Title
Cognitive Primitives and Bayesian Number Word Learning

Journal
Proceedings of the Annual Meeting of the Cognitive Science Society, 45(45)

Authors
Riley, Sean
Savelson, Zachary M.
Singh, Raj

Publication Date
2023

Peer reviewed

Cognitive Primitives and Bayesian Number Word Learning

Sean N. Riley, Zach Savelson, & Raj Singh
Department of Cognitive Science
Carleton University
Ottawa, Canada
{sean.riley, zach.savelson, raj.singh}@carleton.ca

Abstract

We use the computational Bayesian learning model in Piantadosi, Tenenbaum, and Goodman (2012) to explore how different combinations of cognitive primitives and frequency distributions affect the learning of natural numbers. We find that the model converges on the natural numbers through attested developmental stages only under very restricted sets of primitives and frequency distributions. Assuming the size principle familiar from Bayesian approaches to inductive generalization, it would be natural to conclude that there are sharp constraints on the primitives out of which humans build natural numbers, some of which we hope to elucidate below.

Keywords: natural numbers; primitives; learning; Bayesian reasoning; natural language semantics

Introduction

This paper is concerned with characterizing the remarkable human capacity to acquire the natural numbers. This knowledge in the adult state can be quite complex. Within linguistic syntax, numerals appear to belong to several categories, including adjectives (e.g., *the* {**happy/three**} *students arrived*; *the students were* {**happy/three**}), quantificational determiners (e.g., {**some/three**} *students arrived*), and proper names (e.g., *the president of Company X is* **Sam**/*the number of students is* **three**). We also come to possess knowledge of various complex properties of the numbers, ranging from their deep and exotic properties proved in pure mathematics to their applications in physics and engineering. Most fundamentally, knowledge of numbers entails knowledge of a discrete infinity of objects that are abstract and yet also allow you to do things in the real world, such as count objects, worry about how much debt you have accumulated, and so on. What does the child come born with, and what do they need to experience, in order to converge on the natural numbers as well as the symbols that denote them?

Children typically learn the count sequence $1,2,3\ldots$ before they actually know what numerals *mean* (e.g., Fuson & Fuson, 1988). They eventually learn to assign numerals to the associated number concepts (the semantics of the expressions), typically passing through a set of *knower* levels (e.g., Wynn, 1992). They start as *one-knowers*: they know that numeral *one* picks out singleton sets but are unable to systematically identify higher numerosities. They then become *two-knowers*: they associate *one* with singleton sets and *two* with two-membered sets (doubletons), but fail at higher numerosities. They then become *three-knowers*: they associate *one* with singleton sets, *two* with doubtleton sets, and *three* with tripleton sets, but fail at higher numbers. Then, all of a sudden, they transition from these so-called *subset-knower* stages to become full-fledged *cardinal-principle-knowers* (hf. *CP-knowers*): they know that each number has a unique successor, and properly associate numerals with their corresponding number.[1]

Piantadosi et al. (2012) present a Bayesian learning algorithm that provides a computational implementation of Carey (2009)'s *bootstrapping* model of number learning. Under Carey's model, the child begins with the capacity to represent and count small sets (up to around three), and they transition to CP-knowers by noticing that as they transition through subset-knower stages the next numeral denotes a set that has one more element than the set denoted by the previous numeral. As we discuss in more detail in the next section, Piantadosi et al. (2012) formalize Carey's insight by developing a learner that creates a hypothesis space of functions in the lambda-calculus that map sets to a numeral expressing their cardinality. For example, the function corresponding to the concept 'three' would map a set like $\{a,b,c\}$ to the word *three*.[2] The functions are built out of an assumed set of logical primitives, operations on the count sequence, and a recursive operator; of particular interest are primitives *singleton*, *doubleton*, and *tripleton* that map sets to `true` if they have one, two, or three members, respectively. Prior probabilities of hypotheses are determined by and inversely related to their complexity (Goodman, Tenenbaum, Feldman, & Griffiths, 2008), and they are updated by Bayes' Rule. The data are the frequencies of numerals in the CHILDES database (MacWhinney, 2000; frequency decreases with the size of the number). The system finds the most probable hypothesis in light of the data.

As we will soon see, the system in Piantadosi et al. (2012) manages to converge on the number system through attested knower-level stages. This is a proof of concept of the

[1] See Barner and Bachrach (2010) for evidence that implicature computation affects the characterization of knower-levels, as well as Spector (2013) for a more general characterization of the semantics and pragmatics of numerals in the adult steady state. We largely set implicature computation aside here, but see note 7.

[2] It is not immediately obvious how this characterization connects to the linguistic syntax and semantics of numerals in their occurrences as adjectives, names, or determiners. We put this aside for now but briefly return to this in the discussion section.

In M. Goldwater, F. K. Anggoro, B. K. Hayes, & D. C. Ong (Eds.), *Proceedings of the 45th Annual Conference of the Cognitive Science Society*.

Bayesian approach to bootstrapping, and in particular it manages to converge on the numbers without making use of the notion of *successor* as a primitive concept. The approach also extends to other interesting systems concerned with counting, such as singular-plural distinctions (among others).

Our paper aims to explore which aspects of the model are *essential* to characterizing numerical systems in human language and thought. Piantadosi et al. (2012) assumed a set of primitives along with an empirically motivated frequency distribution that served as input to a machine that reasons along Bayesian principles. Holding the reasoning fixed, would the model continue to converge if the primitives were different? And would it converge if frequency distributions were modified? By choosing various combinations of primitives and frequency distributions, we aim to discover which combinations would converge on the natural numbers, and of those that would, which would pass through attested stages in child development. The model itself allows for a large array of possible numerical systems, only some of which turn out to be empirically attested; motivated by considerations like the *size principle* familiar from Bayesian learning (e.g., Tenenbaum, 1999), this in turn suggests the need to constrain the system to rule out unattested stages and inventories. The approach taken here thus shares with other works concerned with characterizing logical and numerical lexical inventories the assumption that an adequate characterization needs to account not only for what is attested but also for what is unattested; this inevitably leads to a theory that must provide explicit assumptions about the primitive building blocks out of which logical and numerical words and concepts are built (see, e.g., Feldman, 2000; Piantadosi, Tenenbaum, & Goodman, 2016; Katzir & Singh, 2013; Buccola, Križ, & Chemla, 2022; Enguehard & Spector, 2021; Uegaki, 2022). Since not all primitives will do, we are also faced with the task of motivating the privileged status that some primitives play in human language and thought.

Modelling Number Learning

When it comes to modelling number learning, Piantadosi et al. (2012) simplify the complex problem by focusing on how learners often hear number words spoken in specific contexts (e.g., "there are four ducks"), and then attempt to learn structured representations of what those number words mean. In other words, learners attempt to systematically map sets of objects to specific number words (see also note 2). To model this process, Piantadosi et al. (2012) leverage lambda calculus as the means of structured representation, with complex expressions being built from a set of primitives assumed to be inherent to the learner.[3] The task for the learner, then, is to use their real-world exposure to both object sets and number words to figure out how to combine said primitives such that the learner can map sets of arbitrary size to their appropriate number word.

To facilitate this, Piantadosi et al. (2012) deploy a Bayesian model that assigns probabilities to expressions given the word, context, and object(s) heard by the learner. However, the space of possible expressions is infinite, so to hone in on a single expression that best fits the data, their model performs a stochastic search over the hypothesis space whereby for each hypothesized lexicon a lambda expression is sampled and a single change is made to it. This change is then either accepted or rejected with some probability, such that expression complexity is penalized. This sampling processes is completed for one million iterations, with up to 1,000 pairs of object sets, number words, and contexts serving as the input data, and the frequencies of each number word being derived from real-world data (see MacWhinney, 2000).

When it comes to primitives, as noted earlier Piantadosi and colleagues propose three functions that map sets to truth values: *singleton*, *doubleton*, and *tripleton*, which return `true` if the set has one, two, or three members, respectively. They also propose four functions that act on sets: *set-difference* XY, which returns the set that results from removing Y from X; *select* X, which returns a set containing a single element from X; as well as union and intersection functions. They also propose the logical functions *and*, *or*, *not*, and *if*; recursion, denoted C, which returns the result of evaluating an entire lambda expression on a set; and functions that drive the counting routine, specifically *next*, *prev*, and *equal-word*, which move the learner forward/backwards through the counting numbers, and checks for equivalence between number words.[4]

Overall, Piantadosi et al. (2012)'s model is able to generate a CP-Knower that passes through observed developmental stages, first becoming a One-Knower, then a Two-Knower, then a Three-Knower, and finally a CP-Knower. Moreover, their model is also able to learn other numerical systems, such as Mod-5 and singular-plural.

Methods & Results

To explore the sensitivity of Piantadosi et al. (2012)'s model, we performed a variety of experiments that altered the model's assumptions, either by changing word frequencies or adding/removing model primitives. We assumed that the set-manipulation primitives stipulated in the model like union and intersection have prior motivation (see e.g., the references in the penultimate sentence of the introduction). Hence we focused our attention on numerical primitives that mapped sets to truth-values depending on the cardinality of the set (cf. Denić and Szymanik (2022)). We tried to find combinations of primitives that are unattested in any natural number system we know of, and (perhaps partly for that reason) struck us as somewhat unnatural but that – as far as the model is

[3] We assume familiarity with basic concepts from the lambda calculus, but for textbook treatments with applications in mind see (among others) Heim and Kratzer (1998) and Abelson and Sussman (1996).

[4] Piantadosi et al. (2012) point out that their formulation does not rely on having a successor function as primitive, but it does seem to presuppose the notion of a predecessor function (it is a theorem of Peano Arithmetic that each non-zero number has a unique predecessor).

concerned – would be as good as any other. For example, we tested a model that has a singleton and tripleton but no doubleton and we also tested a dual-only system that mapped sets of 2 to *two* and all other sets to *one*. We know of no language with these patterns of morphological number marking, nor are we aware of any natural cognitive system that counts like this. But so far as the model goes these are just as good as the attested system as a set of primitives. By manipulating assumed sets of primitives in this way, we hope to sharpen the characterization of which primitives are essential to human cognition, which also raises the question of why evolution chose the primitives we have instead of some others that allow you to converge on the natural numbers. The goal throughout was to examine what number patterns the system converges on under our manipulations, as well as the stages it passes through, and compare the result with what is actually attested. How does the choice of primitives – what is a priori – affect what a human may become?

For each experiment (50,000 iterations with 300 pairings), we tracked the knower-sequence the model passed through and the knower-formula the model generated. Word frequencies and probabilities were derived from the CHILDES database (Table 2; MacWhinney, 2000), as per Piantadosi et al. (2012). Overall, seven experiments generated a CP-Knower formula, with four experiments failing to do so. The system also managed to converge on our made-up dual-only system. The top-3 formulae for our baseline run of Piantadosi et al. (2012)'s original model can be seen in Table 1, with the model converging on a CP-Knower formula after approximately 8,700 iterations.

Baseline Knower-Sequence Columns are set sizes. Rows are stages. Cells contain number words ascribed to set sizes (U = undefined/unknown).

	1	2	3	4	5	6	7	8	9
1	one	one	one	one	one	one	one	one	one
2	one	U	U	U	U	U	U	U	U
3	one	two	two	two	two	two	two	two	two
4	one	two	U	U	U	U	U	U	U
5	one	two	three	U	U	U	U	U	U
6	one	two	three	four	five	six	seven	eight	nine

Table 1: Top-3 formulae for baseline results. Note: C denotes recursion.

Rank	Formula
1	λ S. (if (singleton? S) "one" (next (C (set-difference S, (select S)))))
2	λ S. (prev (if (singleton? S) "two") (next (C (set-difference S, (select S)))))
3	λ S. (if (singleton? S) "one" (next (C (set-difference S, select(select S)))))

Table 2: Word frequencies and probabilities of CHILDES database.

Number	Frequency	Probability
1	7187	0.68
2	1484	0.14
3	593	0.06
4	334	0.03
5	297	0.03
6	165	0.02
7	151	0.01
8	86	0.01
9	105	0.01
10	112	0.01
Total	10,514	1.00

CP-Knower Generation

Overall, seven manipulations (see Table 3) were able to generate a CP-Knower formula. Of these seven, five were alterations to primitives (adding/removing primitives) and two were alterations of frequencies (swapping frequencies in Table 1).

Swap 1 & 3 Knower-Sequence Converged after 26,000 iterations via non-attested development sequence.

	1	2	3	4	5	6	7	8	9
1	three	three	three	three	three	three	three	three	three
2	U	U	three	U	U	U	U	U	U
3	U	two	three	U	U	U	U	U	U
4	U	two	three	four	five	six	seven	eight	nine
5	one	two	three	four	five	six	seven	eight	nine

Swap 2 & 3 Knower-Sequence Converged after 17,000 iterations via non-attested development sequence.

	1	2	3	4	5	6	7	8	9
1	one	one	one	one	one	one	one	one	one
2	one	U	U	U	U	U	U	U	U
3	one	three	three	three	three	three	three	three	three
4	one	one	three	one	one	one	one	one	one
5	one	U	three	U	U	U	U	U	U
6	one	two	three	four	five	six	seven	eight	nine

Remove Doubleton Knower-Sequence Converged after 35,100 iterations via non-attested development stages.

	1	2	3	4	5	6	7	8	9
1	one	one	one	one	one	one	one	one	one
2	one	U	U	U	U	U	U	U	U
3	one	two	three	one	one	one	one	one	one
4	one	two	three	four	five	six	seven	eight	nine

Remove Tripleton Knower-Sequence Converged after 6,200 iterations via non-attested development sequence.

	1	2	3	4	5	6	7	8	9
1	one	U	U	U	U	U	U	U	U
2	one	two	U	U	U	U	U	U	U
3	one	two	three	four	five	six	seven	eight	nine

Add Quadrupleton Knower-Sequence Converged after 33,100 iterations via attested development sequence, plus a new unattested stage 'four-knower'.

	1	2	3	4	5	6	7	8	9
1	one	one	one	one	one	one	one	one	one
2	one	two	U	U	U	U	U	U	U
3	one	two	three	U	U	U	U	U	U
4	one	two	three	four	U	U	U	U	U
5	one	two	three	four	five	six	seven	eight	nine

Remove Doubleton & Tripleton Knower-Sequence Converged after 24,300 iterations via non-attested development sequence.

	1	2	3	4	5	6	7	8	9
1	one	one	one	one	one	one	one	one	one
2	one	U	U	U	U	U	U	U	U
3	one	two	three	four	five	six	seven	eight	nine

Remove Tripleton & add Quadrupleton Knower-Sequence Converged after 19,100 iterations via non-attested development sequence.

	1	2	3	4	5	6	7	8	9
1	one	one	one	one	one	one	one	one	one
2	one	two	one	one	one	one	one	one	one
3	one	U	U	U	U	U	U	U	U
4	one	U	U	four	U	U	U	U	U
5	one	two	two	four	two	two	two	two	two
6	one	two	U	four	U	U	U	U	U
7	one	two	three	four	five	six	seven	eight	nine

Failed CP-Knower Generation

Here, four manipulations (see Table 4) failed to generate a CP-Knower formula. Of these four, two were alterations to primitives and two were alterations of frequencies.

Swap 1 & 2 Knower-Sequence Failed to converge after 50,000 iterations.

	1	2	3	4	5	6	7	8	9
1	two	two	two	two	two	two	two	two	two
2	one	two	two	two	two	two	two	two	two
3	one	two	U	U	U	U	U	U	U
4	one	two	three	U	U	U	U	U	U

Swap 2 & 8 Knower-Sequence Failed to converge after 50,000 iterations.

	1	2	3	4	5	6	7	8	9
1	one	one	one	one	one	one	one	one	one
2	one	U	U	U	U	U	U	U	U
3	one	U	three	U	U	U	U	U	U

Remove Singleton Knower-Sequence Failed to converge after 50,000 iterations.

	1	2	3	4	5	6	7	8	9
1	one	one	one	one	one	one	one	one	one
2	one	two	three	one	one	one	one	one	one

Remove Singleton & Doubleton Knower-Sequence Failed to converge after 50,000 iterations.

	1	2	3	4	5	6	7	8	9
1	one	one	one	one	one	one	one	one	one
2	one	one	three	one	one	one	one	one	one
3	one	one	three	four	one	one	one	one	one

Singular-plural and dual-only

Piantadosi et al. (2012) trained their model to map singleton sets to *one* and all other cardinalities to *two*, hence capturing the English singular-plural system. We amended this by mapping two-membered sets to *two* and all other cardinalities (1, and 3-9) to *one*. Our system managed to converge on this 'dual-only' target system after 1,500 iterations.

Dual-Only System Knower-Sequence Converge after 1,500 iterations.

	1	2	3	4	5	6	7	8	9
1	one	one	one	one	one	one	one	one	one
2	one	two	one	one	one	one	one	one	one

Discussion

Our testing of Piantadosi et al. (2012)'s model revealed a number of noteworthy results. Broadly construed, it seems possible that given enough time, all frequency manipulations would eventually converge on a CP-Knower formula. Although the present research failed to achieve convergence when the frequencies for 1 & 2, and 2 & 8, were swapped, that the model was able to converge in the face of similar swaps suggests the stochastic nature of their model is a plausible explanation, and that given enough iterations the model would converge. By contrast, our results suggest that removing the *singleton* primitive will result in the model failing to converge on a CP-Knower formula. Removal of other primitives like *doubleton* and *tripleton* were not fatal for convergence, but removal of *singleton* was. This makes intuitive sense as *singleton* is, arguably, the foundation of number learning; not only does it provide a natural 'base case', but removing it could in addition remove the ability to recognize that you can

Table 3: Manipulations & formulae generating CP-Knower. Note: C denotes recursion.

Manipulation	Top Formula
Swap 1 & 3	λ S. (if (True) (if (singleton? S) "one" (next (C (set-difference S, (select S))))) C (S))
Swap 2 & 3	λ S. (if (singleton? S) "one" (next (C (set-difference S, (select S)))))
Remove doubleton	λ S. prev (if (singleton? S) "two" (if (singleton? S) C (S) (next (next (C (set-difference S, (select S))))))
Remove tripleton	λ S. (if (singleton? S) "one" (if (singleton? S) C (S) (next (C (set-difference S, (select S))))))
Add quadrupleton	λ S. (if (singleton? S) "one" (next (C (set-difference S, (select S)))))
Remove doubleton & tripleton	λ S. (if (singleton? S) "one" (next (C (set-difference S, (select S)))))
Remove tripleton, add quadrupleton	λ S. (if (singleton? S) "one" (if (singleton? S) C (S) (next (C (set-difference S, (select S))))))

Table 4: Manipulations & formulae failing to generate CP-Knower. Note: C denotes recursion.

Manipulation	Top Formula
Swap 1 & 2	λ S. (if (singleton? S) "one" prev(next(if (doubleton? S) "two" next(if (doubleton? S) "two" "U"))))
Swap 2 & 8	λ S. prev(if (tripleton? S) "four" prev(prev(if (singleton? S) "four" "U")))
Remove singleton	λ S. (if (tripleton? S) "three" (if (doubleton? S) "two" "one"))
Remove singleton & doubleton	λ S. prev(next(if (tripleton? S) "three" (if (tripleton? (set-difference S, (select S))) "four" "one")))
Dual-only system	λ S. (if (doubleton? S) next("one") "one")

always add one to a count number to get the next count number. Moreover, our findings confirm that recursion – but crucially not necessarily a built-in successor function – is also a prerequisite for becoming a CP-Knower, as all of the successful manipulations, and none of the unsuccessful ones, leveraged a formula containing it.

Perhaps more interesting are the Knower-Sequences produced by the model. Here, the way in which the learner becomes a CP-Knower (if they do at all) depends in part on word frequencies. In all our simulations, the first number word learned by the model was the word with the highest frequency, and the only models to successfully transition through observed developmental stages were the baseline model and the 'add quadrupleton' model. Interestingly, after passing through the usual knower-levels, this 'add quadrupleton' manipulation also produced a new unattested stage, namely, a 'four-knower' stage that sits in between three-knowers and CP-knowers.

This reveals another general property of the system: it appears to begin its lexicalization by selecting from its primitives and then generalizes from there, as the bootstrapping model would predict.[5] If this is correct, this means that our assumptions about primitives are an essential component of the system. For example, note that the model allows for the existence of 'dual-only' systems (see Table 4), namely, those that would mark two-membered sets with *two* (or plural or some other morphological marker) and everything else with *one* (or singular or some other morphological marker). So far as we know, no such morphological system is attested in natural language, suggesting again that this kind of system is unavailable to humans.

More generally, in order to properly capture universals – such as Greenbergian universals about number morphology – we need to assume that not all combinations of primitives are available to the child. In this framework, we might need statements such as that if the system accesses *tripleton* it cannot do so without also accessing *doubleton*, or that if it accesses *doubtleton* it cannot do so without also accessing *singleton*. These kinds of constraints do not seem to follow as theorems of the model. If that is correct, then some separate statements need to be incorporated to adequately constrain the system to rule out these unattested inventories. Furthermore, with assumptions about primitives being evidently essential to characterizing possible numerical systems in human language and thought, we hope that we can eventually come to understand why the primitives are what they are. For example, *singleton* seems required for CP-knower convergence and *doubleton* and (then) *tripleton* seem required for capturing knower-levels. Is there some principled reason for this, and if so, does this shed insight into why morphological number marking systems also broadly obey this pattern? Similar questions arise for the other primitives assumed in the model. We hope to return to these questions in future work.

We would also like to briefly remark here on the connection between the analysis of numbers in Piantadosi et al. (2012) and some of the linguistic analyses of numerals discussed earlier, such as their use as quantificational determiners in sentences like *three students arrived*. In this kind of sentence, it would be natural to assume that *three* belongs to the category D (for 'determiner') and semantically denotes a function like this: $[[three]] = \lambda P.\lambda Q.|(P \cap Q)| = 3$. This is a function that takes two predicates, P and Q, and returns `true` if the cardinality of their intersection is exactly three.[6] It is not immediately obvious how this determiner analysis relates to the analysis in Piantadosi et al. (2012), where it is unclear what category a numeral belongs to, nor is it clear what a numeral *means*. In the Piantadosi et al. (2012) model, a numeral like *three* is the output of a counting function that maps sets to a word, but it remains somewhat unclear what that word itself means, nor how it relates to other uses of the word such as its use as a determiner.

One tentative thought could be that *three* in its counting sense, $three_C$, denotes the collection of sets that the counting procedure, C, maps to *three*: $[[three_C]] = \lambda P.C(P) = three$. Here, C could be any of the CP-knower functions in Tables 2 and 3, for example. There is an uncomfortable circularity here, in that the denotation itself references the symbol whose meaning we are analyzing. However, we don't think the circularity is vicious. We are not assuming anything about the meaning of the symbol; the symbol is just the output of a counting procedure.

This tentative proposal might suggest a natural way to relate the Piantadosi et al. (2012) model with other uses of numerals. Suppose we take the counting analysis as basic. Then the determiner analysis of *three*, call this use $three_D$, could be rewritten as follows: $[[three_D]] = \lambda P.\lambda Q.[[three_C]](P \cap Q)$. The meaning of $three_D$ here continues to be a function of type $< et, < et, t >>$ and it maps the exact same sets P and Q to `true` as the classical determiner analysis above. We have just replaced the truth-condition $|P \cap Q| = 3$ with the truth-condition $C(P \cap Q) = three$.[7]

[5] As would be expected, the elimination of primitives typically results in a more complex CP-knower formula. For example, note that the CP-knower formulae for 'remove doubleton' and 'remove tripleton' are more complex than the one for the baseline. Interestingly, removing both *doubleton* and *tripleton* results in the same simple CP-knower formula as the baseline. The only differences between these systems has to do with the stages they go through: the baseline goes through the usual knower-levels but the 'remove doubleton and tripleton' system does not, and the baseline converges in around a third of the iterations required by the 'remove doubleton and tripleton' system.

[6] Strictly speaking, P and Q are functions of type $< e,t >$, but given well-known relations between such functions and their characteristic sets it is harmless to switch between function talk and set talk. See Heim and Kratzer (1998).

[7] It is commonly assumed that the basic meaning of a *sentence* like *three students arrived* is not that exactly three students arrived, but rather that at least three students arrived. One way to implement this together with the assumption that the word *three* means 'exactly three' is to assume the existence of a covert existential operator elsewhere in the structure. The exactly-three interpretation would then result from the computation of a scalar implicature (see e.g., Spector, 2013, for discussion). A challenge for this view is that, unlike some other implicatures like the 'not all' interpretation

Acknowledgements

We would like to thank Steven Piantadosi and colleagues for providing access to their original model, and the three anonymous reviewers for their detailed and thoughtful commentary.

References

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

Aloni, M. (2022). Logic and conversation: the case of free choice. *Semantics and Pragmatics*, *15*(5), 1-39.

Bar-Lev, M., & Fox, D. (2017). Universal free choice and innocent inclusion. In *Proceedings of salt 27.*

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

Bergen, L., Levy, R., & Goodman, N. (2016). Pragmatic reasoning through semantic inference. *Semantics and Pragmatics*, *9*, Early Access.

Buccola, B., Križ, M., & Chemla, E. (2022). Conceptual alternatives: Competition in language and beyond. *Linguistics and Philosophy*, *45*(2), 265–291.

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

Chemla, E. (2009). Universal implicatures and free choice effects: Experimental data. *Semantics and Pragmatics*, *2*, 1-33.

Chemla, E., & Singh, R. (2014a). Remarks on the experimental turn in the study of scalar implicature, Part I. *Language and Linguistics Compass*, *8*(9), 373-386.

Chemla, E., & Singh, R. (2014b). Remarks on the experimental turn in the study of scalar implicature, Part II. *Language and Linguistics Compass*, *8*(9), 387-399.

Degen, J. (2023). The rational speech act framework. *Annual Review of Linguistics*, *9*, 519–540.

Denić, M., & Szymanik, J. (2022). Reverse-engineering the language of thought: A new approach. In *Proceedings of the annual meeting of the cognitive science society* (Vol. 44).

Enguehard, E., & Spector, B. (2021, June). Explaining gaps in the logical lexicon of natural languages: A decision-theoretic perspective on the square of Aristotle. *Semantics and Pragmatics*, *14*(5), 1–31.

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

Frank, M. C., & Goodman, N. D. (2012). Predicting pragmatic reasoning in language games. *Science*, *336*(6084), 998–998.

Fuson, K. C., & Fuson, K. C. (1988). The number-word sequence: an overview of its acquisition and elaboration. *Children's counting and concepts of number*, 33–60.

Goodman, N. D., Tenenbaum, J. B., Feldman, J., & Griffiths, T. L. (2008). A rational analysis of rule-based concept learning. *Cognitive science*, *32*(1), 108–154.

Heim, I., & Kratzer, A. (1998). *Semantics in generative grammar*. Oxford: Blackwell.

Katzir, R., & Singh, R. (2013). Constraints on the lexicalization of logical operators. *Linguistics and Philosophy*, *36*(1), 1-29.

MacWhinney, B. (2000). *The childes project: Tools for analyzing talk: Volume i: Transcription format and programs, volume ii: The database.* MIT Press.

Piantadosi, S. T., Tenenbaum, J. B., & Goodman, N. D. (2012). Bootstrapping in a language of thought: A formal model of numerical concept learning. *Cognition*, *123*(2), 199–217.

Piantadosi, S. T., Tenenbaum, J. B., & Goodman, N. D. (2016). The logical primitives of thought: Empirical foundations for compositional cognitive models. *Psychological Review*, *123*(4), 392-424.

Singh, R. (2019). Context, content, and the occasional costs of implicature computation. *Frontiers in Psychology*.

Singh, R., Wexler, K., Astle-Rahim, A., Kamawar, D., & Fox, D. (2016). Children interpret disjunction as conjunction: Consequences for theories of implicature and child development. *Natural Language Semantics*, *24*(4), 305-352.

Spector, B. (2013). Bare numerals and scalar implicatures. *Language and Linguistics Compass*, *7*(5), 273-294.

Tenenbaum, J. B. (1999). *A Bayesian framework for concept learning* (Unpublished doctoral dissertation). MIT, Cambridge, MA.

Uegaki, W. (2022, 10). The Informativeness/Complexity Trade-Off in the Domain of Boolean Connectives. *Linguistic Inquiry*, 1-23.

Wynn, K. (1992). Children's acquisition of the number words and the counting system. *Cognitive psychology*, *24*(2), 220–251.

of *some*, the exactly reading tends to be faster to develop in children, is less costly to process, and is easier to detect in embedded positions (for a review of some relevant literature, see Chemla & Singh, 2014a, 2014b). These observations can be made consistent with the implicature-analysis with appropriate auxiliary assumptions about parsing (Singh, Wexler, Astle-Rahim, Kamawar, & Fox, 2016; Singh, 2019). Also relevant here are processing facts about free-choice inferences (Chemla, 2009; Bar-Lev & Fox, 2017; Aloni, 2022). A reviewer points out that Piantadosi et al. (2012)'s model used strong sampling, but one could leverage a more pragmatically-informed likelihood function and model number acquisition through the lens of a Rational Speech Act (see e.g., Bergen, Levy, & Goodman, 2016; Degen, 2023; Frank & Goodman, 2012). This is an interesting proposal we hope to explore in future work.