other sets are mapped to *undef* ("U"). Thus, this behavior is consistent with a 2-knower. As this table makes clear, the MCMC algorithm used here searches a wide variety of LOT expressions. Most of these hypotheses have near-zero probability, indicating that the data are sufficient to rule out many bizarre and non-attested developmental patterns.

Figure 2-4 shows the behavioral pattern for different values of γ and α. Figures 2-4a and 2-4b demonstrate that the learning rate depends on α: when α is small, the model takes more data points to arrive at the correct grammar. Intuitively this is sensible because α controls the degree to which the model is penalized for an incorrect mapping from sets to words, meaning that when α is small, the model takes more data to justify a jump to the more complex CP-knower hypothesis.

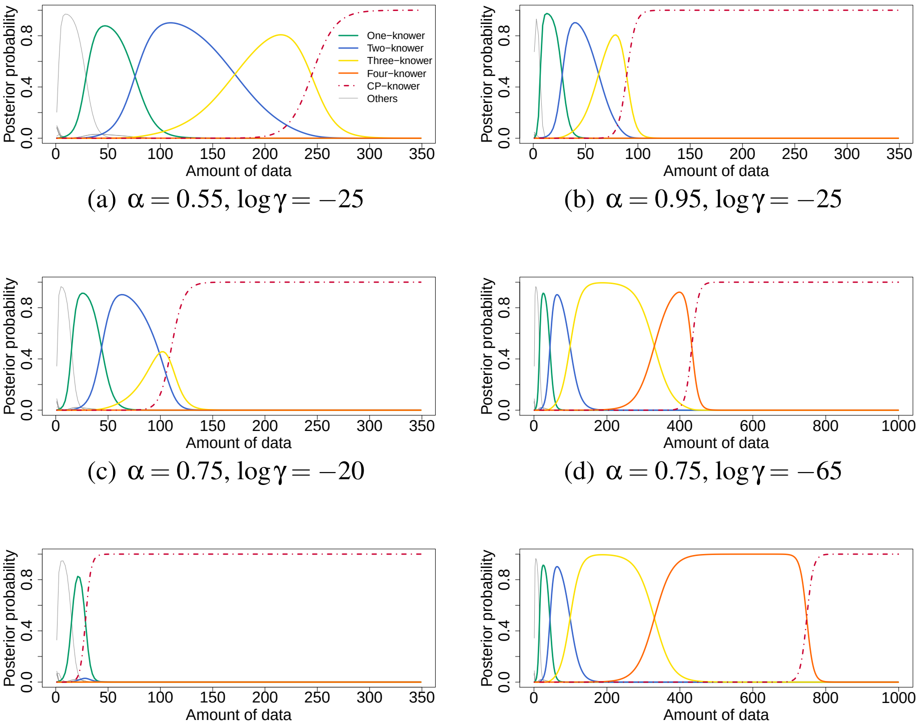
Figures 2-4c-2-4f demonstrate the behavior of the model as γ is changed. 2-4c and 2-4d show that a range of $\log\gamma$ that roughly shows the developmental pattern is from approximately -20 to -65, a range of forty-five in log space or over nineteen orders of magnitude in probability space. Even though we do not know the value of γ, a large range of values show the observed developmental pattern.

Figures 2-4e and 2-4f show what happens as $\log\gamma$ is made even more extreme. The plot for $\gamma = \frac{1}{2}(\log\gamma = -0.69)$ corresponds to *no* additional penalty on recursive hypotheses.

[13] For conciseness, we use "U" for "undef", the numeral 1 for "one", 2 for "two," etc. in this table.

Rank	Log Posterior	Behavioral Pattern	LOT expression
1	-0.93	(1 2 3 4 5 6 7 8 9 10)	λ *S* . (*if* (*singleton? S*) “*one*” (*next* (*C* (*set-difference S* (*select S*)))))
2	-2.54	(1 2 3 4 5 6 7 8 9 10)	λ *S* . (*if* (*singleton? S*) “*one*” (*next* (*C* (*set-difference S* (*select* (*select S*))))))
3	-3.23	(1 2 3 4 5 6 7 8 9 10)	λ *S* . (*if* (*not* (*singleton? S*)) (*next* (*C* (*set-difference S* (*select S*)))) “*one*”)
1423	-11.92	(1 2 3 U U U U U U U)	λ *S*. (*if* (*tripleton? S*) “*three*” (*if* (*singleton? S*) “*one*” (*if* (*doubleton? S*) “*two*” *U*))))
1604	-14.12	(1 2 3 U U U U U U U)	λ *S* . (*if* (*tripleton? S*) “*three*” (*if* (*doubleton?* (*union S S*)) “*one*” (*if* (*doubleton? S*) “*two*” *U*)))
4763	-20.04	(1 2 U U U U U U U U)	λ *S* . (*if* (*doubleton?* (*union S S*)) “*one*” (*if* (*doubleton? S*) “*two*” *U*))
7739	-39.42	(1 U 3 U U U U U U U)	λ *S* . (*if* (*tripleton? S*) “*three*” (*if* (*singleton? S*) “*one*” *U*))
7756	-44.68	(1 U U U U U U U U U)	λ *S* . (*if* (*singleton? S*) “*one*” *U*)
7765	-49.29	(1 2 4 4 4 4 4 4 4 4)	λ *S* . (*if* (*doubleton? S*) “*two*” (*if* (*singleton? S*) “*one*” “*four*”))
9410	-61.12	(1 2 1 1 1 1 1 1 1 1)	λ *S* . (*if* (*not* (*not* (*doubleton? S*))) “*two*” “*one*”)
9411	-61.12	(1 2 2 2 2 2 2 2 2 2)	λ *S* . (*if* (*not* (*not* (*singleton? S*))) “*one*” “*two*”)
9636	-71.00	(1 2 7 1 1 1 1 1 1 1)	λ *S* . (*prev* (*prev* (*if* (*doubleton? S*) “*four*” (*if* (*tripleton? S*) “*nine*” “*three*”))))
9686	-100.76	(1 3 3 3 3 3 3 3 3 3)	λ *S* . (*if* (*singleton? S*) “*one*” “*three*”)
9695	-103.65	(1 1 3 1 1 1 1 1 1 1)	λ *S* . (*if* (*tripleton? S*) (*prev* “*four*”) “*one*”)
9765	-126.01	(1 1 1 1 1 1 1 1 1 1)	λ *S* . (*next* (*prev* “*one*”))
11126	-396.78	(2 U U U U U U U U U)	λ *S* . (*if* (*singleton? S*) “*two*” *U*)
11210	-444.68	(2 U 2 2 2 2 2 2 2 2)	λ *S* . (*if* (*not* (*doubleton? S*)) “*two*” *U*)

Table 2.2: Several hand-selected example hypotheses at 200 data points.



(a) $\alpha = 0.55, \log\gamma = -25$

(b) $\alpha = 0.95, \log\gamma = -25$

(c) $\alpha = 0.75, \log\gamma = -20$

(d) $\alpha = 0.75, \log\gamma = -65$

(e) $\alpha = 0.75, \log\gamma = -0.69$

(f) $\alpha = 0.75, \log\gamma = -100$

Figure 2-4: Behavioral patterns for different values of α and γ. Note that the X-axis scale is different for 2-4d and 2-4f.

This shows a CP-transition too early—roughly after becoming a 1-knower. The reason for this may be clear from Figure 2-1: the CP-knower hypothesis is not much more complex than the 2-knower in terms of overall length, but does explain much more of the observed data. If $\log\gamma = -100$, the model is strongly biased against recursive expressions and goes through a prolonged 4-knower stage. This curve also shows a long three-knower stage, which might be mitigated by including *quadrupleton?* as a primitive. In general, however, it will take increasing amounts of data to justify moving to the next knower-level because of the power-law distribution of number word occurrences—higher number words are much less frequent. The duration of the last knower-level before the CP-transition, though, depends largely on γ.

These results show that the dependence of the model on the parameters is fairly intuitive, and the general pattern of knower-level stages preceding CP-transition is a robust property of this model. Because γ is a free parameter, the model is not capable of predicting or explaining the precise location of the CP-transition. However, these results show that the

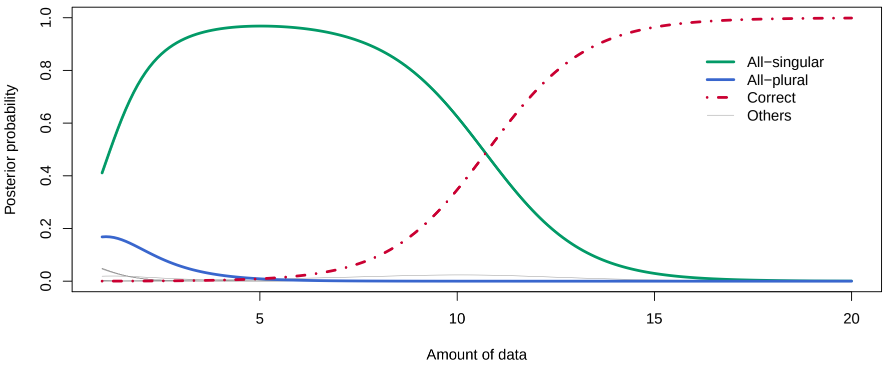


Figure 2-5: Learning results for a singular/plural system.

behavior of the model is not extremely sensitive to the parameters: there is no parameter setting, for instance, that will make the model learn low-ranked hypotheses in Table 2.2. This means that the model can explain why children learn the correct numerical system instead of any other possible expression which can be expressed in the LOT.

Next, we show that the model is capable of learning other systems of knowledge when given the appropriate data.

2.4.2 Learning singular/plural

An example singular/plural system is shown in Figure 2-1. Such a system differs from subset-knower systems in that all number words greater than *"one"* are mapped to *"two"*. It also differs from the CP-knower system in that it uses no recursion. To test learning for singular/plural cognitive representations, the model was provided with the same data as in the natural number case, but sets with one element were labeled with *"one"* and sets with two or more elements were labeled with *"two"*. Here, *"one"* and *"two"* are just convenient names for our purposes—one could equivalently consider the labels to be singular and plural morphology.

As Figure 2-5 shows, this conceptual system is easily learnable within this framework. Early on in learning this distinction, even simpler hypotheses than singular/plural are considered: λS . *"one"* and λS . *"two"*. These hypotheses are almost trivial, but correspond to learners who initially do not distinguish between singular and plural markings—a simple, but developmentally attested pattern (Barner, Thalwitz, Wood, Yang, & Carey, 2007). Here, $\lambda\, S\, .\ \textit{“one”}$ is higher probability than $\lambda\, S\, .\ \textit{“two”}$ because sets of size 1 are more frequent in the input data. Eventually, the model learns the correct hypothesis, corresponding to the singular/plural hypothesis shown in Figure 2-1. This distinction is learned very quickly by the model compared to the number hypotheses, matching the fact that children learn the singular/plural distinction relatively young, by about 24 months (Kouider, Halberda, Wood, & Carey, 2006). These results show one way that natural number is not merely “built in”: when given the different kind of data—the kind that children presumably receive in learning singular/plural morphology—the model infers a singular/plural system of knowledge.

2.4.3 Learning Mod-*N* systems

Mod-*N* systems are interesting in part because they correspond to an inductive leap consistent with the correct meanings for early number words. Additionally, children do learn conceptual systems with Mod-*N*-like structures. Many measures of time—for instance, days of the week, months of the year, hours of the day—are modular. In numerical systems, children eventually learn the distinction between even and odd numbers, as well as concepts like “multiples of ten.” Rips, Asmuth, and Bloomfield (2008) even report anecdotal evidence from Hartnett (1991) of a child arriving at a Mod-1,000,100 system for natural number meanings[14].

[14] They quote,

> D.S.: The numbers only go to million and ninety-nine.
> Experimenter: What happens after million and ninety-nine?
> D.S.: You go back to zero.
> E: I start all over again? So, the numbers do have an end?
> Or do the numbers go on and on?
> D.S.: Well, everybody says numbers go on and on because you start over again with million and ninety-nine.
> E: . . .you start all over again.
> D.S.: Yeah, you go zero, one, two, three, four—all the way up to million and ninety-nine, and then you start all over again.
> E: How about if I tell you that there is a number after that?
> A million one hundred.
> D.S.: Well, I wish there was a million and one hundred, but there isn’t.

When the model is given natural number data Mod-*N* systems are given low probability because of their complexity and inability to explain the data. A Mod-*N* system makes the wrong predictions about what number words should be used for sets larger than size N. As Figure 2-1 shows, modular systems are also considerably more complex than a CP-knower lexicon, meaning that they will be dispreferred even for huge N, where presumably children have not received enough data. This means that Mod-*N* systems are doubly dispreferred when the learner observes natural number data.

To test whether the model could learn a Mod-*N* system when the data support it, data was generated by using the same distribution of set sizes as for learning natural number, but sets were labeled according to a Mod-5 system. This means that the data presented to the learner was identical for *"one"* through *"five"*, but sets of size 6 were paired with *"one"*, sets of size 7 were paired with *"two"*, etc. As Figure 2-6 shows, the model is capable of learning from this data, and arriving at the correct Mod-5 system[15]. Interestingly, the Mod-learner shows similar developmental patterns to the natural number learners, progressing through the correct sequence of subset-knower stages before making the Mod-5-CP-transition. This results from the fact that the data for the Mod system is very similar to the natural number data for the lower and more frequent set sizes. In addition, since both models use the same representational language, they have the same inductive biases, and thus both prefer the "simpler" subset-knower lexicons initially. A main difference is that hypotheses other than the subset-knowers do well early on with Mod-5 data. For instance, a hypothesis which maps all sets to *"one"* has higher probability than in learning number because *"one"* is used for more than one set size.

The ability of the model to learn Mod-5 systems demonstrates that the natural numbers are no more "built-in" to this model than modular-systems: given the right data, the model will learn either. As far as we know, there is no other computational model capable of arriving at these types of distinct, rule-based generalizations.

[15] Because of the complexity of the Mod-5 knower, special proposals to the MCMC algorithm were used which preferentially propose certain types of recursive definitions. This did not change the form of the resulting probabilistic model.

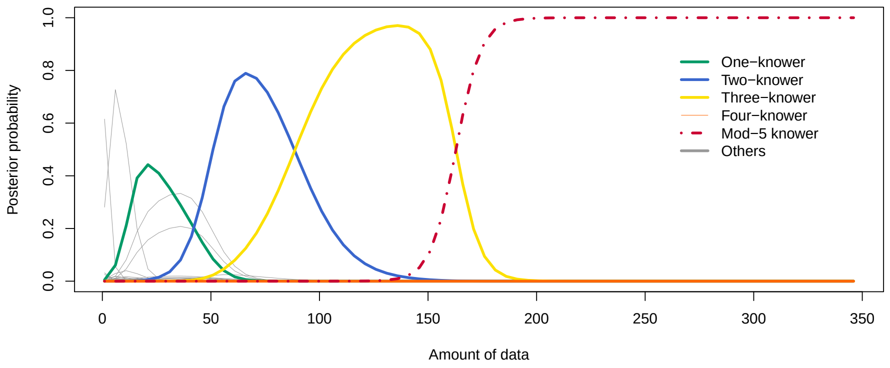


Figure 2-6: Learning results for a Mod-5 system.

2.5 Discussion

We have presented a computational model which is capable of learning a recursive numerical system by doing statistical inference over a structured language of thought. We have shown that this model is capable of learning number concepts, in a way similar to children, as well as other types of conceptual systems children eventually acquire. Our model has aimed to clarify the conceptual resources that may be necessary for number acquisition and what inductive pressures may lead to the CP-transition. This work was motivated in part by an argument that Carey's formulation of bootstrapping actually presupposes natural numbers, since children would have to know the structure of natural numbers in order to avoid other logically plausible generalizations of the first few number word meanings. In particular, there are logically possible modular number systems which cannot be ruled out given only a few number word meanings (Rips et al., 2006; Rips, Asmuth, & Bloomfield, 2008; Rips, Bloomfield, & Asmuth, 2008). Our model directly addresses one type of modular system along these lines: in our version of a Mod-N knower, sets of size k are mapped to the $k \mod N$th number word. We have shown that these circular systems of meaning are simply less likely hypotheses for learners. The model therefore demonstrates how learners might avoid some logically possible generalizations from data, and furthermore demonstrates that dramatic inductive leaps like the CP-transition should be expected for ideal learners. This provides a "proof of concept" for bootstrapping.

In implementing a fully working version of this model we have had to make several design choices about the representational system and statistical model. These choices of course affect the final “adult state“ of the model. It is useful to distinguish between the choices made in our specific implementation of the model, and our general approach to understanding numerical development. It might turn out, for instance, that children’s number word meanings are lower-bounded (with “one” meaning “one or more”) (Barner & Bachrach, 2010) or that the representational system we assume is either too weak or too powerful. While such discoveries may be inconsistent with our specific implementation, one could modify the model’s representational basis to accommodate such facts.

We have discussed empirical data that provide support for our modeling approach, but it is also important to ask what kind of empirical findings would or would not weigh against our model. Evidence against our most fundamental claim—numerical knowledge comes from statistical inference over a structured representational system—could be found by discovering developmental patterns that are inconsistent with the predictions of a statistical model operating on a plausible representational system—for instance, representational changes that cannot be explained as a response to evidence. Or, one might show that children’s representational primitives give an inductive bias unlike that required for the model to work[16].

It is also tempting to suppose that because the model discovers a recursive form of numerical meaning, it must have knowledge of “the successor principle.” However, the model produces explicit representations only of functions which map sets to truth values, not of, for instance, counting principles. Knowing that “one more element” is “one more on the count list” is only *implicit* in the computations performed by the model. Explicit knowledge of successorship and counting might require “looking at” the representations learned by the model and noticing that, for instance, every additional element makes the model return one higher word on the count list. Knowledge of such abstract properties of numbers does not come for free, even to learners who have discovered a function that can correctly map sets to number words.

This work leaves open the question of whether our approach can learn the *full* concept of natural number. This is a subtle issue because it is unclear what it means to have this

[16] For instance, as above, if *Mod* was a primitive.

concept (though see Leslie et al., 2008). Does it require knowledge that there are an infinity of number concepts, or number words? What facts about numbers must be explicitly represented, and which can be implicitly represented? The model we present learns natural number concepts in the sense that it relates an infinite number of set-theoretic concepts to a potentially infinite list of number words, using only finite evidence. However, the present work does not directly address what may be an equally interesting inductive problem relevant to a full natural number concept: how children learn that *next* always yields a new number word. If it was the case that *next* at some point yielded a previous number word—perhaps (*next "fifty"*) is equal to *"one"*—then learners would again face a problem of a modular number system. Rips, Asmuth, and Bloomfield's arguments about the need to rule out alternative modular number-system hypotheses could apply to both the Mod-*N* systems we considered earlier or the challenge described here, in which *next* is defined cyclically. The latter framing may be closer to their intended challenge, and is not directly addressed by our work here. But it is likely that similar methods to those that we use to solve the inductive problem of mapping words to functions could also be applied to learn that *next* always maps to a new word. It would be surprising if *next* mapped to a new word for 50 examples, but not for the 51st. Thus, the most concise generalization from such finite evidence is likely that *next* never maps to an old linguistic symbol[17].

In addition, the model developed here suggests a number of hypotheses for how numerical acquisition may progress:

2.5.1 The CP transition may result from bootstrapping in a general representation language.

This work was motivated in large part by the critique of bootstrapping put forth by Rips, Asmuth, and Bloomfield (Rips et al., 2006; Rips, Asmuth, & Bloomfield, 2008; Rips, Bloomfield, & Asmuth, 2008). They argued that bootstrapping presupposed a system isomorphic to natural numbers; indeed, it is difficult to imagine a computational system which could not be construed as "building in" natural numbers. Even the most basic syntactic

[17] In some programming languages, such as Scheme, there is even a single primitive function *gensym*, for creating new symbols in this manner.

formalisms—for instance, finite-state grammars—can create a discrete infinity of expressions which are isomorphic to natural numbers. This is true in our LOT: for instance the LOT can generate expressions like (*next x*), (*next* (*next x*)), (*next* (*next* (*next x*))), etc.

However, dismissing the model as "building in" natural number would miss several important points. First, there is a difference between representations which could be interpreted *externally* as isomorphic to natural numbers, and those which play the role *internally* as natural number representations. An outside observer could interpret LOT expressions as isomorphic to natural numbers, even though they do not play the role of natural numbers in the computational system. We have been precise that the space of number meanings we consider are those which *map sets to words*: the only things with numerical content in our formalization are functions which take a set as input and return a number word. Among the objects which have numerical content, we did not assume a successor function: none of the conceptual primitives take a function mapping sets of size N to the N'th word, and give you back a function mapping sets of size $N+1$ to the $N+1$'st word. Instead, the correct successor function is embodied as a recursive function on sets, and is only one of the potentially learnable hypotheses for the model. Our model therefore demonstrates that a bootstrapping theory is not inherently incoherent or circular, and can be both formalized and implemented.

However, our version of bootstrapping is somewhat different from Carey's original formulation. The model bootstraps in the sense that it recursively defines the meaning for each number word in terms of the previous number word. This is representational change much like Carey's theory since the CP-knower uses primitives not used by subset knowers, and in the CP-transition, the computations that support early number word meanings are fundamentally revised. However, unlike Carey's proposal, this bootstrapping is *not* driven by an analogy to the first several number word meanings. Instead, the bootstrapping occurs because at a certain point learners receive more evidence than can be explained by subset-knowers. According to the model, children who receive evidence only about the first three number words would never make the CP-transition because a simpler 3-knower hypothesis could better explain all of their observed data. A distinct alternative is Carey's theory that children make the CP-transition via analogy, looking at their early number word meanings and noticing a correspondence between set size and the count list. Such a theory might predict a CP-transition even when the learner's data only contains the lowest number words, although it is unclear what force would drive conceptual change if all data could be explained by a simpler system[18].

2.5.2 The developmental progression of number acquisition may result from statistical inference.

We have shown that the developmental trajectory of the model tracks children's empirically observed progression through levels of numerical knowledge. In the model, this behavior results from both the prior and the likelihood, which were chosen to embody reasonable inferential principles: learners should prefer simple hypotheses which can explain observed patterns of word usage.

The fact that these two ingredients combine to show a developmental progression anything like children is surprising and we believe potentially quite informative. One could imagine that the model, operating in this somewhat unconstrained hypothesis space, would show a developmental pattern nothing like children—perhaps learning other bizarre hypotheses and not the subset- and CP-knower hypotheses. The fact that the model does look like children provides some evidence that statistical inference may be the driving force in number-word learning. This theory has the advantage that it explains why acquisition proceeds though the observed regular stages—why don't children exhibit more chaotic patterns of acquisition? Why should children show these developmental stages at all? Children should show these patterns because, under the assumptions of the model, it is the way an ideal learner should trade off complexity and fit to data.

Additionally, the model addresses the question of why the CP-transition happens so suddenly. Why isn't it the case that children successively acquire meanings for later number words? Why is the acquisition all-or-none? The answer provided by the model is that the representational system may be discrete: at some point it becomes better to restruc-

[18] These two alternatives could be experimentally distinguished by manipulating the type of evidence 3-knowers receive. While it might not be ethical to deprive 3-knowers of data about larger number words, analogy theories may predict no effect of *additional* data about larger cardinalities, while our implementation of bootstrapping as statistical inference does.

ture the entire conceptual system and jump to a different LOT expression. Our model also predicts recent findings that the amount of evidence children receive about numbers correlates with their knowledge of cardinal meanings, even controlling for other factors such as socioeconomic status (S. Levine, Suriyakham, Rowe, Huttenlocher, & Gunderson, 2010): unlike maturational or strongly nativist theories, idealized statistical learners are highly sensitive to amount of evidence.

2.5.3 The timing of the CP-transition may depend on the primitives in the LOT and the price of recursion.

Carey (2009) suggests that the limitations of children's small-set representational system may drive the timing of the CP-transition. In what appears to be an otherwise puzzling coincidence, children become CP-knowers roughly at the capacity limit of their small-set representational system—around 3 or 4. At this point, children may need to use another strategy like counting to understand exact numerosities, and thus decide to transition to a CP-knower system.

In the model, the timing of the CP-transition depends both on the primitives in the LOT, and the value of γ, the free parameter controlling the additional cost of recursion. The fact that the timing of the model's CP-transition depends on a free parameter means that the model does not strongly predict when the CP-transition will occur. However, the point at which it does occur depends on which primitives are allowed in the LOT. While a full exploration of how the choice of primitives impacts learning is beyond the scope of the current paper, our own informal experimentation shows an intuitive relationship between the primitives and the timing of the CP-transition. For instance, removing *doubleton?* and *tripleton?* will make the CP-transition occur earlier since it makes 2-knowers and 3-knowers more complex. Including *quadrupleton?* and *quintupleton?* pushes the CP-transition beyond *"four"* or *"five"* for appropriate settings of γ. Fully and quantitatively exploring how the choice of primitives impacts the learning is an interesting and important direction for future work. In general, the model can be viewed as realizing a theory very much like Carey's proposal: the CP-transition occurs when the learner runs out of small primitive set operations which allow them to simply define early number word meanings, though, in our model the primitives do not solely determine the timing of the CP transition since the transition also depends on the cost of recursion.

Alternatively, it may turn out that algorithmic considerations play a role in the timing of the CP transition. The simple stochastic search algorithm we used can discover the correct numerical system if given enough time, and the difficulty of this search problem may be one factor which causes number-word learning to take such a long time. Algorithmic accounts are not incompatible with our approach: any approximate probabilistic inference algorithm may be applied to the model we present, producing quantitative predicted learning curves.

2.5.4 The count list may play a crucial role in the CP-transition.

The CP-knower lexicon that the model eventually infers crucially relies on the ability to "move" on the counting list using *next*. This function operates on a list of words that children learn long before they learn the words' meanings. For the CP-knower, this function allows learners to return the *next* number word after the word they compute for a set with one fewer element. Without this function, no CP-knower could be constructed since the LOT would not be able to relate cardinalities to successive words in the count list. The necessity of *next* makes the interesting prediction that children who learned to recite the counting words in a different order—perhaps *next* returns the next word in alphabetical order—would not make the CP-transition, assuming the words have their standard English numerical meaning. Such a reliance on the count list matches Carey (2009)'s suggestion that it provides a key component of bootstrapping, a *placeholder structure*, which guides the critical induction. Indeed, cultures without number words lack the capacity to encode representations of exact cardinalities (Frank, Everett, Fedorenko, & Gibson, 2008; Gordon, 2004; Pica, Lemer, Izard, & Dehaene, 2004)[19], although this does not prevent them from exactly matching large cardinalities (Frank et al., 2008).

Reasoning about LOT expressions may allow learners to understand more explicitly what happens in moving forward and backward on the count list—e.g that adding one element to the set corresponds to one additional word on the count list. This potentially

[19] Though see R. Gelman and Butterworth (2005) and see also Hartnett and Gelman (1998).

explains why subset-knowers do not know that adding or subtracting an element from a set corresponds to moving forward and backward on the count list (Sarnecka & Carey, 2008): subset knowers' representations do not yet use *next*.

2.5.5 Counting may be a strategy for correctly and efficiently evaluating LOT expressions.

As we have presented it, the model makes no reference to the act of counting (pointing to one object after another while producing successive number words). However, counting has been argued to be the key to understanding children's numerical development. Gallistel and Gelman (1992); R. Gelman and Gallistel (1978) argue children suddenly infer the meanings of number words greater than "three" or "four" when they recognize the simple correspondence between their preverbal counting system and their budding verbal counting system. They posit an innate preverbal counting system governed by a set of principles, along with an innate successor function which can map any number N onto its successor $N+1$. The successor function allows the learner to form concepts of all natural numbers, given a meaning for "one." Under this theory, the ability to count is a central, core aspect of learning number, and children's main task in learning is not creating numerical concepts, but discovering the relationship between verbal counting and their innate numerical concepts.

In our model there is a role for counting: counting may be used to keep track of which number word is currently in line to be returned by a recursive expression. For instance, perhaps each time *next* is computed on a number word, children say the number word aloud. This helps them keep track of which number word they are currently on. This may simplify evaluation of sequences like (*next* (*next* (*next* ...))), which may be difficult to keep track of without another strategy. Similarly, the act of pointing to an object is also interpretable under the model. In the recursive CP-knower hypothesis, the model must repeatedly compute (*set-difference S* (*select S*)). It may be that each time an element of a set is *select*ed, it is pointed to in order to individuate it from other objects. This interpretation of counting differs from Gelman & Gallistel's view in that counting is only a technique for helping to evaluate a conceptual representation. It may be an essential technique, yet distinct from the crucial representational systems of numerical meaning.

This interpretation of counting explains why children generally do not count when they are subset-knowers, but do count when they are CP-knowers (Wynn, 1992)[20]. Because the subset-knower lexicons make use of LOT operations like *singleton?* and *doubleton?*, they do not need to use *next*, and therefore do not need to keep track of a location in the recursion along the list of counting words. When the transition is made to a CP-knower lexicon, all number words are computed using *next*, meaning that children need to use counting as a strategy.

2.5.6 Innovation during development may result from compositionality in a LOT.

One of the basic mysteries of development is how children could get something fundamentally new—in what sense can children progress beyond systems they are innately given? Indeed, this was one of the prime motivations in studying number since children's knowledge appears very different before and after the CP-transition. Our answer to this puzzle is that novelty results from compositionality. Learning may create representations from pieces that the learner has always possessed, but the pieces may interact in wholly novel ways. For instance, the CP-knower lexicon uses primitives to perform a computation which is not realized at earlier subset-knower levels, even though the primitives were available.

The apparent novelty seen in other areas across development may arise in a similar way, from combining cognitive primitives in never-before-seen ways. This means that the underlying representational system which supports cognition can remain unchanged throughout development, though the specific representations learners construct may change[21].

Such compositionality is extremely powerful: as in lambda calculus or programming languages, one need only assume a very small set of primitives in order to allow Turing-

[20] Though see Bullock and Gelman (1977); R. Gelman and Meck (1992); R. Gelman (1993) for contrary evidence.

[21] This hypothesis provides an alternative view on the distinction between continuity and discontinuity in development. The model's learning is continuous in that the representation system remains unchanged; it is discontinuous in that the learner substantially changes the specific representations that are constructed.

complete computational abilities[22]. This provides an interesting framework for thinking about Fodor's radical concept nativism (Fodor, 1975, 1998a). Fodor argues that the only sense in which new concepts can be learned is by composing existing concepts. He says that because most concepts are not compositional (e.g., *carburetor*, *banana*, *bureaucrat*), they could not have been learned, and therefore must be innate (see also Laurence & Margolis, 2002). We have demonstrated that learning within a computationally powerful LOT, lambda calculus, may be a viable developmental theory. If development works this way, then concepts that apparently lack a compositional form–like *bureaucrat*–could be learned by only requiring a set of primitives of sufficient computational power. The reason for this is that if a computational theory of mind is correct, all such concepts can be expressed in lambda calculus or other Turing-complete formalisms, with the appropriate primitives. Such concepts could be learned *in principle* by the type of model we present, by searching through lambda expressions.

2.5.7 A sufficiently powerful representation language may bind together cognitive systems.

The LOT we use is one of the most important assumptions of the model. We chose to include primitive LOT operations—like *and*, *not*, and *union*—which are simple and computationally basic. These simple primitives provide one plausible set of conceptual resources 2-year olds bring to the task of learning number-word meanings.

We also included some operations—for instance *singleton?*, *doubleton?* and *tripleton?*—which seem less computationally and mathematically basic. Indeed, they are technically redundant in that, for instance, *doubleton?* can be written using *singleton?* as λ *S* . (*singleton?* (*set-difference S* (*select S*))). However, we include all three operations because there is evidence that even very young children are capable of identifying small set sizes. This indicates that these three operations are all cognitively "basic," part of the conceptual core that children are born with or acquire very early in childhood. The inclusion of all three of these operations represents an important assumption of the model, since excluding

[22] The subset of lambda calculus we use is not Turing-complete, though it may not halt. It would be simple to modify our learning setup to include untyped lambda expressions, making it Turing-complete.

some would change the inductive bias of the model and lead to a different developmental pattern.

Thus, the model presented here formalizes and tests assumptions about core cognitive domains by the primitives it includes in the LOT. Importantly, the core set operations must interface with basic operations in other domains. The ability to compute *singleton?* is only useful if it can be combined with an ability to use the result of applying *singleton?* to a set. This requires a representational system which is powerful enough to operate meaningfully across domains. The LOT we use contains several of these cross-domain operations: for example, *L* maps a set to a word, transforming something in the domain of sets to one in the domain of number words. The primitive *equal-word?* transforms something in the domain of words to the domain of truth values, which can then be manipulated by logical operations. The model therefore demonstrates that core cognitive operations can and must be integrated with general computational abilities and learning, potentially through a LOT. As future work elaborates our understanding of children's cognitive capacities, this class of model can provide a means for exploring how different abilities may interact and support learning.

2.5.8 Further puzzles

There are several phenomena in number learning which further work is required to understand and model.

First, since our model is not intended to be an algorithmic theory, it leaves open the question of why learning number takes so long. This is puzzling because there are many instances in which children learn novel words relatively quickly (Heibeck & Markman, 1987; Carey & Bartlett, 1978). It is possible that it takes so long because children's space of possible numerical meanings is large, and the data is consistent with many different hypotheses—as in this model. The difficulty with learning number words may even point to algorithm theories under which children stochastically search or sample the space of hypotheses (Ullman et al., 2010), as we did with the model. Such an algorithm typically considers many hypotheses before arriving at the correct one.

A second puzzle is to formalize the role of counting and its relationship to the abstract principles of counting (Gallistel & Gelman, 1992; R. Gelman & Gallistel, 1978). We suggested that counting may play a role in helping to evaluate LOT expressions, but this would require a level of meta-reasoning that is not captured by the current model. Even subset knowers appear to understand that in specific circumstances, counting can be used to determine cardinality, although it is not clear how able they are to generally use counting (Sarnecka & Carey, 2008). It may be that the knowledge of when and how to count is initially learned as a pragmatic—not numerical—principle, and children have to discover the relationship between counting behavior and their system of numerical representation. Future work will be required to address counting and understand how it might interact with the types of numerical representations presented here. Additionally, it will be important to understand the role of social and pedagogical cues in number learning. These are most naturally captured in the likelihood function of the model, perhaps by increasing the importance of certain socially salient data points.

The model we have presented assumed that number word meanings are *exact*: "two" is true of sets containing exactly two elements, and not more. In natural language use, though, numerals can often receive an interpretation of *at least*: if there are six professors named "Mark," then it is also true that there are three named "Mark." There is some evidence that children's meanings of numbers are exact (Papafragou & Musolino, 2003; Barner, Chow, & Yang, 2009; Huang, Snedeker, & Spelke, 2004), in which case the exactness assumptions of the model would be justified. This leaves open the question of how children learn the scalar implicatures common in numerical meanings, and how such pragmatic phenomena relate to pragmatics in other determiners and language more generally. One way to capture this would be to modify the likelihood to include pragmatics, or perhaps even to learn the correct form of the likelihood—learn how number words are used. On the other hand, it may turn out that early number words meanings are not exact (Barner & Bachrach, 2010). If this were true, one could construct a model very similar to the one presented here, but modify the primitives to non-exact versions. For instance, *singleton?* would change to *at-least-singleton?* and would be true of sets of cardinality one or greater. Such a system might require pragmatic factors to be included in the likelihood, and could be used to explore learning patterns assuming non-exact operations as primitives.

Another key question is how children make the mapping between approximate numerosity (Feigenson, Dehaene, & Spelke, 2004; Dehaene, 1999) and their counting routine. One way the representations of continuous quantity could interface with the representations assumed here is that continuous quantities may have their own operations in the LOT. It could be, for instance, that children also learn a compositional function much like the CP-knower hypothesis which maps approximate set representations to approximate number words, and this process may interact with the LOT representations of exact number.

2.6 Conclusion

In number acquisition, children make one of their most elegant inductive leaps, suddenly inferring a system of numerical competence which is, in principle, infinitely productive. On the surface, generalization involving such a simple conceptual system is puzzling because it is difficult to see how such a leap might happen, and in what way children's later knowledge could go beyond their initial knowledge.

The model we have provided suggests one possible resolution to this puzzle. Children's initial knowledge may be characterized by a set of core cognitive operations, and the competence to build a vast set of potential numerical systems by composing elements of their representational system. The hypothesis space for the learner might be, in principle, infinite, including many types of Mod-*N* systems, singular/plural systems, even/odd systems, and other systems even more complex. We have shown that not only is it theoretically possible for learners to determine the correct numerical system from this infinity of possibilities, but that the developmental pattern such learning predicts closely resembles observed data. The model we have proposed makes sense of bootstrapping, providing a way of formalizing the roles of key computational systems to explain how children might induce natural number concepts.

2.7 Acknowledgments

We'd like to thank Lance Rips, Sue Carey, Ted Gibson, Justin Halberda, Liz Spelke, Irene Heim, Dave Barner, Rebecca Saxe, Celeste Kidd, Leon Bergen, Eyal Dechter, Avril Kenney, and members of CoCoSci and TedLab for helpful discussions and feedback. This work was supported by an NSF Graduate Research Fellowship and AFOSR grant FA9550-07-1-0075.

Chapter 3

Quantifiers and the learnability of language[1]

3.1 Introduction

In learning language, children achieve a remarkable feat. They come to representations that support an extraordinarily intricate and productive system for communication, involving multiple aspects of meaning and complex subtleties. They do this from seemingly impoverished evidence, arriving at a linguistic system that appears to go beyond what is directly observable in their input. This is especially striking in children's acquisition of function words like "the," "both," and "and." Content words—which map onto objects and actions—are plausibly learned through tracking co-occurrences between words and things in the world, perhaps using a sufficiently powerful cross-situational word-learning model (Siskind, 1996; Vogt & Smith, 2005; Yu & Ballard, 2007; Yu & Smith, 2007; Frank et al., 2007a). But function words do not correspond to any plainly observable perceptual phenomena and express their meaning only through combination with other words. They therefore embody two of the most challenging, interesting, and fundamental aspects of language learning: abstractness and compositionality.

Indeed function words appear as a striking gap in most theories of language learning, with little to no attention from statistical approaches, and relatively incomplete and non-

[1] This work is joint with Noah D. Goodman and Joshua B. Tenenbaum.

computational (non-implemented) nativist theories. Even for a maximally nativist theory under which all function word meanings are innately specified, children still face a problem of mapping them to their corresponding arbitrary phonetic forms. As we show, this is no easy task.

Because function word meanings are abstract and compositional—yet likely must be inferred from impoverished and non-explicit evidence—they provide a clear case for integrating structured representations with statistics: structure is needed to explain adult competence, and statistics is needed to explain learning. Here, we develop a statistical model of one interesting and representative class of function words: quantifiers. We present a model that can use naturalistic data to infer rich lexical meanings for quantifier meanings, including literal meaning and presupposition. We use the model to study the learnability of quantifier meanings, and show that our approach provably learns the adult meanings, using only positive evidence if necessary. We use the model to evaluate claims about the utility of various constraints on these meanings, and the potential role of negative evidence. We then relate the learning model to developmental data, arguing that broad patterns in the acquisition of these words can be captured by the types of inductive methods we propose.

3.2 Learnability and quantification

At first pass, the inductive problems faced by language learners seem perplexing. Perhaps the most-cited formal result about language learning is Gold (1967), who demonstrated conditions under which even very simple classes of formal languages cannot be identified by idealized learners (see K. Johnson, 2004; Bertolo, 2001). Gold's theorem shows how, in the worst case, learners could receive infinitely many positive examples of utterances from a formal language $\mathcal{L}$, and never be able to identify that $\mathcal{L}$ was the correct target language. This was taken to be relevant to natural language since natural languages seem to satisfy necessary assumptions for Gold's theorem to apply—in particular, language learners appear not to receive negative evidence, explicit instruction about what constructions are not grammatical (Braine, 1971; Marcus, 1993; Brown & Hanlon, 2004)[2]. Gold's theo-

[2] Though see Pullum and Scholz (2002).

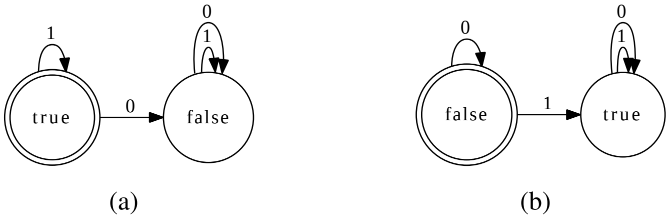


Figure 3-1: The representation of “every” or “all” (a) and “some“ (b) in Clark (1998)’s learning model.

rem therefore provides a puzzle for the study of language acquisition (Wexler & Culicover, 1983; Osherson, Stob, & Weinstein, 1984): how might children learn language if many languages cannot even be identified *in theory* by the available evidence? One approach to resolving this dilemma is to build in sets of innate constraints, supposing that the major task for children during acquisition is to determine the correct way of setting simple parameters that capture cross-linguistic variation (Wexler & Culicover, 1983; Niyogi & Berwick, 1996; Gibson & Wexler, 1994; Sakas & Fodor, 2001; J. Dresher & Elan, 1990; B. Dresher, 1999; Yang, 2002; Fodor, 1998b; Kohl, 1999). These approaches have typically posited rich sets of language-specific rules and constraints, and simple learning mechanisms that change states of a grammar depending on patterns in the input. However, in thinking about quantifier meanings, it is not clear that any simple acquisition theory based on, for instance, parameter setting is sufficiently powerful: the meaning of quantifiers (and function words in general) requires computation and semantic composition. Learners must therefore consider potential quantifier meanings to be some space of computational devices, consisting of at least those present in natural languages. This search through the space of computational representations is likely much harder than, for instance, a search through grammatical parameters would be (e.g., Gibson & Wexler, 1994), and is made especially difficult by the fact that such function words are not captured by easily perceptible phenomena in the world, like objects, properties, and events.

As a result, quantifier learning has previously been studied using tools from computability theory. Quantifier meanings can be associated with computational devices of varying degrees of complexity theoretically, ranging from finite-state automata, to more complex computational devices like pushdown automata (van Benthem, 1984, 1986; M. Mostowski, 1998; Tiede, 1999; Florêncio, 2002; Gierasimczuk, 2007). This basic approach builds on computational ideas developed by van Benthem (1984), who noted that the computations required by many quantifiers—in particular, those like "every" and "some" that can be written in first-order logic—can be captured by finite-state automata (for extensions up the automata hierarchy, see M. Mostowski, 1998). The meaning of "every" can be captured by a finite-state machine like that shown in Figure 3-1(a). Here, a language user wishing to check if "every A is B" would start in the double-circled state *true* and proceed to look at elements of A. Each element $a \in A$ is processed and if it is an element of B, a 1 link is followed; if it is not, a 0 link is followed. Thus, as long as every element in A is in B, the learner will stay in the *true* state; otherwise they will fall inextricably into the *false* state. A similar example for "some" is shown in Figure 3-1(b), only here one positive example is enough to change the automaton permanently to an accepting state.

Clark (1996) presents a detailed account of quantifier acquisition based on finite state models, and provides similar automata for even more complex meanings, such as "none," "at least two," and "an even number of" (see also Clark, 2010). By formalizing meanings as finite-state automata, Clark is able to apply learnability results for regular languages (Angluin, 1987) to show that first-order quantifiers[3] can be learned jointly using positive and negative evidence. Unfortunately, this approach requires the adults to provide explicit counterexamples for the learner, perhaps an unrealistic expectation since it involves knowing the learner's hypothesized meanings. These results were extended by Tiede (1999), who showed that first-order *left increasing monotonic* quantifiers are identifiable in the limit (i.e. Gold-learnable) from *positive* evidence alone[4]. Not all first-order quantifiers are identifiable in the limit from positive evidence: Tiede shows that quantifiers that are left *decreasing* (e.g., "few") monotonic, right increasing monotonic (e.g., "several"), or right decreasing monotonic (e.g., "no") do not come with guarantees of learnability[5]. Florêncio

[3] Those which can be expressed in first-order logic; not, e.g., "most."

[4] Left increasing monotonic quantifiers are those quantifiers Q such that $(QAB) \rightarrow (QA'B)$ where $A \subseteq A'$. In other words, quantifiers that, if true, can generalize to any more inclusive first set. For instance, "several" is left-upward-monotonic since if "several angry lawyers are fools" is true, then "several lawyers are fools": by increasing the size of the first set from "angry lawyers" to "lawyers," we do not make the sentence false. Note that this is not true for "few": "few angry lawyers are fools" does *not* imply "few lawyers are fools."

[5] Tiede also shows how all quantifiers with a certain form in *Presburger arithmetic* are learnable in the limit.

(2002) extends these results to what he argues are “psychologically plausible” restrictions on learning algorithms, such as algorithms that do not care about the order of sets or only change hypotheses when they are incorrect.

While these results have mapped out the space of learnability for some quantifiers in a mathematically sophisticated way, this general approach is lacking in several important respects. First, these learning theories only apply to subsets of natural language quantifiers (for instance, the left-upward-monotonic ones), yet a full learning theory should handle at least everything observed in natural language. In addition, it is not clear how these learning frameworks might be extended to handle noisy evidence. In the case of quantifiers, this means perhaps incorrectly identifying the relevant sets, and also occasionally hearing quantified expressions which are false. The learning theories are not implemented, meaning that it is unclear if the amount of data required to learn quantifier meanings is at all plausible. These accounts are disconnected from data, as the authors do not use them to make empirical predictions, much less explain developmental phenomena. The correct learning theory should ideally predict the developmental trajectory of learning, including the types of errors that children make. These theories only capture literal meanings and to our knowledge have not been extended to other aspects of meaning, such as presupposition. Relatedly, these learning theories use representations which are wholly unlike anything else used in semantics. Finite-state machines are standard computational formalisms, but it is not clear how they relate to more standard machinery of linguistic theories like lambda expressions; indeed, the choice of these representations—though mathematically elegant—seems largely *ad hoc*. Finally, these approaches typically do not explicitly address or discuss what we see as one of the most interesting and challenging aspects of learning semantics from noisy positive evidence, *the subset problem*.

3.2.1 The subset problem in semantics

The *subset problem* is that learners may incorrectly infer and under-restrictive word meaning, and positive data cannot provide direct evidence that they have done so. For instance, SOME is logically weaker than EVERY: every time “Every accordion is heavy” is true it is also true that "Some accordion is heavy." This means that if at any stage of acquisition, children incorrectly guess that "every" has the denotation of "some," then positive evidence would never lead them to change their mind. This problem appears in many areas of language acquisition including syntax (Wexler & Manzini, 1987; Berwick, 1985) and and phonology (Smolensky, 1996; Hale & Reiss, 2003).

The subset problem also appears in learning compositional semantic structures. Crain, Ni, and Conway (1994) discuss the ambiguous sentence, "The big elephant eats only peanuts." This could either mean, (i) the only thing the big elephant does is eat peanuts, or (ii) the only food the big elephant eats is peanuts. Importantly, in any context where (i) is true, (ii) is also true: if all the elephant does is eat peanuts, peanuts must be the only food it eats. Crain et al. (1994) discuss the subset problem that this poses: if learners initially thought meaning (i) was not permissible but (ii) was, no positive evidence could ever compel them to accept (i). This is because they would never have a truth-functional reason for extending their meaning to include interpretation (i) since learners who interpreted the meaning as (ii) would already think the utterance was true[6].

In these domains—syntax, phonology, and compositional semantics—one proposed solution is the *subset principle*, which holds that learns should have a strong innate bias for logically stronger hypotheses (Wexler & Manzini, 1987; Berwick, 1985; Smolensky, 1996; Crain, 1992, 1993; Crain & Philip, 1993; Gualmini & Schwarz, 2009; Crain & Thornton, 2000; Crain et al., 1994; Musolino, 2006). Positive evidence then compels learns to move to logically weaker hypotheses. In the case of "every" and "some," learners' initial state would be to prefer the correct meaning of "every" for both words, and then uses in other context would eventually show them that "some" has its logically weaker meaning. In Crain et al. (1994)'s example, learners would innately prefer interpretation (i) and it would take positive evidence to convince them that (ii) must also be possible.

This theory requires learners to have a very specific initial state with meanings or hypotheses ordered by logical strength. Innately preferred specific meanings are then broad-

[6] Another possibility that has been suggested is that positive evidence is enough to solve the subset problem for sufficiently sophisticated learners. Gualmini and Schwarz (2009) argue that other syntactic and semantic principles can resolve the learnability problems with these types of sentences and others, although it is not clear that a similar approach could be applied to the case of quantifiers or verbs discussed above. Musolino (2006) argues the problem does not exist in the first place.

ened to a more inclusive hypothesis in order to explain observed data. Unfortunately, even within quantifiers, it is not always possible to order hypotheses by logical strength, as with "most" and "many." Subset principle learning does not provide an account of how learners sort out these types of word meanings, which cannot be ordered by logical strength. Additionally, the subset principle approach seems much less plausible when one considers that the subset problem is faced even more generally in acquisition—for instance, in learning lexical semantics. Surely learners do not have an innate specification that PUSHING is more specific than TOUCHING, GREYHOUND is more specific than DOG, etc. Perhaps more problematic is that the subset principle appears unable to handle noisy input because it implicitly formalizes an irreversible process. Once learners come to think that a word means SOME, no (positive) evidence can ever convince them that it really meant EVERY. But occasionally learners will receive incorrect evidence—for instance, hearing "every scientist is an accordionist" in a context where it is untrue, but "some" is true. How might learners deal with this problem? Strictly taken, the subset principle holds that learners should change their meaning of "every" to SOME, since EVERY is now no longer consistent with their evidence. A related problem, the "triggering problem" (Borer & Wexler, 1987), is that these types of accounts must explain why observed data often does not quickly change learners' hypotheses. One can imagine versions of the subset principle to solve both of these problems, where learners have some threshold for the amount of evidence required to change their meaning. To our knowledge such a system has never been formalized or shown to learn correctly.

Finally, the subset principle appears to make falsified predictions about learning trajectories in learning semantics. Musolino (2006) reviews some predictions of the semantic subset principle that fail in experiments. For instance, sentences such as "Every student can't afford a new car." could mean either (i) for each student s, s cannot afford a new card, or (ii) it is not the case that every study can afford a new card. Since (i) implies (ii), the subset principle implies early learners should interpret the sentences as (i), not (ii); but, the opposite is true (Musolino, Crain, & Thornton, 2000). Similar incorrect predictions can be found in studies of word learning. Xu and Tenenbaum (2007) presented children with examples of objects chosen from either subordinate (dalmatian), basic-level (dog), or superordinate (animal) categories. In the context of the subset problem, children must have some way of discovering that "dalmatian" does not mean DOG and "dog" does not mean ANIMAL. One way to test the predictions of a learner who uses the subset principle is to look at conditions where evidence is provided that is only consistent with the subordinate level category, the most specific generalization children could make. Xu and Tenenbaum (2007) find that children shown one example of the subordinate level category generalized its label to higher level categories 31% of the time in one experiment and 40% of the time in another. In other words, 31-40% of the time children do not make the most specific generalization from data that is possible. With 3 examples from the subordinate level category, generalization to the basic level category dropped to 13% in one experiment, and 6% in another, showing that children were sensitive to the amount of data as well. Note that this differential pattern of generalization with only positive evidence is not predicted by the subset principle. Subset-principle learners should, from the earliest amount of evidence, make the most specific generalizations, and this pattern should not change as additional consistent evidence is provided.

We take the above arguments as compelling problems with using the subset principle to solve the subset problem in language acquisition. It requires a complex innate set of language-specific hypotheses and computations, is not clearly able to handle noisy data, and appears to make already falsified predictions. Moreover no implemented models exist demonstrating the computational tractability and theoretical soundness of these proposals.

However, the subset principle does motivate the appealing intuition that more specific hypotheses should be preferred when they are right. A learner should dis-prefer for "every" to mean SOME because it makes the incorrect, broad prediction that "every" should be a possible option in all situations where "some" is. We capture this intuition in a probabilistic model described in the next section. In contrast to subset-principle proposals, our proposal draws on potentially domain-general techniques to solve the subset problem: the subset problem is solved by any model which engages in statistically sound reasoning. We additionally show this model is provably learnable from arbitrarily noisy data, and does not require specific innate ordering of hypotheses. We apply these techniques to learning quantifier meanings, but they are considerably more general, applicable in theory to subset problems in syntax and phonology as well.

3.3 Learning quantifiers

We begin our approach to quantifier learning by first articulating the types of representations learners must eventually acquire. We are motivated by the types of representations often posited in semantic theories, since these constitute our "best guess" for adults' knowledge of quantifier meanings. However, like all models, our target meanings are a simplification. They are intended to capture many of the most interesting aspects of quantifier meanings.

The most basic assumption we make in this paper is that learners express hypotheses about word meanings using a *language of thought* (LOT) (Fodor, 1975), a structured representational system in which complex semantic representations are built by composing a small set of cognitive primitives. There are several motivations for this approach. First, semanticists often express word meanings using a structured, compositional representation system (e.g., Montague, 1973; Heim & Kratzer, 1998; Steedman, 2000), because doing so allows complex word meanings to be formalized precisely in terms of simple, known, and well-defined logical operations. Second, as we show, a compositional representation system provides a compelling account of learning: learning consists of appropriately combining (composing) simpler logical capacities. This type of system need not be language-specific and indeed has been proposed to explain learning and development in other domains, including kinship relations (Katz et al., 2008), abstract relational concepts (Kemp et al., 2008a), boolean rule-based concepts (Goodman et al., 2008), lexical semantics (Siskind, 1996), number-word acquisition (Piantadosi, Tenenbaum, & Goodman, submitted), compositional semantics (Zettlemoyer & Collins, 2005; Kwiatkowski et al., 2009; Piantadosi et al., 2008), intuitive notions of causality (Goodman et al., 2009), and magnetism (Ullman et al., 2010).

Given that we posit learners acquire representations consisting of logical expressions "built" out of conceptual primitives, there are three aspects of quantifier meanings that our learning model aims to acquire: literal meaning, presupposition, and word production probability.

Literal meaning

We follow Heim and Kratzer (1998) in supposing that to a first approximation, the literal meaning of quantifiers can be captured with *generalized quantifiers*, logical operations that denote relations between sets (see also Montague, 1973; Barwise & Cooper, 1981; Keenan & Stavi, 1986; Keenan & Westerståhl, 1997)[7]. For instance, a sentence like "Some reporter is a liar" might be mapped to a logical expression like

$$(\textit{nonempty?}\ (\textit{intersection reporters liars})). \tag{3.1}$$

Throughout this paper we use *prefix notation*, meaning that a function f applied to an argument x is written $(f\ x)$. Expression (3.1) is an expression that says that the intersection of the set of reporters and liars is not empty. It is built using two logical operations: *intersection* computes the set-intersection of its arguments, and *nonempty?* checks if a set is not empty[8]. To arrive at such a meaning, comprehenders, for instance, would hear "Some reporter is a liar," parse the sentence and use their compositional semantics to appropriately compose the word meanings in the sentence to arrive at (3.1). Simple formalized systems for this type of language understanding can be found in Blackburn and Bos (2005); detailed linguistic accounts can be found in Heim and Kratzer (1998) and Steedman (2000)[9]. In a very simplified system, "reporters" would map to the set of reporters, "liar" would map to the set of liars, and "some" would have a special denotation, a function of two sets:

$$\lambda\, A\ B\ .\ (\textit{nonempty?}\ (\textit{intersection}\ A\ B)). \tag{3.2}$$

[7] We note that this is a nontrivial assumption, but see Heim and Kratzer (1998) for arguments that this is indeed a good way to characterize quantifier meanings. Learning that quantifiers denote these types of meanings is not addressed here.

[8] In standard set-theoretic notation this would be $\textit{reporters} \cap \textit{liars} \neq \emptyset$; in first-order logical notation it might be written $\exists x.\textit{reporter}(x) \wedge \textit{lied}(x)$. We use prefix notation keeping in line with previous work (e.g Piantadosi et al., submitted; Piantadosi, Tenenbaum, & Goodman, 2009), and the computer-programming language *scheme*, which we use to implement these models.

[9] Here, we will focus only on the meaning of the quantifier and not how it compositionally combines with other words, though see Piantadosi et al. (2008) and Zettlemoyer and Collins (2005) for theories of learning compositional structures.

This notation, lambda calculus, provides a convenient formalism for expressing *functions*. Here, "$\lambda\, A\ B$." denotes that the expression after the period is a function of the variables A and B. The compositional semantics of English would have to pass *reporter* for "reporter" as the argument A, and *lied* as the argument B in order to arrive at (3.1). Many quantifier meanings can be written down as lambda expressions like this that take two sets and return a truth value. For instance, "every" might be denoted

$$\lambda\, A\ B\ .\ (\mathit{subset}\ A\ B) \tag{3.3}$$

where *subset* is a function which is true if the first set is a subset of the second. "No" (or "none of the") might be written as

$$\lambda\, A\ B\ .\ (\mathit{empty?}\ (\mathit{intersection}\ A\ B)). \tag{3.4}$$

We note that we could have written down each of the above quantifiers in first-order logic, using $\forall$ and $\exists$. The use of set-theoretic operations is motivated by other quantifiers meanings which provably cannot be expressed in first-order logic (A. Mostowski, 1957; Barwise & Cooper, 1981). For instance "most" cannot be written down using $\exists$ and $\forall$, intuitively because "most" requires comparing potentially arbitrarily large cardinalities ("Most A are B" if there are more As that are B than those that are not), but a finite expression in first-order logic can only manipulate finitely many cardinalities. However, it can be expressed very naturally with set-operations:

$$\lambda\, A\ B\ .\ (\mathit{card}{>}\ (\mathit{intersection}\ A\ B)\ (\mathit{set\text{-}difference}\ A\ B)) \tag{3.5}$$

where *card>* is a function that compares the cardinality of its first argument to the cardinality of its second. This formal insufficiency of first-order logic for natural language semantics is a deeply interesting and nontrivial property of human language—language apparently involves rich and complex types of quantification. This move from first-order logical operations, which have no explicit notion of number or cardinality, to set-based representations with cardinality operations seems to fit with evidence that the interpretation of quantifiers draws on neural systems for processing number (McMillan, Clark, Moore, Devita, & Grossman, 2005; McMillan, Clark, Moore, & Grossman, 2006; Clark & Grossman, 2007), although there may be differences between quantifiers which can and cannot be written down in first-order logic (Troiani, Peelle, Clark, & Grossman, 2009; Szymanik & Zajenkowski, 2010).

We note that for "most" and all the other quantifiers studied here, there are many equivalent ways of writing their meanings. Alternative formalizations, when treated as explicit theories of the computational processes underlying these word meanings have been argued to give rise to different behavioral hallmarks (Hackl, 2009; Pietroski, Lidz, Hunter, & Halberda, 2009), but these distinctions will not be addressed in this work.

Presupposition

The second aspect of quantifier meaning is *presupposition*, which captures the assumptions that are required for a statement to receive a truth value (see Heim & Kratzer, 1998, section 6.7, for an overview). Our primary representational choices build off proposals in semantics and philosophy of language dating back to B. Russell (1905) and Strawson (1950)[10], who argued about the correct way to handle presuppositions in the definite determiner, "the." Russell argued that the meaning of sentences like "The *A* is *B*" asserts that *A* is true of exactly one element and that element is in *B*. In other words "The accordionist is cooking" is true if and only if there is exactly one accordionist and that accordionist is cooking. This proposal captures the notion that "the" can only be used in situations where there is a unique referent. However, as argued by Strawson (1950), this account is lacking in that it seems to assign truth values to sentences which intuitively may not even have truth values. Strawson's sentence, "The present king of France is bald" would be strictly false under Russell's account, since it is not true that there is exactly one present king of France. Strawson argues that our intuitions really say this sentence *does not have* a truth value (see also B. Russell, 1957; Von Fintel, 2004). Strawson argues that sentences like "The present king of France ... " *presuppose* the existence of a king of France, rather than assert

[10] We do not wish to get bogged down in the details of the semantic analyses of these words, or the large philosophical and linguistic literature devoted to more thoroughly developing theories of semantics, reference, and presupposition; for a detailed description, see (see Ludlow & Neale, 2008).

it. That is, in order for such sentences to be true or false, there must exist a present king of France. If there is no king of France, the sentence is neither true nor false. Indeed, violations of such background assumptions appear to have different behavioral hallmarks than truth-value violations of asserting something false (Langford & Holmes, 1979).

Such presupposed meanings are an important aspect to the semantics of many quantifiers. For instance, in a situation where there is exactly one sailor, it is bizarre to assert

$$\text{“Both sailors are happy.”} \tag{3.6}$$

regardless of whether the one sailor is happy. Sentence (3.6) appears to require as part of its background assumption that there are exactly two sailors, and it is difficult to say whether it is strictly true or false if there are not exactly two. Moreover, if there were two sailors, the background assumptions do not intuitively change if the sentence is negated—in contrast to literal meaning: "It is not the case that both sailors are happy" still assumes there are two sailors.

Representationally, we can capture presuppositional aspects of meaning by assuming that semantic representations have two parts: the presupposed content and the asserted content (see Karttunen & Peters, 1979; Heim, 1991). For instance, "the" would presuppose exactly one element in A, and assert that the element of A is in B. In this case, both the presupposed and asserted content can be expressed as lambda expressions:

Presupposed $\lambda\, A\ B\ .\ (\mathit{singleton?}\ A)$

Asserted $\lambda\, A\ B\ .\ (\mathit{nonempty?}\ (\mathit{intersection}\ A\ B))$

Here, *singleton?* is a function which is true if given a set of size one, a singleton. In principle these two aspects of meaning could be combined within one single LOT expression. For instance,

$$\lambda\, A\ B\ .\ (\mathit{presup}\ (\mathit{singleton?}\ A)\ (\mathit{nonempty?}\ (\mathit{intersection}\ A\ B))) \tag{3.7}$$

where *presup* is a function which returns undefined if its first argument is false, and it returns its second argument if the first argument is true. Here, we choose to separate these two aspects of meaning because the probabilistic model we present will need access to both the presupposed and asserted aspects of meaning, allowing us to separately evaluate the learning of each.

Probability of production

Not all true and presuppositionally valid quantifiers are equally appropriate in each situation. For instance, in every situation where “a” is true, “the” is also true: if “the mayor is cheating” then “a mayor is cheating.” However, it is intuitively somewhat odd to use “a” in this situation when “the” is true. Heim (1991) proposed explaining these types of intuitions—and others that are unrelated to quantification—with a pragmatic principle known as *maximize presupposition*. This principle holds that all else being equal, speakers will prefer utterances with the strongest presuppositions (see also Sauerland, 2003; Schlenker, 2006; Singh, 2009). For instance, since “the” presupposes exactly one, it will be uttered over the presuppositionally weaker “a” when both are true. This intuitively captures something very similar to Grice (1975)’s maxim to be informative, except that maximize presupposition is argued to be valid even when the quantifiers convey the same amount of information.

Maximize presupposition provides an interesting challenge for computational learning theories. Even if maximize presupposition is assumed to be known to young learners, they would still have to be able to correctly score the probability that an adult with their grammar would have produced the utterances they observe. This would involve computing the relative logical strength of any of their hypothesized expressions, which is likely to be extremely costly—if not uncomputable—depending on the space of primitives in their representations. This means that fully implementing maximize presupposition on an unrestricted space of meanings is likely untractable, both for children and computational models. Worse still, if learners do not know maximize presupposition, they would additionally have to find this principle from some larger space of principles, which would make learning substantially harder.

An alternative to maximize presupposition—or other explicit principles in pragmatics—

is to assume that each word's probability of production is lexicalized. Here, we assume that learners store a single number, a *weight*, for each word that quantifies how likely that word is to be uttered when it is true. For instance, "the" would have a higher weight than "a" so that whenever both are true, "the" is more likely to be uttered. Speakers in this system only need to "look up" a word's weight in order to determine what is most likley to be said, and learners would only need to search through weights. This system is chosen here for reasons of computational tractability, and for the right settings of the weights, it is possible to emulate any other measure on words. However, it is not unreasonable to think that real language-processing systems might work this way, also for reasons of computational tractability[11]. It will be informative for future work to study the way in which abstract principles like maximize presupposition might be learned, and the ways in which early knowledge of pragmatics—through explicit principles or lexicalized weights—affects learning trajectories.

3.3.1 The target space of meanings

Given the above three aspects of meaning, we can define a *lexicon* to be a mapping from words to literal meanings, presuppositional meanings, and weights. To our knowledge, previous learning models have not tackled the complexity of learning these multiple aspects of meaning, simultaneously and for a plausible set of function words. Figure 3-2 shows the target set of words the learning model will acquire. The general learning setup is that learners hear utterances like "All A are B" in some context containing a set of objects, and must use the cross-situational occurrences to infer the best lexicon. The details of this statistical inference are described in the modeling section.

For this target lexicon we chose eight words that differ on their literal meanings and presuppositions. The specific meanings and presuppositions in Figure 3-2 are meant to provide an interesting approximation to English, but available data does not distinguish between this grammar and close alternatives. For instance, the literal meaning of "the"

[11] In other words, this can be taken as a hypothesis for how learners "implement" maximize presupposition. Lexicalized word weights could be experimentally distinguished from principles like maximize presupposition by teaching people a novel quantifier and seeing if it is used in accordance with maximize presupposition from the start, or whether people would also need to learn its probability of production.

Word	**Presupposition**	**Literal meaning**	**Weight**
the	λ *A B* . (*singleton? A*)	λ *A B* . (*singleton?* (*intersection A B*))	1000
a	λ *A B* . (*nonempty? A*)	λ *A B* . (*nonempty?* (*intersection A B*))	100
two	λ *A B* . (*nonempty? A*)	λ *A B* . (*doubleton?* (*intersection A B*))	250
both	λ *A B* . (*doubleton? A*)	λ *A B* . (*doubleton?* (*intersection A B*))	25
every	λ *A B* . (*nonempty? A*)	λ *A B* . (*subset? A B*)	10
most	λ *A B* . (*nonempty? A*)	λ *A B* . (*card>* (*intersection A B*) (*set-difference A B*))	1
none	λ *A B* . (*nonempty? A*)	λ *A B* . (*empty?* (*intersection A B*))	1
neither	λ *A B* . (*doubleton? A*)	λ *A B* . (*empty?* (*intersection A B*))	1

Figure 3-2: Target quantifier meanings for the learning model.

might be captured with either λ *A B* . (*singleton?* (*intersection A B*)) or λ *A B*. (*nonempty?* (*intersection A B*)). In this table, the meanings for "a" and "every" could alternatively mean "some" and "all" and learning to distinguish these respective pairs surely presents an interesting problem, but this is not addressed here.

The "weights" in this table were set by hand to make the target lexicon approximate English production frequencies of these words, on randomly generated sets of objects. This makes the adult productions approximate English, and so the task for learners—who do not know these weights—is as close as possible to learning English. Figure 3-3 shows the probability (normalized frequency) of each word according to CHILDES, and when the lexicon in Figure 3-2 is used to generate words. Using just a few different possible weight

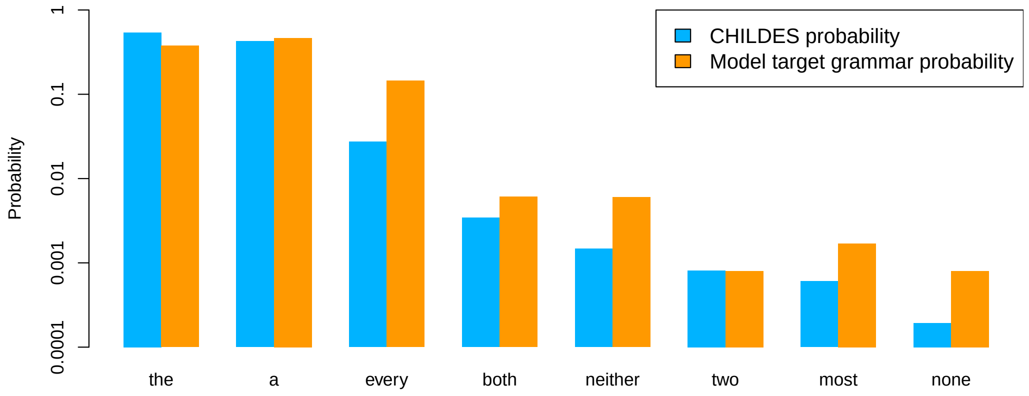


Figure 3-3: Word frequencies from CHILDES (MacWhinney, 2000) compared with probability-of-mention according to the target grammar in Figure 3-4, when labeling randomly generated sets. The model frequency distribution is used as the "adult" utterances for testing the learning model.

Nonterminal		Expansion	Gloss
START	→	λ *A B* . *BOOL*	Function of *A* and *B*
BOOL	→	*true*	Always true
	→	*false*	Always false
	→	(*total-intersection A B*)	The intersection of *A* and *B* exhaust the entire set.
	→	(*card> SET SET*)	Compare cardinalities (>)
	→	(*card= SET SET*)	Check if cardinalities are equal
	→	(*and BOOL BOOL*)	Boolean conjunction
	→	(*or BOOL BOOL*)	Boolean disjunction
	→	(*not BOOL BOOL*)	Boolean negation
	→	(*subset? SET SET*)	Is a subset?
	→	(*empty? SET*)	Is a set empty?
	→	(*nonempty? SET*)	Is a set not empty?
	→	(*singleton? SET*)	Contains 1 element?
	→	(*doubleton? SET*)	Contains 2 elements?
	→	(*tripleton? SET*)	Contains 3 elements?
SET	→	(*union SET SET*)	Union of sets
	→	(*intersection SET SET*)	Intersection of sets
	→	(*set-difference SET SET*)	Difference of sets
	→	(*complement SET*)	Complement of a set
	→	*A*	Argument *A*
	→	*B*	Argument *B*

Figure 3-4: A grammar that generates quantifier meanings.

values allows us to approximate the observed word distribution so that we can study words which occur with a range of frequencies spanning several orders of magnitude.

Each word in the adult "target" lexicon is written as a lambda expression that uses set-theoretic primitives. For the learner, these lambda expressions can be modeled as being generated from a *grammar* of concepts, corresponding to a fully productive, compositional, representation language for expressing word meanings. Figure 3-4 presents a context-free grammar to generate these expressions. To see how the grammar could generate an expression such as,

$$\lambda\, A\ B\ .\ (\textit{subset? } A\ (\textit{complement } B)), \tag{3.8}$$

we first start with the *START* symbol. We then recursively expand nonterminals according to the possible rules in Figure 3-4 until no more nonterminals remain. The only possible way to expand *START* is to λ *A B* . *BOOL*, meaning we always will generate an expression representing a function of two arguments, *A* and *B*. This function returns a boolean (*BOOL*) since *BOOL* only expands to functions which return boolean values. For instance, a *BOOL* can expand to (*subset? SET SET*), yielding the expression λ *A B* . (*subset? SET SET*). Here, we can expand the first *SET* to *A*, and the second set to (*complement SET*), and expand this last *SET* to *B*. This yields the full expression in (3.8).

This grammar includes a number of primitive operations that manipulate logical values (*and*, *or*, *not*), sets (*union*, *intersection*, *set-difference*, *complement*), small-set cardinalities (*singleton?*, *doubleton?*, *tripleton?*), and which can relate properties of sets to truth values (*subset?*, *empty?*, *nonempty?*). We also include the ability to form trivial expressions such as $\lambda\, A\ B\ .\ true$, and also include one special expression, $\lambda\, A\ B$. (*total-intersection A B*) which is true if the intersection of *A* and *B* exhaust the entire context. The cardinality primitives and special form of $\lambda\, A\ B$. (*total-intersection A B*) are motivated by the errors that children commonly make. In particular, on Give-N tasks, children often respond from a "base" distribution that is biased toward small cardinalities and biased towards responding with the entire set (Sarnecka & Lee, 2009; Lee & Sarnecka, 2010b, 2010a). That is, Sarnecka and Lee show with a Bayesian data analysis that after controlling for numerical knowledge, children's chance responses can be modeled as coming from a distribution that assigns small sets and the entire set large probability mass. This suggests that both small set primitives, and representations of *A* and *B* exhausting the entire set should be conceptually preferred, which here means including these primitive functions. Inclusion of the *total-intersection* rule in the grammar will be important later for explaining errors children make with learning "every."

There are a vast number of potential hypotheses that can be generated according to this grammar. Some of these are relatively complex. For instance,

$$\lambda\, A\ B\ .\ (or\ (empty?\ (set\text{-}difference\ B\ A))\ (not\ (doubleton?\ (intersection\ A\ B)))) \qquad (3.9)$$

is a quantifier which could be expressed in this language. This is true if all elements of *A* are in *B*, or if the intersection of *A* and *B* does not contain exactly two elements. Of course, one can also generate hypotheses like those necessary for natural language, including all of those shown in Figure 3-2. The challenge for the learning model is to infer which representations are best given some evidence. This inductive problem is formalized and solved using a probabilistic model over the space of meanings defined by the grammar in Figure 3-4. This approach is best understood in the context of *program induction*, an area of machine learning which attempts to learn programs—here, lambda expressions—that generate observed data (e.g., Koza, 1992).

While our grammar for concepts defines a capacity for creating new representations, the grammar is qualitatively unlike those typically posited as linguistic theories. First, it requires no additional principles or parameters—the system is itself extremely simple, only a context-free grammar. This means that the amount of information such a grammar "builds in" for learners is very small. Second, this grammar uses only plausibly domain-general primitives, and is highly overlapping with representational systems posited in other domains, for instance, in learning the meaning of number words (Piantadosi et al., submitted)[12]. This means that this approach potentially provides a simple unified way of understanding how learners acquire many types of structured conceptual systems in language.

3.3.2 Challenges for learning

To summarize, we have outlined several challenges for accounts of quantifier learning. Most basically, quantifier meanings are abstract and the representational system that supports them cannot simply be based on associations between, say, words and objects or events. It is also difficult to see how such meanings could be captured by simply remembering previous instances, as in prototype or exemplar models. Instead, learners must have some mechanism for representing and inferring unseen abstract relations between sets of objects. Second, quantifiers have more than just a literal meaning. Each quantifier also has assumptions it carries and different probabilities of being uttered in any situation, and learners must have a mechanism to learn these meanings, and indeed to correctly distinguish presuppositions from literal meanings. The space of possible meanings—both presuppositional and literal—is complex and rich, with word meanings potentially in subset relations to each other. The evidence that children get is bound to be noisy—especially for function words—since children may lack knowledge of the relevant content words, they

[12] The major exception to this is that the primitives here include operations for exact cardinality, the very thing Piantadosi et al. (submitted) are trying to learn. The results from Pietroski et al. (2009) indicate that these word meanings are expressed using the approximate, not exact systems—perhaps providing indirect evidence for the claim of Piantadosi et al. (submitted) that exact meanings are constructed from other logical primitives. Here, we choose to include exact meanings rather than approximate because it substantially simplifies implementation of the model: approximate meanings return distributions (or samples from them) on set cardinalities, but exact cardinality returns a single number. However, we expect that our learnability results will generalize to representation systems that use approximate numerosity.

may misunderstand the intended referents, or adults may occasionally say wrong or infelicitous things. It is therefore unrealistic to put hope in a theory that is irreversible, such as the subset principle.

We aim to address all of these challenges with a fully implemented[13] learning model presented in the next section. This model formulates quantifier acquisition as a problem of learning the correct LOT representation to describe observed quantifier usages. As we argue, this can solve all of the above problems learners face, providing a working hypothesis for the representational and computational substrate that supports quantifier learning, and perhaps function word learning in general. Our aim here is breadth—to show how learning in a sufficiently powerful representation system can explain both the ability of children to learn these word meanings at all, and the general types of errors they make. The details of this account like depend on many factors we have simplified away, including memory, other cognitive systems, syntax, and the precise nature of the input.

3.3.3 The probabilistic model

Here we present the learning model is a relatively abstract form by supposing that learners must map an arbitrary set of words to an arbitrary set of meanings. We assume that there is some collection of words $w_1, w_2, \ldots, w_k$ that the learner is trying to discover meanings for. For the purposes of this section, this could be *any* set of words, even those that are not quantifiers. We denote the meaning of word w_i by m_i, and the collection of all meanings as $m = (m_1, m_2, \ldots, m_k)$. In the case of quantifiers, for instance, m_5 might be the literal meaning, presupposition, and word weight for the word "every"[14]. It might seem peculiar that we are formalizing the learning of *sets* of words, rather than individual words. This turns out to be an important part of a generative statistical model of language acquisition, since the probability of a particular word or utterance can only be computed by using what other utterances possibly could have been produced.

We will assume that the learner hears a sequence of uttered words $u_1, u_2, \ldots, u_n$ each in

[13] Running code is available from the first author.

[14] The meanings need not necessarily be semantic—they could also include pieces of syntactic structure as in, for instance, Combinatory Categorial Grammar (Steedman, 2000).

a corresponding linguistic context c_i. We will here consider the simplest case, where each u_i is only a single word from $w_1,\ldots,w_k$. This is equivalent to assuming that each w_i occurs in contexts containing only other known words. For instance, the parent might say "most snakes are hungry animals", with u_i = "most" and c_i = "____ snakes are hungry animals". In our quantifier learning model, we assume children already know (or can guess) the meanings of "snakes," "are," "hungry," "animals," and are trying to figure out the meaning of "most." However, the results we describe here could be extended to utterances of any form, containing any number of unknown words.

If m_i is a hypothesized meaning of w_i, with $m = (m_1, m_2, \ldots, m_k)$, then we can write using Bayes rule,

$$P(m \mid u_1,\ldots,u_n,c_1,\ldots,c_n) \propto P(u_1,\ldots,u_n \mid m,c_1,\ldots,c_n) \cdot P(m). \tag{3.10}$$

This equation says that the probability m is the correct set of meanings, given the u_i and c_i, depends on two things. First, it depends on $P(m)$, a prior probability on meanings. The prior formalizes the expectations learners have about the correct meaning before any data has been observed. In this paper, we do this by converting the grammar in Figure 3-4 to a *probabilistic* context-free grammar (PCFG) by supposing that each nonterminal is equally likely to expand by any of its rules, except the rules that generate A and B, the arguments to the function, are 10 times as likely than other rules. This probabilistic grammar induces a probability distribution on expressions, assigning a probability to an expression corresponding to how likely it is to sample jointly each of the rules required to generate the expression. This assigns short expressions higher prior probability, corresponding to the intuition that simple (concise) representations should be preferred a-priori by rational learners (Feldman, 2000; Chater & Vitányi, 2003; Goodman et al., 2008; Piantadosi et al., submitted). This prior is similar to more sophisticated versions of rule-length priors developed in other work (Goodman et al., 2008). The prior on any set of meanings m can be found by assuming that each was chosen independently from this PCFG. The up-weighting of rules that generate A and B is necessary to ensure that the grammar does not generate infinitely long expressions, and also to bias the learner to preferentially use the sets that are arguments to the function.

The other term in (3.10), $P(u_1, \ldots, u_n \mid m, c_i, \ldots, c_n)$, is the *likelihood.* The likelihood measures the probability that $u_1, \ldots, u_n$ would be produced in their corresponding contexts if m was the correct meaning. It makes sense to assume that each utterance u_i depends on m and c_i, and is independent of the other utterances and contexts once m and c_i are known. This means that the likelihood can be rewritten as

$$P(u_1, \ldots, u_n \mid m, c_i, \ldots, c_n) = \prod_{i=1}^{n} P(u_i \mid m, c_i). \tag{3.11}$$

In establishing learnability, it is important that learners know the correct form of the likelihood, $P(u_i \mid m, c_i)$, although they do not know the correct target meanings. Learners must be able to say if a particular hypothesized set of meanings were correct, what the probability of an utterance would be. Learners' knowledge may be very weak: for our meanings, part of the meanings m are the weights, which determine how likely each true word is to be uttered. These weights need not be known to learners, so learners have to discover each word's probability of being uttered in each context. Instead, learners must only understand the *structure* of this model—for any guess at the correct weights, adult utterances are typically chosen by sampling from the true and presuppositionally valid words, with probability proportional to their weights. Our implementation is therefore *Gricean*: speakers tend to say things which are true and relevant to the current context (Grice, 1975).

Formally, we will suppose that adults first choose with some probability α_p whether or not to say something which has met presuppositions. Assuming they do, then they choose from the true words, proportional to their weights, with probability α_t, and something at random (proportional to its weights) otherwise. If the adult chooses not to say something with a met presupposition, they choose from all words proportional to their weights. Thus, two parameters α_p and α_t characterize the probability that true and presuppositionally valid words are uttered. For simplicity we assume α_p and α_t are high and known to learners, but neither of these is necessary for the learnability proof. To compute $P(u_i \mid m, c_i)$ we must sum over all the ways words could be generated. So if u_i is true and presuppositionally valid in context c_i, then

$$P(u_i \mid m, c_i) = \frac{\alpha_p \cdot \alpha_t \cdot w(u_i)}{W_{p\wedge t}} + \frac{\alpha_p(1-\alpha_t) \cdot w(u_i)}{W_p} + \frac{(1-\alpha_p) \cdot w(u_i)}{W}. \tag{3.12}$$

Here, $w(u_i)$ is the weight of the quantifier u_i, according to the learner's hypothesized meanings m. $W_{p\wedge t}$ is the sum of the weights of all true and presuppositionally valid words for c_i, W_p is the sum of all weights of presuppositionally valid words for c_i, and W is the sum of weights of all words. So, for instance, the second term is included because a word could have been generated by choosing a word at random from the presuppositionally valid words, ignoring truth values. This happens with probability $\alpha_p \cdot (1-\alpha_t)$, and generates the word u_i with probability $w(u_i)/W_p$. To compute words which are either false or presuppositionally invalid, the corresponding terms from (3.12) are dropped. For instance, if u_i is not presuppositionally valid, only the last term in (3.12) is included since the word could not have been generated by choosing from presuppositionally valid words, or true words. Our learnability results do not depend on this specific choice of likelihood function, but we use it here because it captures the notion that true and presuppositionally valid words tend to be uttered.

Equation 3.12 embodies an important principle for Bayesian statistical learning: the *size principle* (Tenenbaum, 1999). The size principle has been argued for in other models of word learning (Xu & Tenenbaum, 2007; Piantadosi et al., 2008; Frank et al., 2007a; Piantadosi et al., submitted), and here means that the probability that any particular word u_i is used in c_i depends on the weight of the words which alternatively could have been uttered[15].

3.3.4 How the size principle solves the subset problem

To illustrate how the size principle solves the subset problem, it is useful to consider the example of "every" and "some," and suppose that they are the only words in the lexicon. For simplicity, in this section, we also assume that $\alpha_p = \alpha_t = 1$, so that the only utterances under consideration are true and presuppositionally valid. In this case, the likelihood $P(u_i \mid m, c_i)$

[15] This is why in Equation 3.11, the probability of u_i depends on all of m, and not just m_i.

is only the first term of (3.12), which reduces to

$$P(u_i \mid m, c_i) = \frac{w(u_i)}{W_{p\wedge t}}. \tag{3.13}$$

The key is to look at the likelihood of observed instances of "some" when "every" means SOME, compared to when "every" means EVERY. If "every" meant EVERY, then most of the time it would not be true in a context where "some" was uttered, since it is logically stronger. This means that it will typically be the case that

$$P(\text{“some”} \mid m, c_i) = \frac{w(\text{“some”})}{w(\text{“some”})} = 1. \tag{3.14}$$

since if "every" is not true, $W_{p\wedge t}$ is only the weight of "some."

In contrast, if "every" meant SOME, then the observed instances of "some" would be *less likely*:

$$P(\text{“some”} \mid m, c_i) = \frac{w(\text{“some”})}{w(\text{“some”}) + w(\text{“every”})} < 1, \tag{3.15}$$

since now "every" would be true in all the situations where "some" is. Letting "every" mean SOME decreases the likelihood of the observed instances of "some." The reason for this is intuitive: if "every" meant SOME, each instance of "some" would have to have been sampled from two possible true utterances rather than just one. Analogously, in a sequence of coin flips, ten heads in a row are more likely under a hypothesis of a coin with heads on both sides, than a coin with heads on one side and tails on another. Intuitively if both heads and tails were possible, the sequence of heads is less likely to occur; if "every" and "some" could both be used in many contexts, the observed instances of "some" would have to be less likely. Probability mass should not be held out for events that don't occur. This is an application of the size principle: hypotheses can assign the observed utterances higher likelihood if they predict that fewer words are true in each context.

The size principle is similar to the subset principle proposed previously in that it prefers meanings which are logically strong, or true less often. However, it differs from the subset principle in the root cause of this preference. The size principle prefers meanings which are true less often because they can assign the observed utterances a higher *likelihood*, all else being equal. In contrast, the subset principle puts the bias in the prior, assuming that learner's innate expectations lead them to prefer stronger logical meanings. This is undesirable for a few reasons. First, it requires positing a rich set of innate logical orderings, a claim which seems unlikely without strong empirical evidence. Second, it cannot handle noise: an ideal learner will always use the data to arrive at the correct meaning, overwhelming any prior expectations. This means that enough misunderstood or noisy occurrences will overwhelm any bias for logically stronger meanings in the prior. The advantage of putting the preference in the likelihood is that it falls out very naturally by positing that learners think about how language is generated. All they must realize is that utterances are generated using a set of meanings, and that the total probability of all possible utterances must sum to 1. In this sense, the size principle is a simple consequence of formalizing a fully generative statistical model. One could imagine alternative models that, for instance, set $P(u_i \mid m, c_i) = \alpha$ if u_i is true, and $1 - \alpha$ if u_i is false. Such a model is intuitive in penalizing incorrect meanings, but doesn't specify a valid probability distribution and would fail to solve the subset problem.

In the next section we show that this Bayesian framework is considerably more powerful than only solving the subset problem: it can always learn the correct set of meanings.

3.3.5 The Bayesian model is provably learnable

This section is meant to introduce a simple proof of the learnability of meanings in a Bayesian framework. The proof is not novel—it is well-known that in the limit, the data will support the correct model. We present it here because we hope to refine the debate on learnability, moving away from questions of what is in principle learnable, to questions of what can be learned by plausible computational models on realistic data. The proof is different from, for instance, Chater and Vitányi (2007)'s very general proof of language learnability in that it does not require Kolmogorov complexity (Li & Vitányi, 2008), and is more obviously applicable to psychologically plausible representations like those we use for quantifier meanings. Moreover, it explicitly formalizes the learning problem as one of inferring unseen linguistic structures, rather than asymptotically matching the statistics of observed language.

To show learnability with this setup, we will consider the *Bayes factor*, a measurement which quantifies the strength of belief an ideal learner should have for one model over another (Jeffreys, 1998). The Bayes factor is simply the log ratio of the posterior probabilities of two statistical models. In this case, one statistical model will be data generated with the correct set of meanings, $\hat{m}$. The alternative model will be any other set of meanings, m. The Bayes factor in favor of $\hat{m}$ is then given by

$$BF = \log \frac{P(\hat{m} \mid u_1, \ldots, u_n, c_1, \ldots, c_n)}{P(m \mid u_1, \ldots, u_n, c_1, \ldots, c_n)}. \tag{3.16}$$

The Bayes factor ranges from negative infinity (definitive support of m) to positive infinity (definitive support of $\hat{m}$), and equals zero when $\hat{m}$ and m have the same posterior probability. We will show that as the amount of data gets large, the Bayes factor in support of the correct model over any alternative goes to infinity. Thus, with enough positive examples, learners will accumulate an arbitrarily large amount of evidence supporting the correct set of meanings.

Using Bayes rule, we can rewrite the Bayes factor as

$$\log \frac{P(\hat{m})P(u_1, \ldots, u_n \mid \hat{m}, c_1, \ldots, c_n)}{P(m)P(u_1, \ldots, u_n \mid m, c_1, \ldots, c_n)}. \tag{3.17}$$

As above, we assume that each u_i depends only on c_i and is conditionally independent of all other u_j and c_j $(j \neq i)$. In other words, each utterance depends only on the context it occurs in and not any other utterances or contexts. This means that we can factor (3.17), as

$$\log \left[\frac{P(\hat{m})}{P(m)} \prod_{i=1}^{n} \frac{P(u_i \mid \hat{m}, c_i)}{P(u_i \mid m, c_i)} \right]. \tag{3.18}$$

which can be re-written to

$$\log \frac{P(\hat{m})}{P(m)} + \sum_{i=1}^{n} \log \frac{P(u_i \mid \hat{m}, c_i)}{P(u_i \mid m, c_i)}. \tag{3.19}$$

This says that the Bayes factor can be re-written as the sum of the log ratio between the prior on m and $\hat{m}$, a constant, plus the sum of the ratio between the likelihoods on each data point. We are concerned with what happens for learners who get increasing amounts of data generated from the correct model. This means that for each i, u_i is chosen according to the correct adult meaning, $P(u_i \mid \hat{m}, c_i)$. Clearly, as n gets large, the behavior of (3.19) depends on the expected value of $\log \frac{P(u_i|\hat{m},c_i)}{P(u_i|m,c_i)}$ under sampling utterances from the correct set of meanings. If this term tends to be positive, then (3.19) will grow to positive infinity as n gets large—increasing amounts of evidence will support the correct meaning $\hat{m}$. This will eventually overwhelm any effect of the prior log ratio $\log \frac{P(\hat{m})}{P(m)}$, meaning that learners will eventually assign $\hat{m}$ the highest posterior probability. This would establish learnability because it would show that the correct model will eventually be preferred over any alternative hypothesis.

The expected value of this term can be found by averaging or integrating over contexts c_i and utterances u_i. Let's assume that each context c_i has a probability given by $P(c_i)$. In c_i, u_i has a probability of $P(u_i \mid \hat{m}, c_i)$, since u_i is generated from the adult grammar. Thus,

$$\mathbb{E}_{u_i,ci}\left[\log \frac{P(u_i \mid \hat{m}, c_i)}{P(u_i \mid m, c_i)}\right] = \sum_{c_i} P(c_i) \sum_{u_i} P(u_i \mid \hat{m}, c_i) \log \frac{P(u_i \mid \hat{m}, c_i)}{P(u_i \mid m, c_i)}. \tag{3.20}$$

A standard theorem in information theory and probability, known as the *Gibbs inequality*, holds that

$$\sum_x A(x) \log \frac{A(x)}{B(x)} > 0 \tag{3.21}$$

if A and B are different distributions on elements x. A basic proof of this is provided in Cover and Thomas (2006, Theorem 2.6.3) . This applies to (3.20), by letting $A(u_i) = P(u_i \mid \hat{m}, c_i)$ and $B(u_i) = P(u_i \mid m, c_i)$. Thus, for any c_i the term $\sum_{u_i} P(u_i \mid \hat{m}, c_i) \log \frac{P(u_i|\hat{m},c_i)}{P(u_i|m,c_i)} > 0$ meaning that the entire expected values in (3.20) is greater than 0. This means that on average, the next data point provides support in favor of the correct meanings $\hat{m}$. This completes the proof, since it shows that with enough data, an ideal learner will favor $\hat{m}$ over any alternative m. Note that we have made no assumptions about the form of $P(u_i \mid m, c_i)$—that is, about *how* the set of meanings m give rise to utterances. Under *any* such system, corresponding to any linguistic system, the above argument will hold: negative evidence is not necessary.

3.4 The implemented learning model

More important than establishing learnability in theory is showing that the correct meanings are learnable with a developmentally plausible amount of data. Here, we use an implemented version of the model to study how many example utterances are necessary to correctly learn the meanings in Figure 3-2.

3.4.1 Methods

Because naturalistic data consisting of quantifiers used by parents in the presence of sets of objects is not available, we constructed simulated data by creating sets at random, and sampling adult meanings according to the likelihood process described above, with $\alpha_p = \alpha_t = 0.9$. This means that the data is fairly noisy, with roughly 10% of the utterances not satisfying presuppositions, and of those that do, 10% are false. We generated sets at random, each containing between 1 and 8 objects. Each object in a set was one of three animals (mouse, pig, rabbit) that was one of three colors (white, brown, or pink). Here, the argument A was an animal and B was a color. For each set, we sampled utterances according to the target grammar: for instance, for a set containing a pink mouse and two brown rabbits, we might sample the quantifier "some" in the context " ____ mouse is pink."

The model described in the previous section is a *computational* theory of quantifier learning, not an algorithmic one (Marr, 1982). We combine several techniques from probabilistic modeling to implement a working version of this model. This provides us with learning curves as the amount of data for the model varies, which represent the learning curves for an idealized statistical learner, operating over the space of meanings we describe. In principle, learners should be able to consider any hypothesis generated by the grammar in Figure 3-4. In practice, most of the hypotheses this grammar generates are very low probability, either by being long (small prior) or not explaining the data (low in the likelihood). The first approximation that we make is that our algorithm looks only at hypotheses that use 10 or fewer rule expansions[16]. We enumerate this space of hypotheses, and, for computational tractability, collapse equivalent hypotheses. Thus, for instance,

[16] Hypotheses which are excluded this way have a prior probability less than 1 in 10 million.

$\lambda\ A\ B$. (*or* (*singleton? A*) *false*) is not treated distinctly from $\lambda\ A\ B$. (*singleton? A*). This results in 699 hypotheses (or equivalence classes of hypotheses) that represent distinct functions on sets. This space was treated as a fixed, finite hypothesis space of expressions for purposes of inference. We note that this still represents a huge effective hypothesis space for the learner since the number of possible ways of mapping 8 word meanings and presuppositions to 699 hypotheses is $699^{8+8} \approx 10^{45}$. In addition, we allowed each word to have a weight chosen from $\{1, 5, 10, 25, 50, 100\}$, corresponding to a superset of weights necessary for the target meanings. Again, each mapping from words to meanings is a *lexicon*. To search through lexicons, we first ran Gibbs sampling (Geman & Geman, 1984) for varying amounts of data from 0 to 2000 sets. For each set size we ran 100 separate Gibbs sampling runs, storing 10 lexicons with highest posterior for each run at each amount of data. This finite space of lexicons was treated as the finite hypothesis space for constructing the learning curves and results here. Note that this method means that the target lexicon had to be found at some amount of data by Gibbs sampling. However, once it is found by one run, it will be included in the final finite hypothesis space of lexicons, allowing for better statistical estimation. This is a form of selective model averaging (Madigan & Raftery, 1994), that we have used in other similar learning models (Piantadosi et al., submitted). It amounts to using sampling techniques as essentially search for finding high probability hypotheses, and then using the high probability hypotheses as a finite space for performing statistical inference.

3.4.2 Idealized learnability of quantifiers

Figure 3-5 shows learning curves for learning model, broken down by each of its three aspects of meaning: presupposition, literal meaning, and probability of production. This shows the model's probability of correctly learning each aspect of word meaning, as a function of the number of examples observed. Note that the *x*-axis in this plot does not represent the number of times that each of these words is heard; instead it shows the total number of labeled sets, only some of which are labeled with the word. This is an important point because a contributor to learning is both the number of times a word is heard, and

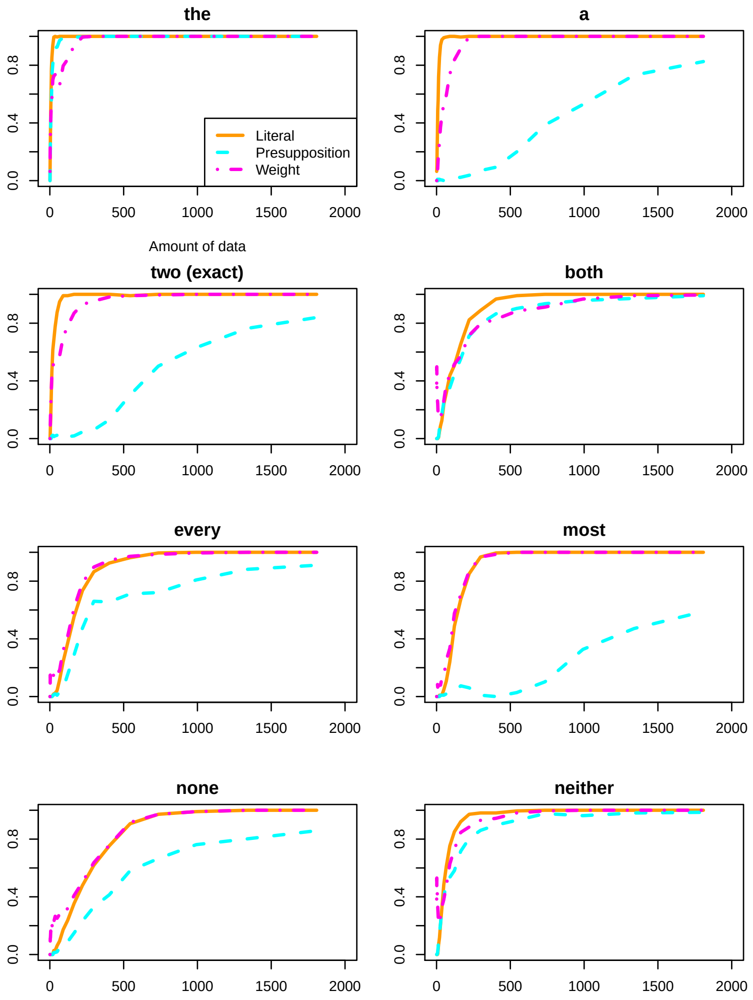


Figure 3-5: Learning curves for $\alpha_p = \alpha_t = 0.9$, showing model proportion correct (y-axis) versus amount of data (x-axis) for each aspect of meaning.

the number of times it is not, using the implicit negative evidence provided by the size principle.

The figure shows that the most frequent word in the input, “the,” is learned extremely quickly, within around 100 labeled sets, while infrequent words like “most” and “none” take several hundred. The absolute scale of the learning rates and variability is quite important since it shows that a set of 8 determiners can be learned simultaneously from noisy positive evidence, using a developmentally plausible amount of data. It would not take children hundreds of thousands or millions of determiners to learn the correct meaning—hundreds to one or two thousand examples are sufficient. To put this amount of data in perspective, determiners or quantifiers are used in Adam's section of the Brown corpus in CHILDES (MacWhinney, 2000) over 8000 times, and this corpus represents only a small subset of the data Adam heard. This quantity of data would be enough for an ideal learner to discover all aspects of meaning studied here, even assuming that only a quarter of instances have clear and known referents.

An important facet of this learning account is that it can handle noisy evidence. Learning proceeds quickly even in the presence of about 10% noise consisting of utterance produced at random. This is possible because the model works cross-situationally, aggregating evidence from multiple utterances in multiple contexts in order to determine the most likely word meanings. Cross-situational word learning (Yu & Smith, 2007; Siskind, 1996; Vogt & Smith, 2005; Yu & Ballard, 2007; Frank et al., 2007a) is likely especially important for function word meanings because their meaning is never unambiguous from a single context.

This plot reveals a strong tendency for literal meanings to be learned before presuppositions, and around the same time as the word weights. That is, under our form of the likelihood, the assumptions required for a meaning to get a truth value are hardest for ideal learners to determine. For the model, the majority of these errors are thinking that the utterance always has met presuppositions (λ *A B. true*), and other likely errors are to presuppose that the intersection of *A* and *B* is nonempty (λ *A B.* (*nonempty?* (*intersection A B*))), or that *A* and *B* exhaust the entire set (λ *A B.* (*total-intersection A B*)). We note that the difficulty of lexical presuppositions appears to be a robust prediction of this form of the likelihood; although to our knowledge lexical presuppositions have not been studied empirically in detail other than for the word "the," discussed below.

3.4.3 Constraints on quantifier meanings

One interesting aspect of the learning model is that its hypothesis space is relatively unrestricted. On one hand, this is a strength because it means that fewer constraints need to be posited as part of children's innate linguistic repertoire. On the other hand, it might seem that an unrestricted hypothesis space would make learning unnecessarily difficult because there are so many hypotheses to consider. One potential constraint on quantifier meanings that is proposed be universal is *conservativity* (Keenan & Stavi, 1986; Barwise & Cooper, 1981). In our notation, a quantifier is conservative if it depends only on the elements of A, the first argument to the function. Thus, "Most men are happy" can be checked by looking only at the set of men[17]. Keenan and Stavi (1986) argue that conservativity provides a useful constraint for language learners; in a simple example involving sets of two individuals, they count $65,536$ possible quantifiers, only 512 of which are conservative. Intuitively, learners should benefit by narrowing down the space of possible meanings by a factor of 128. On the other hand, it may be the case that most of the quantifiers that are ruled out with a conservativity constraint are already low-probability. Furthermore, a factor of 128 is not necessarily very useful: it corresponds to just 7 *bits* of information about the correct meaning.

Figure 3-6 shows a model-based analysis of how much conservativity helps under the assumptions of the idealized learning model. The line labeled 'C' (conservative) shows learners who only consider conservative quantifiers[18]. This shows that on average conservativity helps very little; quantifier meanings are almost equally as learnable in the unconstrained space. The reason for this is that the data evidently provides a much more useful constraint than conservativity. Observed usages of quantifiers quickly provides much more information about their meaning than a priori constraints. There is still evidence that children prefer to learn quantifiers that are conservative (Hunter & Conroy, 2009), but this figure demonstrates that this constraint should not be posited for reasons of learnability.

This raises the question of whether any constraints on quantifier meanings could substantially aid learning. One way to determine this is to compare the unconstrained learner to what might be considered the most constrained learner possible. The blue 'R' (restricted) lines in Figure 3-6 show the performance of a learner who only considered expressions necessary for the literal meanings or presuppositions in Figure 3-2 as possible hypotheses. That is, for such a learner, there are only the necessary semantic concepts, and the task

[17] Conservativity is perhaps best understood by a potential counter-example to it, "only." "Only men are happy" depends on the set of things which are *not* men, violating conservativity. However, "only" is argued not to be a quantifier due to the fact that it patterns differently in some syntactic constructions.

[18] For computational tractability, conservativity was evaluated "empirically" by evaluating the quantifier on 200 randomly generated sets.

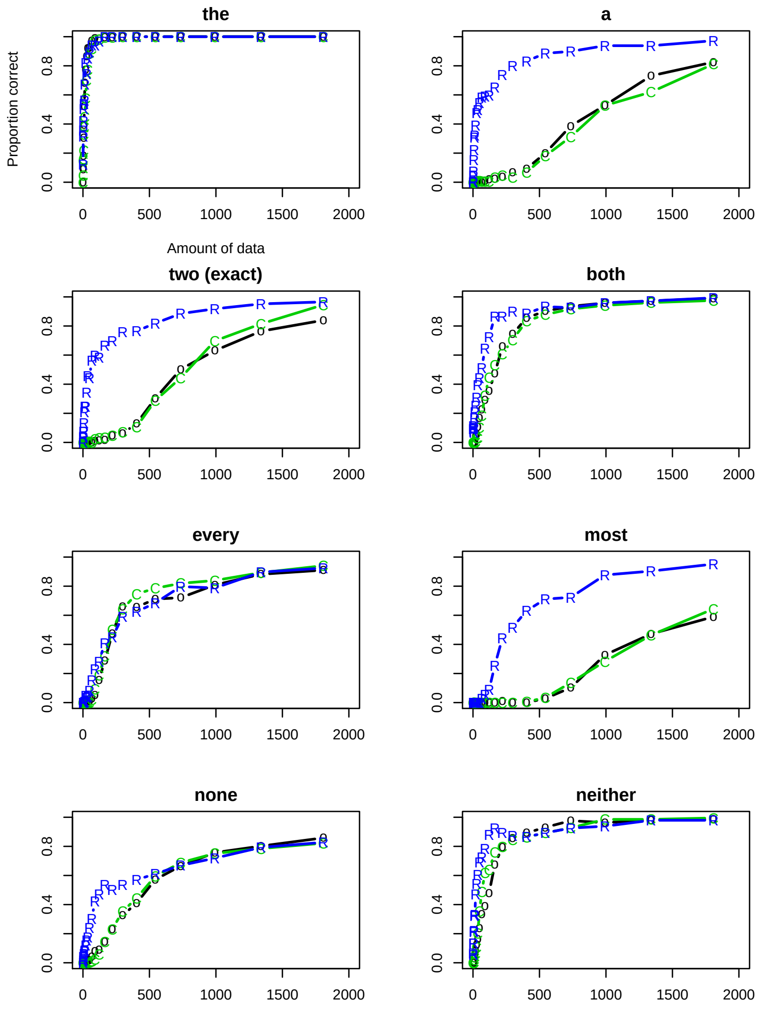


Figure 3-6: Learning curves for the basic unrestricted model (black), conservative quantifiers (C, green), and the maximally restricted model (R, blue). The y-axis shows probability of correct acquisition of all aspects of meanings (literal, presupposition, production probability).

of learning is to figure out which maps to which words. Note that this is the minimum amount of learning which must happen, since even under such an extreme nativist theory, learners must still figure out the arbitrary mapping between phonological forms and meanings. This line in Figure 3-6 shows that such a learner does not have a much easier time than the unconstrained learner who considers all possible representations for each word. Such a constrained space only substantially helps for “a,” “two,” and “most.” This indicates that in many cases the hard part of learning is narrowing down the correct meaning from among several high-probability competitors. The problem is not in weeding out implausible hypotheses. This is potentially a non-obvious and important point for language acquisition: restricted hypothesis spaces—often posited to reduce computational demands on learners—do not always substantially improve acquisition, especially if they still include many of the likely competitors included in the unrestricted space. Said another way, learning in unrestricted hypothesis spaces is not necessarily much harder for ideal learners than restricted ones, at least for models where the data is capable of quickly providing a substantial amount of information about the target meanings. Constraints need not be posited for reasons of learnability.

3.5 Detailed patterns of acquisition

Here we show that the model additionally provides a plausible account of quantifier meaning from a developmental perspective by showing that the learning patterns of the model are similar to empirically observed patterns.

3.5.1 Probability of production

Figure 3-5 showed that learning the probability of production—word weights—was as easy for the model as learning the literal meanings. This makes the prediction that children's probability of producing each of these words should match their parents' probability of production, starting from earliest utterances. That is, once children have figured out when to use each word, they should also know the relative frequency of each word. This would not be predicted by the model if, for instance, the word weights took substantially longer than the literal meanings to learn. Figure 3-7 shows the parental relative frequency for the eight determiners, plotted against childrens' relative frequency, for every year from 1 to 5 years old (point size). This shows a strong linear correlation of $R^2 = 0.91$ between parents' production frequency and children's, spanning several orders of magnitude in frequency and ranging throughout development. These frequencies do not change substantially for parents or children throughout development, indicating that children's earliest productions follow the relative distribution of the adult grammar. Thus, as predicted by the model, children who know the meaning of determiners well enough to use them, also know their rela-

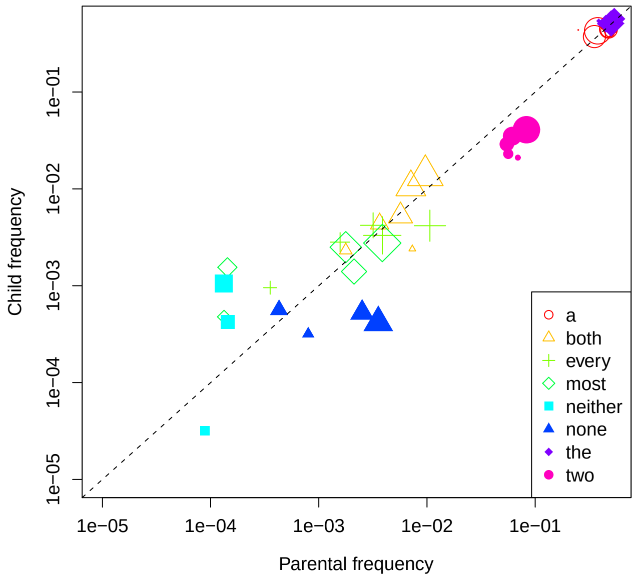


Figure 3-7: Data from CHILDES (MacWhinney, 2000) showing parental production frequencies compared to child production frequencies, binned every 12 months from 1 to 5 years (point size small to large). The dotted line is $y = x$. These frequencies are correlated in the log domain at $R^2 = 0.91$ ($p < 0.001$).

tive frequencies. Of course, there are many other ways to get these patterns—for instance, if children often repeated what their parents said. Such nearly perfect correlations are not always found, however: children's probability of saying the function words "who," "what," "where," "when," "how," or "why" has a substantially lower correlation of $R^2 = 0.73$ with their parents' probability.

3.5.2 The definite article

Although "the" is the most frequent determiner in child-directed speech (Figure 3-3), children fail some tasks that test their comprehension of "the" until surprisingly late. English-speaking children as old as 9 *years* systematically arrive at non-adult semantics for "the" (Maratsos, 1974, 1976; Warden, 1974, 1976; Karmiloff-Smith, 1981; Modyanova & Wexler, 2007). For instance, after one turtle in an array of turtles is explicitly identified, children will fail to consistently put a star on "the turtle," instead putting it on any of the turtles (consistent with "a turtle") (Modyanova & Wexler, 2007). This represents a failure to understand that “the” requires a unique referent in the singular case (e.g., “the turtle”). In the plural case, this notion of uniqueness generalizes to *maximal* sets, so that “the turtles” refers to all of the turtles in the locally salient set, not a subset of them (Heim, 1991). Maratsos (1976) argues that analogous production errors result from children acting egocentrically, broadly in line with Piaget (1955): children may choose a unique referent and fail to recognize that others would not recognize it as unique. Under this view, children have full semantic competence for “the” but have difficulty integrating the necessary semantic representations with their understanding of others’ knowledge. An alternative is argued for by Wexler (2011): children may lack the uniqueness or maximality presupposition of “the.” In our formalization as lambda calculus, this incorrect presupposition might be

$$\lambda\, A\ B\ .\ true \tag{3.22}$$

instead of the correct form,

$$\lambda\, A\ B\ .\ (singleton?\ A). \tag{3.23}$$

But there are many other formulations of their knowledge of the presuppositions of “the” that are consistent with their errors, such as λ *A B*. (*nonempty? A*), or λ *A B* . (*nonempty?* (*intersection A B*)). Wexler (2011)’s account is specifically *maturational*, meaning that he argues children’s lexical entry for “the” is initially wrong because of biological facts about the computational system that supports language learning and use, rather than, for instance, difficulties with learning. One difficulty with a maturational account is that similar effects are seen in L2 language learners (Ko, Ionin, & Wexler, 2006; Ko, Perovic, Ionin, & Wexler, 2008), meaning that the effects are seen in learners who have already fully matured.

Our model suggests an alternative: perhaps children’s errors result, at least in part, from the difficulties faced during learning. Intuitively, *singleton?* may be difficult to distinguish from meanings such as *true* or the *nonempty?* because of subset-superset relationships among these hypotheses. To evaluate this, we can examine the errors made by the model over the course of learning. Figure 3-8 shows the top 10 most frequent presuppositions learned by the model. Note that because “the” is learned so quickly by the model, we have logarithmically transformed the *x*-axis (amount of data) to show more detail early

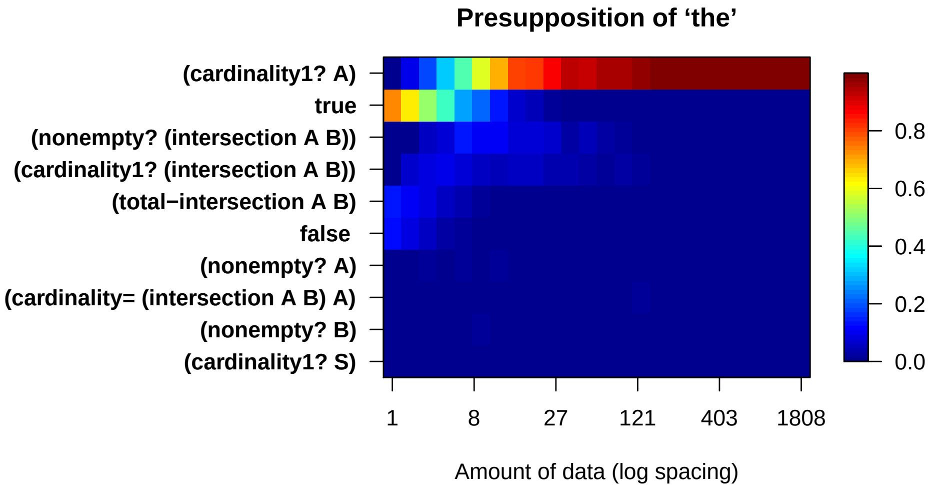


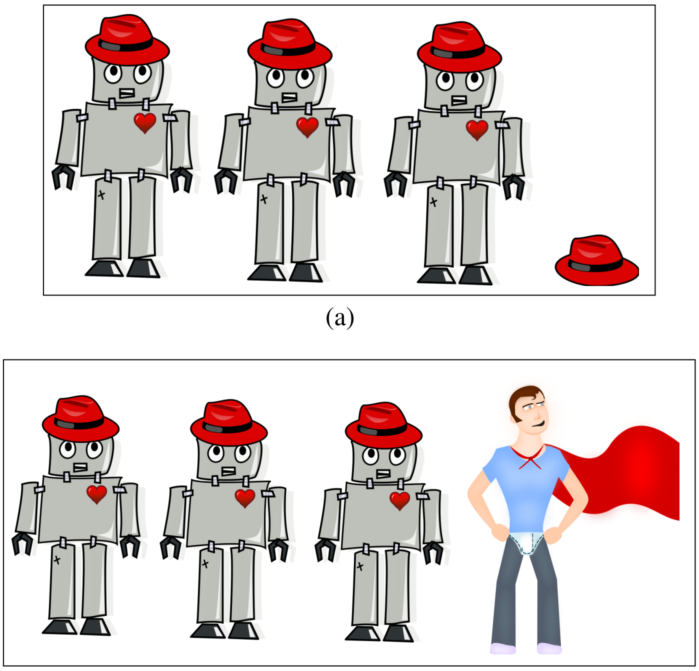
Figure 3-8: Posterior probability (z-axis) of the most-likely presuppositions of “the” (y-axis) over the course of acquisition (x-axis). Note the x-axis has been logarithmically transformed to show more detail early in acquisition.

in learning. The top row of this figure is the correct presupposition, which is eventually learned. The next several rows include the true presupposition, and nonemptyness of the intersection, which are consistent with the observed errors. This shows that the correct presupposition for “the” is indeed hardest to distinguish from meanings like λ *A B . true* and λ *A B* . (*nonempty?* (*intersection A B*)), both of which are consistent with children’s lack of presupposition. Indeed early in acquisition, the model consistently believes that “the” is always presuppositionally valid, represented by λ *A B . true*. This explanation in terms of difficulty for learners additionally explains why similar errors are common in adult L2 language learning. We believe it is substantially informative that a model operating over a relatively unrestricted set of meanings shows errors like children’s.

However, there is one aspect of the learning pattern here which is distinctly not like children: the model learns the correct meaning of “the” within dozens of examples, rather than requiring perhaps a decade of data. It is also possible that more complete versions of the model would show substantially delayed acquisition relative to these results. For simplicity, this implementation of the model assumed that “the” is only used with singular count nouns. In real acquisition, children hear these determiners with plural nouns (“the aliens”) and mass nouns ("the pasta"), and these tokens may substantially complicate acquisition[19]. In particular, learning maximality, a logical form that covers both singular and plural uses of "the," might be substantially more difficult if maximality must be expressed in terms of other primitives; here, uniqueness is "built in" by the primitive *singleton?*. The idea that maximality must be "built" is attractive since it can explain other evidence offered in support of maturational accounts of maximality: for instance, other work has found significant delays until around 7 years on wholly separate syntactic constructions that are argued to involve maximality (Munn, Miller, & Schmitt, 2006; Modyanova & Wexler, 2008; Caponigro, Pearl, Brooks, & Barner, 2010).

An alternative is that children actually do learn much of the meaning of the definite determiner early, but fail on certain tasks for othe reasons. Indeed, the empirical evidence does not unambiguously support slow acquisition of the definite determiner. Karmiloff-Smith (1981, Experiment 12) tested French children's knowledge of the uniqueness presuppositions of "le" using a task that is intuitively much more difficult. Karmiloff-Smith showed French-speaking children scenes in which a boy had, for instance, a number of brushes and a girl had a single brush. Children heard an adult ask for either "le" brush, picking out a unique individual, or "un" brush, picking one from a set, and were asked whether the speaker was talking to the boy or the girl. Children as young as 3 years were 85% accurate at responding correctly. This means that French-speaking 3-year-olds must know "le" picks out a unique individual, and they are also capable of combining this with world knowledge to determine who someone else is speaking to. Karmiloff-Smith describes the key differences between these experiments as pertaining to usage: the experiment in which children succeed is *deictic* since world context establishes the referent, and the other is *anaphoric* since the linguistic context establishes the referent. In general, this indicates that current empirical results do not rule out early knowledge of these types of meanings, and that further empirical work is needed to make sense of the precise timecourse of children's acquisition.

[19] It is tempting to look at languages like French that have different forms for singular (*le*) and plural (*les*), but it is not clear how phonetically distinct these forms are for learners, and, as discussed, the acquisition path even in French is not entirely clear.



(a)

(b)

Figure 3-9: Illustration of the two types of spreading errors common in the acquisition of "every," *classical spreading* (a) and *bunny spreading* (b). In both cases, children incorrectly reject a sentence like "Every robot is wearing a hat," pointing to the unworn hat in (a) and the man with a cape but no hat (b).

3.5.3 Acquisition of "every"

As with "the," the word "every" has received special attention in developmental studies. Unlike "the," children (likely) have trouble learning its literal meaning, as first noted by Inhelder and Piaget (1969). In a situation with several robots wearing hats, children often would think that "Every robot is wearing a hat" would not apply if there was a hat not worn (Figure 3-9(a)). If asked why the sentence is not true, children explain that it is false because there is a hat not being worn (Philip, 1991, 1992, 1995)[20]. This error is known as a *spreading error* because "children extend the quantifier 'all' to the logical predicate of the sentence as well as to its logical subject" (Inhelder & Piaget, 1969, pg 70-71). This spreading error has been replicated cross-linguistically (Takahashi, 1991; Philip, 1995, 1998; Kang, 1999; Philip, 2003; Fiorin, 2010), and in large-scale developmental studies

[20] Children similarly reject "A boy is pulling every wagon" in a situation in which there is a boy not pulling a wagon, but otherwise all wagons are pulled by boys. Here, we will not deal with quantifiers in object position since this requires additional syntactic or semantic machinery.

(e.g., Seymour, Roeper, & De Villiers, 2003; Roeper et al., 2004). Following Roeper et al. (2004), we refer to this spreading error as *classical spreading*. In a second kind of spreading, *bunny spreading* (Roeper et al., 2004), children believe that the intersection of the two sets, A (the set of robots) and B (the set of hat-wearers) must exhaust the entire set observed in the context. So if children were shown a scene with several robots wearing hats, and a man wearing a cape, they will say that it is not true that "Every robot is wearing a hat" because of the man wearing a cape (Figure 3-9(b)). This latter form of spreading can be characterized using the primitive *total-intersection*, discussed above, as simply,

$$\lambda\, A\ B\ .\ (\textit{total-intersection}\ A\ B). \tag{3.24}$$

Note that there is not as simple a representation for classical spreading, since the set of hats is not naturally an argument to "every" (in the way that the set of hat-wearers is). We therefore will focus on bunny spreading here in our modeling work.

Roeper et al. (2004) charts out the developmental trajectory of spreading errors, finding that at age 6, about 40% of children show both errors and 20-30% show only classical spreading. By age 8-9, bunny spreading becomes even less common (about 20% of children), but still 40-50% of children show classical spreading. Indeed, spreading errors persist quite late in development. Even at 8-9 years old, about 30-40% of children show *no* spreading errors. By 12 years old, only 50-60% of children show no spreading errors. Spreading is therefore remarkable in both its consistency cross-linguistically and across studies, as well as its persistence throughout development.

Two primary theories have been offered to explain children's spreading errors. Philip (1995) argues that spreading results from children incorrectly quantifying over events rather than objects. In classical spreading, for instance, children might incorrectly think that "every robot is wearing a hat" would mean that for all events e in the context involving a robot or a hat, then the robot is wearing the hat. Philip observes that children's rough acquisition order from bunny spreading to classical spreading to adult representations follows what would be expected for learners that ordered hypotheses in terms of logical strength, as in the subset-principle learning framework discussed above. Crain et al. (1996) argue for an alternative—though still strongly nativist—account (see also Meroni, Gualmini, & Crain, 2000; Gualmini, Meroni, & Crain, 2003; Rakhlin, 2007). Crain et al. show that children can often provide adult-like interpretations of "every" when the context is more pragmatically felicitous—in particular, when the negation of the target sentence, e.g., "Every robot is wearing a hat," could potentially be false. Under this view, children have a full adult semantic representation for "every" but are pragmatically unable to interpret "every" in contexts used in previous experiments.

Both of these accounts have shortcomings. Most troublingly, neither account explains how children learn which words go with which semantic representations—that is, they are not situated in the context of a sufficiently competent and formalized learning model. Philip (1995)'s account, like other subset-principle accounts, does not explain how children transition between hypotheses in the face of noisy evidence. As far as we can tell, Crain et al.'s account cannot explain the fact that children still do make errors in these quantifier meanings. Their explanation additionally relies on an extremely subtle experimental manipulation. As pointed out by Roeper et al. (2004), another challenge for Crain et al.'s account is that recent work has found similar patterns in L2 acquisition (DellaCarpini, 2003), meaning that adults who presumably have competent pragmatics still show these errors. As Roeper writes, the errors likely result from the "challenge of grammar construction" facing both L1 and L2 learners. Both of these theories are somewhat post-hoc, explaining children's errors but not predicting them a priori from any independently motivated basis.

From the viewpoint of an idealized statistical learner, what is especially interesting about these spreading errors is that they do not appear to be simpler meanings than the correct meaning of "every." However, one possibility is that bunny spreading results from children's bias for expressions involving the entire set. As discussed above, children often have a bias to respond with the entire set (Sarnecka & Lee, 2009; Lee & Sarnecka, 2010b, 2010a), which motivated our inclusion of the *total-intersection* primitive function. The question remains, however, whether this meaning is indeed difficult to rule out as the meaning of "every." Figure 3-10 shows the errors made by the model in learning the literal meaning of "every," analogous to the plot for "the" above. Apart from $true$ and $false$, this meaning is indeed the most common error the model makes, indicating that typical evi-

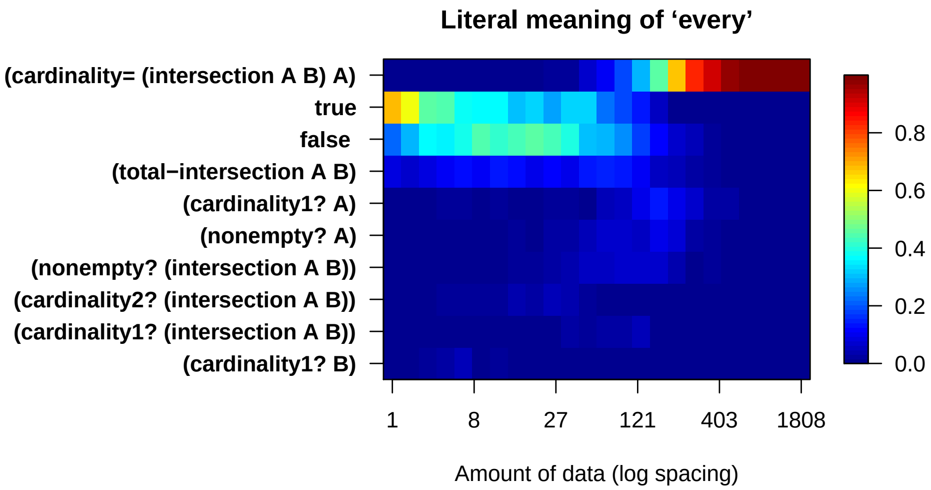


Figure 3-10: Posterior probability (z-axis) of the most-likely literal meanings of "every" (y-axis) over the course of acquisition (x-axis).

dence distinguishes the meaning of "every" poorly from this alternative. This error is still relatively uncommon for the model, mostly because the errors are dominated by $true$ and $false$. However, it is easy to imagine that such meanings may not really be considered by learners as plausible literal meanings since they are semantically vacuous[21].

This account of spreading errors differs from Philip's in that it is a fully implemented acquisition model, using data to search through hypotheses. The fact that this roughly shows the types of errors children make is informative since the hypothesis space is relatively unrestricted and one might imagine that other bizarre hypotheses could be considered by such a learner. It is also informative that an ideal learner with this simple representational system has more trouble ruling out $true$ and $false$ than *total-intersection*, potentially providing indirect evidence that such semantically vacuous meanings are dispreferred by young learners. These results are suggestive that refinements of the learning setup—in terms of data presented to the model and the assumed representational system—could potentially explain children's spreading errors from this type of independently motivated statistical model.

[21] However, *true* at least is potentially high in the prior for the *presupposition* of "the," as required above, since some words do not have presuppositions.

3.6 General Discussion

These results have revealed that some aspects of the learning model are very child-like: the model's mistakes are similar to those made by children, despite the fact that it is operating in a relatively unrestricted hypothesis space. In this sense, the model is to some degree able to *predict* these errors from domain-general, independent principles—the challenge of searching through hypotheses to explain observed utterances. This prediction is not perfect because we have chosen a very simple representational system, but its particular components have not yet been established through independent experimentation. Inclusion of the right representational components like the meaning *total-intersection* is important for modeling children's responses to developmental experiments. Further work to discover what types of set operations children find most natural may lead to a learning model that more accurately captures observed developmental patterns.

One feature of the learning model that differs from children is that it appears to be extremely *easy* for the learning model to discover these word meanings—typically within hundreds to a few thousand labeled sets. We have made several simplifications that lead to this. First, the model assumes the sets A and B are known. If learners are simultaneously learning nouns, then this would introduce more uncertainty, delaying acquisition. Second, we have assumed that the syntactic and semantic compositionality of quantifiers is known—that is, that they take two sets and assert something about their relationship. It is possible that if this also must be learned simultaneously, it would substantially complicate learning and slow acquisition. The model in its present form also has perfect memory of previous data. This assumption is not necessary for this type of LOT statistical model (see Chapter 3), but also leads to increased rates of acquisition. Finally, we have assumed relatively high rates for α_p and α_t. As the learner's settings of these parameters are brought to 0, the model "cares" less about explaining the observed utterances. In this sense, the learning *rate* of the model is essentially a free parameter, and is not strongly predicted by our account. However, it is informative that for high and we believe intuitively plausible settings of these parameters, the model learns substantially faster than children.

The ease of learning for this idealized model may raise one interesting possibility for language acquisition. It may be the case that the key puzzle for language acquisition is not the *poverty* of the stimulus, but the *abundance* of stimulus: why do some aspects of language acquisition take so long, given that an idealized statistical learner would find them so easy? The answer above is that children are non-ideal in all sorts of ways, including memory limitations, and imperfect observations. But it might be the case that even given these, facts, idealized learners find it easier than children; it is not intuitively clear to us if addressing these simplifications in the model would make learning "the" take the amount of data present in 10% of a person's life, rather than the amount of data present in a single conversation. Similarly, abstract syntactic principles may be learnable from surprisingly little data (Perfors et al., 2011). Indeed, the abundance of the stimulus was argued by Babyonyshev, Ganger, Pesetsky, and Wexler (2001) to support a maturational account of other syntactic phenomena, such as A-chain formation, since children are substantially delayed with A-chains despite their prevalence in the input. Addressing the abundance of stimulus problem is an interesting challenge for statistical learning models—one that is the polar opposite of traditional poverty arguments put forth against statistical learning. This paints a different picture of acquisition, one where the environment is full of information sources, and the hard part of language learning is using those information sources effectively—not that fact that the information is not present.

We take the most substantive contribution of this work to be the learnability arguments and simulations. We have contrasted our approach to finite-state, Gold-style learnability proofs for quantifier meanings, which either require positive and negative evidence, or only provably work for a subset of meanings. It is also unclear how these accounts would handle noisy data, or if they could show developmental patterns that are anything like children's. Unlike these approaches, our model was motivated by the types of representations common in semantic theories—i.e. functions on sets. We think of quantifier learning as a problem of composing primitive cognitive operations in order to form conceptual representations that explain observed utterances. This general approach can be viewed as a kind of *program induction*, where learners must create a simple expression in their "language of thought" that assigns observed data high probability. This has the advantage of explicitly formalizing what knowledge learners must bring to acquisition (representational primitives and rules of composition) and how this knowledge interacts with observed evidence (through approximately optimal statistical inference). Our model therefore provides an appealing compromise between nativist and empiricist theories in language development. The model is nativist in that it "builds in" a hypothesis space of potential meanings. This amount of nativism is, in some sense, a necessity for any learning model that can arrive at the correct set of meanings—even models which do not have explicit representations build in spaces of hypotheses (see Griffiths, Chater, Kemp, Perfors, & Tenenbaum, 2010). However, language-specific—or, more precisely, semantics-specific—constraints and learning procedures are *not* necessary for learning: an ideal learner in an unrestricted hypothesis space takes only marginally more data to arrive at the right answer, than a learner with even the most restricted set of potential meanings. The model is empiricist in that it learns in a relatively unrestricted hypothesis space, with the data determining the correct hypothesis in the limit. In this sense, we have directly tackled the question of what must be innate specifically for learning these meanings: the capacity for structure, statistical inference, and the capacity to apply it in a particular domain.

The conceptual foundations of this approach draw heavily on work by Chater and Vitányi (2007). They argued that in language acquisition, learners need not be innately constrained at all in order to learn language, contrasting with Gold's theorem. Chater and Vitányi (2007) demonstrate conditions under which any *computable* language is learnable from positive evidence only, by learners who potentially consider any computable language—as unrestricted a learner as possible. This proof works by assuming that learners attempt to find what are essentially short computer programs to describe the data they say. Chater and Vitányi (2007) apply Solomonoff (1978)'s *Prediction Theorem* to show that such learning can always identify the correct linguistic system given enough data, with relatively little error. The assumptions of this approach differ from Gold's in several key ways. Rather than a worst-case analysis, Chater and Vitányi (2007) present average-case learnability results. Second, their learning criterion differs from Gold's. They consider the task of the learner to be modeling the data, rather than correctly identifying the language in the limit. We take the results of Chater and Vitányi (2007) as providing a definitive theoretical argument that language is learnable from unrestricted hypothesis spaces. However, their work leaves many practicalities to be resolved. For one it is not immediately clear how to apply their results and methods to actual acquisition problems. This is in part because their results are stated using *Kolmogorov complexity*, a formal measure based on description length of, in this case, grammars. This measure is formally uncomputable, and—even if it were computable—seems an unlikely measure for children to compute. The alternative worked out here is that children "compute" using a set of conceptual primitives—in this case, functions that manipulate sets of objects. Their task in acquisition is to determine how to compose these cognitive primitives into the correct representations. The representations can be thought of as computer program that are written in the "language of thought." The standard learnability proof we present shows that in language acquisition such representations can always be recovered from positive evidence, and our implementation shows that the amount of data required is surprisingly small. This type of acquisition from a composition space of semantic primitives can "really work" in explaining language learning. Indeed, the general class of model presented here could be generalized to learning other types of semantic and even syntactic operations, especially those formalized using a lexicalized grammar (e.g., Steedman, 2000); for a simple example of this, see Piantadosi et al. (2008).

3.7 Conclusion

This paper has studied learning problems in semantics as a case study of the logical problem of language acquisition. We have argued that learners of quantifier meanings face many of the complexities that make learning language daunting: non-obvious literal meanings, the subset problem, presuppositional content, and variable word frequencies. The learning model we present shows how fairly rich representations of quantifier meanings could be learned from positive evidence alone, using a developmentally plausible amount of data. These general techniques could be applied to other subset-problems in language, or areas where unseen abstract structure must be inferred. Like most Bayesian models, the one we present is provably learnable. The implementation has also allowed us to test the utility of constrained hypothesis spaces for quantifier learning, and provides a potentially statistical explanation for some of the most-studied effects in quantifier acquisition.

We have argued against several approaches to quantifier learning, namely those which use ad-hoc representations such as finite-state machines, or require innate ordering on hypotheses to solve the subset problem. The model presented here is only one exemplar of a family of theories that explain learning as composition from innate or previously learned primitive functions. Examining the details of this type of model's acquisition patterns will likely require both more sophisticated data sets of naturalistic adult productions (in semantic contexts) and independent experimentation to establish plausible cognitive primitives. Function word acquisition provides a rich test case for understanding how learners come to abstract knowledge in domains like language, and how existing representational theories in linguistics can be combined with sophisticated statistical inference techniques to produce empirically and theoretically tenable learning theories.

Chapter 4

Concept learning and the language of thought[1]

4.1 Introduction

One of the basic puzzles of human cognition is the extraordinary richness of our conceptual systems. We are not restricted to simple similarity-based generalization or rote memorization, but easily create and manipulate abstract, compositional, and structured representations—concepts like *prime number*, *half-sister*, *the tallest building in Cambridge*, or the semantic representations of function words like "every." Such concepts are interesting in part because they appear to be most easily characterized as logical rules. For instance, *A* is the tallest building in Cambridge if for all other buildings *B*, *A* is taller than *B*; two girls are half-sisters if they share one but not both parents; "every" is often formalized in semantics as a logical relation (subset) between two sets (Montague, 1973; Barwise & Cooper, 1981; Keenan & Stavi, 1986; Keenan & Westerståhl, 1997).

The existence of such concepts presents two challenges to cognitive science. The first challenge is understanding how children learn rule-like representations. How might children take observed instances of concepts like *tallest* or *half-sister* and determine the correct rules? Children are likely not born knowing these concepts and so they must be constructed from the knowledge children do bring to acquisition; yet, learning in logical systems is

[1] This work is joint with Noah D. Goodman and Joshua B. Tenenbaum.

nontrivial because the space of possible concepts is large and potentially Turing-complete. Recently, computational work has explored learning concepts in logical domains (Siskind, 1996; Feldman, 2000; Zettlemoyer & Collins, 2005; Goodman et al., 2008; Katz et al., 2008; Kemp et al., 2008a; Piantadosi et al., 2008; Goodman et al., 2009; Kemp, 2009; Kwiatkowski et al., 2009; Ullman et al., 2010; Piantadosi et al., submitted). Most of this work formalizes learning as inductive inference over a compositional representation system, a *language of thought* (LOT) (Fodor, 1975; Boole, 1854), and such systems have been argued to explain the mind's productivity, systematicity, and compositionality (Fodor, 1975; Fodor & Pylyshyn, 1988; Fodor, 2008). Our goal here is to extend LOT learning models to richer languages capable of expressing the types of quantification and logical operations that we argue will be necessary for capturing human learning patterns.

The second challenge posed by these types of logical concepts is that of discovering the right representational system for these concepts—what logical primitives and laws of combination do people use to construct these types of representations? Is our representation of *tallest* built out of concepts like *taller than* and quantification? Or is *tallest* a conceptual primitive, perhaps accessible even to the youngest learners? A coarse characterization of the right system for these concepts can be made in terms of computational power: representations must be capable of supporting the knowledge people have and the computations they perform (Marr, 1982). For instance, people's representational systems must extend beyond simple Boolean propositional logic since such systems that lack quantification provably cannot express concepts like *tallest*. However, descriptions based only on computational power are always under-determined. Two representations can be equally expressive—capable of solving the same computational problems—yet distinct in how they achieve that computational power (see, e.g., Hackl, 2009; Pietroski et al., 2009, for examples in semantics). A full understanding of human conceptual systems must therefore aim to characterize the precise components of rule-like concepts.

Here we present a formal learning model for learning compositional, rule-based concepts that aims to address these challenges. We model learning of compositional rules as Bayesian inference over a hypothesis space of concepts that are generated by a probabilistic grammar. The grammar can generate concepts of any expressivity and computational power, allowing the model to, in principle, learn concepts that require arbitrarily complex computations. This work therefore pushes studies of rule-based learning beyond the domain of simple Boolean concepts (e.g., Feldman, 2000; Goodman et al., 2008). The use of a full probabilistic model allows us to capture detailed learning curves and patterns of generalization participants exhibit while learning novel concepts. This shows how these two ideas—a grammar for generating expressions that represent arbitrary computations and a statistical learning model over thse representations—can be combined in a way that captures some of the richness of human inductive processes over computationally complex representations.

One key idea behind our approach is that representational *simplicity* is a major determinant of learnability (Neisser & Weene, 1962; Haygood, 1963; Feldman, 2000, 2003c, 2003b; Chater & Vitányi, 2003; Goodman et al., 2008; Kemp et al., 2008a): people prefer to make representationally simple generalizations from data. In learning, a bias for simplicity allows the model to narrow down the vast space of possible rules. Intuitively, simplicity does provide a compelling way to decide between possible generalizations: in machine learning and statistics, for instance, simplicity plays a key role in model selection because simple models are more parsimonious (e.g., Conklin & Witten, 1994), explaining the data with fewer free parameters or arbitrary stipulations. As we show, when simplicity is measured in the right way, the model generalizes much like human participants. Second, the existence of a bias for representational simplicity allows us to reverse engineer likely components of people's representations. Different hypothesized LOTs will likely have different measures of simplicity even if they have equivalent computational power, meaning that we can compare representational systems to see which ones measure simplicity most like humans.

A key example for our purposes is Feldman (2000), who showed that people's difficulty with learning Boolean concepts is well-modeled by the concept's description length in logic. For example, subjects would find it harder to learn the concept[2],

[red and [not square]] or [[not red] and square] (4.1)

[2] We will use brackets to group/disambiguate English descriptions of concepts.

compared to

$$\textit{red or square} \tag{4.2}$$

since the former has a longer description in standard Boolean logic. This suggests that subjects who learn such concepts actively compose logical operators (*and*, *or*, *not*) to express concepts, and that they have a bias to prefer concise representations in this system. Goodman et al. (2008) extend this idea by presenting a probabilistic model over Boolean logical expressions. They show that many experimental results could be captured by modeling learning as idealized statistical inference over a Boolean hypothesis space, assuming a bias for simpler expressions.

However, as is often pointed out in philosophical discussions of induction, what counts as "simple" is not at all straightforward (Goodman, 1955). For instance if people's representational system included the exclusive-or function (*XOR*) as a primitive, then the complexity—and therefore learning biases—for the above two concepts would be equal. Concept (4.2) could be expressed the same way, but (4.1) becomes

$$\textit{red XOR square} \tag{4.3}$$

which is as simple as (4.2) in terms of total number of logical operations. Importantly, adding *XOR* to the representational system does not change its computational power[3]. This demonstrates that (4.1) is not more complex than (4.2) in any independent, objective sense; indeed, it is difficult to avoid the problem that what counts as simple is to some extent arbitrary[4]. Here, we turn this philosophical puzzle into an experimental *tool*: if subjects actually do find (4.2) easier than (4.1), that provides evidence that their representational system measures simplicity according to a logical language that lacks the logical connective *XOR*.

The outline of this paper is as follows: first, we present a massive concept learning ex-

[3] Since in general *A XOR B* can be expressed as *[A and [not B]] or [[not A] and B]*.

[4] Although Kolmogorov complexity (see Li & Vitányi, 2008) is a good attempt. As Feldman (2003a) points out, Boolean description length is independent of the representational system up to a multiplicative constant, much as Kolmogorov complexity is independent of the representation system up to an additive constant. Unfortunately, even within these constraints, there is still a huge range of possible ways in which people might measure simplicity.

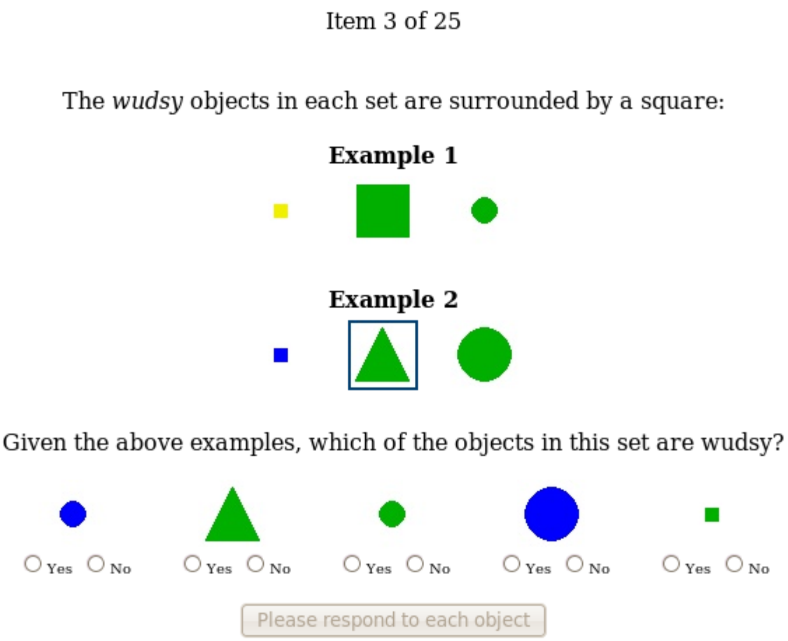
periment that taught subjects novel rule-based concepts, ranging from those studied previously in Boolean concept learning experiments (Bruner, Goodnow, & Austin, 1956; Shepard et al., 1961; Feldman, 2000; Goodman et al., 2008) to those involving richer types of quantification (e.g Kemp, 2009). We then describe how we formalize the LOT in terms of lambda calculus, and we develop two kinds of models. The first is a learning model that—like participants in the study—takes observed data and infers likely LOT expressions. As we show, the learning model is capable of inferring quite complex concepts from data, and the generalizations it makes closely track those of participants in the experiment. Second, we develop a data analysis model that uses participants' experimental data to infer unknown parameters of the learning model—for instance, the probability of different primitives or participants' memory-decay parameters. These models allow us to quantitatively evaluate different LOTs to see which ones best explain participants' learning curves. We first apply these methods to only Boolean concepts in the experiment, and then to concepts involving quantification.

4.2 Experimental paradigm

Simple Boolean concepts can be understood as mapping a given set of objects to a subset. For instance, you might be handed a set of balloons and be asked to give back the *red or green* ones. This can easily be generalized to concepts involving quantification: for instance, you might be asked to hand back all balloons such that *every other balloon in the set is not the same color* or every balloon such that *there exists another balloon in the set of the same shape*. Note that for quantificational concepts to make sense, they must quantify over some domain of objects—here, either all objects in the set or all *other* objects in the set. The choice of concepts that map sets to subsets is not exactly what is necessary for natural language quantifiers, since quantifiers are often relations on sets—they map two sets to a truth value. We chose concepts mapping sets to subsets because this most naturally generalizes previous Boolean concept learning experiments, and the representational machinery needed for these concepts can naturally be extended to generalized quantifiers. In deciding on the space of concepts, one might attempt to enumerate all possible concepts up to a given length, as in Feldman (2003a) and Kemp (2009). Unfortunately the number of such concepts quickly becomes intractable. We therefore chose to construct a space by hand consisting of 108 different concepts chosen to span an interesting and wide range of possible concepts involving quantification and relational terms. To do this, we considered different kinds of basic Boolean concepts (e.g., *blue objects*) and incoporated quantificational and relational terms (e.g., *the unique blue object*, *same shape as a blue object*, *every other object with the same shape is blue*, etc.). We additionally took these quantificational and relational concepts and added Boolean terms (e.g., *[same shape as a blue object] and circle*). The full set of concepts is listed in Figures 4-2 to 4-4.

In the experiment, subjects were told that they had to discover the meaning of "wudsy," a word in an alien language. They were explicitly told that this word applied to some objects in a set, and that whether or not an object was wudsy might depend on what other objects were present in the set. The learning paradigm was then sequential: subjects were shown a set and asked to guess at which items were "wudsy." After responding, they were shown the right answers. The correctly labeled sets stayed on the screen, and subjects moved on to the next set. So, on set N, a subject could still see the correct answers to the previous $N-1$ sets. Thus, the Nth subject's response represents their inferences conditioned on the previous $N-1$ labeled data points. This continuous measure of generalization contrasts with previous Boolean concept learning paradigms which have typically tested only after a fixed amount of training. Our paradigm allows a substantial amount of inductive generalizations to be gathered, providing a detailed picture of human learning curves.

An example experimental item is shown in Figure 4-1, showing subjects being asked to generalize to a set containing five elements after seeing the two preceding sets, only one of which contained a positive instance of the concept. To aid in motivation, subjects were required to wait 5 seconds when they made a mistake in any element of a set. The space of objects included squares, circles, and triangles, that were either green, blue or yellow. Object sizes ranged through 3 logarithmically spaced sizes, labeled size 1 (smallest), 2, and 3 (largest). Sets were generated from this space of objects at random, by first uniformly choosing a set size between 1 and 5, and then randomly sampling objects without replacement. Random generation was used to ensure that subjects do not assume sets were chosen



Item 3 of 25

The *wudsy* objects in each set are surrounded by a square:

Example 1

Example 2

Given the above examples, which of the objects in this set are wudsy?

Figure 4-1: An example item from the concept learning experiment. Here, the subject has seen two example sets of objects, and is asked to generalize to a new set. A likely response here would be to answer in accordance with the simple concept *triangles*.

to be informative about target concepts (as in, e.g., Shafto & Goodman, 2008). Subjects were shown 25 sets of objects in total.

Subjects were randomly assigned to a concept and one of two lists in that concept, where each list was a different sequence labeled according to the target concept. Subjects were allowed to do multiple concepts, but could not repeat the same concept twice. The small number of lists allowed us to run more subjects within each list to get higher confidence in the exact learning curves for any particular sequence of labeled data. The specific shapes and colors in each target concept were randomized across subjects. For example, *red and circle* was randomized to *blue and triangle*, *blue and square*, *green and circle*, etc. across subjects. Subjects were run online using Amazon's Mechanical Turk. Subjects who fell more than 2 standard deviations below the mean accuracy in their concept were removed. Data from subjects who completed fewer than 5 sets of a given concept was also removed, but otherwise partial data from subjects was included in this analysis.

At sets 5, 10, and 25, subjects were asked to describe what they thought "wudsy" meant. In general, verbal descriptions proved extremely difficult to analyze because subjects often wrote ambiguous descriptions. For instance, we ran concepts such as *the unique tallest* (cannot be tied for tallest shape in the set) and *one of the tallest* (can be tied for tallest shape in the set). Subjects with both of these concepts wrote “tallest,” which, in English, might mean either concept[5].

4.2.1 Results

A total of 1596 subjects were run on 108 concepts. Individual subjects completed an average of 4.24 concepts (median 2), with the maximum number of concepts run by a subject at 80. Overall accuracy in the experiment was 78% with a chance rate of 56%, though the accuracies varied substantially by concept. Mean accuracies on concepts were highly consistent across the two lists, with a correlation of $R^2 = 0.81$. Subjects were run until each concept and list was completed by approximately 20 subjects.

Figures 4-2 to 4-4 list the 108 concepts tested here, as well as raw subject performance on these concepts. Each horizontal line in these figures shows a compressed learning curve with two points, representing mean subject performance on the first and last quarter of the experiment. Each figure shows one third of the total concepts tested, sorted by overall mean accuracy: Figure 4-2 shows the most easily learned concepts, Figure 4-3 shows concepts that are likely learnable with some difficulty, and Figure 4-4 shows concepts that are extremely difficult to learn, some of which may not have been learned by anyone over the course of the experiment. These plots also include blue bars corresponding to chance performance. Chance was computed by assuming that subjects guess with the correct base rate: thus, if the concept is true of set elements 30% of the time, then subjects matching the base rate would be correct with probability $0.3 \cdot 0.3 + (1 - 0.3) \cdot (1 - 0.3)$. The horizontal lines in these plots are green for simple Boolean concepts and black for concepts that have no equivalent in Boolean logic.

This graph demonstrates several basic patterns previously found in Boolean concept learning. For instance, complex concepts (*circle and blue*) are learned less quickly than

[5] Indeed, the ambiguity in subject descriptions provides some evidence that subjects did not represent target concepts in natural language—doing so often leaves the target concept underspecified. On the other hand, it is possible that the descriptions subjects provided were incomplete characterizations of unambiguous linguistic representations.

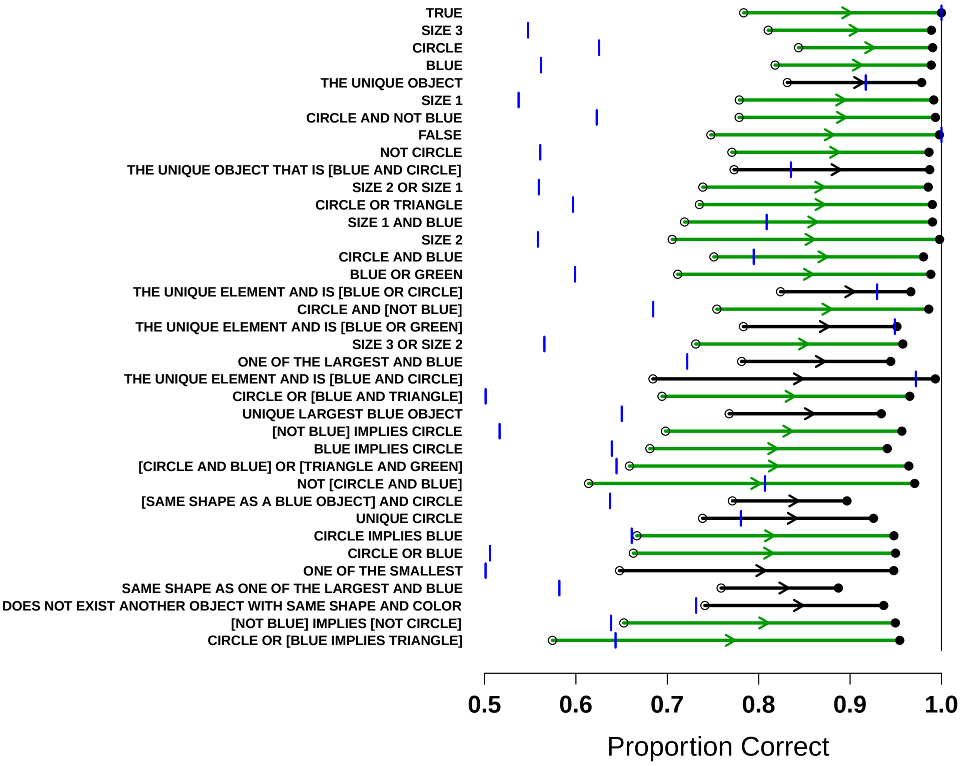


Figure 4-2: Proportion correct on the first 25% of the experiment (open circle) and last 25% (closed circles) for the top third of concepts most easily learned. Green lines denote concepts that can be written in simple Boolean (propositional) logic. Blue bars denote chance guessing at the correct base rate.

simple ones (*circle*) (Feldman, 2000). The graph also shows that conjunctions (*circle and blue*) are easier than disjunctions (*circle or blue*). The *and/or* asymmetry is one of the oldest findings in rule-based concept learning (Bourne, 1966; Shepard et al., 1961) and has been replicated across cultures and levels of education (Ciborowski & Cole, 1972). These results also suggest selective attention effects where multiple references to the same feature dimension (*blue or green*) are easier than references across dimensions (*circle or blue*). Figure 4-2 also suggests that many concepts that are not Boolean but can nonetheless be easily learned. For instance, *the unique element and is [blue or green]*, *one of the smallest*, *exists another object with the same shape and color*.

However, direct comparisons on this data are not straightforward. For one, the concepts vary in their base rate accuracy (blue points) and so it is difficult to know if differences in

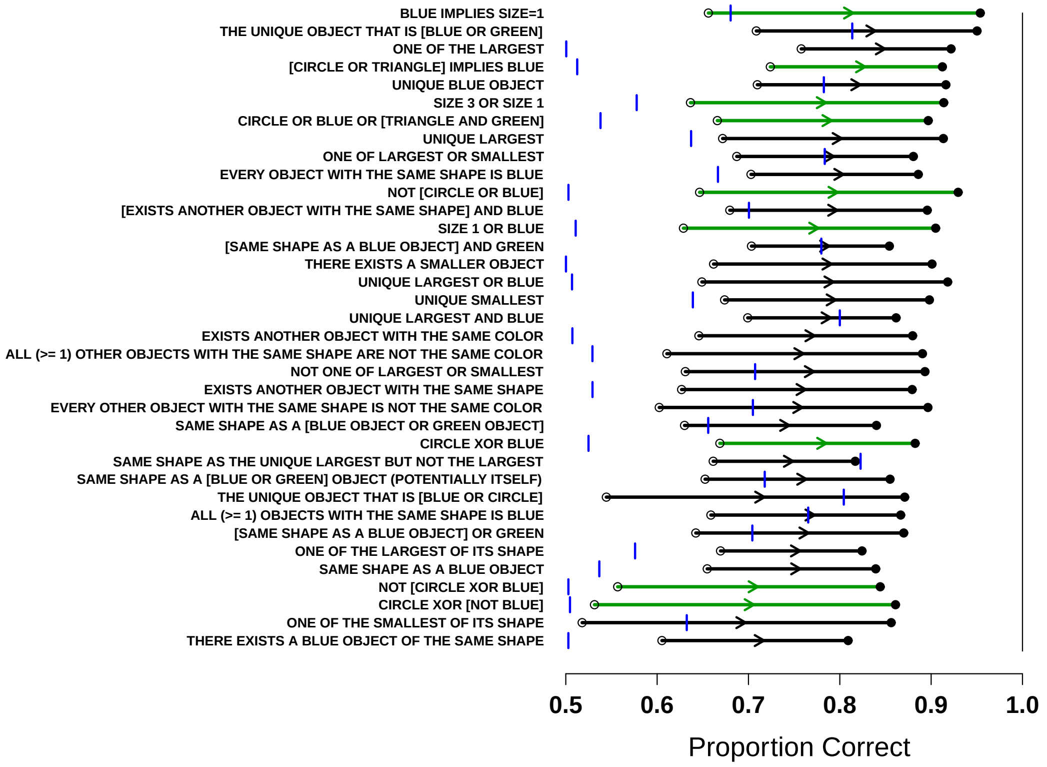


Figure 4-3: Proportion correct on the first 25% of the experiment (open circle) and last 25% (closed circles) for the second third of concepts most easily learned. Green lines denote concepts that can be written in simple Boolean (propositional) logic. Blue bars denote chance guessing at the correct base rate.

accuracy result from difference in chance performance. Worse, though, is that subjects can achieve high accuracy on concepts not by learning the correct concept, but by learning a closely related one. Subjects may, for instance learn *one of the tallest* (the object can be tied for tallest) for *the unique tallest* (the object cannot be tied for tallest). A third problem is that it is not clear how informative learning rates are for comparisons since the observed data may be differentially informative as to the target concept. For instance, an ideal learner who is equibiased between *circle and blue* and *circle or blue* may nonetheless find the former easier to learn because it is true less often, meaning that the positive examples may be more diagnostic for the target concept or maybe more psychologically available. In general, then, these types of learning curves are not directly informative as to the prior biases of learners. To address these issues, we develop a model in the next section that

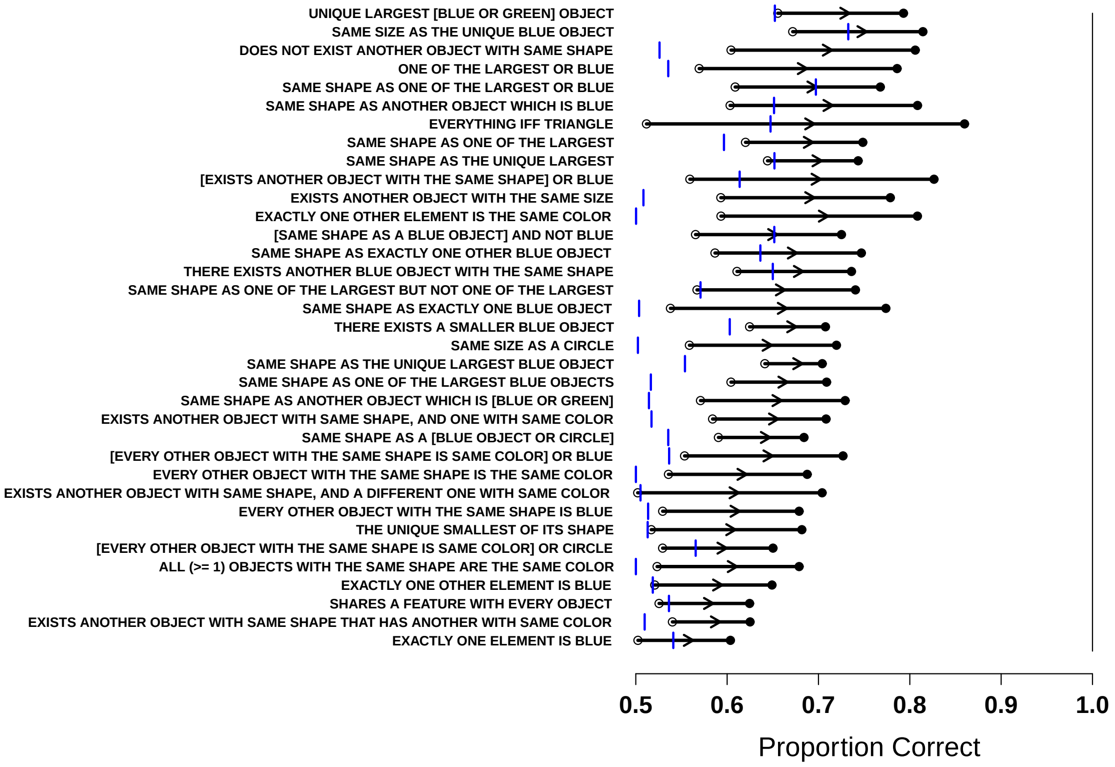
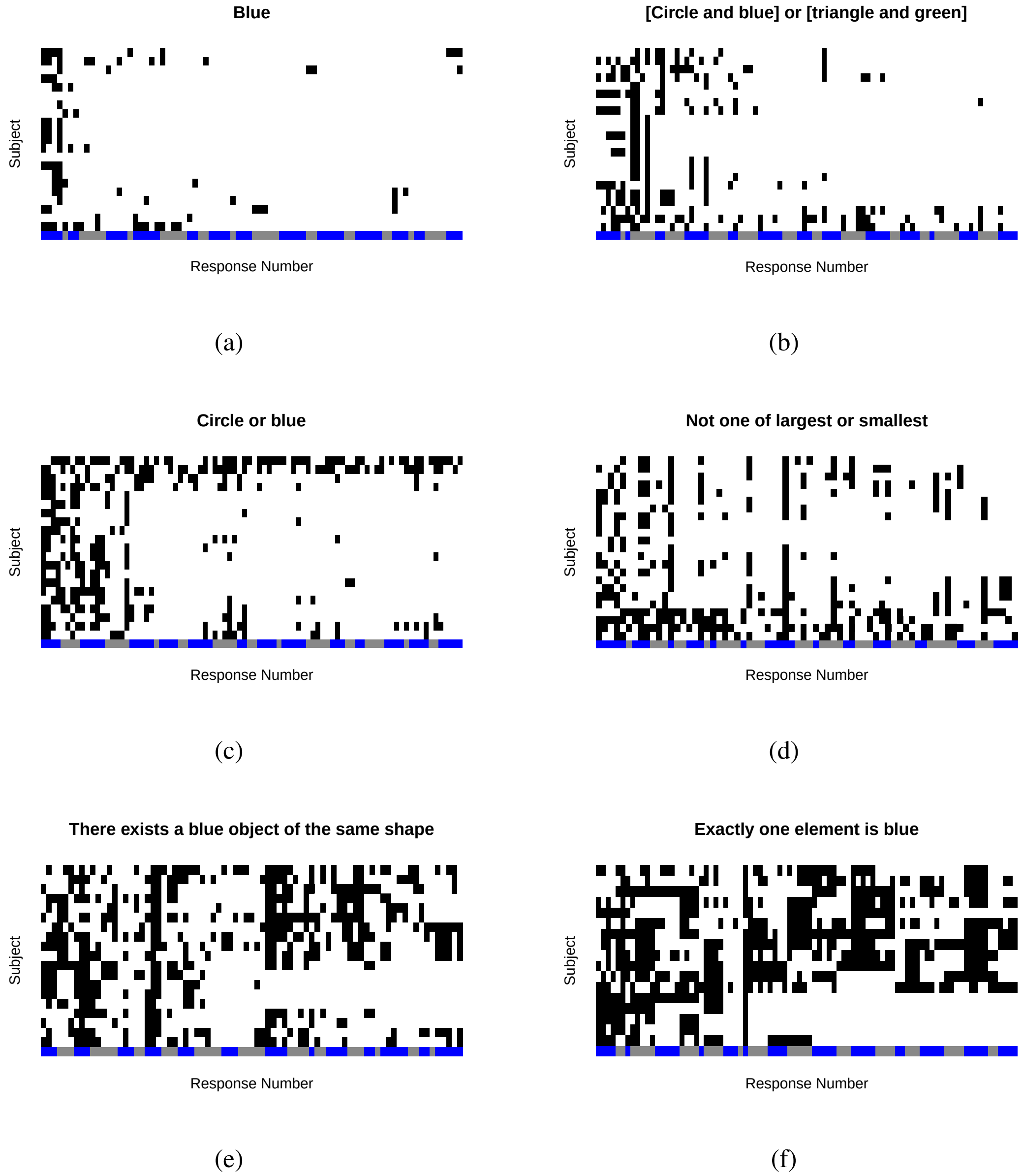


Figure 4-4: Proportion correct on the first 25% of the experiment (open circle) and last 25% (closed circles) for the third of concepts hardest to learn, none of which are simple Boolean expressions. Blue bars denote chance guessing at the correct base rate.

attempts to capture the details of the learning curves, rather than such coarse patterns.

Our modeling is in part inspired by the richness of individual subject response patterns. Figure 4-5 shows responses to 6 examples concepts: Figure 4-5(a) and 4-5(b) from the upper most accurate third of concepts, 4-5(c) and 4-5(d) from the middle third, and 4-5(e) and 4-5(f) from the lower third. Each sub-figure shows a subject on a row, and their response to each object in each set over the course of the experiment. Black squares in this plot represent incorrect responses, and white represent correct responses. Columns on the left of these plots correspond to early responses in the experiment, columns on the right correspond to later ones, when correct answers for all previous (leftward) examples have been observed. So for example, the top two rows of Figure 4-5(c) shows two subjects who did not learn the target concept and made mistakes throughout the entire experiment. The rows (subjects) in these plots are sorted by clustering to reveal subjects whose patterns


Blue

[Circle and blue] or [triangle and green]

Circle or blue

Not one of largest or smallest

There exists a blue object of the same shape

Exactly one element is blue

Figure 4-5: This figure shows subjects on each row, and elements of each set in columns, throughout the course of the experiment (left to right). The key at the bottom shows which elements are grouped together in each set. This shows systematic patterns of mistakes during learning, and often all-or-none acquisition by individual subjects.

group together. The blue and gray bar is a key that shows which individual responses were responses to objects in the same set: adjecent columns with the same color in the key were correspond to objects presented in the same set.

This plot demonstrates several interesting patterns. Even in concepts that subjects readily learn (e.g., 4-5(a)), they still make occasional errors. These errors, however, appear not to be systematic across subjects or sets. In other situations, such as Figure 4-5(d), subjects make highly systematic patterns of mistakes, often incorrectly labeling the same elements of the same sets. There are three subjects in the middle of this plot, however, who appear to correctly get the target early concept and answer perfectly for most of the experiment. This pattern is also found in harder concepts, 4-5(e) and 4-5(f), where only a few subjects achieve high accuracy and the rest appear to make similar patterns of responses. Note that even though mean accuracy is low on these concepts, these figures demonstrate that one would not want to classify these concepts as impossible for participants to learn in this setting since some actually do learn them.

These plots also demonstrate that while the average subject may show graded performance, individual subjects likely have very rule-like hypotheses in mind (Nosofsky, Palmeri, & McKinley, 1994): at some point, subjects appear to "get it" and respond perfectly or nearly perfectly for the remainder of the experiment. Subject averages, though, show more gradual patterns of learning since subjects often "get it" in slightly different places. This means that while it may be reasonable to model averaged learning data, it is important to recognize that a model of average learning does not necessarily represent how individuals act. It would be revealing to try to understand what hypotheses subjects have in mind at each point in time (e.g., M. Levine, 1966), yet also difficult since any set of responses they make at one time is consistent with a large number of hypotheses.

Our experimental results generally suggest that there is a wide range of potential concepts that people are able to learn, and in learning these concepts they show intricate, clustered, patterns of responses. A question is then what type of cognitive system could ever accommodate such general capacities for representation and learning? Indeed, for these models, it appears difficult to apply standard exemplar (e.g., Medin & Schaffer, 1978; Nosofsky, 1984) or prototype (e.g., Rosch & Mervis, 1975) models to this data. There is no exemplar or prototype of *objects that are the same color as a triangle*, because this concept is highly context-dependent. It cannot be captured by simple properties of objects, or even by most obvious properties of sets. The use of these concepts is by design because many concepts in language have this character or not being characterizable in terms of prototype or exemplar theory—it is hard to imagine a sense in which there could be a prototype representation of “most” or “of.”

The rest of this paper presents a learning framework that makes predictions about learning curves, conditioned on labeled data and allows us to compare different representation systems and models on this learning data. Our learning framework explicitly represents hypotheses at each iteration through the 25 labeled sets, and updates them according to an ideal Bayesian analysis. We primarily vary the priors used by this model, corresponding to different LOTs for expressing these types of concepts. We provide validation for our method by showing that intuitively implausible bases for thought (such as the NAND-basis for Boolean logic) can be excluded based on quantitative data analysis. In addition, we vary the form of the model used in order to determine what probabilistic frameworks best capture learning of Boolean and more difficult concepts. After all, it is possible that people’s inferences in our experiment actually *are* best explained by an exemplar or prototype theory—perhaps their generalizations over the course of learning use some similarity metric between sets or objects, and perhaps they never actually acquire the rich set of structured concepts we had in mind. We begin by describing how we formalize different representational systems for learners.

4.3 Languages of thought

Desiderata for a theory of the concepts appear daunting (see Prinz, 2004): (i) the representational system should support the type of *learning* that occurs in this experiment and more generally in cognitive development; (ii) the representation system should allow suitable *expressivity*, including the concepts subjects learn in this experiment and those necessary for natural language; (iii) the representation system should be able to explain the *systematicity* of generalization across subjects; (iv) the representation system should ideally *interface with linguistic systems*. These desiderata appear to be difficult to satisfy simultaneously—more expressive systems are likely more difficult to do learning in, for instance.

We argue that these desiderata can all be satisfied by rule-like representations, which we express in *lambda calculus*. In general, lambda calculus allows for flexability in theorizing that is not possible in other systems such as propositional, first-order, or second-order logic. Lambda calculus allows for any amount of computational power, ranging from the expressiveness of these logical systems, up to Turing-completeness. In previous work, we (Piantadosi et al., submitted, 2008; Piantadosi, Tenenbaum, & Goodman, in prep) and others (Zettlemoyer & Collins, 2005; Liang, Jordan, & Klein, 2010) have argued for the developmental and computational plausibility of learning lambda calculus expressions. In particular, previous results show that learning in lambda calculus is computationally tractable, both in terms of the computational resources and amount of data necessary (i). Moreover, lambda calculus allows for arbitrary expressivity: the choice of cognitive primitives allows us to specify representational systems with any degree of power, ranging from predicate logic, to Turing-complete systems (ii). In simple cases—some of which are presented here—it reduces to formalisms previously posited in Boolean concept learning, including propositional logic. The discrete, rule-like nature of lambda calculus means that in an appropriate learning model, subjects may come to correct and incorrect rules based on their observed data, potentially explaining systematicity in generalization and mistakes (iii). Finally, lambda calculus is often used in compositional semantics (e.g., Heim & Kratzer, 1998; Steedman, 2000) because it provides a convenient way for capturing the systematic compositional patterns of natural language. This means that our system automatically interfaces with contemporary theories of linguistic meaning (iv).

4.3.1 Lambda calculus

Lambda calculus can be viewed as a formalism for expressing hypotheses about how a compositional LOT may work. As such, it provides a framework-level theory, the particular instances (grammars) of which can be evaluated empirically. Lambda calculus was developed by Church (1936) as part of his work investigating computability and foundations of mathematics. It is especially convenient here because it provide a simple and uniform syntax for composing hypothesized elementary cognitive operations into more complex concepts. An example *lambda expression* is

$$\lambda\, x\,.\,(\mathit{and}\ (\mathit{red?}\ x)\ (\mathit{circle?}\ x)). \tag{4.4}$$

Each lambda expression has two parts and represents a *function*. To the left of the period is "λ x". This signifies that the expression represents a function of the variable named x. Here, x is an object in one of the presented sets[6]. To the right of the period is an expression representing what to compute. For convenience, we write functions in *prefix notation*, with the function the first symbol in the parentheses, followed by its arguments. In this case, we first compute (*red? x*), a predicate that checks if x is colored red[7]. Then, we compute (*circle? x*), which checks if x is a circle. We combine the two values (*red? x*) and (*circle? x*) using logical conjunction, *and*, yielding a function that returns true iff x is a *red circle*. An equivalent expression in propositional logic might be written as $red(x) \wedge circle(x)$. However, the reason not to use propositional logic is that lambda calculus provides a much richer and more powerful formalism. Lambda expressions can manipulate other lambda expressions and define *higher-order functions* (functions that manipulate functions). One example is

$$\lambda\, x\, S\ .\ (or\ (blue?\ x)\ (exists\ (\lambda\, x_2\ .\ (and\ (red?\ x_2)\ (equal\text{-}shape?\ x\, x_2)))\ S)) \tag{4.5}$$

Here, we have λ x S, representing that this function takes *two* arguments, in this case an object denoted x and a set denoted S. This function checks if x is blue, via (*blue? x*). It also computes (*exists* (λ x_2 . (*and* (*red?* x_2) (*equal-shape? x* x_2))) *S*). Here, *exists* is a function which takes two arguments, another predicate and a set. It implements existential quantification, returning true if the predicate is true of any element of the set. In this case, the predicate is (λ x_2 . (*and* (*red?* x_2) (*equal-shape? x* x_2))), which is true when x_2 is red and x_2 is the same shape (*equal-shape?*) as x. Note that this predicate depends on what x is, and so is actually a different function depending on what object has been passed in as x. Overall, then, (4.5) returns true if *x is blue or there is a red object of the same shape*.

This compositional representation system allows a relatively small number of primitives to be used in a combinatorially large number of ways. Once a set of primitive operations is defined, they inductively define a vast space of potential concepts. This implicitly provides an explanation for where part of the richness of cognition comes from: we are able

[6] We are therefore working with *typed* lambda calculus, in which every variable has a type.

[7] Predicates are given a "?" to indicate that they return Boolean values.

SIMPLE-BOOLEAN		
START	→	λ *x . BOOL*
BOOL	→	(*and BOOL BOOL*)
		(*or BOOL BOOL*)
		(*not BOOL*)
BOOL	→	(*F OBJECT*)
OBJECT	→	*x*
F	→	COLOR
		SHAPE
		SIZE
COLOR	→	*blue?*
		green?
		yellow?
SHAPE	→	*circle?*
		rectangle?
		triangle?
SIZE	→	*size1?*
		size2?
		size3?

(a)

NAND		
START	→	λ *x . BOOL*
BOOL	→	(*nand BOOL BOOL*)
		true
		false
BOOL	→	(*F OBJECT*)
OBJECT	→	*x*
F	→	COLOR
		SHAPE
		SIZE
COLOR	→	*blue?*
		green?
		yellow?
SHAPE	→	*circle?*
		rectangle?
		triangle?
SIZE	→	*size1?*
		size2?
		size3?

(b)

Figure 4-6: Two bases for Boolean logic: (a) writes expressions using the standard logical connectives (and, or, not), while (b) uses only one connective (not-and). Both are universal, in that all propositional formulas can be written using either set of primitives.

to recombine our perhaps small set of cognitive operations in new ways, allowing for systematicity, productivity, and compression in conceptual space. We define a *representation language* as the set of all structures which can be built in lambda calculus, assuming some set of primitives. The structures built must respect the types of arguments each primitive requires: *exists*, for instance, could not be given two Boolean arguments. To formalize this, we use a context-free grammar.

4.3.2 Grammars for lambda calculus

An example grammar is shown in Figure 4-6(a). Each row in this table represents an expansion rule: the left hand side is a *type* and the right hand side is an expression that that type expand to. Thus, for instance, we could create the expression λ *x* . (*or* (*green? x*) (*blue? x*)) by first expanding the *START* symbol with *START* → λ *x* . *BOOL*. We then expand the *BOOL* in the right hand size of λ *x* . *BOOL* to (*or BOOL BOOL*), yielding the intermediate expression λ *x*. (*or BOOL BOOL*). Then, each of these *BOOL*s are expanded to (*F x*) yielding λ *x*. (*or* (*F OBJECT*) (*F OBJECT*)) . Finally, the first *F* is expanded to *COLOR*, then *green?* and the second *F* is expanded to *COLOR* and then *blue?*. Both *OBJECT*s are expanded to *x*es, yielding the full expression.

In general, we expand expressions until they contain no more nonterminals, the uppercase symbols on the left side of this grammar. These grammars are meant to capture a core generative capacity of learners—that learners can in principle construct a huge number of potential concepts. Note that the majority of rules in the grammar are actually methods of accessing perceptual primitives. The core logical or computation parts of the grammars are relatively small. In this case, there are just three logical connectives for Boolean expressions. In practice, these grammars more accurately model people's learning curves if we additionally include rules $START \rightarrow true$ and $START \rightarrow false$, corresponding to trivial concepts that are always true or false. These rules are included in all grammars but, for conciseness, not shown.

There are many other ways to write down expressions in Boolean logic, corresponding to different languages of thought. Figure 4-6(b) shows one other: the NAND grammar uses only a single logical connective, *NAND*, yet can provably express all concepts the SIMPLEBOOLEAN grammar 4-6(a) can[8]. These two grammars provide distinct representational hypotheses of equivalent computational power, but distinct computational processes. Importantly, 4-6(a) and 4-6(b) measure simplicity in different ways. A concept like *red and circle* might be written using the SIMPLEBOOLEAN grammar as

$$\lambda\, x\, .\, (\mathit{and}\ (\mathit{red?}\ x)\ (\mathit{circle?}\ x)). \tag{4.6}$$

Using the NAND basis, this would have to be written as

$$\lambda\, x\, .\, (\mathit{nand}\ \mathit{true}\ (\mathit{nand}\ (\mathit{red?}\ x)\ (\mathit{blue?}\ x))). \tag{4.7}$$

Expressing this concept in the NAND basis requires two logical connectives, and an expansion to *true*; expressing it in SIMPLEBOOLEAN requires only one logical connective. In contrast, the concept *not [red and circle]* requires only one logical primitive in NAND and

[8] We include *true* and *false* as expansions of *BOOL* only for the NAND grammar, because NAND requires *true* in order to express the equivalent of *not*.

SIMPLE-FOL		
(SIMPLE-BOOLEAN *rules not shown*)		
SET	→	*S*
	→	(*non-Xes S*)
BOOL	→	(*forall F SET*)
		(*exists F SET*)

(a)

FOL		
(SIMPLE-BOOLEAN *rules not shown*)		
F	→	(λ x_i . *BOOL*)
SET	→	*S*
	→	(*non-Xes S*)
BOOL	→	(*forall F SET*)
		(*exists F SET*)
		(*size>= OBJECT OBJECT*)
		(*size> OBJECT OBJECT*)
		(*squal-size? OBJECT OBJECT*)
		(*equal-color? OBJECT OBJECT*)
		(*equal-shape? OBJECT OBJECT*)

(b)

Figure 4-7: Two grammars for generating expressions with quantification. Both build on FULLBOOLEAN by adding primitives: (a) adds quantifiers, and (b) adds quantifiers and lambda abstraction, allowing for quantification over arbitrary predicates.

two in SIMPLEBOOLEAN. As above, this means that the relative difficulty of concepts may be informative about which representational system better-approximates human notions of simplicity. These two grammars also make different predictions about how the difficulty of *red and circle* is related to other concepts: SIMPLEBOOLEAN predicts that its difficulty should be predicted by the difficulty of using the operation *and*, whereas NAND predicts this difficulty should depend on the difficulty of using *nand* twice. Teasing apart the types of grammars in Figure 4-10 is the goal of the computational model presented in the next section.

We can also define grammars that go beyond simple Boolean logic. Figure 4-7(a), defines a grammar that involves simple quantification. Here, we introduce two more functions that return truth values, *exists* and *forall*. These correspond to the standard logical quantifier ∃ and ∀. These quantifiers are somewhat nonstandard in that they operate only over a very restricted domain, the objects in the current set. These functions themselves take a *function*, *F*, as an argument, as well as a set. So *exists* returns true if its argument *F* evaluates to *true* on any element of the set; *forall* returns true if *F* evalutes to *true* on all elements of the set. We must therefore include sets to quantify over. Here we choose two that are natural: the set of all elements in the current set, *S*, and the set of all elements in the context other than *x*, (*non-Xes S*). The nonterminal *F* expands as in SIMPLEBOOLEAN to some set of primitive functions. Thus we can write concepts such as *There exists a red* *object*:

$$\lambda\, x\, .\, (\textit{exists red? S}). \tag{4.8}$$

All elements in a set would be "wudsy" under this concept, if the set contained a single red element. Similarly, we can write concepts like *There exists a red object other than x*:

$$\lambda\, x\, .\, (\textit{exists red?}\ (\textit{non-Xes S})). \tag{4.9}$$

Note that in SIMPLE-FOL the only predicates that can quantify over *S* are those that *F* can expand to—namely the primitive feature accessors for size, shape, and color.

A much more interesting kind of quantification can be created if the grammar can potentially define new functions from sets to objects. FOL is one such grammar, where now *F* can expand to a *new* lambda expression using the rule $F \rightarrow (\lambda\, x_i\, .\, \textit{BOOL})$. Such a grammar is shown in 4-7(b). This means we can create concepts like (4.5) above, or,

$$\lambda\, x\, S\, .\, (\textit{exists}\ (\lambda\, x_2\, .\, (\textit{and}\ (\textit{red?}\ x_2)\ (\textit{equal-shape?}\ x\, x_2)))\ S). \tag{4.10}$$

Here, the *F* on the right hand size of the rule for *exists* was expanded to a new lambda expression, $(\lambda\, x_2\, .\, (\textit{and}\ (\textit{red?}\ x_2)\ (\textit{equal-shape?}\ x\, x_2)))$. We note one small technical complication, which is that when new lambda expressions are created, they introduce a new bound variable, x_i (for $i = 1,2,3,\ldots$). To deal with this, any time a lambda expression is generated, we also add a rule that allows $\textit{OBJECT} \rightarrow x_i$. That is, inside the *BOOL* generated on the right hand side of the first rule of FOL, we modify the PCFG to generate $\textit{OBJECT} \rightarrow x_i$. For simplicity, we make all expansions of *OBJECT* to *x* or any of the existing x_i equally likely. In this setup, our actual grammar is not a context-free grammar, but is closely related. Once new variables x_i are introduced, it is natural to include comparison operators such as *size*> and *equal-shape?*, which respectively check if an object is larger than another object or if two objects are the same shape[9]. In this example, we therefore expand the innermost lambda expression by choosing rules for *and*, *red?*, and *equal-shape?*, and expanding *OBJECT* to *x* in some places and x_2, the variable introduced by the $\lambda\ x_2$,

[9] In the case of SIMPLEBOOLEAN, there is no way to call functions on anything except *x*, meaning that it would be useful to have primitives for size, shape, and color comparison.

in others. Unlike the Boolean languages above, the FOL languages add expressive power: concepts involving *exists* and *forall* cannot be written down without quantifiers. As a representational theory, they therefore predict that people should be able to go beyond Boolean logic to learn these types of concepts that involve quantification.

4.4 Inference and the language of thought

So far we have defined several *languages*, spaces of lambda expressions compositionally built out of a given set of primitives. Our goal in this section is to develop an inferential theory around these representations. For learners, such rule-based concepts are only cognitively useful if they can be inferred from data. In other work we have argued that learning in this type of system predicts acquisition patterns and rich adult-knowledge in number-word learning (Piantadosi et al., submitted), and also can explain learning of multiple aspects of quantifier meaning (Piantadosi et al., in prep). We first present a learning theory that captures how expressions in a grammar like those above may be learned from labeled data. This is meant to model subjects' cognitive processes in the experiment: they take labeled examples and infer a target concept. On top of this model, we then introduce a Bayesian data analysis model (A. Gelman, Carlin, Stern, & Rubin, 2004), which takes subjects' responses and determines the probability of any particular representational system, given the structure of the learning model. This allows us to evaluate the ability of any grammar to explain the human learning curves.

Figure 4-8 shows the a graphical model (Pearl, 1998) that describes the relationships between the random variables in the learning model. The blue nodes in this figure denote variables that are known to learners. Let s_i and l_i respectively denote the i'th set of objects observed and their corresponding labels. For $i < n$, the sets and the labels are both known to learners since they have been provided feedback on previous sets. The main variable of interest, h, is a lambda expression in some representation language, G. The true value of h is the lambda expression that generates the $true$ / $false$ labels for each past set and the current set s_n. It is assumed that learners know a grammar G that generates expressions h, as well as variables that parameterize the likelihood (α, β, γ), and the prior (D_{**}), both

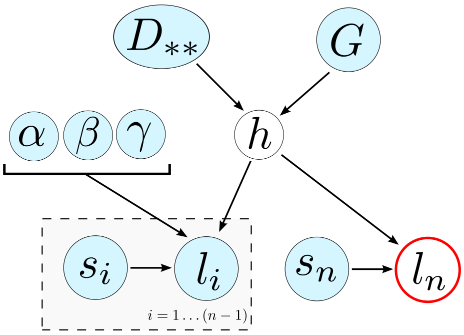


Figure 4-8: Graphical model representing the variables of the learning model. Here, the expression for the target concept h depends on Dirichlet parameters D_{**} and the grammar G. The specific labels observed for the i'th object of the n'th set depend on the hypothesis, set, and likelihood parameters, α and γ. In responding, the labels for the n'th set of objects are not observed, but the n'th set is.

of which are discussed later. Thus, learners must take their grammar and the previously observed labels to infer a hypothesis h, and apply this to the current set s_n to find l_n. We first show how the probability of h can be computed given any set of previously labeled data.

For convenience, denote the sequence of sets $(s_1, s_2, \ldots, s_i)$ by $\vec{s}_i$ and the corresponding sequence of sets of labels $\vec{l}_i$. We are interested in scoring the probability of h given these previously observed sequences, $\vec{s}_{n-1}$ and $\vec{l}_{n-1}$, and the other known variables. Using Bayes rule, this probability is given by

$$\begin{aligned} P(h \mid \vec{s}_{n-1}, \vec{l}_{n-1}, G, D_{**}, \alpha, \gamma, \beta) &\propto P(h \mid G, D_{**}) P(\vec{l}_{n-1} \mid h, \vec{s}_n, \alpha, \gamma, \beta) \\ &= P(h \mid G, D_{**}) \prod_{i=1}^{n-1} P(l_i \mid h, s_i, \alpha, \gamma, \beta). \end{aligned} \tag{4.11}$$

Equation (4.11) makes use of several natural conditional independences shown in Figure 4-8, for instance, the fact that l_n is independent of G once the hypothesis h is known, and that the l_i are independent once h is known. This equation has two parts, a prior and likelihood, which we now address in turn.

4.4.1 Priors on expressions

The prior $P(h \mid G, D_{**})$ embodies the core assumptions that learners bring to learning. Here we choose the prior to capture the assumption that learners should prefer representationally *simple* hypotheses. What is simple may depend on several factors: simplicity might depend on an expression's description length, corresponding to the number of primitives used in the expression h. Or, it may not be the case that all primitives are equally costly, meaning that the prior might depend on which primitives are used, not just how many. Additionally, Goodman et al. (2008) show that a model that prefers re-use can capture selective attention effects in concept learning, where subjects prefer concepts that use the same dimension (e.g., color) multiple times to those that use different dimensions. Thus, *circle or square* is easier than *circle or red* since the former references two shape dimensions, and our prior should potentially incorporate notions of re-use.

One simple way to capture all of these factors is to first imagine converting one of the above context-free grammars (e.g., Figure 4-6(a)) to a *probabilistic context-free grammar* (PCFG). This amounts to assigning a probability that each nonterminal will be expanded according to each of its rules (see Manning & Schütze, 1999). For instance one could make all rule expansions equally likely, meaning that learners would have equal preferences for using any primitive (given a nonterminal type). However, we might also assign the probabilities non-uniformly, corresponding to varying expectations about the probability of any particular expansion or primitive. Any such choice of probabilities induces a distribution on expressions, with the probability of any expression given by product of the probabilities of each of its expansions. Following Goodman et al. (2008) and M. Johnson, Griffiths, and Goldwater (2007) we use a variant of PCFGs that potentially encourage re-use of rules: a multinomial Dirichlet PCFG (see also O'Donnell et al., 2009). This is best understood as integrating over the rule production probabilities, using a Dirichlet prior on the (multinomial) rule expansions. Suppose that $C_{AB}(h)$ is the count of how many times an expression h uses the rule $A \rightarrow B$, and that C_{A*} is a vector of counts of how many times the nonterminal

A expands to *each* B. Then, for a given grammar G,

$$P(h \mid D_{**}, G) \propto \prod_{\text{nonterminals } A} \frac{\beta(C_{A*}(h) + D_{A*})}{\beta(D_{A*})}, \tag{4.12}$$

where D_{A*} is a vector of parameters of the same length as C_{A*}, and D_{**} is the set of all Dirichlet parameters (for each A). Here, β is the multinomial beta-function, which is given in terms of the Gamma function:

$$\beta(c_1, c_2, \ldots c_n) = \frac{\prod_{i=1}^{n} \Gamma(c_i)}{\Gamma(\sum_{i=1}^{n} c_i)}. \tag{4.13}$$

This prior uses a single Dirichlet-multinomial for each set of rule expansions for each nonterminal A. This Dirichlet-multinomial is parameterized by a set of real numbers, D_{A*}. If we re-normalize D_{A*}, we get the expected probability of each rule expansion from A, using the basic properties of the Dirichlet-distribution. Importantly, however, the Dirichlet parameters also characterize re-use: if the Dirichlet parameters are small in magnitude, then observing a rule used once will substantially increase its probability of being re-used in the future. In contrast, if the magnitude of D_{A*} is large, then adding additional rule counts does not change the probability an expansion will be re-used, so the model does not prefer re-use strongly. When $D_{A*} = \mathbf{1}$ for all A, this prior recovers the rational rules model of Goodman et al. (2008). By doing inference over the D_{**} we are therefore able to infer both the relatively probabilities of each rule expansion, and how much their probabilities of use in the future are influenced by whether or not they were used already in an expansion.

We also include one more parameter, a *temperature* T, which controls the strength of this prior by raising it to the $1/T$'th power. As $T \to 0$ the prior assigns most probability mass to short expressions, and as $T \to \infty$ the prior approaches a uniform distribution. For notational simplicity, this is left out of our equations.

4.4.2 The likelihood of data given an expression

The *likelihood* $P(\vec{l}_n \mid h, \vec{s}_n, \alpha, \gamma, \beta)$ in Equation (4.11) quantifies how well each expression h explains previously observed labels. Following the set-up of the experiment, we can consider sets of objects and learners who have observed $true$ and $false$ labels on some collection of previously observed data points. Given h, we assume labels are noisily generated for the current set by choosing the correct label (according to h) for each item with high probability α, and with probability $(1-\alpha)$ choosing from a baseline distribution on labels, parameterized by γ. Thus, when the labeling is not done according to h, $true$ is chosen with probability γ and $false$ is chosen with probability $(1-\gamma)$. This captures the intuition that the labels typically come from h, but occasionally noisy labels are generated from some baseline distribution. This process is therefore parameterized by two variables, α and γ. This process gives that

$$P(l_i = x \mid h, s_i, \alpha, \gamma) = \begin{cases} \alpha + (1-\alpha)\cdot\gamma \text{ if } h \text{ returns } x \text{ for } s_i \text{ and } x = true \\ \alpha + (1-\alpha)\cdot(1-\gamma) \text{ if } h \text{ returns } x \text{ for } s_i \text{ and } x = false \\ (1-\alpha)\cdot\gamma \text{ if } h \text{ returns } y \neq x \text{ for } s_i \text{ and } y = true \\ (1-\alpha)\cdot(1-\gamma) \text{ if } h \text{ returns } y \neq x \text{ for } s_i \text{ and } y = false. \end{cases} \tag{4.14}$$

Equation (4.14) simply adds up all the ways that x could be generated for s_i. When x is the correct label generated by h, then x could be generated by labeling from h with probability α, or by choosing from the baseline distribution with probability $(1-\alpha)$. This choice from the baseline depends on whether x is $true$ (probability γ) or x is $false$ (probability $1-\gamma$). If x is not the label returned by applying h to s_i, then it has to have been generated from the baseline distribution. This equation embodies the assumption that learners reason about the statistical process that generates their observed data, allowing them to imagine how likely any particular hypothesis h would make the observed data, given the noisy labeling process.

Equation (4.14) allows us to score the likelihood of the label for any particular labeled set of objects. But in the experiment, subjects see a sequence of labeled sets and objects. It is likely that learners have better memory for more recent examples, so we include a *memory-decay* on the likelihood, so that learners prefer more strongly to get more recent examples correct. Motivated by power law decays in memory (Anderson & Schooler,

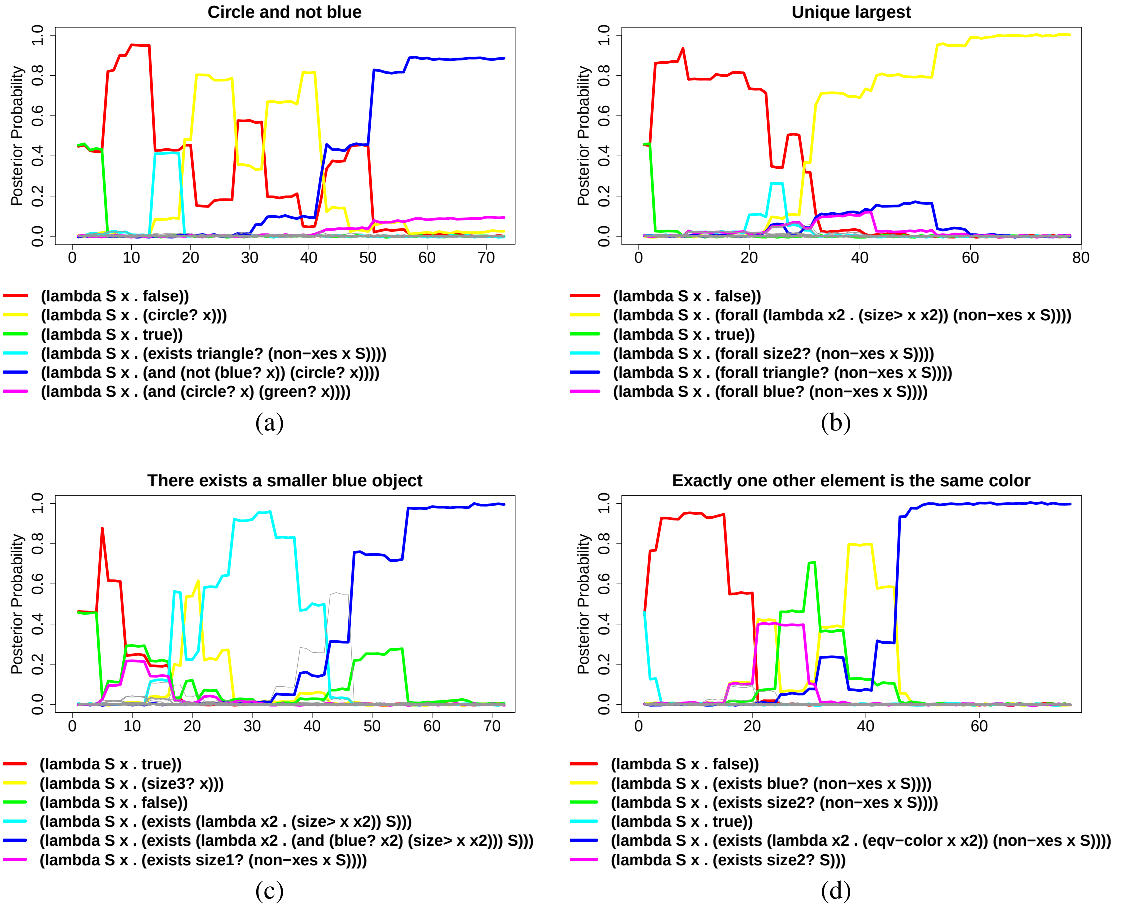


Figure 4-9: Learning curves with expressions from FOL-OTHER, with $\alpha = 0.75, \gamma = 0.5, \beta = -0.1$. The top six hypotheses are shown in color and all other hypotheses are in gray.

1991), this takes the form of a power law decay on the *log* likelihood:

$$\log P(\vec{l}_n \mid h, \vec{s}_n \alpha, \gamma, \beta) = \sum_{i=1}^{n} (n-i+1)^{-\beta} \log P(l_i \mid h, s_n, \alpha, \gamma) \tag{4.15}$$

Here, we have weighted the likelihood of an individual set from (4.14) by a power-law term, $(n-i+1)^{-\beta}$ which makes earlier data less important. This introduces one free parameter, $\beta > 0$, which controls the amount of memory-decay in the model. For $\beta > 0$, learners prefer hypotheses that explain more recent data, but as $\beta \to 0$, this preference is reduced.

4.4.3 The dynamics of LOT learning

In this section, we illustrate some simple dynamics of learning with the prior and likelihood described above. In its current form, this learning model takes labeled examples and returns a distribution over hypotheses, which are expressions in a representation language. Figure 4-9 shows the posterior probability of several hypotheses as the amount of training data increases for four target concepts. This figure collapses across logically equivalent hypotheses: for instance, (λ *x S.* (*and* (*blue? x*) (*circle? x*))) is logically equivalent to (λ *x S.* (*and* (*circle? x*) (*blue? x*))). We have merged both into a single line meaning that these plots essentially show behavioral predictions, collapsing across the specific form of the representation. This figure shows hypotheses from the FOL grammar shown in Figure 4-7(b).

These plots were produced by running $250,000$ steps of the Metropolis Hastings algorithm (Metropolis et al., 1953; Hastings, 1970; MacKay, 2003) on each amount of data, ranging through the 25 sets run in the experiment. At each amount of data, the top 250 hypotheses were stored, forming a large finite hypothesis space that was used for all further analysis. This means that any hypothesis that was found to be high probability at any amount of data was re-evaluated on the entire set of data, producing the learning curves shown in Figure 4-9.

These plots illustrate several important aspects of the learning dynamics. First, they show that in many cases, the learning model can arrive at the correct concept. This is true even when the target concept is quite complex: for instance, in *the unique largest* (Figure 4-9(b)) the model correctly constructs a lambda expression that quantifies over all elements other than x and asserts that all other objects x_2 are strictly smaller than x. In this sense, the learning model "really works" and is capable of narrowing down a vast space of hypotheses using only a few labeled examples—in this case, around 30 labeled sets.

Second, these plots demonstrate the model's simplicity bias: the expressions that are learned early are often simplified approximations of the correct target concept. For instance, for *circle and not blue* (Figure 4-9(a)) the model initially learns *circle*; for *there exists a smaller blue object* the model first learns to pick out objects of size 3, the maximum size, then picks out objects that have a smaller object in the set, and finally it converges on the correct answer. Such learning patterns demonstrate that “errors” subjects make in the experiment may be rational: the ideal learner does not immediately jump to *there exists a smaller blue object* when shown only two examples. Instead, simpler and thus more likely a priori hypotheses must be eliminated first.

These plots also illustrate the fact that for any given set of data, there are relatively few hypotheses relevant at any given time. Nearly all the probability mass in the model is split between at most the top 10 hypotheses. This is fortunate for a theory of concept learning based in such an unrestricted space, because it means that humans would only need to consider in depth a handful of relevant hypotheses at any given stage of learning. It is also fortunate for performing inference in this model: the full distribution on hypotheses can be approximated reasonably well using the top 10 or 100 hypotheses at each point in time.

Figure 4-9(d) shows an interesting concept that is not easily expressed in the representation language FOL. Expressing *exactly one other...* requires two quantifiers in FOL and this intuitively should take a considerable amount of data to justify. Indeed, with this amount of data the model does not learn the correct concept, but comes to *there exists another object of the same color*. This shows that the representation language chosen may substantially influence what concepts are easily learnable. In the next section, we formalize a Bayesian data analysis method for taking these types of learning curves and predicting human response patterns in the experiment. This will allow us to compare different representation languages, and make inferences about unknown parameters in the model.

4.5 Inferring the language of thought

The learning model described in the previous section specifies a probability of any expression, given some set of labeled data. This is intended as our psychological theory of how human learners react to evidence, given the assumed structure of the model, choice of grammar, and choice of parameter values. However, we are really interested in the right representational system—what grammar G and grammar parameters D_{**} are most likely, given people’s learning curves. We structure this problem as a Bayesian data analysis problem (A. Gelman et al., 2004).

4.5.1 Inference for data analysis

For each item in each set (for a given concept and list in the experiment) we observe a number of counts of how often subjects respond $true$ and $false$. Let $r_n(x)$ be the number of subjects who labeled set s_n with the set of labels x, and R the set of all human responses. In analyzing the data, we are interested in scoring the probability of any particular set of parameters given the subject responses. By Bayes rule, the probability

$$P(G, D_{**}, \alpha, \gamma, \beta \mid R, \vec{s}_n, \vec{l}_n) \propto P(R \mid G, D_{**}, \alpha, \gamma, \beta, \vec{s}_n, \vec{l}_n) P(G, D_{**}, \alpha, \gamma, \beta). \tag{4.16}$$

The first term here is the likelihood of the human responses for any given setting of the parameters. Under the assumption that subjects choose labelings according to the predictions of the learning model, this term is a multinomial likelihood,

$$P(R \mid G, D_{**}, \alpha, \gamma, \beta, \vec{s}_n, \vec{l}_n) = \prod_{i=1}^{n} \prod_{x} \left[P(l_i = x \mid \vec{s}_{i-1}, \vec{l}_{i-1}, G, D_{**}, \alpha, \gamma, \beta) \right]^{r_i(x)} \tag{4.17}$$

where product over x runs over all possible labelings x of s_i (all values of l_i). Equation (4.17) says that the probability of all responses according to the learning model is a product over all sets observed, and for each set a product over all possible labelings raised to the number of times that labeling is observed in subjects. The key term of (4.17) is the probability of any labeling given all previous data, $P(l_i = x \mid \vec{s}_{i-1}, \vec{l}_{i-1}, G, D_{**}, \alpha, \gamma, \beta)$, since this is the model's predicted distribution of responses to set i. This term is important because it characterizes the model predictions: the model works well if it generalizes like people do, labeling new data from typically ambiguous evidence in the same way as our study participants. The target expression h does not appear here because the right h is not known to participants; however, it can be computed using the previously defined prior (4.12) and likelihood (4.14):

$$P(l_i = x \mid \vec{s}_{i-1}, \vec{l}_{i-1}, G, D_{**}, \alpha, \gamma, \beta) = \sum_{h \in G} P(l_n = x \mid h, s_i, \alpha, \gamma, \beta) P(h \mid \vec{s}_{n-1}, \vec{l}_{n-1}, G, D_{**}, \alpha, \gamma, \beta). \tag{4.18}$$

Intuitively, subjects' distribution of guesses at l_n is given by the probability of l_n given each hypothesis h, times the probability that h is correct according to all previously labeled data.

The second term in Equation (4.16) is the prior on parameters. We choose these priors to have a very simple form: D_{**} is chosen according to a $gamma(1,2)$ prior and the priors on α, β, and γ are taken to be uniform. In practice, the amount of data the model is fit to makes these priors largely irrelevant.

Together, these parts of Equation (4.16) specify a probabilistic data analysis model that allows us to use the learning model to infer a distribution on unknown parameters that characterize the grammar and the likelihood. Until this point, we have presented this model as though there is only one concept and list; this was for notational convenience since otherwise all variables would have to be indexed by a concept (and potentially a list). Very importantly, however, the main parameters of interest—D_{**}, α, γ, β, and most importantly G—are psychological variables that are true *across* concepts. Most of the power of our analysis comes from finding parameters that work for a wide variety of concepts. So in reality, (4.17) involves a product over concepts[10].

4.5.2 Data analysis algorithm

Full Bayesian model comparison would compute $P(G \mid R)$, the probability of any grammar given all responses, marginalizing over all the unknown parameters; this is unfortunately an intractable integral. We therefore use two other standard measures: first, we compute the Bayesian Information Criterion (BIC), which scores how likely the data is according to the model, penalizing for the number of free parameters (Schwarz, 1978)[11]. The BIC is convenient because it relies only on the *maximum-likelihood* fits of parameters. If P_{ML} is

[10] And so $\vec{s}_n$ and $\vec{l}_n$ must also be indexed by concept and list correspondingly.

[11] BIC is similar to Akaike Information Criterion (Akaike, 1974), except that it more strongly penalizes free parameters and has a Bayesian justification.

the probability of the responses fitting D_{**}, α, γ, and β, then, the BIC is given by

$$BIC(G) = -2\log P_{ML} + k\log n, \tag{4.19}$$

where k is the number of free parameters, and n is the total number of data points. This is essentially the probability that the best fitting parameters assign to the data, penalized by the number of free parameters.

The second measure is more direct: the likelihood of held-out data. We only train the model (fit the parameters) on one of the two *lists* for each concept. The held-out scores represent the ability of the model to predict human learning curves on entire sequences of data that it has received no training on. This does not directly penalize free parameters since over-fitting will result in poor held-out performance. We note that the two lists of sets for each concept are generated at random, meaning that training on one list provides no information about the other list, other than the parameter values in the grammar and likelihood. This therefore provides a strong test of each model or grammar's performance and is the primary method we use for model comparison.

This still leaves the issue of how to fit the model parameters. In the data analysis algorithm, this is a *doubly intractable* problem, with an infinite search over hypothesized expressions in the grammar for each of an infinite number of choices of the parameter values. We approximate a solution to this problem by first constructing a finite space of hypotheses to approximate the infinite one, and then using this finite space in our data analysis algorithm. To make the finite space, we run $100,000$ Markov Chain Monte Carlo (MCMC) steps on each concept, list, and amount of data, using a version of the Metropolis-Hastings algorithm (Metropolis et al., 1953; Hastings, 1970; MacKay, 2003). These MCMC runs search over expressions using typical values of the likelihood parameters and $D_{**} = \mathbf{1}$, and produce a finite sample of hypotheses. Any hypothesis that occurs in the top 100 hypotheses for any amount of data on a particular concept and list is stored and added to the finite hypothesis space for the model. Thus, the finite space includes a large number of hypotheses that are high-probability at some point throughout the experiment. This is justified because the learning results above (Figure 4-9) shows most hypotheses are very low probability at each amount of data, so the top 100 form a reasonable approximation to the infinite space.

Given this finite space of hypotheses we then do MCMC to approximately fit the parameters α, β, γ, and D_{**}. To do this, we run 6000 iterations, each alternating between 10 MCMC steps over the likelihood parameters, and 10 MCMC steps over the prior parameters. In trial runs, most of the "burn in" time was used increasing the prior temperature parameter, which we initialize to a higher value of 3.5. This, along with hand-tuned proposal distributions, means that the model mixes within several hundred of the outer-loops, or several thousand total MCMC steps. Because we do inference over the prior but our finite space was constructed with a particular prior, we also update the finite hypotheses by, every 5 steps, sampling $10,000$ times from the prior and adding the top 25 hypotheses to the finite space. This keeps the finite approximation "current" to the inferred prior parameters, and generally gives replicable results across multiple inference runs.

To summarize, we have defined two statistical models. The first captures the behavior of an ideal learner over the space of lambda expressions, showing how for any particular choice of its prior and parameter values, one can compute predicted learning curves. To do this, we formalized a prior probability on lambda expressions, and a likelihood measuring how well each lambda expression explains previously observed labels on sets. By trading rationally between the prior and likelihood, we construct an idealized learning model on lambda expressions. In this section, we showed how to take the output of the learning model and predict the distribution of human responses. We can use this ability to predict human responses to compare representation languages: for each language G we can compute a BIC score and a held-out likelihood score, corresponding to performance on trained and untrained data. This ability to assign each potential LOT a number representing how well it predicts human learning will allow us to work backwards from human responses to see which representational systems best explain human learning curves. In the next section, we first apply this to simple Boolean representation languages, before moving on to languages with quantification.

4.6 Boolean concept analysis

The Boolean concepts studied here are shown by the green lines in Figures 4-2 to 4-4. These target concepts involved simple conjunctions and disjunctions of features, as well as concepts that most naturally involve other logical connectives. We analyze Boolean concepts first because they present a simple test case with several intuitively implausible bases that we show can be ruled out using our data analysis method. We first describe the Boolean languages compared here.

4.6.1 Boolean languages

First, we include the two grammars discussed earlier, SIMPLEBOOLEAN and NAND. The SIMPLEBOOLEAN grammar was the one used by (Feldman, 2000) and in addition corresponds naturally to the way that these logical words are used in natural language. The NAND basis is natural because it corresponds to a minimal set of logical operations[12]. A cognitive scientist who had strong expectations that the set of cognitive primitives was small, simple, and non-redundant might find this the most plausible basis. The NOR grammar, including only the operation *not-or* is similarly minimal and is also included for comparison. There are several natural extensions of SIMPLEBOOLEAN to consider. First, we might add logical operations such implication (*implies* or ⇒) or the biconditional (*iff* or ⇔). These operations are redundant in that they can be written using primitives in SIMPLEBOOLEAN: *implies* is λ *x y* . (*or* (*not x*) *y*), and *iff* is λ *x y* . (*or* (*and x y*) (*and* (*not x*) (*not y*))). The "claim" of a representational system including these primitives is that they are so simple for learners, they must be cognitive primitives rather than compositionally derived from other connectives. We include three additional languages, shown in Table 4.1: IMPLIES adds *implies*, BICONDITIONAL adds *iff*, and FULLBOOLEAN adds both.

All these languages allow for free-form recombination of logical connectives in that there are no restrictions on the compositional structure. However, there are ways of writing Boolean expressions that force everything into a *normal form*. Figure 4-10(a) shows one example: a DNF grammar for *disjunctive normal form*, in which all concepts are written as

[12] NAND is equivalent to a system expressed using only the *Sheffer stroke* (Sheffer, 1904).

Language	**Description**
SIMPLEBOOLEAN	*and*, *or*, *not*, used in any composition.
IMPLICATION	Same as SIMPLEBOOLEAN, but with logical implication (⇒).
BICONDITIONAL	Same as SIMPLEBOOLEAN, but a biconditional operation (⇔).
FULLBOOLEAN	Same as SIMPLEBOOLEAN, but with logical implication (⇒) and biconditional (⇔).
HORNCLAUSE	Expressions must be conjunctions of Horn clauses (e.g., (*implies* (*and* (*and a b*) *c*) *d*)).
DNF	Expressions are in disjunctive normal form (disjunctions of conjunctions).
CNF	Expressions are in conjunctive normal form (conjunctions of disjunctions).
NAND	The only primitive is *NAND* (not-and).
NOR	The only primitive is *NOR* (not-or).
ONLYFEATURES	No logical connectives; the only hypotheses are primitive features (*red?*, *circle?*, etc).
RESPONSEBIASED	Learners only infer a response bias on $true$ / $false$.

Table 4.1: Summary of Boolean languages compared here.

disjunctions of conjunctions. This might be natural if people paid attention to conjunctions of features, and preferentially stated concepts in terms of these conjunctions; indeed, this system was the representation language used by Goodman et al. (2008). Similarly, we can also consider a CNF grammar that expresses concepts as conjunctions of disjunctions.

Next, Figure 4-10(b) shows a grammar for conjunctions of *Horn clauses* (Horn, 1951; McKinsey, 1943), which generate expressions of the form $x_1 \wedge x_2 \wedge \ldots \wedge x_k \rightarrow y$. Horn clauses are often used in artificial intelligence systems due to their desirable computational properties (e.g., Hodges, 1993; Makowsky, 1987; S. Russell & Norvig, 2009, section 7.5.3). In particular, they support efficient algorithms for inference and satisfiability (Dowling & Gallier, 1984; S. Russell & Norvig, 2009), and thus provide a plausible basis for Boolean reasoning in any computational system. Indeed, recent cognitive models have assumed the plausibility of Horn clauses in human learning of theories about the world (Katz et al., 2008; Kemp et al., 2008a).

DNF		
START	→	λ *x . DISJ*
DISJ	→	*CONJ*
		(*or CONJ DISJ*)
CONJ	→	*BOOL*
		(*and BOOL CONJ*)
BOOL	→	(*F OBJECT*)
		(*not* (*F OBJECT*))
OBJECT	→	*x*
F	→	COLOR
		SHAPE
		SIZE
COLOR	→	*blue?*
		green?
		yellow?
SHAPE	→	*circle?*
		rectangle?
		triangle?
SIZE	→	*size1?*
		size2?
		size3?

(a)

Horn Clause		
START	→	λ *x . CONJ*
CONJ	→	*CLAUSE*
		(*and HORN-CLAUSE CONJ*)
CLAUSE	→	(*implies CONJ PRIM*)
CLAUSE	→	(*implies CONJ* $false$)
PRIM	→	(*F OBJECT*)
OBJECT	→	*x*
F	→	COLOR
		SHAPE
		SIZE
COLOR	→	*blue?*
		green?
		yellow?
SHAPE	→	*circle?*
		rectangle?
		triangle?
SIZE	→	*size1?*
		size2?
		size3?

(b)

Figure 4-10: Two additional bases for Boolean logic. The DNF grammar expresses concepts as disjunctions of conjunctions; the HornClause grammar expresses concepts as conjunctions of Horn clauses.

For baseline measures, we include ONLYFEATURES which corresponds to learners with no logical connectives, but only access to primitive features. Poor performance of this would indicate Boolean compositional abilities, as opposed to an ability to just select feature values. An even simpler base, RESPONSEBIASED corresponds to learners who only try to learn the correct response bias, which is equivalent here to a representation language who only expressions are $true$ and $false$.

Finally, we evaluate several other types of models, described in the Appendix. First, an EXEMPLAR model measures each set's similarity to all previously observed sets and attempts to generalized previous labels based on this similarity. There is a LOGISTIC model that performs a simple logistic regression within each concept and list, providing a type of "psychophysicist's baseline": if we can predict learning curves better than a freely fit logistic curve, then that provides good evidence for a real representational theory[13]. We also include a version of the model that incorporates no prior: the UNIFORM model has an improper, flat prior on expressions, corresponding to no prior bias for simplicity. As with the LOT models, the free parameters (e.g., the distance metric for the EXEMPLAR model) are fit using Bayesian data analysis.

4.6.2 Model comparison results

Table 4.2 shows a model comparison of these representation languages in predicting human responses. This table shows the held-out likelihood score (H.O.LL) and BIC, described above. Better model fit on the held-out likelihood corresponds to numbers closer to positive infinity; better model fit on BIC corresponds to lower numbers. The main measure we use for evaluating languages is held-out likelihood, since this quantifies generalization performance and does not require any additional assumptions about the models being tested.

The next column gives the model's number of free parameters, counting the several parameters in the likelihood and the D_{**} parameters of the grammar. The last two columns of Table 4.2 give two intuitive measures of the model's performance. $R^2_{response}$ gives the model's overall R^2 value to individual responses, quantifying the amount of variation in proportion of people who select $true$ for each single response that can be explained by the

[13] Note that this procedure cannot generalize to new, held-out lists or concepts.

Grammar	H.O. LL	BIC	FP	$R^2_{response}$	R^2_{mean}
FullBoolean	-16296.84	33758.22	27	0.88	0.60
Biconditional	-16305.13	33653.88	26	0.88	0.64
CNF	-16332.39	34094.68	26	0.89	0.69
DNF	-16343.87	33595.98	26	0.89	0.66
SimpleBoolean	-16426.91	34050.60	25	0.87	0.70
Implies	-16441.29	34030.89	26	0.87	0.70
HornClause	-16481.90	33989.62	27	0.87	0.65
NAND	-16815.60	35082.25	24	0.84	0.61
NOR	-16859.75	35415.95	24	0.85	0.58
uniform	-19121.65	39168.89	4	0.77	0.06
exemplar	-23634.46	46605.23	5	0.55	0.15
OnlyFeatures	-31670.71	64519.83	19	0.54	0.14
response-biased	-37912.52	75930.60	4	0.03	0.04

Table 4.2: Model comparison results on all Boolean concepts.

model. R^2_{mean} gives the model's ability to predict each concept's average difficulty, across all concepts. These correlation measures provide a more intuitive way of understanding the relative performance of each model and are computed only on held-out data.

Table 4.2 shows grammars sorted by the main measure, held-out likelihood. The worst performing models and grammars are ones that lack structured representation: the exemplar model, OnlyFeatures grammar, and Response-Biased grammar. The best of these, the exemplar model, can explain only around half of the variance in human responses. The failure of these models provides evidence that such unstructured approaches miss fundamental facts about people's patterns of generalization. The next worst model is the uniform model that has no simplicity bias[14]. Again, this provides strong evidence for a simplicity bias in concept learning, in line with Feldman (2003c, among others).

Next, we have the NAND and NOR grammars. These fare poorest of the real representation languages, providing some validation for this approach since we can rule out languages with the wrong inductive bias, even if they have the necessary computational power. In some cases the R^2 measures for these poor performers are substantial. NOR for instance has an $R^2_{response}$ that is only 0.03 away from the best grammar, even though its held-out likelihood is several hundred points worse. This is likely because R^2 is not as sen-

[14] The language used in this model was the best-performing language, FullBoolean.

sitive a measure as the likelihood scores, and the R^2 of the best grammars is upper-bounded by the noise of the responses. Additionally, many simple functions like λx . (*red? x*) can be directly expressed in NAND and NOR grammars, and others like λx . (*not* (*red? x*)) can be relatively easily expressed: λx. (*nand true* (*red? x*)). If people spend much of their time considering these very simple concepts, then most of their responses will not distinguish NAND from, say, FULLBOOLEAN.

Next, we have the HORNCLAUSE, which provides a relatively poor model of people's inductive bias in this task. This indicates that this common representation for AI and machine learning research does not accurately capture human inductive biases. The next best languages are IMPLIES and SIMPLEBOOLEAN. SIMPLEBOOLEAN allows for free-form combination of *and*, *or*, and *not*; IMPLIES additionally includes logical implication. The fact that SIMPLEBOOLEAN performs worse than languages with more primitives like FULLBOOLEAN and BICONDITIONAL means that people likely have a richer set of logical connectives than just *and*, *or*, and *not*. In particular, the grammars that perform best according to Table 4.2 are the grammars that add *iff* to SIMPLEBOOLEAN, as well as the normal-form grammars, DNF and CNF. The largest differences between languages with an without *iff* appears to be in concepts that require exclusive-or (*XOR*), such as *red XOR circle*[15]. Even for those concepts, differences in learning curves for languages with and without *iff* are not very large, resulting in very close BIC and held-out likelihood scores.

The best grammar, FULLBOOLEAN, scores about 8 points better on held-out likelihood than its closest competitor, BICONDITIONAL, and about 160 points *worse* than the best language in terms of BIC, DNF. This indicates that the choice of "best" grammar depends somewhat on the measure chosen; here we choose held-out likelihood since it makes the fewest assumptions. However, the convergence of scores for these top grammars indicates that the data set does not have sufficient resolution to distinguish among the best performing grammars.

In some sense, we may take the held-out data scores as "final" measures of how well each grammar performs. However, we might also wonder if these differences between the

[15] This indicates, concepts with *XOR* will be important for future work testing different representational bases.

top grammars are statistically significant—after all, we tested only finitely many concepts out of an infinity of possible concepts. Also, our inference algorithms make several approximations, and it would be good to know if these approximations are good enough to maintain sensitivity to 8 point differences in likelihood. First, we computed a Wilcoxon signed-rank test, a nonparametric paired comparison, on the likelihood that each pair of models assigned to each held-out data point. With Bonferroni correction for multiple comparisons[16], this reveals no difference between the FULLBOOLEAN language and BICONDITIONAL, CNF, DNF, or SIMPLEBOOLEAN. However, it does show that FULLBOOLEAN performs significantly better than the others ($p < 0.05$, corrected)—in particular, IMPLIES, HORNCLAUSE, NAND, and NOR. We also ran the inference algorithms multiple times on the set of Boolean concepts[17]. This revealed some variation in the order of the top four grammars, but consistency in ranking these four better than the rest, typically with either DNF, BICONDITIONAL, or FULLBOOLEAN ranked first. Though this shows that the resolution of the present data set cannot distinguish between the top four grammars, it does indicate that FULLBOOLEAN, CNF, DNF, and BICONDITIONAL are better than the other languages for capturing people's inductive bias.

We note that the best grammars can explain an impressive amount of variation in the individual subject responses. This is especially compelling because this correlation is computed only on held-out data: with no parameters fit to the held-out data, the learning model described above can explain 88% of the variation in subjects' response patterns. This is further demonstrated by Figure 4-11, which shows FULLBOOLEAN's probability of responding *true* compared to participants in the experiment. This shows substantial noise in the individual object (in a particular set, list, and concept) responses, shown in gray. The binned data for which we have much less measurement error shows a strong and almost perfectly linear relationship between model predictions and human responses. This holds across both training and held-out data suggesting no over-fitting in the model. This relationship is noisiest in the middle of the range—where humans and models respond *true* and

[16] It is not clear what the best statistical testing procedure is to use here, since we are selecting the main comparison grammar, FULLBOOLEAN by good performance, and it should be more difficult to find statistical differences between the top performing grammars. Bonferroni correction here is likely highly conservative.

[17] A single "run" takes about 50 processor-days of CPU time, making gathering a large sample of runs impractical.

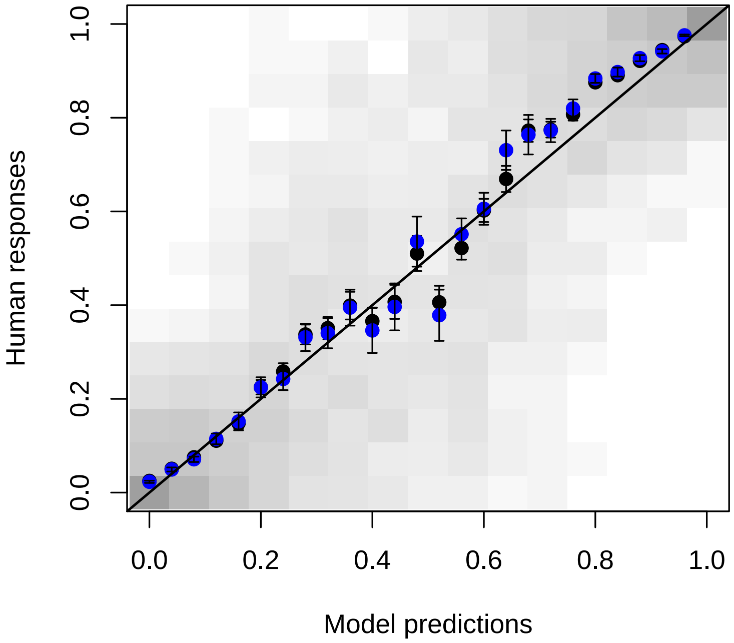
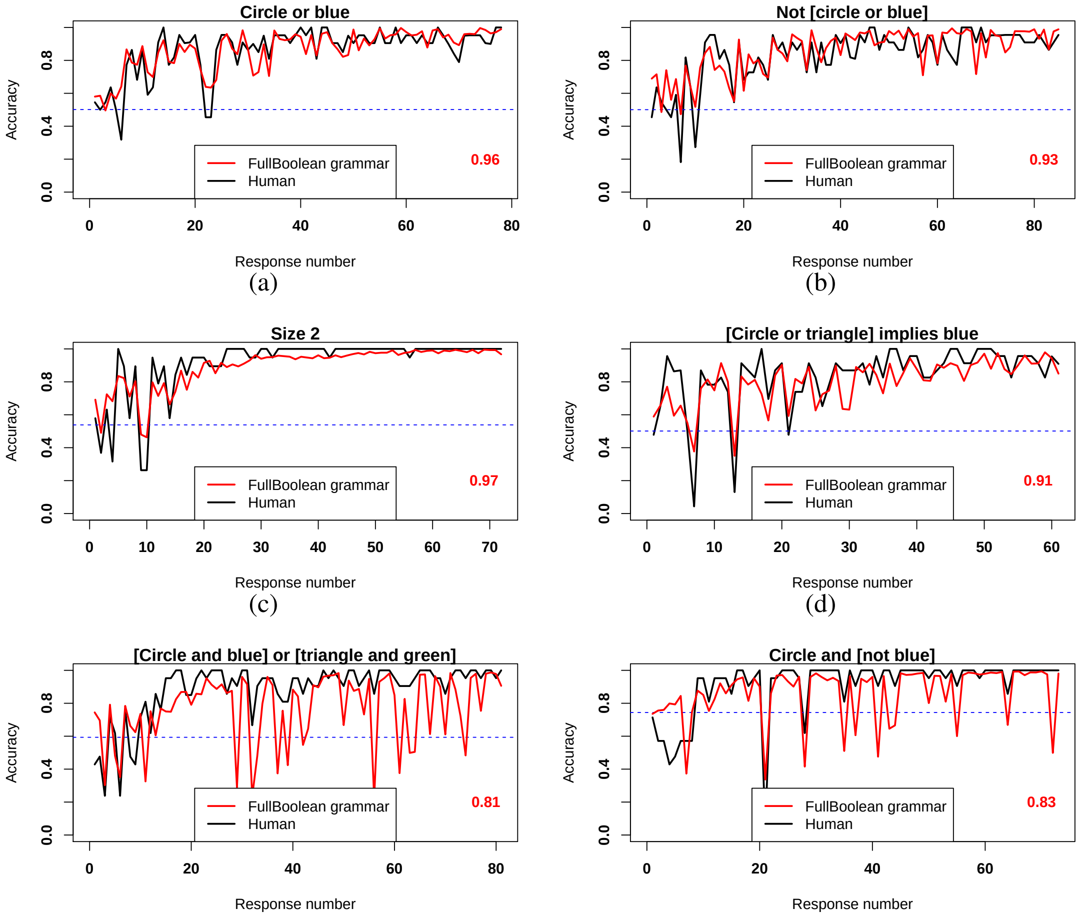


Figure 4-11: Relationship between model predicted probability of responding *true* (x-axis) and participants' probability (y-axis). The gray background represents unbinned data, corresponding to raw responses on each object in each set, list, and concept, of the experiment. Black points are binned training data and blue are binned held-out data.

false with roughly equal proportion—and it is not clear if this variability is due to non-human characteristics of the model, or simply the increased variance of binomial outcomes when $p \approx 0.5$. The other well-performing grammars appear similar when plotted in this manner.

4.6.3 Learning Curves

Importantly, the model is capable of capturing many of the qualitative phenomena that learners exhibit. In particular, learners in the experiment tended to make systematic patterns of errors (see Figure 4-5). Because we have implemented a full learning model, we can see if the model makes similar errors. Figure 4-12 shows six typical learning curves. The x-axis here shows the response number for List 2 in the experiment, the held-out data. The y-axis shows human subjects' proportion correct at this object, and the model's proportion



(a) (b) (c) (d) (e) (f)

Figure 4-12: Human (black) versus predicted learning curves on four example concepts. The numbers in the lower right give R^2s between FULLBOOLEAN's predicted accuracies and humans' observed accuracies. Note the human data for these sequences of data were held-out from training all models.

correct. Thus, each y value represents the accuracy of learners and the model, conditioning on having seen the correct labels for all previous sets. The dotted blue line here represents the base rate of the concept. This figure shows that through the experiment, both learners and the model make systematic patterns of errors, corresponding to dips in accuracy for the black and red lines. Figures 4-12(a) to 4-12(d) show concepts where the agreement of the model and people is quite close, and Figures 4-12(e) to 4-12(f) show cases where the agreement is less good. The fact that the model and people tend to agree indicates that what people are doing is largely rational, generalizing in a way that is similar to our model based on previous labeled examples. In this sense, their "errors" are not really mistakes, but only cases where previous data has led them to hypotheses that give answers different from what the target concept says. Even in cases where the model predictions differ from human participants, many of the differences tend to be in the magnitude of an error and not presence or absence of an error. For instance, many of the model "dips" in 4-12(f) line up with places where people do have an increased rate of errors, they just do not make errors as often as the model. It is important to emphasize that these model curves are *not* fit to this data. There is no parameter of the models, for instance, that makes the learning curves dip around item 20 of 4-12(a). This dip is caused by the model's learned prior (D_{**}) on independent training data, combined with the fact that at this particular item the observed data leads both models and people to make an incorrect generalization.

4.6.4 The inferred grammar

A more detailed picture of the grammars inferred from the experimental results is shown in Figure 4-13. This shows the D_{**} parameters found by the data analysis model for FULLBOOLEAN. The red points correspond to MAP estimates of the parameters, and the intervals are *highest-posterior density* ranges, using the Chen and Shao (1999) algorithm from the R package *boa* (Smith, 2007). These numbers can, roughly, be interpreted by re-normalizing for each nonterminal type to yield a PCFG. Thus, *COLOR* expands to *yellow?* and *green?* with approximately equal probability. *blue?* is much more salient—it is more likely to be used in a concept. Similarly, for *SHAPE* expansions, subjects are roughly twice as likely to expand to *circle?* as the others, indicating a bias in the prior for concepts using circular shapes. The (unnormalized) magnitude of these numbers shows the role of re-use in an expression: roughly, each time a rule is used in creating an expression, its parameter value is increased by 1 for later expansions in the same expression, and the nonterminals are re-normalized. For example, F is equally likely to expand to a *SIZE*, *SHAPE* or *COLOR* since all of these have equal magnitudes (≈ 2.0). Roughly, once F is expanded one way once—say to *SHAPE*—in an expression, *SHAPE* is preferentially re-used with probability $(2.0+1)/(2.0+1+2.0+2.0) = 0.43$ next time a F expansion is followed. Here, since *SHAPE* was used once, 1 has been added to its initially unnormalized probability of 2.0, and the expansions re-normalized. Thus, as the magnitude of these parameters gets large,

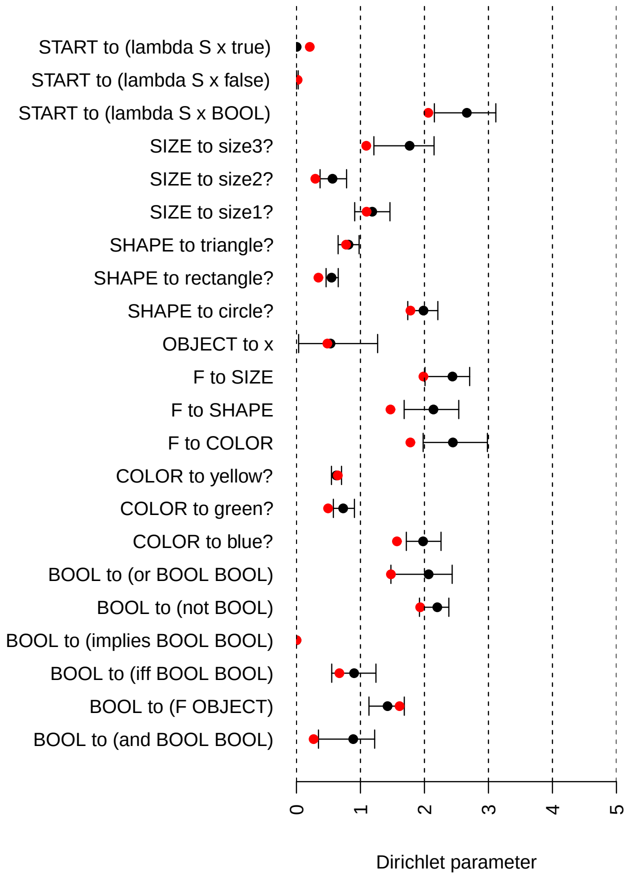


Figure 4-13: Posterior parameters D_{**} found by the inference algorithm for the FULLBOOLEAN grammar. The red dots are MAP grammar parameters and the intervals are 95% HPD intervals computed using the Chen & Shao (1999) algorithm.

re-use of rules is less preferred; the general low values of these parameters indicate that re-use is likely preferred by learners, consistent with Goodman et al. (2008).

Figure 4-13 reveals several interesting trends. First, the *START* symbol is expanded with very high probability to an expression involving *BOOL* instead of *true* or *false* hypotheses, indicating a prior bias against truth-functionally trivial expressions. Plausibly, *SIZE* is preferentially expanded to the most salient sizes, *size3?* and *size1?*. Not surprisingly, this grammar assigns substantial probability to *iff*, the primitive that only the top two grammars, FULLBOOLEAN and BICONDITIONAL include. Lower probability is assigned to *implies*. However, *implies* is not zero probability—otherwise these probabilities for FULLBOOLEAN would essentially yield the grammar BICONDITIONAL. The MAP probability of *implies* is about 0.001, so use of *implies* would yeild a prior about 7 log points lower, requiring getting 10 to 15 additional example objects (not sets) correct throughtout the experiment for typical values of α. Said another way, this difference in the prior could easily be overcome in the likelihood with just a handful of examples; this is why FULLBOOLEAN can outperform BICONDITIONAL despite the fact that the only primitive FULLBOOLEAN additionally has (*implies*) is low probability.

Examination of grammars for CNF and DNF reveal trends of some preference for reuse, especially of feature primitives. These grammars also tend to set probabilities to generate primarily conjunctive concepts, rather than disjunctive concepts, leading to a stronger prior conjunction bias than FULLBOOLEAN[18].

4.6.5 Boolean summary

These results showed that models that treat learning as inference in a rich representational system can capture participants' detailed patterns of errors (Figure 4-12) as well as their patterns of graded generalizations (Figure 4-11). These best rule-like representations outperform other types of baselines, such as simple exemplar models, logistic curves, and response-biased models (Table 4.2). Importantly, we are also able to provide evidence against intuitively implausible representations bases, such as the NAND basis, and even formalisms popular in AI like Horn clauses, although the amount of data at present does not distinguish between the best grammars.

4.7 More complex languages

Building off of previous experimental and computational studies (Kemp, 2009; Piantadosi et al., 2009), we extend the modeling results to the wider range of concepts from our experiment that involve quantification and relational terms. To model these concepts, we must consider spaces of representation languages that include these additional operations such as existential and universal quantification, or cardinality operations. Ideally we might write a set of primitives and construct a grammar either including or excluding each pos-

[18] This may be a hallmark of over-fitting, potentially explaining why CNF and DNF performed better on training data but trended worse on held-out data.

sible primitive. The problem with this approach is that the number of possible grammars is exponential in the number of primitives we consider including or not. We might alternatively consider writing a large grammar including all of the primitives and positing that low-probability primitives are likely not components of the LOT. The problem with this is demonstrated by FULLBOOLEAN and SIMPLEBOOLEAN above, where low-probability operations (*iff* and *implies*) nonetheless improve the model fit. It is difficult to tell from the values of D_{**} which primitives should be considered "in" the grammar. We therefore take a middle road by constructing several plausible *collections* of primitives and either including each or not. This gives a small family of grammars, and for each grammar, we run inference to fit the D_{**} parameters. Table 4.3 shows five different families of primitive functions. The primitives in each family can be added or not to the best Boolean language, FULLBOOLEAN, to form a new and more powerful language.

First we can consider adding first-order quantifiers, *exists* ($\exists$) and *forall* ($\forall$), as in the FOL grammar. These primitives strictly increase the expressive power of any of the Boolean representation languages, allowing for existential and universal quantification. We allow each type of quantification to operate either over the entire set S, or over the elements other than x in S, and the probabilities of each of these types of quantification are fit in the PCFG.

As mentioned above, only adding *exists* and *forall* yields relatively impoverished quantificational abilities. This is because the expansion of F is restricted to the primitive functions shown in FULLBOOLEAN above (Table 4.1). Thus, we can only form expressions like

$$\lambda\, x\, S\, .\ (\textit{exists red? } S) \tag{4.20}$$

for *There exists a red object in S*, or

$$\lambda\, x\, S\, .\ (\textit{exists circle? } (\textit{non-Xes } S)) \tag{4.21}$$

for *there exists a circle in the set* $S \setminus \{x\}$. True quantification abilities would allow an arbitrary predicate F to range over a set, not just the primitive features. This can be accomplished by allowing F to expand to a new lambda expression using the rules in

FOL	
(*exists F SET*)	There exists some $x \in S$ such that (F x)
(*forall F SET*)	For all $x \in S$, (F x)
LAMBDA-AND-RELATIONAL	
($\lambda\, x_i$. *BOOL*)	Lambda abstraction (also introduces a new bound variable x_i)
(*equal? x y*)	x and y are the same object
(*same-shape? x y*)	x and y are the same shape
(*same-color? x y*)	x and y are the same color
(*same-size? x y*)	x and y are the same size
(*size> x y*)	x is larger than y
(*size>= x y*)	x is large than or equal to y
ONE-OR-FEWER	
(*exists-one-or-fewer F SET*)	There exists one or zero $x \in S$ such that (F x)
SMALL-CARDINALITIES	
(*exists-exactly-one F SET*)	There exists exactly one $x \in S$ such that (F x)
(*exists-exactly-two F SET*)	There exists exactly two $x \in S$ such that (F x)
(*exists-exactly-three F SET*)	There exists exactly three $x \in S$ such that (F x)
SECOND-ORDER-QUANTIFIERS	
(*exists-shape P*)	There a shape predicate $s \in \{$ *circle?*, *rectangle?*, *triangle?*$\}$ such that (P s)
(*exists-color P*)	There a color predicate $s \in \{$ *blue?*, *green?*, *yellow?*$\}$ such that (P s)
(*exists-size P*)	There a size predicate $s \in \{$ *size1?*, *size2?*, *size3?*$\}$ such that (P s)

Table 4.3: Five sets of primitives which can each be independently included or not to form a space of possible grammars. All grammars include expansions mapping $SET \rightarrow S$ and $SET \rightarrow$ (*non-Xes S*), respectively the context set S and the set $S \setminus \{x\}$.

LAMBDA-AND-RELATIONAL. This introduces a rule for defining new *functions F*:

$$F \rightarrow \lambda\, x_i \ .\ BOOL. \tag{4.22}$$

This rule says that a nonterminal of type F can be expanded into a lambda expression $\lambda\, x_i$ followed by an expansion of *BOOL* (for $i = 1, 2, 3, \dots$) . For instance, F could expand to $\lambda\, x_2$. (*or* (*red?* x_2) (*blue?* x_2)). Quantifiers such as *exists* then can take this function and a set:

$$\lambda\, x\, S\ .\ (exists\ (\lambda\, x_2\ .\ (or\ (red?\ x_2)\ (blue?\ x_2)))\ S). \tag{4.23}$$

Here, *exists* returns true iff the *function* ($\lambda\, x_2$. (*or* (*red?* x_2) (*blue?* x_2))) is true for some element of S. As described above, creating a new function in this way requires introducing a new bound variable—here x_2—and we must therefore allow for these newly introduced bound variables to be generated by the grammar. We suppose that each nonterminal that could expand to a bound variable has a certain probability of doing so, but that this probability mass is split equally between all possible bound variables at the current depth. This complication means that the grammars we use are not strictly probabilistic context-*free* grammars. However, the expression minus the bound variables are context-free, and the bound variables are uniformly generated from those that are possible at each depth.

The FOL operators correspond to those in classical logic. It has also been suggested, though, that other types of quantification actually provide a better account of people's inductive learning. For instance, Kemp, Goodman, and Tenenbaum (2008b) introduces two quantifiers, *exists-exactly-one* and *exists-one-or-fewer*. Both of these quantifiers are analogous to *exists*, except that *exists-exactly-one* is true if there is only one element of the set satisfying the predicate, and *exists-one-or-fewer* is true if there is at most one element of the set satisfying the predicate. These quantifiers can be written using the more standard *exists* and *forall* predicates. For instance, *exists-exactly-one*, is a function of a function F and a set S, and can be written as,

$$\lambda\, F\; S\;.\;(\mathit{exists}\;(\lambda\, x_1\;.\;(\mathit{and}\;(F\;x_1)\;(\mathit{forall}\;(\lambda\, x_2\;.\;(\mathit{implies}\;(F\;x_2)\;(\mathit{equal?}\;x_1\;x_2)))\;S)))\;\;S). \tag{4.24}$$

In other words (*exists-exactly-one* F S) is true if there is one element x_1 in S satisfying F, and for each x_2 in S, if x_2 satisfies F it must be x_1. Importantly, these quantifiers are quite complex to express using *exists* and *forall*, so including them as primitives substantially changes the inductive bias of the model. More generally, we have argued elsewhere (Piantadosi et al., submitted, in prep) that small-set cardinalities (1, 2, and 3) should also be included as representation primitives, in line with very young children's abilities to manipulate small sets (Wynn, 1992). These novel quantifiers are included as ONE-OR-FEWER and SMALL-CARDINALITIES.

Finally, we also compare a simplified version of *second-order* quantification. In standard logic, second-order quantification allows for quantification over predicates, or equivalently subsets of the domain of discourse. For instance, a typical second-order expression is $\exists P \forall x . P(x)$, which is true if there exists a predicate P such that $P(x)$ for all x. Here, it is difficult to allow for quantification over all predicates, but we can allow quantification over the primitive feature predicates, *red?*, *blue?*, *triangle?*, *size1?*, etc. This is not formally powerful enough for capturing "real" second-order logic since this type of quantification can be expressed in first-order logic, but it does capture an intuitive sense of quantifying over predicates rather than objects (see also Kemp, 2009). Thus, an expression such as

$$\lambda\, x\, S\, .\, (\textit{exists-color}\ (\lambda\, P\, .\, (\textit{forall}\ (\lambda\, x_2\, .\, (P\ x_2))\ S))) \tag{4.25}$$

says that there exists a color predicate P (e.g., a predicate in *red?*, *green?*, *blue?*) such that $(P\ x_2)$ is true for all x_2 in S. In other words, all of the elements of S are the same color, no matter what that color happens to be. We note that—unlike second-order logic in general—this could be written using only disjunctions of *forall*, although it would be substantially more complex. Also, it is the case that these second-order predicates require an additional bound variable to be interesting: concepts such as $\lambda\, x\, S\, .\, (\textit{exists-color}\ (\lambda\, P\, .\, (P\ x)))$ is true for all objects.

To summarize, we have introduced several sets of primitive functions, each of which may or may not be included in a hypothesized LOT. Because the learning model we developed is powerful enough to handle learning in any representation system, we can apply the same methods as the previous Boolean section to see which combination of these primitives best captures people's learning curves.

4.8 Results

Grammars without the LAMBDA-AND-RELATIONAL operations generally performed poorly, so all grammars compared here include these primitives. Thus, by including or excluding each of 4 sets of primitives, we form a hypothesis space that encompasses a total of $2^4 = 16$ different grammars. We additionally include the top-performing Boolean grammars on this wider space of concepts to test whether people's inductive machinery goes beyond these simple Boolean predicates.

4.8.1 Model comparison results

FOL	One-Or-Fewer	Small-Cardinalities	2nd-Ord.-Quan.	H.O. LL	BIC	FP	$R^2_{response}$	R^2_{mean}
✓	✓	·	·	-79023.25	160305.43	41	0.66	0.78
✓	✓	✓	·	-79096.47	160468.65	44	0.66	0.78
✓	·	·	·	-79329.61	160745.98	40	0.65	0.78
✓	✓	·	✓	-79347.52	161385.44	46	0.65	0.77
·	✓	·	·	-79463.06	161547.28	39	0.64	0.80
✓	·	✓	·	-79518.84	161772.87	43	0.65	0.77
·	✓	✓	·	-79863.95	162343.18	42	0.63	0.77
✓	·	·	✓	-79908.25	162201.60	45	0.64	0.78
✓	✓	✓	✓	-79997.35	162615.70	49	0.64	0.74
·	✓	·	✓	-80261.60	163076.90	44	0.63	0.77
✓	·	✓	✓	-80366.78	162942.38	48	0.63	0.75
·	·	✓	·	-80392.52	163807.06	41	0.63	0.72
·	✓	✓	✓	-80435.33	164010.82	47	0.62	0.76
·	·	✓	✓	-80604.70	164353.55	46	0.63	0.71
BICONDITIONAL				-81790.49	167998.05	26	0.59	0.72
FULLBOOLEAN				-81844.53	168048.55	27	0.58	0.71
SIMPLEBOOLEAN				-82134.87	168911.78	25	0.58	0.73
DNF				-82380.87	168931.47	26	0.59	0.73
CNF				-82597.55	169541.39	26	0.58	0.73
·	·	·	✓	-82745.12	169779.55	43	0.56	0.72

Table 4.4: Model comparison results on all languages with quantifiers.

Table 4.4 shows the results of the model comparison on all languages. Beyond the primitives in FULLBOOLEAN, the best grammar here includes only primitives from FOL and ONE-OR-FEWER. This grammar performs substantially better than the Boolean languages, across all measures. Using a Wilcoxon signed rank test on held-out likelihoods, the top grammar is significantly better than the second place grammar and all others, conservatively correcting for 16 comparisons ($p < 0.01$, corrected). This provides strong evidence for quantification in the LOT, in line with Kemp (2009); the superiority of a grammar with multiple types of quantifiers indicates that, like the Boolean results, quantificational operations in the LOT do not make use of a “minimal” basis of operations (such as just FOL).

These results suggest that SMALL-CARDINALITIES are potential primitives since the second-place grammar includes them; note that the concepts studied here do not include many operations on small cardinalities. Most concepts here required checking only for the existence of single elements, which is a cardinality operation captured by FOL. Grammars with these operations might do better if more of the target concepts require them.

These results provide strong evidence against SECOND-ORDER-QUANTIFIERS: for every other choice of primitives, addition of SECOND-ORDER-QUANTIFIERS reduced the model fit. Indeed addition of *only* SECOND-ORDER-QUANTIIFERS to FULLBOOLEAN resulted in a language that performed worse than any of the Boolean languages. This provides some evidence that people tend not to quantify over properties, consistent with Kemp (2009). This result contrasts with SMALL-CARDINALITIES, which improves over Boolean LOTs, though not as much as the inclusion of other quantifiers.

These results also provide evidence that the non-normal-form Boolean grammars (e.g., BICONDITIONAL, FULLBOOLEAN, SIMPLEBOOLEAN) better describe concept learning in general, with these langauges performing better than CNF and DNF on this full set of concepts.

The ability of the quantificational grammars here to predict human responses on all the concepts is substantially worse than the previous analysis on solely Boolean concepts. The $R^2_{response}$ values show that the grammars explain around 66% of the variance, compared to the capability of the Boolean grammars to explain around 88% of the variance for Boolean concepts. This could indicate that the representation languages we consider here do not as accurately model people’s conception of quantificational concepts, or it could be that people give more variable responses on such complex concepts. Interestingly, the ability of the model to predict each concept’s mean difficulty (R^2_{mean}) is actually higher, around 78% of the variance compared with 60-70% on Boolean concepts. This is potentially due to greater and more systematic variance in the concept mean difficulties. As with the Boolean concepts we can plot the model-predicted performance versus the subjects’

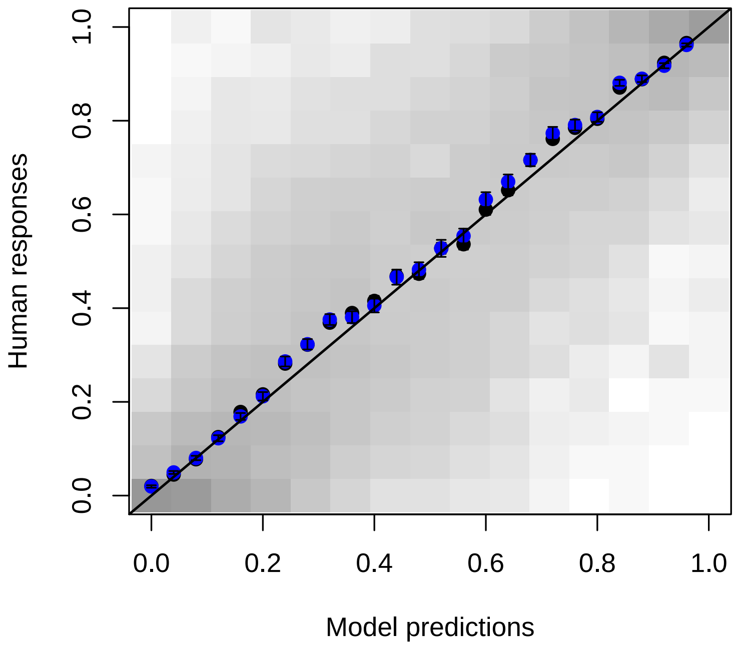


Figure 4-14: Relationship between model predicted probability of responding *true* (x-axis) and participants' probability (y-axis). The gray background represents unbinned data, corresponding to raw responses on each object in each set, list, and concept, of the experiment. Black points are binned training data and blue are binned held-out data.

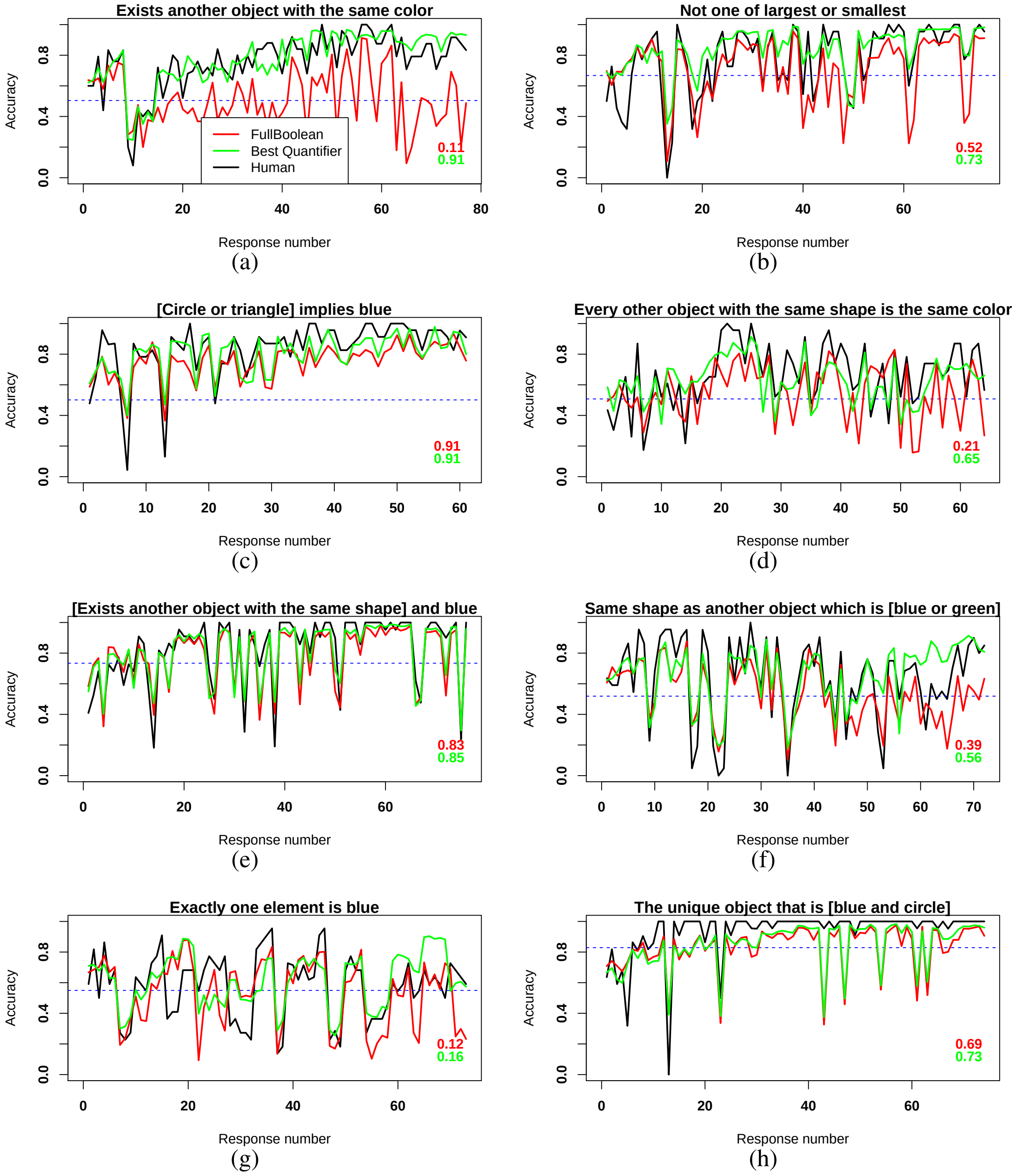
actual performance, collapsing across all concepts. Figure 4-14 shows this relationship and demonstrates the model's ability to predict fine gradations in human response probabilities.

4.8.2 Learning curves

Again, like the Boolean analysis, the quantificational model is capable of predicting detailed patterns of human learning curves. Figure 4-15 shows eight different learning curves: 4-15(a)-4-15(f) show well-fit concepts and 4-15(g)-4-15(h) show relatively poorly fit concepts. Some plots show concepts in which the best grammar with quantifiers and other operations is substantially better than FULLBOOLEAN. In 4-15(a), for instance, FULLBOOLEAN is incapable of expressing *exists another object with the same color*, yet people learn this relatively quickly. Interestingly, on this concept both grammars fit equally well for the first few sets, during which people would not have observed enough data to justify using quantifiers and so therefore would respond with simple Boolean expressions. Once enough data has been seen to cause people to learn the concept (around 20-40 sets), the predictions of the quantifier language and the Boolean one diverge substantially. This is exactly the type of concept for which we find strong quantitative evidence in favor of a representation language that is capable of quantification. However, these types of clear and intuitive cases are relatively uncommon; most of the target quantifier concepts are difficult for people to learn. Importantly, 4-15(c) demonstrates that for the Boolean concepts, both grammars are capable of performing equally well, here with an R^2 of about .91. By adding quantifiers, we do not decrease the model's ability to fit simpler concepts.

Figures 4-15(e) to 4-15(f) show two concepts that are not simple Boolean predicates, but for which the qualitative fits for the Boolean and quantificational grammars are approximately the same. Indeed, most of the concepts studied here are like this, and do not strongly distinguish between these types of grammars. The reason is likely that in these concepts people do not appear to learn the target concept, as evidenced by the fact that they make systematic patterns of mistakes even near the end of the experiment. The fact that FULLBOOLEAN can capture these mistake patterns indicates that people learn Boolean concepts that are similar to the target concepts, but incorrect. That is, people do not learn the targets, instead inferring some simpler Boolean expression. For instance, in 4-15(e) (*[exists an object with the same shape] and blue*) they might learn the concept *blue* since it will often be the case that there is an object of the same shape, and so *blue* provides a good approximation to the target. In 4-15(f) (*same shape as another object which is [blue or green]*), people may eventually learn *same shape as another object*, which can only be expressed as quantifiers. Using people's response patterns to infer what concept they may have learned is an important future direction of this work.

The curves shown in 4-15(g)-4-15(h) are particularly poor fits for the model. Both grammars seem to mischaracterize learning late in 4-15(g), yielding low correlations. Even though the correlation is high for 4-15(h), both grammars predict patterns of mistakes later that are not observed in human subjects. This latter example suggests potential for improvement in how the model handles uniqueness and small cardinalities.



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 4-15: Human (black) versus predicted learning curves according to the best grammar in Figure 4.4 and FULLBOOLEAN.The numbers in the lower right give R^2s between each language's predicted accuracies and humans' observed accuracies. Note the human data for these sequences of data were held-out from training all models.

4.8.3 The inferred grammar

A better understanding of the grammar that is inferred with quantifiers is provided by the probabilistic version for the grammar that is parameterized by D_{**}. As with the Boolean grammar, the relative size of each of these parameters characterizes the probability of using

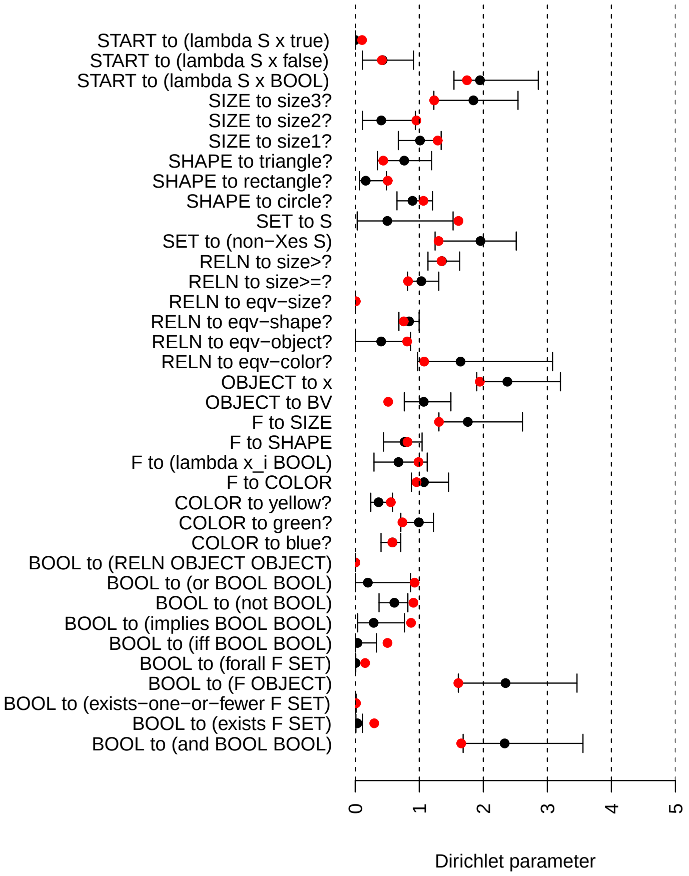


Figure 4-16: Posterior parameters D_{**} found by the inference algorithm for the best grammar in Figure 4.4, including only FOL operations. The red dots are MAP grammar parameters and the intervals are 95% HPD intervals computed using the Chen & Shao (1999) algorithm.

each primitive, and the magnitude of these values inversely relates to the degree that re-use is preferred.

Figure 4-16 shows the D_{**} parameters for the full data set and illustrates similar patterns to Figure 4-13: for instance, *size2?* is a low-probability operation, *F* is about equally likely to expand to *SIZE*, *SHAPE* and *COLOR*. Unlike the Boolean results, this shows that *and* is higher probability than *or*, agreeing with concept learning asymmetries between conjunctive and disjunctive concepts.

Most interestingly, however, are the probabilities assigned to the more complex primitives. For instance, the quantifiers *exists* and *forall* are given relatively low probability, and the low magnitude of all expansions of *BOOL* in general indicate a stronger preference for re-use. The relational terms in LAMBDA-AND-RELATIONAL vary substantially in probability, indicating preferences to compare equality of shapes, colors, and objects, not sizes. As with *implies* in the FULLBOOLEAN above, the primitive *exists-one-or-fewer* is given a MAP probability of 0.013, which is low in the prior, but easily overcome with a few examples, allowing it to improve model fit. Additionally, *SET*s are more likely to be expanded to (*non-Xes S*) than *S*, indicating that quantification tends to occur over all *other* objects in the set; this makes it more natural to express concepts such as *exists another object with the same color* and less natural to express *everything iff there is a triangle in the set*, concepts which people find easy and hard respectively.

4.9 Discussion

These results have begun to elaborate the representational systems that support rule-based concept learning. We have shown that it is possible to take people's learning curves and "work backwards" to infer likely representational systems, both for simple Boolean concepts and for concepts that extend Boolean logic with richer types of quantification. Our method was based on the idea that learners prefer concepts that are representationally simple, and that representation systems give different measures of simplicity even if they have the same expressive capability. This allowed several theory-internal comparisons to determine which LOTs best capture human learning curves. We found that systems with rich sets of Boolean connectives and quantifiers best described human learning. Importantly, we also showed that the same analysis can distinguish across levels of computational ability, building on Boolean logic to include first-order operations. We were able to rule out intuitively implausible bases like the NAND-basis, and provide quantitative evidence supporting more reasonable representations systems. We believe this represents progress towards narrowing down the range of psychologically plausible, rule-based theories.

Our approach gained much of its power by aggregating results across concepts—the probability of a primitive such as *and* should be found by seeing how well grammars performs on all concepts as the probability of *and* is varied. We could imagine simpler "pairwise" comparisons for instance comparing average learning rates on concepts like *red and* *circle* and *red or circle*. The difficulty with this approach is that it seems difficult or impossible to control other variables like the competing hypotheses and the informativeness of the data with respect to the target concept. However, by implementing a full model, we are able to construct a plausible learning theory (Figure 4-9) and test its quantitative predictions. The learning model allowed different representational systems to be "plugged in" without requiring any modification. We then could recover a single score corresponding to how well the best-fitting parameters of the model generalized to unseen human response patterns. This provides a quantitative standard and allows for an effective comparison of representations that cannot be directly observed in behavior. In general, this approach can be extended to any type of representational system or learning task, although we note that in many cases—such as for the top Boolean grammars—it appears difficult to achieve the necessary resolution to distinguish the best representational theories.

It may seem obvious that human conceptual systems involve quantification since we are able to think thoughts with quantifiers—like "Some dog adored Lindsay." But the experiments and analysis here have tested a more subtle point of whether quantification is a natural part of people's *inductive* machinery. It could be the case that our mechanisms of learning operate only over very simple representations like conjunctions of features or continuous spaces. Or it might be that Bayes-optimal statistical reasoning is found only in low-level cognition, for which evolution has had millions of years to optimize computational processes. What we have shown here adds to a growing body of work that demonstrates, either empirically or theoretically, that learning mimics ideal Bayesian inference in rich representational systems (Siskind, 1996; Tenenbaum, 1999; Piantadosi et al., 2008; Kemp et al., 2008b; Kemp & Tenenbaum, 2008; Perfors et al., 2011; Kemp, 2009; O'Donnell et al., 2009; Kemp, Tenenbaum, Niyogi, & Griffiths, 2010; Piantadosi et al., submitted). People have a capacity for logical induction over rules of considerable computational power.

Our results have also provided quantitative evidence in support of rule-based theories. Indeed, even the worst rule-based theories have substantially higher correlations with human responses than alternatives like the exemplar model. The types of effects typically offered in support of non-rule-based approaches may result from "averaging" over rules

(Tenenbaum, 2000), a probabilistic rule-based system (Stuhlmüller, Tenenbaum, & Goodman, 2010), or a model that incorporates both rule-like behavior and exemplar behavior (Nosofsky, 1991). Indeed, subjects' success with many of these rule-based concepts illustrates that rule-learning may be a viable developmental theory both for word meanings (Piantadosi et al., in prep) and syntax (Perfors et al., 2011)—and perhaps more generally for conceptual theories (Kemp et al., 2008b; Ullman et al., 2010). This type of rule induction is not extremely difficult for people or models: here we used a simple Monte-Carlo method to search through concepts, and this was efficient enough to allow comparison of dozens of potential grammars. It is unlikely that people use such simple search methods, but the fact that they are successful on these spaces indicates that learning lambda expressions is not intractably difficult.

The quantitative results in this paper motivate a challenge to the other paradigmatic approaches to cognitive science—such as connectionism (e.g., Rumelhart & McClelland, 1986; Smolensky & Legendre, 2006) or cognitive architectures (e.g., Anderson, 1993; Newell, 1994)—to provide a model that quantitatively out-performs theories based on near-ideal statistics and explicit representations. We believe several design features of our experiment make this especially difficult for, e.g., connectionist models: the concepts learned are rule-like and relational, the set sizes are variable, and participants learn the concepts from relatively little data. The important aspect of this challenge is that we have an explicit and principled quantitative measure: performance on held-out data. Other fields such as natural language processing find standardized data sets critical for comparing approaches. This work provides one standardized data set that we hope will prove useful in refining debates about what types of representations and architectures support the richness of human cognitive capacities.

4.10 Conclusion

At its core, the human mind is a computational system capable of fluidly creating and manipulating concepts. As in our experiment, these concepts are often induced—not deduced—from data and can easily combine with existing other concepts. The question of what type of system supports these abilities is a fundamental one for cognitive science. Here, we have argued that one plausible hypothesis is that learners have a compositional representation system, combined with approximately ideal statistical inference mechanisms. When these two components are put together, we are able to model key phenomena in a massive concept-learning experiment, including patterns of errors, graded responses, and eventual learning of complex, compositional concepts. A compositional representational system models the course of learning and the eventual end-state of our participants, providing a computational theory for how basic logical abilities might be elaborated into complex systems of structured concepts throughout learning and development.

4.11 Appendix

Here we describe several additional models compared in the Results sections:

UNIFORM This model assigns all hypotheses a uniform prior:

$$P(h) \propto 1. \tag{4.26}$$

This prior is also improper and is an interesting baseline that corresponds to no substantive expectations about the form of concepts.

RESPONSE-BIASED This model corresponds to a simple response-biased model which uses labeled data to do inference over the proportion of time the hypothesis is true. This can be interpreted as a special representation language where there are only two possible expressions: one that is always true and one that is always false.

LOGISTIC This model provides another baseline which fits a logistic learning curve within each concept. This model therefore has no interesting representational capacities or predictive abilities, but comparison to it reveals the degree to which LOT models can surpass how a statistician or psychophysicist might model performance in this task.

EXEMPLAR This is an exemplar model on the set-based stimuli. It is difficult to know exactly how exemplar models might be applied to sets of objects, since such models are generally stated in terms of similarity of object features, not similarities of collections of objects. Here, we begin by defining an object-wise distance metric:

$$\begin{aligned} d(x,y) = \quad & \delta_{shape(x)=shape(y)} \cdot W_{shape} \\ & +\delta_{color(x)=color(y)} \cdot W_{color} \\ & +\delta_{size(x)=size(y)} \cdot W_{size}, \end{aligned}$$

where *shape*, *color*, and *size* are functions that map objects to their shapes, colors, and sizes, and W_{shape}, W_{color}, and W_{size} are free parameters. Given two sets, we can then consider all possible ways of aligning their objects. This is necessary because if the next set is similar to a previously observed set, we need to know how objects in the current set correspond to objects in the previous one. Since their orders may change, this can only be accomplished by finding an alignment between the sets. For convenience, let $d^*(s_j, s_k)$ be the total distance according to d of the best alignment of elements of *sets* s_j and s_k. If the sets are different sizes, then some elements may be dropped. Then we define a distance metric on sets by

$$D(s_j, s_k) = abs(|s_j| - |s_k|) \cdot W_{length} + d^*(s_j, s_k). \tag{4.27}$$

Intuitively, this says that sets are penalized W_{length} for differences in cardinality, and then according to the distance of their elements in the best alignment via $d(x, y)$. D is used to define the probability of generalizing labels from a previously labeled s_j to the next set, s_n, according to the best alignment between the two. This *log* probability is proportional to

$$-\beta^{n-j+1} \cdot \log D(s_j, s_n). \tag{4.28}$$

Like the Bayesian models, this includes a power law memory-decay parameter, β.

Chapter 5

Afterword

This work has attempted to show how ideas about induction and representation can be combined into a coherent picture of developmental change. The three papers in this thesis have presented three slightly different versions of what is essentially the same model of inductive learning. This approach formalizes the idea that learners bring a capacity for conceptual combination to the problem of learning, and construct representations of the world much as programmers write programs or scientists develop theories.

The specific representation languages used in these papers are similar but not identical. All include logical operations, simple notions of sets and operations on sets, and the capacity for structurally rich representations by using one structure-building operation: function composition. Some differences in languages resulted from studying different aspects of acquisition at different times in the development of this work. Other differences resulted from the fact that the papers each study distinct types of conceptual systems, ranging from word meanings formalized as relations on sets, to novel rules in propositional and first-order logic, to recursive systems for counting. These different systems used distinct input and output types, with representations in the first paper mapping sets to words, the second mapping two sets to a truth value, and the third mapping sets to subsets. In principle, one could express all of these operations with the same language. Developing and testing such a model is an important direction for future work, though the predictions of a unified model would not be substantially different from those presented if the additional primitives did not create new, more concise ways of expressing any of the target concepts. For instance, including a recursion primitive *L* into the quantifier model would not substantially affect the model's predictions, so long as the target concepts can be expressed more concisely without *L*. Alternatively, it may turn out that a single representation system does not capture the expectations that learners bring to these tasks. For instance, in the concept-learning experiment, adult participants may bring different types of expectations to the task than children learning quantifiers, and these differences may be well captured by distinct representation systems. Indeed real cognition may employ different kinds of structured representational systems for different domains—for instance, in learning linguistic concepts like function-word meanings as compared to learning Boolean concepts. A challenge, then, would be to understand how such distinct systems interface. In general, though, this work has demonstrated the plausibility of learning in these types of systems by showing that developmental patterns and adult generalizations can be captured as rational statistical inferences over the right kind of representations.

As such, this approach provides a compelling middle ground between nativist and empiricist theories of development and language acquisition. It is sometimes tempting to view this work as "even more" nativist than contemporary nativist theories, since rather than building in a few particular representations, we build in an effectively infinite number. There is some truth to this, as what we build is almost certainly more computationally powerful than standard nativist theories. But there is an important difference too: we build in only the *capacity* for representations, not particular representations themselves. Perhaps counterintuitively, the amount of information required to specify an infinite space of concepts (a capacity) can easily be considerably less than that required to specify a handful of particular concepts. The situation is similar to Jorges Luis Borges' *Library of Babel* (Borges, 1941/1970) that consists of all possible books—all possible sequences of characters printed on a page. The library contains essentially no information at all, since its algorithmic complexity (Li & Vitányi, 2008) is close to zero[1] (see also Quine, 1989, pp. 223–225). In the models we presented, all that would need to be, for instance, genetically encoded is the grammar of concepts. Because the grammar is very concise, this amounts

[1] It must contain very little information since its entire content can be conveyed by saying it consists "of all possible sequences of characters."

to "building in" very little—a minimally nativist theory with explicit representation. This theory has appropriate places for structure, early conceptual knowledge, and induction. The results presented pose challenges for theories that lack any of these components. We address each in turn.

First, it appears to be very difficult to capture the results of the adult concept learning experiments with systems that lack structured representations. The exemplar model fared poorly even compared to the *worst* representation languages. Though it may be possible to construct better exemplar models, it is hard to imagine that any model that lacked structural components could more compellingly capture the essence of functions that can operate on sets of arbitrary size or the capacity for linguistic compositionality, as is needed to explain quantifier learning[2]. The pattern of errors subjects made suggests a similar trend: subjects clumped together into systematic patterns of behavior, apparently changing rules in the face of new data. This systematicity and quick revision is more obviously consistent with rule-like representations than graded representations in a continuous space. In the number learning model, structured representations also appeared necessary to explain children's quick developmental change, since learning mechanisms based only on associations would require increasing amounts of evidence to learn the higher, more infrequent, number cardinalities. Indeed, it does not seem possible to capture adult knowledge of natural numbers—their structure and infinite cardinality—without positing some type of generative abstraction like the CP-knower lexicon that the model learns.

Second, the capacity for structure is not sufficient: learners must also have a mechanism to infer "good" structures for explaining their observed data. This is most basically an *inductive* problem, requiring learners to take their observed data and make strong predictions about unseen data. Much of the early work in rule-based concept learning treated concept learning as a deductive sequential decision-making problem (Bruner et al., 1956), or algorithmically as the problem of inferring what were essentially decision-tree represen-

[2] A second advantage of the types of explicit structural representations we provide is that it is intuitively easy to manipulate these representations even lacking concrete examples of sets. I could tell you that "blark A are B" if the the number of elements in $A \cap B$ is a power of 2. You immediately know that if "blark" applies, so do a whole range of other quantifiers—*an even number of*, *at least two*, *not exactly fifteen*, etc. It is difficult to know how such richly integrated knowledge of "blark" could be determined without having seen any examples of its use, and from only hearing a verbal description of its meaning. This is not difficult in principle for rule-like representations.

tations of Boolean concepts (Hunt, Marin, & Stone, 1966). The contribution of the present work has been to more fully develop the idea that concept learning is a statistically rational inference problem, along the lines of Goodman et al. (2008). Induction is philosophically troublesome (Hume, 1748/2000; Goodman, 1955), and squaring the philosophical problem with its psychological realization has been a significant goal for—or perhaps achievement of—modern cognitive science and machine learning (see, e.g., Chater & Vitányi, 2003; Hayes, Heit, & Swendsen, 2010). Like most previous attempts to solve the problem of induction in cognition, this work has argued that simplicity (Chater & Vitányi, 2003) is the key bias that allows for human-like inductive inferences. This was argued for by the fact that the model of Boolean concept learning that includes no simplicity bias performs poorly, and indirectly by the number learning model that shows plausible developmental patterns only when given the right simplicity bias. Many other approaches to learning have failed to formally specify the necessary inductive machinery, including accounts of number learning based on innate counting principles (R. Gelman & Gallistel, 1978) and Carey's bootstrapping account under which children make the CP-transition by analogy (Carey, 2009). Theories of quantifier acquisition that use Gold-style learnability (Clark, 1996) also lack a real statistical inductive approach and generally appear incapable of solving the subset problem using solely positive evidence. The provable success of a rational statistical model in solving these learning problems suggests that well-formulated statistical models provide a good starting point for developmental theories, much in the spirit of rational analysis (Anderson & Milson, 1989; Chater & Oaksford, 1999).

The third aspect of this work was to formalize a place for early cognitive capacities. We have talked about these primitives as though they are innate, but strictly speaking the primitives could be learned very early. The key assumption required is that these operations are "simple" enough for learner that they have a high probability of being used in new conceptual representations. It is easy to imagine why a cognitive system would make use of such primitive functions: as in computer science (see, e.g., Abelson & Sussman, 1996) definition of primitives allows for abstraction, re-use, encapsulation, and compression. This type of encapsulation of more complex operations is even attested in behavioral neuroscience, with, for instance, single neurons in motor cortex coding for complex behaviors (Graziano, 2006). Of course, one might imagine versions of these models that include only the minimal, logically necessary set of cognitive primitives. Just as logicians and mathematics attempt to find minimal sets of axioms, here we might consider attempting to do as much as possible with as few primitives as possible. For instance, "numbers" can be built out of nothing more than the syntax of the lambda calculus, as *Church numerals* (Church, 1932): "one" equal to $\lambda f\ x\ .\ (f\ x)$, "two" equal to $\lambda f\ x.\ (f\ (f\ x))$, etc. In this there are no explicit representations of sets, cardinalities, or set-theoretic functions, so we would not need to posit primitives like *singleton?* and *doubleton?*. There are at least three challenges for this kind of approach. The first is to explain infant abilities in—for instance—manipulating and representing small set cardinalities (e.g., Wynn, 1992), including their competence and their systematic patterns of errors, for instance, tracking three objects in a bucket, but not four (Feigenson & Carey, 2005). A set of functional primitives provides a way to explain these data by positing only small cardinalities are "built in."[3] Second, representations like Church numerals still require a linking function to behavior to specify how to apply these lambda expressions. With our setup, it is natural to assume that primitives like *singleton?* are evaluated directly by perceptual systems. In the case of Church numerals, it is non-obvious how to relate them to perceptual systems in a plausible way. Finally, such minimal systems likely give a "wrong" inductive bias for learners, under the assumption of a preference for simplicity. The recursive form of "counting" in Church numerals is very simple—in fact, just a function composition. So why would children first learn "two," "three," and sometimes "four" before learning to count in Church numerals? The reason provided by the number model is that these representations are simpler if small cardinality primitives are cheap and recursion is costly; it is not clear that a similar bias could be constructed for a basis like Church numerals. These types of considerations equally apply to the quantifier model and the set-function model. Indeed, in the set-function model we showed that representations with minimal sets of primitives (e.g., *NAND*) actually do not capture people's simplicity bias well, providing empirical evidence against this kind of

[3] Although, this is admittedly post-hoc since we posit representational primitives based on the experimental results. What is not post-hoc, though, is the demonstration in our results that use of these primitives as compositional elements in many cases can explain developmental patterns in other tasks like quantifier learning or numerical acquisition.

minimalism.

In sum, these results support a view of cognitive development in which learning consists of applying a compositional statistical model to a set of early conceptual primitives. With this setup we have developed computational and empirical methods for studying the interaction of learning and the language of thought. Further, we have argued that this approach can help bring clarity to several deep issues in the study of cognition, including the cognitive problem of induction, questions of innateness and learnability, and the basic puzzle of how complex representations may arise in cognition.

References

Abelson, H., & Sussman, G. (1996). *Structure and interpretation of computer programs*. Cambridge, MA: MIT Press.

Akaike, H. (1974). A new look at the statistical model identification. *Automatic Control, IEEE Transactions on*, *19*(6), 716–723.

Anderson, J. (1993). *Rules of the mind*. Hillsdale, NJ: Lawrence Erlbaum.

Anderson, J., & Milson, R. (1989). Human memory: An adaptive perspective. *Psychological Review*, *96*(4), 703.

Anderson, J., & Schooler, L. (1991). Reflections of the environment in memory. *Psychological Science*, *2*(6), 396.

Angluin, D. (1987). Learning regular sets from queries and counterexamples. *Information and computation*, *75*(2), 87–106.

Aslin, R., Saffran, J., & Newport, E. (1998). Computation of conditional probability statistics by 8-month-old infants. *Psychological Science*, *9*(4), 321–324.

Babyonyshev, M., Ganger, J., Pesetsky, D., & Wexler, K. (2001). The maturation of grammatical principles: Evidence from Russian unaccusatives. *Linguistic Inquiry*, *32*(1), 1–44.

Barner, D., & Bachrach, A. (2010). Inference and exact numerical representation in early language development. *Cognitive psychology*, *60*(1), 40–62.

Barner, D., Chow, K., & Yang, S. (2009). Finding one's meaning: A test of the relation between quantifiers and integers in language development. *Cognitive psychology*, *58*(2), 195–219.

Barner, D., Thalwitz, D., Wood, J., Yang, S., & Carey, S. (2007). On the relation between the acquisition of singular–plural morpho-syntax and the conceptual distinction between one and more than one. *Developmental Science*, *10*(3), 365–373.

Baroody, A. (1984). Children's difficulties in subtraction: Some causes and questions. *Journal for Research in Mathematics Education*, *15*(3), 203–213.

Barwise, J., & Cooper, R. (1981). Generalized quantifiers and natural language. *Linguistics and philosophy*, *4*(2), 159–219.

Bertolo, S. (2001). *Language acquisition and learnability*. Cambridge, UK: Cambridge University Press.

Berwick, R. (1985). *The acquisition of syntactic knowledge*. Cambridge, MA: MIT Press.

Blackburn, P., & Bos, J. (2005). *Representation and inference for natural language: A first course in computational semantics*. Center for the Study of Language and Information.

Bloom, P., & Wynn, K. (1997). Linguistic cues in the acquisition of number words. *Journal* *of Child Language*, *24*(03), 511–533.

Bod, R. (2009). From exemplar to grammar: A probabilistic analogy-based model of language learning. *Cognitive Science*, *33*(5), 752–793.

Boole, G. (1854). *An investigation of the laws of thought: on which are founded the mathematical theories of logic and probabilities*. London, UK: Walton and Maberly.

Borer, H., & Wexler, K. (1987). The maturation of syntax. In T. Roper & E. Williams (Eds.), *Parameter setting* (Vol. 4, p. 123). Norwell, MA: Kiuwer Academic Publishers.

Borges, J. (1941/1970). The library of babel. In *Labyrinths.* Harmondsworth: Penguin.

Bourne, L. (1966). *Human conceptual behavior*. Allyn and Bacon.

Braine, M. (1971). On two types of models of the internalization of grammars. *The ontogenesis of grammar*, 153–186.

Brown, R., & Hanlon, C. (2004). Derivational complexity and order of acquisition in child speech. *First language acquisition: the essential readings*, 155.

Bruner, J., Goodnow, J., & Austin, G. (1956). *A study of thinking*. New Brunswick, NJ: Transaction Publishers.

Bullock, M., & Gelman, R. (1977). Numerical reasoning in young children: The ordering principle. *Child Development*, *48*(2), 427–434.

Caponigro, I., Pearl, L., Brooks, N., & Barner, D. (2010). On the acquisition of maximality. In *Proceedings of SALT* (Vol. 20, pp. 508–524).

Carey, S. (2009). *The origin of Concepts*. Oxford: Oxford University Press.

Carey, S., & Bartlett, E. (1978). Acquiring a Single New Word. *Papers and Reports on Child Language Development*, *15*, 17–29.

Chater, N., & Oaksford, M. (1999). Ten years of the rational analysis of cognition. *Trends in Cognitive Sciences*, *3*(2), 57–65.

Chater, N., & Vitányi, P. (2003). Simplicity: A unifying principle in cognitive science? *Trends in cognitive sciences*, *7*(1), 19–22.

Chater, N., & Vitányi, P. (2007). Ideal learning'of natural language: Positive results about learning from positive evidence. *Journal of Mathematical Psychology*, *51*(3), 135–163.

Chen, M., & Shao, Q. (1999). Monte Carlo estimation of Bayesian credible and HPD intervals. *Journal of Computational and Graphical Statistics*, *8*(1), 69–92.

Church, A. (1932). A set of postulates for the foundation of logic. *The Annals of Mathematics*, *33*(2), 346–366.

Church, A. (1936). An unsolvable problem of elementary number theory. *American journal of mathematics*, *58*(2), 345–363.

Ciborowski, T., & Cole, M. (1972). A cross-cultural study of conjunctive and disjunctive concept learning. *Child Development*, *43*(3), 774–789.

Clark, R. (1996). Learning first order quantifier denotations: An essay in semantic learnability. *IRCS Technical Report 96-19.*

Clark, R. (2010). *Some computational properties of generalized quantifiers.* (Unpublished manuscript)

Clark, R., & Grossman, M. (2007). Number sense and quantifier interpretation. *Topoi*, *26*(1), 51–62.

Condry, K. F., & Spelke, E. S. (2008). The Development of Language and Abstract Concepts: The Case of Natural Number. *Journal of Experimental Psychology: General*, *137*, 22–38.

Conklin, D., & Witten, I. (1994). Complexity-based induction. *Machine Learning*, *16*(3), 203–225.

Cover, T., & Thomas, J. (2006). *Elements of information theory*. Hoboken, NJ: John Wiley and sons.

Crain, S. (1992). The semantic subset principle in the acquisition of quantification. In *Workshop on the Acquisition of WH-Extraction and Related Work on Quantification, University of Massachusetts, Amherst, MA.*

Crain, S. (1993). Semantic subsetutions. In *Invited paper presented at the Center for Cognitive Science Conference: Early Cognition and the Transition to Language, University of Texas, Austin.*

Crain, S., Ni, W., & Conway, L. (1994). Learning, parsing and modularity. *Perspectives on sentence processing*, 443–467.

Crain, S., & Philip, W. (1993). Global semantic dependencies in child language. In *GLOW Colloquium* (Vol. 16). Lund, Sweden.

Crain, S., & Thornton, R. (2000). *Investigations in Universal Grammar: A guide to experiments on the acquisition of syntax and semantics*. Cambridge, MA: MIT Press.

Crain, S., Thornton, R., Boster, C., Conway, L., Lillo-Martin, D., & Woodams, E. (1996). Quantification without qualification. *Language Acquisition*, *5*(2), 83–153.

Dehaene, S. (1999). *The number sense: How the mind creates mathematics*. Oxford: Oxford University Press.

Dehaene, S., & Mehler, J. (1992). Cross-linguistic regularities in the frequency of number words* 1. *Cognition*, *43*(1), 1–29.

DellaCarpini, M. (2003). Developmental stages in the semantic acquisition of quantification by adult L2 learners of English: A pilot study. In *Proceedings of the 6th Generative Approaches to Second Language Acquisition Conference (GASLA 2002): L* (Vol. 2).

Dowling, W., & Gallier, J. (1984). Linear-time algorithms for testing the satisfiability of propositional Horn formulae. *The Journal of Logic Programming*, *1*(3), 267–284.

Dresher, B. (1999). Charting the learning path: Cues to parameter setting. *Linguistic Inquiry*, *30*(1), 27–67.

Dresher, J., & Elan, B. (1990). A computational learning model for metrical phonology. *Cognition*, *34*(2), 137–195.

Elman, J. (1990). Finding structure in time. *Cognitive science*, *14*(2), 179–211.

Elman, J. (1993). Learning and development in neural networks: the importance of starting small. *Cognition*, *48*(1), 71–99.

Elman, J. (1997). *Rethinking innateness: A connectionist perspective on development*. Cambridge, MA: the MIT Press.

Feigenson, L., & Carey, S. (2005). On the limits of infants' quantification of small object arrays. *Cognition*, *97*(3), 295–313.

Feigenson, L., Dehaene, S., & Spelke, E. (2004). Core systems of number. *Trends in cognitive sciences*, *8*(7), 307–314.

Feldman, J. (2000). Minimization of Boolean complexity in human concept learning. *Nature*, *407*(6804), 630–633.

Feldman, J. (2003a). A catalog of boolean concepts. *Journal of Mathematical Psychology*, *47*(1), 75–89.

Feldman, J. (2003b). Simplicity and complexity in human concept learning. *The General Psychologist*, *38*(1), 9–15.

Feldman, J. (2003c). The simplicity principle in human concept learning. *Current Directions in Psychological Science*, *12*(6), 227.

Fiorin, G. (2010). Meaning and dyslexia: a study on pronouns, aspect, and quantification. *Lot Dissertation Series*, *235*.

Fiser, J., & Aslin, R. (2002). Statistical learning of new visual feature combinations by infants. *Proceedings of the National Academy of Sciences of the United States of America*, *99*(24), 15822.

Florêncio, C. (2002). Learning generalized quantifiers. In *Proceedings of Seventh ESSLLI Student Session.*

Fodor, J. (1975). *The language of thought.* Cambridge, MA: Harvard University Press.

Fodor, J. (1998a). *Concepts: Where Cognitive Science Went Wrong*. Oxford: Oxford University Press.

Fodor, J. (1998b). Unambiguous triggers. *Linguistic Inquiry*, *29*(1), 1–36.

Fodor, J. (2008). *LOT 2: The language of thought revisited.* Oxford: Oxford University Press.

Fodor, J., & Pylyshyn, Z. (1988). Connectionism and cognitive architecture: a critical analysis, Connections and symbols. *A Cognition Special Issue, S. Pinker and J. Mehler (eds.)*, 3–71.

Frank, M., Everett, D., Fedorenko, E., & Gibson, E. (2008). Number as a cognitive technology: Evidence from Pirahã language and cognition. *Cognition*, *108*(3), 819–824.

Frank, M., Goodman, N., & Tenenbaum, J. (2007a). A Bayesian framework for cross-situational word learning. *Advances in neural information processing systems*, *20*.

Frank, M., Goodman, N. D., & Tenenbaum, J. B. (2007b). A Bayesian framework for cross-situational word-learning. In *Advances in Neural Information Processing Systems 20.*

Fuson, K. (1984). More complexities in subtraction. *Journal for Research in Mathematics Education*, *15*(3), 214–225.

Fuson, K. (1988). *Children's counting and concepts of number*. New York: Springer-Verlag.

Gallistel, C. (2007). Commentary on Le Corre & Carey. *Cognition*, *105*, 439–445.

Gallistel, C., & Gelman, R. (1992). Preverbal and verbal counting and computation. *Cognition*, *44*, 43–74.

Gasser, M., & Smith, L. (1998). Learning nouns and adjectives: A connectionist account. *Language and cognitive processes*, *13*(2), 269–306.

Gelman, A., Carlin, J., Stern, H., & Rubin, D. (2004). *Bayesian data analysis.* Boca Raton, FL: CRC press.

Gelman, R. (1993). A rational-constructivist account of early learning about numbers and objects. *The psychology of learning and motivation. Advances in research theory*, *30*, 61–96.

Gelman, R., & Butterworth, B. (2005). Number and language: how are they related? *Trends in Cognitive Sciences*, *9*(1), 6–10.

Gelman, R., & Gallistel, C. (1978). *The Child's Understanding of Number*. Cambridge, MA: Harvard University Press.

Gelman, R., & Meck, B. (1992). Early principles aid initial but not later conceptions of number.

Geman, S., & Geman, D. (1984). Gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *6*(2), 721–741.

Gibson, E., & Wexler, K. (1994). Triggers. *Linguistic Inquiry*, *25*(3), 407–454.

Gierasimczuk, N. (2007). The problem of learning the semantics of quantifiers. *Logic, Language, and Computation*, 117–126.

Gold, E. (1967). Language identification in the limit. *Information and control*, *10*(5), 447–474.

Gomez, R., & Gerken, L. (1999). Artificial grammar learning by 1-year-olds leads to specific and abstract knowledge. *Cognition*, *70*(2), 109–135.

Goodman, N. (1955). *Fact, fiction and forecast*. Cambridge, MA: Harvard University Press.

Goodman, N., Tenenbaum, J., Feldman, J., & Griffiths, T. (2008). A Rational Analysis of Rule-Based Concept Learning. *Cognitive Science*, *32*(1), 108–154.

Goodman, N., Ullman, T., & Tenenbaum, J. (2009). Learning a theory of causality. In *Proceedings of the 31st annual conference of the cognitive science society* (pp. 2188–2193).

Gordon, P. (2004). Numerical cognition without words: Evidence from Amazonia. *Science*, *306*(5695), 496.

Graziano, M. (2006). The organization of behavioral repertoire in motor cortex. *Annu. Rev. Neurosci.*, *29*, 105–134.

Grice, H. (1975). Logic and conversation. In P. Cole & J. L. Morgan (Eds.), *Syntax and Semantics, Vol. 3, Speech Acts* (pp. 41–58). New York: Academic Press.

Griffiths, T., Chater, N., Kemp, C., Perfors, A., & Tenenbaum, J. (2010). Probabilistic models of cognition: exploring representations and inductive biases. *Trends Cogn. Sci*, *14*(10.1016).

Gualmini, A., Meroni, L., & Crain, S. (2003). An asymmetric universal in child language. *Arbeitspapier Nr. 114*, 136.

Gualmini, A., & Schwarz, B. (2009). Solving learnability problems in the acquisition of semantics. *Journal of Semantics*.

Hackl, M. (2009). On the grammar and processing of proportional quantifiers: most versus more than half. *Natural Language Semantics*, *17*(1), 63–98.

Hale, M., & Reiss, C. (2003). The Subset Principle in phonology: why the tabula can't be rasa. *Journal of Linguistics*, *39*(02), 219–244.

Hartnett, P. (1991). *The development of mathematical insight: From one, two, three to infinity*. Ph.D. thesis, University of Pennsylvania.

Hartnett, P., & Gelman, R. (1998). Early understandings of numbers: paths or barriers to the construction of new understandings? *Learning and instruction*, *8*(4), 341–374.

Hastings, W. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, *57*(1), 97.

Hauser, M. D., Chomsky, N., & Fitch, W. T. (2002, 11). The Faculty of Language: What Is It, Who Has It, and How Did It Evolve? *Science*, *298*, 1569–1579.

Hayes, B., Heit, E., & Swendsen, H. (2010). Inductive reasoning. *Wiley Interdisciplinary Reviews: Cognitive Science*, *1*(2), 278–292.

Haygood, R. (1963). *Rule and attribute learning as aspects of conceptual behavior*. Ph.D. thesis, University of Utah.

Heibeck, T., & Markman, E. (1987). Word learning in children: An examination of fast mapping. *Child Development*, *58*(4), 1021–1034.

Heim, I. (1991). Artikel und definitheit. *Semantik: Ein internationales Handbuch der zeitgenössischen Forschung*, 487–535.

Heim, I., & Kratzer, A. (1998). *Semantics in generative grammar*. Malden, MA: Wiley-Blackwell.

Hindley, J., & Seldin, J. (1986). *Introduction to combinators and λ-calculus*. Cambridge, UK: Press Syndicate of the University of Cambridge.

Hodges, W. (1993). Logical features of Horn clauses. *Handbook of Logic in Artificial Intelligence and Logic Programming, Logical Foundations*, *1*, 449–503.

Hoeting, J., Madigan, D., Raftery, A., & Volinsky, C. (1999). Bayesian Model Averaging. *Statistical Science*, *14*, 382–401.

Horn, A. (1951). On sentences which are true of direct unions of algebras. *Journal of symbolic logic*, *16*(1), 14–21.

Huang, Y., Snedeker, J., & Spelke, E. (2004). What exactly do numbers mean. In *Proceedings of the 26th Annual Conference of the Cognitive Science Society* (p. 1570).

Hume, D. (1748/2000). *An enquiry concerning human understanding.* New Jersey: Prentice Hall.

Hunt, E., Marin, J., & Stone, P. (1966). *Experiments in induction.* Oxford: Academic Press.

Hunter, T., & Conroy, A. (2009). Children's restrictions on the meanings of novel determiners: An investigation of conservativity. In *BUCLD* (Vol. 33, pp. 245–255).

Inhelder, B., & Piaget, J. (1969). *The early growth of logic in the child* (Vol. 21). Routledge.

Jeffreys, S. (1998). *Theory of probability*. Oxford: Oxford University Press.

Johnson, K. (2004). Gold's theorem and cognitive science. *Philosophy of Science*, *71*(4), 571–592.

Johnson, M., Griffiths, T., & Goldwater, S. (2007). Bayesian inference for PCFGs via Markov chain Monte Carlo. In *Proc. of NAACL-HLT* (pp. 139–146).

Kang, H. (1999). Quantifier spreading by English and Korean children. *Ms., University College, London.*

Karmiloff-Smith, A. (1981). *A functional approach to child language: A study of determiners and reference* (Vol. 24). Cambridge, UK: Cambridge University Press.

Karttunen, L., & Peters, S. (1979). Conventional implicature. *Syntax and semantics*, *11*, 1–56.

Katz, Y., Goodman, N., Kersting, K., Kemp, C., & Tenenbaum, J. (2008). Modeling semantic cognition as logical dimensionality reduction. In *Proceedings of Thirtieth Annual Meeting of the Cognitive Science Society.*

Keenan, E., & Stavi, J. (1986). A semantic characterization of natural language determiners. *Linguistics and philosophy*, *9*(3), 253–326.

Keenan, E., & Westerståhl, D. (1997). Generalized quantifiers in linguistics and logic. *Handbook of logic and language*, 837–893.

Kemp, C. (2009). Quantification and the language of thought. *Advances in neural information processing systems*, *22*.

Kemp, C., Goodman, N., & Tenenbaum, J. (2008a). Learning and using relational theories. *Advances in neural information processing systems*, *20*, 753–760.

Kemp, C., Goodman, N., & Tenenbaum, J. (2008b). Theory acquisition and the language of thought. In *Proceedings of thirtieth annual meeting of the cognitive science society.*

Kemp, C., & Tenenbaum, J. (2008). The discovery of structural form. *Proceedings of the National Academy of Sciences*, *105*(31), 10687.

Kemp, C., Tenenbaum, J., Niyogi, S., & Griffiths, T. (2010). A probabilistic model of theory formation. *Cognition*, *114*(2), 165–196.

Kidd, C., Piantadosi, S. T., & Aslin, R. N. (under review). The Goldilocks Effect: Human infants allocate attention to events that are neither too simple nor too complex.

Kirkham, N., Slemmer, J., & Johnson, S. (2002). Visual statistical learning in infancy: Evidence for a domain general learning mechanism. *Cognition*, *83*(2), B35–B42.

Ko, H., Ionin, T., & Wexler, K. (2006). Adult L2-learners lack the maximality presupposition, too. In *Proceedings of the Inaugural Conference on Generative Approaches to Language Acquisition–North America, Honolulu, HI* (pp. 171–182).

Ko, H., Perovic, A., Ionin, T., & Wexler, K. (2008). Semantic universals and variation in L2 article choice. In *Proceedings of the 9th Generative Approaches to Second Language Acquisition Conference (GASLA 9)* (pp. 118–129).

Kohl, K. (1999). *An analysis of finite parameter learning in linguistic spaces*. Ph.D. thesis, Massachusetts Institute of Technology.

Kouider, S., Halberda, J., Wood, J., & Carey, S. (2006). Acquisition of English Number Marking: The Singular—Plural Distinction. *Language Learning and Development*, *2*(1), 1–25.

Koza, J. (1992). *Genetic programming: on the programming of computers by means of natural selection*. Cambridge, MA: MIT Press.

Kwiatkowski, T., Goldwater, S., & Steedman, M. (2009). Computational Grammar Acquisition from CHILDES data using a Probabilistic Parsing Model. In *Psychocomputational Models of Human Language Acquisition (PsychoCompLA).*

Langford, J., & Holmes, V. (1979). Syntactic presupposition in sentence comprehension. *Cognition*, *7*(4), 363–383.

Laurence, S., & Margolis, E. (2002). Radical concept nativism. *Cognition*, *86*, 25–55.

Le Corre, M., & Carey, S. (2007). One, two, three, four, nothing more: An investigation of the conceptual sources of the verbal counting principles. *Cognition*, *105*, 395–438.

Lee, M., & Sarnecka, B. (2010a). A Model of Knower-Level Behavior in Number Concept Development. *Cognitive Science*, *34*(1), 51–67.

Lee, M., & Sarnecka, B. (2010b). Number-knower levels in young children: Insights from Bayesian modeling. *Cognition*, *120*(3), 391–402.

Leslie, A. M., Gelman, R., & Gallistel, C. (2008). The generative basis of natural number concepts. *Trends in Cognitive Sciences*, *12*, 213–218.

Levine, M. (1966). Hypothesis behavior by humans during discrimination learning. *Journal of Experimental Psychology*, *71*(3), 331.

Levine, S., Suriyakham, L., Rowe, M., Huttenlocher, J., & Gunderson, E. (2010). What Counts in the Development of Young Children's Number Knowledge? *Developmental Psychology*, *46*(5), 1309–1319.

Li, M., & Vitányi, P. (2008). *An introduction to Kolmogorov complexity and its applications*. New York: Springer-Verlag.

Liang, P., Jordan, M., & Klein, D. (2009). Learning semantic correspondences with less supervision. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP* (Vol. 1, pp. 91–99).

Liang, P., Jordan, M., & Klein, D. (2010). Learning Programs: A Hierarchical Bayesian Approach. In *Proceedings of the 27th International Conference on Machine Learning.*

Liang, P., Jordan, M., & Klein, D. (2011). Learning dependency-based compositional semantics. In *Proceedings Association for Computational Linguistics (ACL).*

Lipton, J., & Spelke, E. (2006). Preschool children master the logic of number word meanings. *Cognition*, *98*(3), B57–B66.

Ludlow, P., & Neale, S. (2008). Descriptions. *The Blackwell Guide to the Philosophy of Language*, 288–313.

MacKay, D. (2003). *Information theory, inference, and learning algorithms*. Cambridge, UK: Cambridge University Press.

MacWhinney, B. (2000). *The CHILDES project: Tools for analyzing talk.* Lawrence Erlbaum Hillsdale, New Jersey.

Madigan, D., & Raftery, A. (1994). Model selection and accounting for model uncertainty in graphical models using Occam's window. *J. Amer. Statist. Assoc.*, *89*, 1535–1546.

Makowsky, J. (1987). Why Horn formulas matter in computer science: Initial structures and generic examples. *Journal of Computer and System Sciences*, *34*(2-3), 266–292.

Manning, C., & Schütze, H. (1999). *Foundations of statistical natural language processing* (Vol. 59). Cambridge, MA: MIT Press.

Maratsos, M. (1974). Preschool children's use of definite and indefinite articles. *Child Development*, *45*(2), 446–455.

Maratsos, M. (1976). *The use of definite and indefinite reference in young children: An experimental study of semantic acquisition.* Cambridge, UK: Cambridge University Press.

Marcus, G. (1993). Negative evidence in language acquisition. *Cognition*, *46*(1), 53–85.

Marcus, G., Vijayan, S., Bandi Rao, S., & Vishton, P. (1999). Rule learning by seven-month-old infants. *Science*, *283*(5398), 77.

Margolis, E., & Laurence, S. (2008). How to learn the natural numbers: inductive inference and the acquisition of number concepts. *Cognition*, *106*(2), 924–39.

Marr, D. (1982). *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information.* W.H. Freeman & Company.

McKinsey, J. (1943). The decision problem for some classes of sentences without quantifiers. *Journal of Symbolic Logic*, *8*(2), 61–76.

McMillan, C., Clark, R., Moore, P., Devita, C., & Grossman, M. (2005). Neural basis for generalized quantifier comprehension. *Neuropsychologia*, *43*(12), 1729–1737.

McMillan, C., Clark, R., Moore, P., & Grossman, M. (2006). Quantifier comprehension in corticobasal degeneration. *Brain and Cognition*, *62*(3), 250–260.

Medin, D., & Schaffer, M. (1978). Context theory of classification learning. *Psychological review*, *85*(3), 207–238.

Meroni, L., Gualmini, A., & Crain, S. (2000). A conservative approach to quantification in child language. In *Proceedings of the 24 th Annual Penn Linguistics Colloquium* (pp. 171–182).

Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., Teller, E., et al. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, *21*(6), 1087.

Modyanova, N., & Wexler, K. (2007). Semantic and pragmatic language development: Children know 'that' better. In *Proceedings of the 2nd conference on generative approaches to language acquisition–north america (galana 2)* (pp. 297–308).

Modyanova, N., & Wexler, K. (2008). Maximal trouble in free relatives. In *Proceedings of BUCLD* (Vol. 32, pp. 287–298).

Montague, R. (1973). The Proper Treatment of Quantification in Ordinary English. *Formal Semantics*, 17–34.

Mostowski, A. (1957). On a generalization of quantifiers. *Fundamenta mathematicae*, *44*(1), 12–36.

Mostowski, M. (1998). Computational semantics for monadic quantifiers. *Journal of Applied Nonclassical Logics*, *8*, 107–122.

Munn, A., Miller, K., & Schmitt, C. (2006). Maximality and plurality in children's interpretation of definites. In *Proceedings of the 30th annual boston university conference on language development (bucld 30)* (pp. 377–387).

Musolino, J. (2006). On the semantics of the Subset Principle. *Language Learning and Development*, *2*(3), 195–218.

Musolino, J., Crain, S., & Thornton, R. (2000). Navigating negative quantificational space. *Linguistics*, *38*(1), 1–32.

Neisser, U., & Weene, P. (1962). Hierarchies in concept attainment. *Journal of Experimental Psychology*, *64*(6), 640.

Newell, A. (1994). *Unified theories of cognition*. Cambridge, MA: Harvard University Press.

Niyogi, P., & Berwick, R. (1996). A language learning model for finite parameter spaces. *Cognition*, *61*(1-2), 161–193.

Nosofsky, R. (1984). Choice, similarity, and the context theory of classification. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *10*(1), 104–114.

Nosofsky, R. (1991). Typicality in logically defined categories: Exemplar-similarity versus rule instantiation. *Memory & Cognition*, *19*(2), 131–150.

Nosofsky, R., Palmeri, T., & McKinley, S. (1994). Rule-plus-exception model of classification learning. *Psychological review*, *101*(1), 53.

O'Donnell, T., Tenenbaum, J., & Goodman, N. (2009). *Fragment Grammars: Exploring*

Computation and Reuse in Language (Tech. Rep.). MIT Computer Science and Artificial Intelligence Laboratory Technical Report Series, MIT-CSAIL-TR-2009-013.

Osherson, D., Stob, M., & Weinstein, S. (1984). Learning theory and natural language. *Cognition*, *17*(1), 1–28.

Papafragou, A., & Musolino, J. (2003). Scalar implicatures: Experiments at the semantics-pragmatics interface. *Cognition*, *86*(3), 253–282.

Pearl, J. (1998). Graphical models for probabilistic and causal reasoning. *Handbook of Defeasible Reasoning and Uncertainty Management Systems: Quantified representation of uncertainty and imprecision*, *1*, 367.

Perfors, A., Tenenbaum, J., & Regier, T. (2011). The learnability of abstract syntactic principles. *Cognition*, *118*.

Philip, W. (1991). Spreading in the Acquisition of Universal Quantifiers. *West Coast Conference on Formal Linguistics*, *10*, 359–373.

Philip, W. (1992). Distributivity in the Acquisition of Universal Quantifiers. *Proceedings of the 2nd Conference on Semantic and Linguistic Theory, Ohio State Working Papers in Linguistics*, *40*, 327–346.

Philip, W. (1995). *Event Quantification in the Acquisition of Universal Quantification*. Ph.D. thesis, University of Massachusetts, Amherst.

Philip, W. (1998). The wide scope interpretation of postverbal quantifier subjects: QR in the early grammar of Spanish. *Proceedings of GALA 1997*.

Philip, W. (2003). Specific indefinites & quantifier scope for children acquiring Dutch and Chinese. In *Meeting of the Linguistic Society of the Netherlands. In V. van Geenhoven (ed.), Semantics MeetsAcquisition.* Dordrecht, The Netherlands: Kluwer.

Piaget, J. (1955). *The language and thought of the child.* New York: Routledge and Kegan Paul.

Piantadosi, S., Goodman, N., Ellis, B., & Tenenbaum, J. (2008). A Bayesian model of the acquisition of compositional semantics. In *Proceedings of the Thirtieth Annual Conference of the Cognitive Science Society.*

Piantadosi, S., Tenenbaum, J., & Goodman, N. (2009). Beyond Boolean Logic: Exploring Representation Languages for Learning Complex Concepts.

Piantadosi, S., Tenenbaum, J., & Goodman, N. (in prep). Quantifiers and the learnability of language.

Piantadosi, S., Tenenbaum, J., & Goodman, N. (submitted). Bootstrapping in a language of thought.

Pica, P., Lemer, C., Izard, V., & Dehaene, S. (2004). Exact and approximate arithmetic in an Amazonian indigene group. *Science*, *306*(5695), 499.

Pietroski, P., Lidz, J., Hunter, T., & Halberda, J. (2009). The Meaning of 'Most': Semantics, Numerosity and Psychology. *Mind & Language*, *24*(5), 554–585.

Prinz, J. (2004). *Furnishing the mind: Concepts and their perceptual basis*. Cambridge, MA: MIT Press.

Pullum, G., & Scholz, B. (2002). Empirical assessment of stimulus poverty arguments. *The Linguistic Review*, *18*(1-2), 9–50.

Quine, W. (1989). *Quiddities: An intermittently philosophical dictionary*. Belknap Press.

Rakhlin, N. (2007). A new pragmatic account of quantifier-spreading. *Nanzan Linguistics*, *3*, 239–282.

Rips, L., Asmuth, J., & Bloomfield, A. (2006). Giving the boot to the bootstrap: How not to learn the natural numbers. *Cognition*, *101*, 51–60.

Rips, L., Asmuth, J., & Bloomfield, A. (2008). Do children learn the integers by induction? *Cognition*, *106*, 940–951.

Rips, L., Bloomfield, A., & Asmuth, J. (2008). From numerical concepts to concepts of number. *Behavioral and Brain Sciences*, *31*, 623–642.

Roeper, T., Strauss, U., & Pearson, B. (2004). The acquisition path of quantifiers: Two kinds of spreading. *Current Issues in Language Acquisition, UMOP*, *34*.

Rosch, E., & Mervis, C. (1975). Family resemblances: Studies in the internal structure of categories. *Cognitive psychology*, *7*(4), 573–605.

Rumelhart, D., & McClelland, J. (1986). *Parallel distributed processing*. Cambridge, MA: MIT Press.

Rumelhart, D., & McClelland, J. (1987). Learning the past tenses of English verbs: Implicit rules or parallel distributed processing. *Mechanisms of language acquisition*, 195–248.

Russell, B. (1905). On denoting. *Mind*, *14*(56), 479–493.

Russell, B. (1957). Mr. Strawson on referring. *Mind*, *66*(263), 385.

Russell, S., & Norvig, P. (2009). *Artificial intelligence: a modern approach*. Prentice hall.

Saffran, J., Aslin, R., & Newport, E. (1996). Statistical learning by 8-month-old infants. *Science*, *274*(5294), 1926.

Saffran, J., Johnson, E., Aslin, R., & Newport, E. (1999). Statistical learning of tone sequences by human infants and adults. *Cognition*, *70*(1), 27–52.

Sakas, W., & Fodor, J. (2001). The structural triggers learner. *Language acquisition and learnability*, 172–233.

Sarnecka, B., & Carey, S. (2008). How counting represents number: What children must learn and when they learn it. *Cognition*, *108*(3), 662–674.

Sarnecka, B., & Gelman, S. (2004). Six does not just mean a lot: Preschoolers see number words as specific. *Cognition*, *92*(3), 329–352.

Sarnecka, B., & Lee, M. (2009). Levels of number knowledge during early childhood. *Journal of Experimental Child Psychology*, *103*, 325–337.

Sauerland, U. (2003). Implicated presuppositions. In *Proceedings of the conference on Polarity, Scalar Phenomena, Implicatures. University of Milano Bicocca.*

Schlenker, P. (2006). Maximize presupposition and Gricean reasoning. *Manuscript, UCLA and Institute Jean-Nicod, Paris.*

Schwarz, G. (1978). Estimating the dimension of a model. *The annals of statistics*, *6*(2), 461–464.

Seymour, H., Roeper, T., & De Villiers, J. (2003). *Diagnostic evaluation of language variation*. Thieme Medical Publishers, NY.

Shafto, P., & Goodman, N. (2008). Teaching games: Statistical sampling assumptions for learning in pedagogical situations. In *Proceedings of the thirtieth annual conference of the cognitive science society.*

Sheffer, H. (1904). A set of five independent postulates for Boolean algebras, with application to logical constants. *Transactions*, *5*, 288–309.

Shepard, R., Hovland, C., & Jenkins, H. (1961). Learning and memorization of classifications. *Psychological monographs*.

Singh, R. (2009). Maximize Presupposition! and local contexts. *Natural Language Semantics*, 1–20.

Siskind, J. (1996). A Computational Study of Cross-Situational Techniques for Learning Word-to-Meaning Mappings. *Cognition*, *61*, 31–91.

Smith, B. J. (2007). boa: An R Package for MCMC Output Convergence Assessment and Posterior Inference. *Journal of Statistical Software*, *21*(11), 1–37.

Smolensky, P. (1996). *The initial state and 'richness of the base' in Optimality Theory* (Tech. Rep.). Johns Hopkins University, Department of Cognitive Science. JHU-CogSci-96-4.

Smolensky, P., & Legendre, G. (2006). *The Harmonic Mind.* Cambridge, MA: MIT Press.

Solomonoff, R. (1978). Complexity-based induction systems: comparisons and convergence theorems. *IEEE Transactions on Information Theory*, *24*(4), 422–432.

Spelke, E. (2003). What makes us smart? Core knowledge and natural language. In D. Gentner & S. Goldin-Meadow (Eds.), *Language in Mind.* Cambridge, MA: MIT Press.

Spelke, E. (2004). Core Knowledge. In N. Kanwisher & J. Duncan (Eds.), *Attention and perforamnce, vol. 20: Functional neuroimaging of visual cognition.* Oxford: Oxford University Press.

Spelke, E., & Kinzler, K. (2007). Core knowledge. *Developmental Science*, *10*(1), 89–96.

Steedman, M. (2000). *The syntactic process* (Vol. 131). Cambridge, MA: MIT Press.

Strawson, P. (1950). On referring. *Mind*, *59*(235), 320–344.

Stuhlmüller, A., Tenenbaum, J. B., & Goodman, N. D. (2010). Learning Structured Generative Concepts. In *Proceedings of thirty-second annual meeting of the cognitive science society.*

Szymanik, J., & Zajenkowski, M. (2010). Comprehension of simple quantifiers: empirical evaluation of a computational model. *Cognitive Science*, *34*(3), 521–532.

Takahashi, M. (1991). *Children's interpretation of sentences containing every*. Amherst, MA: GLSA.

Téglás, E., Vul, E., Girotto, V., Gonzalez, M., Tenenbaum, J. B., & Bonatti, L. L. (2011). Pure Reasoning in 12-Month-Old Infants as Probabilistic Inference. *Science*, *27*(332), 1054–1059.

Tenenbaum, J. (1999). *A Bayesian Framework for Concept Learning*. Ph.D. thesis, Massachusetts Institute of Technology.

Tenenbaum, J. (2000). Rules and similarity in concept learning. *Advances in neural information processing systems*, *12*, 59–65.

Tiede, H. (1999). Identifiability in the limit of context-free generalized quantifiers. *Journal of Language and Computation*, *1*(1), 93–102.

Troiani, V., Peelle, J., Clark, R., & Grossman, M. (2009). Is it logical to count on quantifiers? Dissociable neural networks underlying numerical and logical quantifiers. *Neuropsychologia*, *47*(1), 104–111.

Ullman, T., Goodman, N., & Tenenbaum, J. (2010). Theory Acquisition as Stochastic Search. In *Proceedings of thirty second annual meeting of the cognitive science society.*

van Benthem, J. (1984). Semantic automata. In J. Groenendijk, D. d. Jongh, & M. Stokhof

(Eds.), *Studies in discourse representation theory and the theory of generalized quantifiers.* Dordrecht, The Netherlands: Foris Publications Holland.

van Benthem, J. (1986). *Essays in logical semantics*. Dordrecht, The Netherlands: Reidel.

Vogt, P., & Smith, A. (2005). Learning colour words is slow: A cross-situational learning account. *Behavioral and Brain Sciences*, *28*(04), 509–510.

Von Fintel, K. (2004). Would you believe it? The king of France is back! Presuppositions and truth-value intuitions. *Descriptions and beyond*, 315–341.

Warden, D. (1974). *An experimental investigation into the child's developing use of definite and indefinite referential speech*. Ph.D. thesis, University of London.

Warden, D. (1976). The influence of context on children's use of identifying expressions and references. *British Journal of Psychology*.

Wexler, K. (2011). Maximal trouble: cues don't explain learning. In E. Gibson & N. Pearlmutter (Eds.), *The Processing and Acquisition of Reference.* Cambridge, MA: MIT Press.

Wexler, K., & Culicover, P. (1983). *Formal principles of language acquisition*. Cambridge, MA: MIT Press.

Wexler, K., & Manzini, M. (1987). Parameters and learnability in binding theory. *Parameter setting*, 41–76.

Wynn, K. (1990). Children's understanding of counting. *Cognition*, *36*, 155–193.

Wynn, K. (1992). Children's Acquisition of the Number Words and the Counting System. *Cognitive Psychology*, *24*, 220–251.

Xu, F., & Denison, S. (2009). Statistical inference and sensitivity to sampling in 11-month-old infants. *Cognition*, *112*(1), 97–104.

Xu, F., & Garcia, V. (2008). Intuitive statistics by 8-month-old infants. *Proceedings of the National Academy of Sciences*, *105*(13), 5012.

Xu, F., & Tenenbaum, J. (2007). Word learning as Bayesian inference. *Psychological Review*, *114*(2), 245–272.

Yang, C. (2002). *Knowledge and learning in natural language*. Oxford: Oxford University Press.

Yu, C., & Ballard, D. (2007). A unified model of early word learning: Integrating statistical and social cues. *Neurocomputing*, *70*(13-15), 2149–2165.

Yu, C., & Smith, L. (2007). Rapid word learning under uncertainty via cross-situational statistics. *Psychological Science*, *18*(5), 414.

Zettlemoyer, L. S., & Collins, M. (2005). Learning to Map Sentences to Logical Form: Structured Classification with Probabilistic Categorial Grammars. In *UAI* (pp. 658–666).

Zettlemoyer, L. S., & Collins, M. (2007). Online Learning of Relaxed CCG Grammars for Parsing to Logical Form. In *EMNLP-CoNLL* (pp. 678–687).