Cognition 232 (2023) 105150

Contents lists available at ScienceDirect

Cognition

journal homepage: www.elsevier.com/locate/cognit

Quantifiers satisfying semantic universals have shorter minimal description length☆

Iris van de Pol [a,*], Paul Lodder [a], Leendert van Maanen [b], Shane Steinert-Threlkeld [c], Jakub Szymanik [d]

[a] Institute for Logic, Language and Computation, University of Amsterdam, the Netherlands
[b] Experimental Psychology, Utrecht University, the Netherlands
[c] Department of Linguistics, University of Washington, United States
[d] Center for Mind/Brain Sciences and Dept. of Information Engineering and Computer Science, University of Trento, Italy

ARTICLE INFO

Keywords:
Semantic universals
Generalized quantifiers
Logical grammar
Complexity
Minimal description length

ABSTRACT

Despite wide variation among natural languages, there are linguistic properties thought to be universal to all or nearly all languages. Here, we consider universals at the semantic level, in the domain of quantifiers, which are given by the properties of *monotonicity*, *quantity*, and *conservativity*, and we investigate whether these universals might be explained by differences in complexity. First, we use a minimal pair methodology and compare the complexities of individual quantifiers using approximate Kolmogorov complexity. Second, we use a simple yet expressive grammar to generate a large collection of quantifiers and we investigate their complexities at an aggregate level in terms of both their minimal description lengths and their approximate Kolmogorov complexities. For minimal description length we find that quantifiers satisfying semantic universals are simpler: they have a shorter minimal description length. For approximate Kolmogorov complexity we find that monotone quantifiers have a lower Kolmogorov complexity than non-monotone quantifiers and for quantity and conservativity we find that approximate Kolmogorov complexity does not scale robustly. These results suggest that the simplicity of quantifier meanings, in terms of their minimal description length, partially explains the presence of semantic universals in the domain of quantifiers.

1. Introduction

If you have ever tried to learn a new language, you will know that this can be a challenge. You have to learn a lot of new things that are different from the language you are used to. While the world's languages have many differences, at the same time, interestingly, most languages also share a striking amount of similarities, called linguistic universals (Croft, 1990; Goddard & Wierzbicka, 2002; Greenberg, 1966). Here we study such universals at the semantic level, in the domain of quantifiers (Barwise & Cooper, 1981; von Fintel & Matthewson, 2008).

Quantifiers are semantic objects that express quantitative relational properties, such as expressed by the words *some*, *most*, or *all*. It has been observed that the quantifiers that are lexicalized (as mono-morphemic words) in natural language share certain semantic properties, namely those of *monotonicity*, *quantity*, and *conservativity* (Barwise & Cooper, 1981; Keenan & Stavi, 1986; Peters & Westerståhl, 2006). For example, the sentence "some bicycles are red" features the monotone, quantitative, and conservative quantifier `some`. It is monotone because its meaning does not change when making the sentence more specific, e.g., it implies the sentence "some bicycles have a red part." It is quantitative because its meaning does not depend on the order of the bicycles: given that there are indeed some red bicycles, the sentence is true irrespectively of in which order those bikes are placed. It is conservative because the truth of the sentence only depends on the bicycles and not on other red things that are not bicycles, i.e., to verify whether the sentence is true, not everything that is red needs to be checked, only the bicycles. Loosely speaking, a quantifier is monotone when its meaning does not change when moving from a less specific to a more specific meaning (or vice versa), a quantifier is quantitative when its meaning only depends on the number of objects, and not on their order,

☆ This paper is a revised and extended version of non-archival work presented at the 41st and 43rd Conferences of the Cognitive Science Society (Van de Pol et al., 2019, 2021).

* Corresponding author.
E-mail address: ivdpol@protonmail.com (I. van de Pol).

https://doi.org/10.1016/j.cognition.2022.105150
Received 6 August 2021; Received in revised form 22 April 2022; Accepted 25 April 2022
Available online 21 December 2022

and a quantifier is conservative when its meaning only depends on one group of things and everything that falls outside of this group is irrelevant. When looking at the space of all logically possible quantifiers, however, a large majority does not have these properties.

The question arises why these universals hold. Why do quantifiers in natural language have precisely these properties? A possible explanation for these universals lies in the interaction between these properties and our cognitive apparatus (see, e.g., Gibson et al., 2019; Steinert-Threlkeld & Szymanik, 2020). In search of such an explanation in terms of the interaction between linguistics and the specifics of human cognition, several theories have presented some form of a learnability hypothesis: the idea that semantic universals hold because meanings with these properties are easier to learn (see, e.g., Barwise & Cooper, 1981; Van Benthem, 1986; Keenan & Stavi, 1986; Szabolcsi, 2010; Steinert-Threlkeld & Szymanik, 2020). In this paper, we ask whether these semantic universals could also be explained by a bias for simplicity, a concept that is thought to be closely related to learnability (Carr, Smith, Culbertson, & Kirby, 2020; Hsu, Chater, & Vitányi, 2013), which might be understood from the intuition that there is a close relation between how simple or complex the meaning of an expression is and how easy or hard it is to learn the meaning of that expression. Simplicity as an explanatory concept in cognition has been studied in a variety of domains (see Chater & Vitányi, 2003; Feldman, 2016). Here, we explore the following hypothesis: quantifiers that have these semantic properties are *simpler*. Together with the linking hypothesis that the cognitive system favors lexicalizing simple meanings (as monomorphemic words), such differences in complexity could explain the presence of these semantic universals.

While the complexity of quantifiers is a well-studied topic, it remains largely unknown how the complexity of quantifiers relates to the presence of semantic universals. Complexity measures that have traditionally been applied in the area of quantifiers—such as measures derived from logical definability (Szymanik & Zajenkowski, 2010), automata theory (Van Benthem, 1984), and computational complexity theory (Kontinen & Szymanik, 2008; Ristad, 1993)—unfortunately are unsuited for this task. Although these measures can capture some of the cognitive difficulty of quantifier processing (see Szymanik, 2016, for an overview) they are too coarse to pick up on potential differences in complexity between quantifiers that do and do not satisfy semantic universals.

For example, in the semantic automata literature it has been shown that quantifiers that are computable by a *finite-state automaton* are easier to understand in sentence-picture verification experiments than quantifiers that require a *pushdown automaton* (a finite-state automaton augmented with a stack for memory) (McMillan, Clark, Moore, Devita, & Grossman, 2005; Szymanik & Zajenkowski, 2010). However, the finer-grained distinctions that are relevant to the semantic universals under discussion here, in general, cannot be picked up by automata-based complexity measures. For example, the difference between the automaton for the upward monotone quantifier `at least three` and the non-monotone quantifier `exactly two` consists just in which states are marked as final and which not. This means that these automata have the same number of states, i.e., they are of the same size. Most measures for complexity based on automata rely on the number of states and are not sensitive to subtle differences such as which states are marked as final.

Because of these limitations of the complexity measures that have traditionally been applied in the area of quantifiers, we adopt two innovative measures of the complexity of quantifiers—approximate Kolmogorov complexity and minimal description length in a logical grammar—and we investigate their potential to explain the presence of semantic universals. It is an open question how these two measures relate to each other precisely and whether they capture the same aspect of the complexity of quantifiers. We investigate if and how their results differ and which measure best captures potential differences in complexity between quantifiers with versus without universal properties.

We perform two simulation studies, which we call Experiment 1 and Experiment 2. In Experiment 1, following the work by Hunter and Lidz (2013) and Steinert-Threlkeld and Szymanik (2019, 2020), we conduct a minimal pair experiment, where we look at the difference in complexity between minimally differing quantifier pairs, consisting of one quantifier satisfying a universal property and the other not satisfying it. To make fine-grained distinctions between the complexity of the individual quantifiers in the quantifier pairs, we use a measure from the framework of algorithmic information theory—in particular, an approximation to Kolmogorov complexity (Li & Vitányi, 2008) based on the Lempel-Ziv compression algorithm (Lempel & Ziv, 1976). We show that the monotone quantifiers have a lower approximate Kolmogorov complexity and that, overall, complexity and learnability pattern together.

In Experiment 1 we test a handful of quantifier pairs. Unfortunately, there is no principled way to systematically scale up this methodology and automate the selection of a large collection of minimal quantifier pairs. To overcome these limits of scale of the minimal pair methodology we perform a second simulation study, Experiment 2, in which we generate a large collection of logically possible quantifiers and we use a logistic regression model to analyze the relation between the level of complexity of a quantifier and the presence of universal properties at an aggregate level. We generate these quantifiers based on a simple yet expressive grammar, i.e., a language of thought, a framework that has been used, e.g., in the domain of concept learning (Feldman, 2000; Goodman, Tenenbaum, Feldman, & Griffiths, 2008), language acquisition (Piantadosi, Tenenbaum, & Goodman, 2013), and auditory memory (Planton et al., 2021). Following Steinert-Threlkeld (2020, 2021) we measure the complexity of these quantifiers in terms of their minimal description length in this grammar. In addition, for comparison, we also measure their complexities in terms of the approximate Kolmogorov complexity that we use in Experiment 1. We show that quantifiers with universal properties are simpler: they have a shorter minimal description length. We also show that monotone quantifiers have a lower approximate Kolmogorov complexity than non-monotone quantifiers. We find no effect for quantity and an opposite effect for conservativity, which, compared to the results of Experiment 1, raises the question of robust scalability of the approximate Kolmogorov complexity of quantifiers. These results suggest that the simplicity of quantifier meanings, in terms of their minimal description length, partially explains the presence of semantic universals in the domain of quantifiers.

The paper is structured as follows. First, we introduce the framework of generalized quantifiers and the properties of monotonicity, quantity, and conservativity and the semantic universals in relation to these properties. Then, we present the minimal quantifier pair methodology and how we measure the approximate Kolmogorov complexity of quantifiers that we use in Experiment 1. We report on the descriptive statistics of the data for each quantifier pair and we compare these to the learnability results by Steinert-Threlkeld and Szymanik (2019). Next, we define the logical grammar that we use to generate a large collection of logically possible quantifiers and we describe the method that we use to compute their minimal description lengths. We report on both descriptive statistics and bootstrapped logistic regression results, by which we analyze the relation between quantifiers satisfying universal properties and those that do not, and their level of simplicity. Finally, we compare the results of Experiments 1 and 2, discuss their implications and shortcomings, and make suggestions for future research.

2. Quantifiers

Quantifiers are the semantic objects that are expressed by (quantified) determiners,[1] such as *some*, *most*, or *all but one*, which describe quantity in a noun phrase. Determiners are expressions that can combine with common nouns and a verb phrase in simple sentences of the form

[1] Not all determiners express quantifiers. For instance, the demonstrative determiners—like *those*—do not express a quantifier. We refer to determiners that express quantifiers as quantified determiners.

Det N VP, like "some bicycles are red." We assume a division between simple and complex determiners, where *some* and *most* are examples of simple determiners and *all but one* is an example of a complex determiner. Roughly speaking, one can think about this as a division between quantifiers that are lexicalized (as mono-morphemic words) and those that are not (Barwise & Cooper, 1981; Keenan & Stavi, 1986).

Quantifiers form a suitable domain to study semantic universals, both because of the numerous semantic universals that have been identified in this area and because quantifiers lend themself to quite precise and systematic characterization by logical and mathematical tools. Or, as Peters and Westerståhl (2006) eloquently formulate it:

> Quantifiers are one of very few expressive devices of language for which it is known how to break out of the circle of language and explain what a word means other than essentially in terms of other words' meanings. It is possible to explain the meaning of quantifiers in mathematical and other non-linguistic terms. (p. vii).

2.1. Generalized quantifiers

We use the framework of generalized quantifiers to represent the meaning of quantifiers as a collection of models.[2] A model consists of a collection of objects and their properties. It can be seen as a snapshot of a particular part of the world. For instance, to verify the sentence "some bicycles are red," we look at the collection of bicycles and the collection of red things, and when those overlap we know that there are some bicycles that are red.

Formally, a model $\mathcal{M} = \langle M, A, B, < \rangle$ consists of a domain, the set M, two subsets of that domain, sets $A, B \subseteq M$, where the sets A and B possibly overlap, and an ordering $<$ over the domain M. We only consider non-empty domains. Functionally, M is called the domain of discourse, A is called the restrictor, and B is called the scope. The sentence "some bicycles are red" then means that the set A represents the collection of bicyles and the set B represents the collection of red things and that there are some objects (bicyles) in A that are also in B (red things). In addition, we enumerate the objects in the domain, giving us an ordering $<$ over M, which allows to model sentences in which the order of the objects matter, such as "the first 3 bicycles are red." See Fig. 1 for an illustration of a model of the form $\mathcal{M} = \langle M, A, B, < \rangle$.

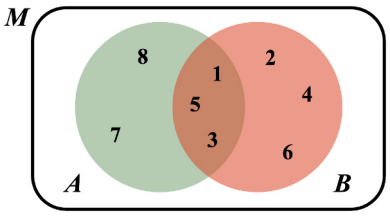

Fig. 1. An example of a model of the form $\mathcal{M} = \langle M, A, B, < \rangle$. The numbers represent the objects in the domain and they correspond to the position of the objects in the order $<$. The sentence "the first 3 bicycles are red" is true in this model. When evaluating that sentence, the objects in A (the left circle in the figure) represent bicycles and the objects in B (the right circle in the figure) represent red things. The sentence is true in $\mathcal{M}$ because the first three objects in A (the objects with position 1, 3, and 5 in the order $<$ over the whole domain), i.e., the first three bicycles, are also in B, i.e., are red.

A quantifier can then be represented and defined by a quantifier expression: a formula in a set-theoretic language. The quantifier is the collection of all models in which that quantifier expression is true. For instance, the meaning of the quantifier `all` can be represented by the expression $A \subseteq B$, meaning that all objects in A are also in B. Formally, we would express the meaning of `all` by $[\![\texttt{all}]\!] = \{\langle M, A, B \rangle : A \subseteq B\}$, which stands for the collection of all models in which the quantifier expression is true.

2.2. Universal properties of quantifiers

Using the framework of generalized quantifiers we can define various properties. A quantifier is *monotone*[3] when it is either *upward* or *downward monotone*. Monotonicity expresses that the meaning[4] of a quantifier does not change when expanding (upward monotone) or contracting (downward monotone) its scope, i.e., the set B. In other words, the meaning of a quantifiers does not change when moving from a more specific to a less specific meaning (upward monotone) or vice versa (downward monotone). For example, the quantifier `some` is upward monotone. To illustrate, the sentence "some logicians climb mountains" implies the more general sentence "some logicians climb."[5] And the quantifier `few`, for example, is downward monotone. To illustrate, the sentence "few logicians climb" implies the more restricted sentence "few logicians climb mountains." The quantifier `exactly two`, on the other hand, is an example of a quantifier that is neither upward nor downward monotone and therefore non-monotone. To illustrate, the sentence "exactly two logicians climb mountains" does not imply the more general sentence "exactly two logicians climb" nor does the sentence "exactly two logicians climb" imply the more restricted sentence "exactly two logicians climb mountains." Formally, monotonicity is defined as follows. Let Q be a generalized quantifier over ordered models, where Q is defined by some quantifier expression q. In other words, Q is the collection of models in which quantifier expression q is true and those models are of the form $\langle M, A, B, < \rangle \in Q$.[6] Then Q is *upward monotone* when: if $\langle M, A, B, < \rangle \in Q$ and $B \subseteq B'$, then $\langle M, A, B', < \rangle \in Q$. Analogously, Q is *downward monotone* when: if $\langle M, A, B, < \rangle \in Q$ and $B' \subseteq B$, then $\langle M, A, B', < \rangle \in Q$. The following universal has been proposed with respect to monotonicity:

- *All simple (quantified) determiners express monotone quantifiers* (Barwise & Cooper, 1981).[7]

The property of *quantity*[8] expresses that the meaning of a quantifier only depends on the sizes of the different subareas of the model, i.e., the sizes of $A \cap B, A \backslash B, B \backslash A$, and $M \backslash (A \cup B)$, which stand for the overlap between A and B, A minus B, B minus A, and the area of M that is outside of A and B, respectively. In contrast, its meaning does not depend on the order of the objects in the domain or on their individual identities or

[2] Specifically, we use the framework of type $\langle 1, 1 \rangle$ generalized quantifiers. Focussing on quantifiers of type $\langle 1, 1 \rangle$ means that we do not consider quantifiers of type $\langle 1 \rangle$, such as proper names. For a textbook treatment of generalized quantifiers see Peters and Westerståhl (2006). For more details on computational representations of quantifiers, see Szymanik (2016).

[3] We use the general term *monotone* to refer to what is also called *right monotone* for type $\langle 1, 1 \rangle$ quantifiers. In our case (that of type $\langle 1, 1 \rangle$ quantifiers) this means that a quantifier is monotone in the set B.

[4] Note that the meaning of a quantifier is defined by the collection of models in which a quantifier expression is true.

[5] We consider the literal meaning of quantifiers, not including their implicatures or presuppositions.

[6] Note that the order $<$ over the models plays no role in the definition of monotonicity. Neither does it play a role in the definition of conservativity. Since it does play a role in the definition of the quantity property, we include it for the sake of uniformity of presentation.

[7] In fact, the original claim by Barwise and Cooper (1981) is a bit weaker, including not just monotone quantifiers, but also conjunctions of monotone quantifiers.

[8] The term *quantity* was introduced by Van Benthem (1984), it refers to the same property that is called *logical* by Keenan and Stavi (1986) and *isomorphism closure* by Peters and Westerståhl (2006).

names. For instance, the quantifier `three` is quantitative. To illustrate, the truth of the sentence "three bikes in the bike shed are red" does not depend on who's bikes they are and in which order they are parked in the bike shed, as long as there are three red bikes, the sentence is true. The quantifier, `the first three`, on the other hand, is not quantitative. The truth of the sentence "the first three bikes in this bike shed are red" depends on the order in which the bikes are parked, knowing only the quantity of red bikes is not enough. Formally, we say that quantifier Q is *quantitative* when: if $\langle M, A, B, <\rangle \in Q$ and for M', A', B' with $A', B' \subseteq M'$ it holds that $A \cap B$, $A \backslash B$, $B \backslash A$, and $M \setminus (A \cup B)$ have the same cardinalities as $A' \cap B'$, $A' \backslash B'$, $B' \backslash A'$, and $M' \setminus (A' \cup B')$, and $<'$ is an order over M', then $\langle M', A', B', <'\rangle \in Q$. In other words, when a quantifier expression is true in some model, it is true in all models that have subareas of the same size as the original model. The following universal has been proposed with respect to quantity:

- *All simple (quantified) determiners express quantitative quantifiers* (Keenan & Stavi, 1986).

The property of *conservativity*[9] expresses that to verify a quantifier, the objects in B that are not in A are not relevant, only the objects that are in A matter. For example, the quantifier `most` is conservative. To illustrate, the truth of the sentence "most logicians like climbing" depends only on logicians and not on climbers who are not logicians. The quantifier `exactly as many A's as B's`, on the other hand, is non-conservative. For example, to verify the sentence "there are exactly as many logicians as there are climbers," it is not just the logicians that are relevant, but also the climbers that are not logicians. Formally, we say that Q is *conservative* when: $\langle M, A, B, <\rangle \in Q$ if and only if $\langle M, A, A \cap B, <\rangle \in Q$. The following universal has been proposed with respect to conservativity:

- *All simple (quantified) determiners express conservative quantifiers* (Barwise & Cooper, 1981; Higginbotham & May, 1981; Keenan, 1981; Keenan & Stavi, 1986).[10]

We interpret these universals as constraints on language in the form of general tendencies or biases, not as fully strict demarcations. An extensive discussion and defense of these universals falls outside of the scope of the current study. Instead, we focus on how these properties relate to the complexity or simplicity of quantifiers and whether simplicity could explain the pervasiveness of these properties in natural language.

3. Experiment 1: minimal pairs

To investigate the relation between the complexity of quantifiers and whether they adhere to universal properties, we first do a small scale study in which we adopt the minimal pair methodology used by Steinert-Threlkeld and Szymanik (2019). This methodology selects minimally differing pairs of quantifiers of which one quantifier satisfies a universal property and the other quantifier does not, and compares a given parameter—in our case simplicity—between the quantifiers in the pair.[11] In each case, the pairs of quantifiers are chosen to be as similar as possible along as many dimensions as possible, while still differing on whether they adhere to the relevant universal. For example, for conservativity, they compare the quantifier `most` ($|A \cap B| > |A \backslash B|$) with a hypothetical quantifier `M`, meaning, `exactly as many A's as B's` ($|A| > |B|$): these both make the same comparison between the cardinalities of two sets, but one is conservative and one is not.

Steinert-Threlkeld and Szymanik (2019) use this methodology to investigate the relation between the learnability of quantifiers and whether they adhere to universal properties, using recurrent neural networks as a model for learning. It is commonly expected that there is a strong relation between learnability and complexity and many theories of learning are built around the idea of such a connection (Hsu et al., 2013; Langley & Stromsten, 2000; Tiede, 1999). At the same time, there are few examples of studies that provide evidence for this expectation in concrete cognitive tasks and capacities. In particular, it remains open which operationalizations of both concepts are most suited to study human cognition, and how those might relate to each other. While our primary focus in this study is on complexity, to facilitate direct comparison between our complexity results and the learnability results by Steinert-Threlkeld and Szymanik (2019), we use the same minimal quantifier pairs as in their study.

3.1. Methods

We use the following methods to compute the complexity of quantifiers in minimal quantifier pairs, to investigate whether we find differences in complexities between quantifiers that do and that do not adhere to the universal properties.

3.1.1. Approximate Kolmogorov complexity of quantifiers

To measure differences in the complexity between individual quantifiers in the minimal pair methodology, we need a fine-grained measure of complexity that is suited for that task. Therefore, in this experiment we use (approximate) Kolmogorov complexity—a measure from the framework of algorithmic information theory that can potentially make such fine-grained distinction—and we investigate its potential to explain semantic universals. This measure has not yet been explored in the domain of quantifiers,[12] though it has previously been shown useful in modeling a cognitive bias towards simplicity in a variety of other cognitive domains (see Chater & Vitányi, 2003; Feldman, 2016; Planton et al., 2021). For example, Feldman (2000) famously showed that boolean concept learning can be predicted by a form of complexity that can be seen as analogous to Kolmogorov complexity (Feldman, 2016).

Roughly speaking, Kolmogorov complexity (K) measures the amount of structure in an individual object: it measures how much a sequence of symbols can be compressed into a shorter sequence without losing information. The intuition behind this is that when a sequence contains regularities, these regularities can be exploited to produce a shorter description of that sequence. For example, the sequence consisting of a thousand zeroes, i.e., `00000...`, has low complexity because it has a lot of structure or regularity: it could be represented by a program like "repeat `0` one thousand times". By contrast, a truly randomly generated sequence of the same length cannot be summarized so succinctly and is

[9] The term *conservativity* was introduced by Keenan (1981), it refers to the same property that is called *lives on* by Barwise and Cooper (1981), and *intersectivity* by Higginbotham and May (1981).

[10] In fact, the original claim is even stronger, namely that *all* (quantified) determiners express conservative quantifiers, not just the simple ones. Whether conservativity is indeed a constraint on the lexicon is an open debate, which we briefly discuss in Section 3.3. The stronger claim about conservativity fits well with the position that conservativity is not a constraint on the lexicon. See also Zuber and Keenan (2019) for an alternative definition of conservativity.

[11] A similar approach was used in an experiment by Hunter and Lidz (2013) to study 4- and 5-year-olds' ability to learn a novel conservative quantifier (not attested in natural language) versus a novel non-conservative quantifier.

[12] Besides the non-archival work by Van de Pol et al. (2019) of which this paper is a revised and extended version.

therefore not as compressible. Such a random sequence will have a higher Kolmogorov complexity than the structured sequence 00000...[13]

More precisely, the Kolmogorov complexity $K(x)$ of a sequence x is defined as the length of the shortest program p that outputs the sequence x (see Li & Vitányi, 2008).[14] A drawback of Kolmogorov complexity is that it has been formally proven to be uncomputable. This means that there exists no algorithm that outputs for any given sequence x, its Kolmogorov complexity $K(x)$ (Li & Vitányi, 2008). Because exact computation is impossible, we need a measure that approximates K. Gauvrit, Zenil, Delahaye, and Soler-Toscano (2014) use a procedure (the "Coding theorem method") for approximating Kolmogorov complexity that gives reliable results for short sequences. Unfortunately, their procedure only works for sequences up to length 50, because it is computationally too expensive for larger sequences. Given that here we look at sequences of length $4^1 + 4^2 + \cdots + 4^{10} = 1,398,100$ (which we will explain in the next section), this procedure is not suitable for our purposes. For these reasons, we use a well-established and tractable approximation to Kolmogorov complexity, that is based on the Lempel-Ziv algorithm for lossless data compression (Lempel & Ziv, 1976).

The Lempel-Ziv compression algorithm measures the number of unique subpatterns, when scanning a sequence from left to right. The Lempel-Ziv complexity $LZ(x)$ of a sequence x is the number of these unique subpatterns of x. For approximate Kolmogorov complexity $\widetilde{K}$, we use $C_{LZ}(x)$, which is defined as $\log_2(\text{len}(x)) \cdot \frac{LZ(x)+LZ(\text{reverse}(x))}{2}$, where $\text{len}(x)$ stands for the length of the sequence x and $\text{reverse}(x)$ stands for the sequence that results from putting sequence x in the reverse order. We use the same version of C_{LZ} as used by Dingle, Camargo, and Louis (2018), which uses the average between $LZ(x)$ and $LZ(\text{reverse}(x))$, instead of just $LZ(x)$.[15] Ziv and Lempel (1978) show that $C_{LZ}(x)$ approximates $K(x)$ in the limit; i.e., when $\text{len}(x)$ approaches infinity. Vitányi (2013) shows that, in practice, lossless compression methods give adequate results also for finite sequences. Furthermore, C_{LZ} is considered particularly adequate as a measure for $\widetilde{K}$ for shorter sequences (Lesne, Blanc, & Pezard, 2009). In the remainder of this paper, we simply use the term Lempel-Ziv complexity to refer to C_{LZ}. This is also what we mean by "approximate Kolmogorov complexity" moving forward.

This framework allows us to compare the complexity of different quantifiers in the minimal quantifier pairs. In doing so, we are not interested in the absolute complexity values of the quantifiers but in the potential difference in complexity between a quantifier that satisfies a universal and its minimally differing counterpart that does not satisfy that universal.

3.1.2. Encoding quantifier meanings as binary sequences

To compute the Lempel-Ziv complexity of quantifiers we first generate a binary representations of that quantifier: a sequence of ones and zeroes that represents the meaning of the quantifier. This works as follows. The meaning of a quantifier is determined by the collection of quantifier models in which the quantifier is true. To order these models into a sequence, we first encode each quantifier model as a sequence of symbols (Mostowski, 1998; Van Benthem, 1986). We give the different subareas in a model a label, say $A \cap B \mapsto d$, $A \backslash B \mapsto e$, $B \backslash A \mapsto f$, and $M \backslash (A \cup B) \mapsto g$, we label the objects in the model by their area, and place each label in a sequence, based on the order of the objects in the model. For example, the model in Fig. 1 is encoded by *dfdfdfee*. Then, we enumerate all models from small to large, up to a maximum model size s, in a fixed order over the encodings of the models. Because of the exponentially large size of the space of possible models[16] we cannot consider *all* possible orderings thereof. Rather, we rely on an especially natural class of orderings, which arises from choosing how to order the subareas of a model, in particular, we consider the class of lexicographical orderings over the encodings of the models, i.e., the dictionary orderings over the labels $\{d, e, f, g\}$.[17] (In the final paragraph of this section we will explain in more detail how we define this class of orderings.) Due to this exponential blow-up, we limit the maximum model size to 10. Finally, for each of the models in the sequence, we put a 1 when the quantifier is true in that model and a 0 otherwise. This results in a unique representation for each quantifier meaning, given the fixed model ordering and a maximum model size. See Fig. 2 for an example of a binary encoding of the quantifier some over models of size 1, i.e., models with only one object.

We can demonstrate how Lempel-Ziv complexity works via this example. The algorithm scans a sequence from left to right and records the number of unique subsequences encountered along the way. For the sequence 1000 from Fig. 2, this will discover the subsequences 1, 0, and 00 (with decomposition 1|0|00), resulting in a complexity of 3 for this particular sequence. The quantifier no would generate the sequence 0111 on these same models and would have complexity 3 as well (with

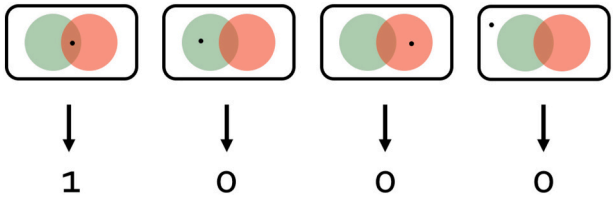

Fig. 2. Encoding of the quantifier some over models of size 1.

[13] Strictly speaking, a procedure for generating a random sequence could also generate the sequence 00000..., which in that case would be considered a randomly generated sequence. However, any procedure for generating random sequences will produce such a sequence with an extremely low probability, so we do not consider that possibility in our explanation here.

[14] The C_{LZ} measure uses a multiplication by $\log_2(\text{len}(x))$ because for lossless compression, for each subpattern a number needs to be stored that identifies the position of a previous subpattern by which the current subpattern can be constructed. This position is upper bounded by $\text{len}(x)$, which can be encoded by a binary sequence of length roughly $\log2(\text{len}(x))$.

[15] Formally, Kolmogorov complexity (K) is defined given a particular universal Turing machine (UTM), but, by *the Invariance Thesis*, K given UTM V or given UTM W will not differ more than some constant c.

[16] The number of possible models grows exponentially in the maximum size of the models, roughly like $|\{d, e, f, g\}|^s$, where $|\{d, e, f, g\}|$ is the number of model areas and s is the maximum model size.

[17] In principle, any fixed ordering of the models can be used for this. However, for our purpose a structured ordering is more suited than a random one. First, it is expected that when a quantifier with a universal property has a lower Lempel-Ziv complexity than its minimally differing counterpart, that is because the universal property causes a regularity in the distribution of truth values across quantifier models. Such a regularity in the distribution over quantifier models might not be visible when those models are placed in a random sequence. Second, for random model sequences, the complexity of a quantifier is expected to be mostly determined by the uniformity of the quantifier (defined by taking the maximum between the ratio of 1's versus 0's and the ratio of 0's versus 1's), which depends only on the number of models in which the quantifier is true and says little about the actual meaning of the quantifier. This is because any two random binary sequences of length n with equal uniformity are likely to have a similarly high Lempel-Ziv complexity. The intuition behind this is that most of the sequences of length n with equal uniformity will have a complexity value close to the maximum Lempel-Ziv complexity for such sequences. This is because having a low Lempel-Ziv complexity means that it can uniquely be compressed to a shorter sequence and there are exponentially few sequences of such short length (i.e., logarithmically many in sequence length n), compared to all possible sequences of length at most n. For these reasons, we evaluate our quantifiers over the lexicographical orderings of models, which is standardly used in the literature on generalized quantifiers.

unique subsequences 0, 1, 11 and decomposition 0|1|11). As a point of contrast, a quantifier that is a falsehood, i.e., $\bot$, would generate the sequence 0000 on these models, which has a complexity of 2, since the only unique subsequences are 0 and 00 (with decomposition 0|00|0). Because the length of the quantifier encodings increases exponentially in the size of the models, we leave longer examples as an exercise to the reader.

Because the four model area labels $\{d, e, f, g\}$ can be ordered in different ways there is not one unique lexicographical ordering of the models. In other words, there are different mappings possible between the labels and the model areas. More precisely, there are 24 such mappings because the four model area labels $\{d, e, f, g\}$ can be ordered in $4! = 24$ ways. Each of these orders can be used as a lexicographical base, i.e., as a dictionary base,[18] on which to order the quantifier models. This result in 24 different sequences of the quantifier models, of which 12 are the reverse of one of the other 12. Since the complexity measure C_{LZ} takes the average between the complexity of a sequence and the complexity of the reverse of that sequence, this leaves 12 uniquely different lexicographical model sequences over which we can compute this measure. For robustness, we look at all 12 uniquely different lexicographical orderings over $\{d, e, f, g\}$. For the quantifier pairs that we look at in the next section, we compare the complexity of the quantifiers in the minimal quantifier pairs for each of these 12 orderings.

3.2. Results

With this framework of Lempel-Ziv complexity as approximate Kolmogorov complexity of quantifiers in place we can now turn to our minimal quantifier pair experiment, to test for the three semantic universals whether quantifiers that adhere to the universal property are simpler, i.e., have a lower Lempel-Ziv complexity, than those that do not. As mentioned, to be able to directly compare our complexity results with the learnability results by Steinert-Threlkeld and Szymanik (2019), we tested the same pairs of quantifiers as in their study.

We computed the following complexity values. Let $x_{i,Q}$ be the binary representation of quantifier Q, based on a sequence of all models up to size i. For each quantifier Q, and for each model size i from 1 to 10, we computed $C_{LZ}(x_{i,Q})$. We repeated this for all 12 lexicographical model sequences. For each quantifier pair we plotted the mean complexity against the maximum model size, with 95% confidence intervals.[19] The 12 individual plots for each of the quantifier pairs can be found in Appendix A. For the sake of readability, in the remainder of this section we will use the phrase "model size" to denote maximum model size. The code that we used for generating these data and the data themselves can be found at https://github.com/ivdpol/quantifier-LZ-complexity.

3.2.1. Monotonicity

To test the property of monotonicity, we looked at two quantifier pairs, one with a downward and one with an upward monotone quantifier. First, we compared the downward monotone quantifier at most three, meaning $|A \cap B| \leq 3$, with the non-monotone quantifier at least six or at most two, meaning $|A \cap B| \geq 6$ or $|A \cap B| \leq 2$. The mean complexity values over all 12 lexicographical model sequences and a 95% confidence interval are plotted in Fig. 3. The descriptive statistics show that for all model sizes larger than 2, monotone at most three has a lower complexity than non-monotone at least six or at most two. (For model size 1 and 2 the differences are 0, because both quantifiers are uniformly true in models of this size.) This holds for each of the 12 different model sequences.

Second, we compared the upward monotone quantifier at least four, meaning $|A \cap B| \geq 4$, with the non-monotone quantifier at least six or at most two, meaning $|A \cap B| \geq 6$ or $|A \cap B| \leq 2$. The mean complexity values over all 12 model sequences and a 95% confidence interval are plotted in Fig. 3. Exactly as in the case of the downward monotone quantifiers, the descriptive statistics show that that for all model sizes larger than 2, monotone at most three has a lower complexity than non-monotone at least six or at most two. (For model size 1 and 2 the differences are 0, because at least four is uniformly false and at least six or at most two is uniformly true in models of this size. This results in binary encodings of all 1's or all 0's, which have the same complexity.) This holds for each of the 12 different model sequences.

These complexity results show a clear pattern of the monotone quantifiers being simpler than the non-monotone quantifiers. This is in line with the learnability results in the study by Steinert-Threlkeld and Szymanik (2019), which found that the monotone quantifiers were easier to learn by a recurrent neural network than the non-monotone quantifiers.

3.2.2. Quantity

To test the property of quantity, we looked at two quantifier pairs with a quantitative and a non-quantitative quantifier. First, we compared the quantitative quantifier at least three, with the non-quantitative quantifier the first three. The mean complexity values over all 12 model sequences and a 95% confidence interval are plotted in Fig. 4. For model size 1, 2, and 3, the differences are 0, since the meanings of these quantifiers are equivalent when evaluated in models of this size (uniformly false in models of size 1 and 2, and uniformly true in models of size 3). For model sizes 4 to 10 the descriptive statistics show that at least three is less complex in 59.5% of the cases, and more complex in 33.3% of the cases.

Second, we compared the quantitative quantifier at least three, with the non-quantitative quantifier the last three. The main complexity values over all 12 model sequences and a 95% confidence interval are plotted in Fig. 4. For model size 1, 2, and 3, the differences are 0, since the meanings of these quantifiers are equivalent when evaluated in models of this size (uniformly false in models of size 1 and 2, and uniformly true in models of size 3). For model sizes 4 to 10 the descriptive statistics show that at least three is less complex in 52.4% of the cases and more complex in 42.9% of the cases.

These complexity results show a tendency towards the quantitative quantifiers being simpler than the non-quantitative quantifiers. The direction of this pattern is in the same direction as the learnability results in the study by Steinert-Threlkeld and Szymanik (2019). However, the pattern of the learnability results was more clearly pronounced, the quantitative quantifiers were significantly easier to learn than the non-quantitative ones.

3.2.3. Conservativity

To test the property of conservativity, we looked at two quantifier pairs with a conservative and a non-conservative quantifier. First, we compared the conservative quantifier most, meaning $|A \cap B| > |A \backslash B|$, with the non-conservative quantifier M, meaning $|A| > |B|$, i.e., meaning that there are more A's than B's, which is non-conservative because to verify its meaning all objects in B are relevant, also the ones that are not in A. The mean complexity values over all 12 model sequences and a 95% confidence interval are plotted in Fig. 5. The descriptive statistics show that that for all model sizes and for all model sequences, conservative most has exactly the same complexity as non-conservative M.

Second, we compared the conservative quantifier not all, meaning $A \not\subseteq B$, with the non-conservative quantifier not only, meaning $B \not\subseteq A$. The mean complexity values over all 12 model sequences and a 95% confidence interval are plotted in Fig. 5. For model size 1 to 10 the descriptive statistics show that not all is more complex in 55.9% of the cases and less complex in 40.8% of the cases.

[18] Note, this dictionary base does not necessarily follow the order of the Latin alphabet.

[19] We report 95% confidence intervals obtained using standard nonparametric bootstrap resampling.

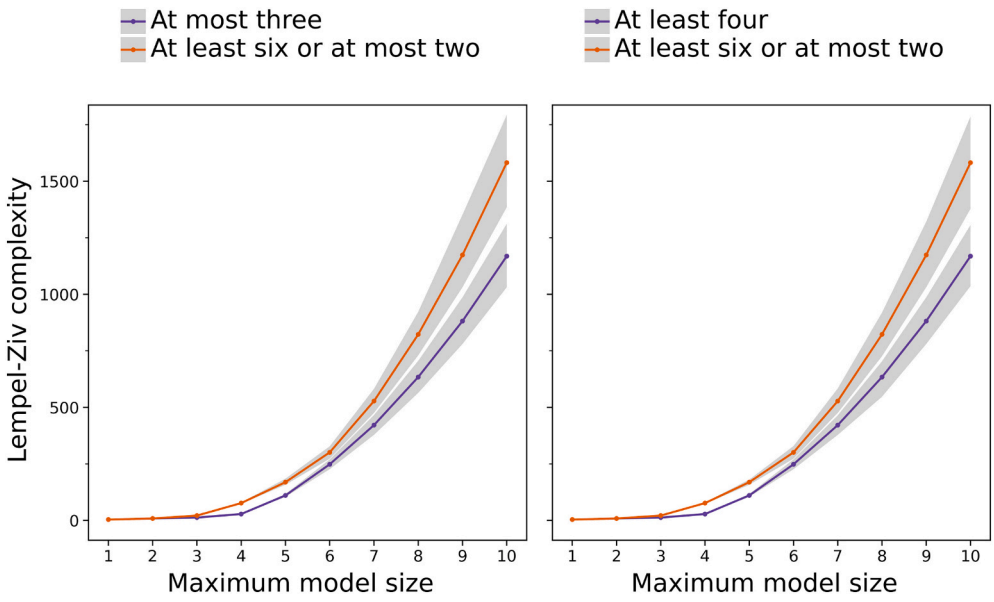

Fig. 3. Quantifier pairs to test the property of monotonicity. Complexity values for `at most three` and `at least six or at most two`, and for `at least four` and `at least six or at most two`. Mean complexity values with a 95% confidence interval over all 12 lexicographical model sequences.

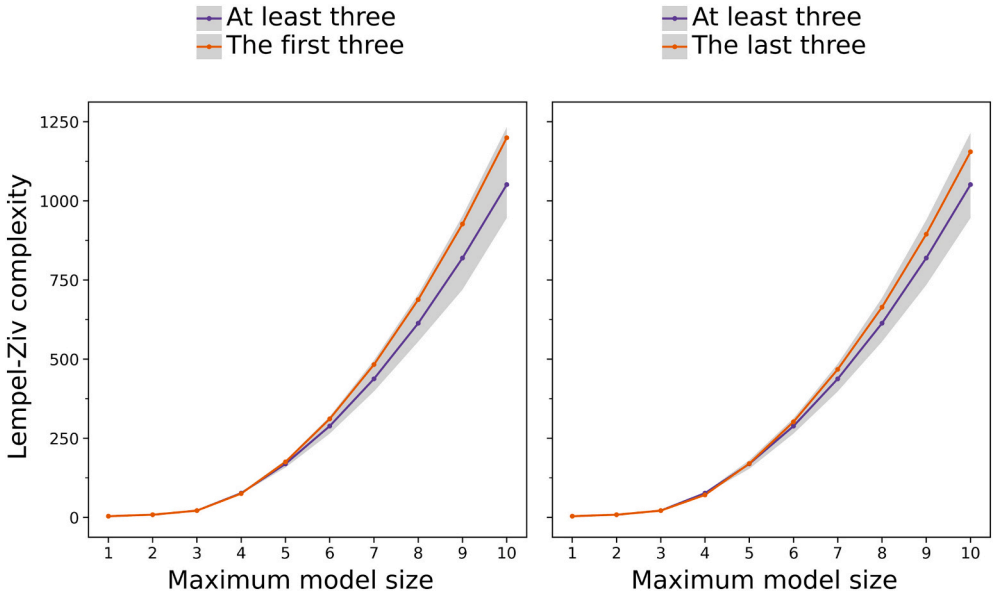

Fig. 4. Quantifier pairs to test the property of quantity. Complexity values for `at least three` and `the first three`, and for `at least three` and `the last three`. Mean complexity values with a 95% confidence interval over all 12 lexicographical model sequences.

These complexity results show a similar level of complexity for the conservative and the non-conservative quantifiers. This is in line with the learnability results in the study by Steinert-Threlkeld and Szymanik (2019), which found that the conservative quantifiers were of similar learnability as the non-conservative ones.

3.3. Discussion of Experiment 1

We applied tools from algorithmic information theory—in particular, approximate Kolmogorov complexity as measured by Lempel-Ziv complexity—to measure the complexity of minimal quantifier pairs, of which one quantifier satisfies a semantic universal and the other does not. We investigated whether quantifiers that satisfy semantic universals are simpler than those that do not and whether complexity could thereby explain the presence of semantic universals for quantifiers. We also looked at whether the complexity results for these quantifiers show similar patterns as existing learnability results.

We found that monotone quantifiers are robustly less complex than non-monotone quantifiers, and that conservative and non-conservative quantifiers have equal or similar complexity. For quantitative quantifiers we found a slight tendency towards being less complex, but this pattern was not robust. The results for monotonicity and conservativity agree with the learnability results for these quantifier pairs in the study by Steinert-Threlkeld and Szymanik (2019). The results on quantitativity are less robust, though they hint to a pattern in the same direction as the learnability results. Steinert-Threlkeld and Szymanik (2019) explain that for conservativity they did not expect a difference in learnability under their framework, because conservativity might rather be explained by different factors than learnabiltiy. It is an ongoing debate whether conservative quantifiers are indeed easier to learn.

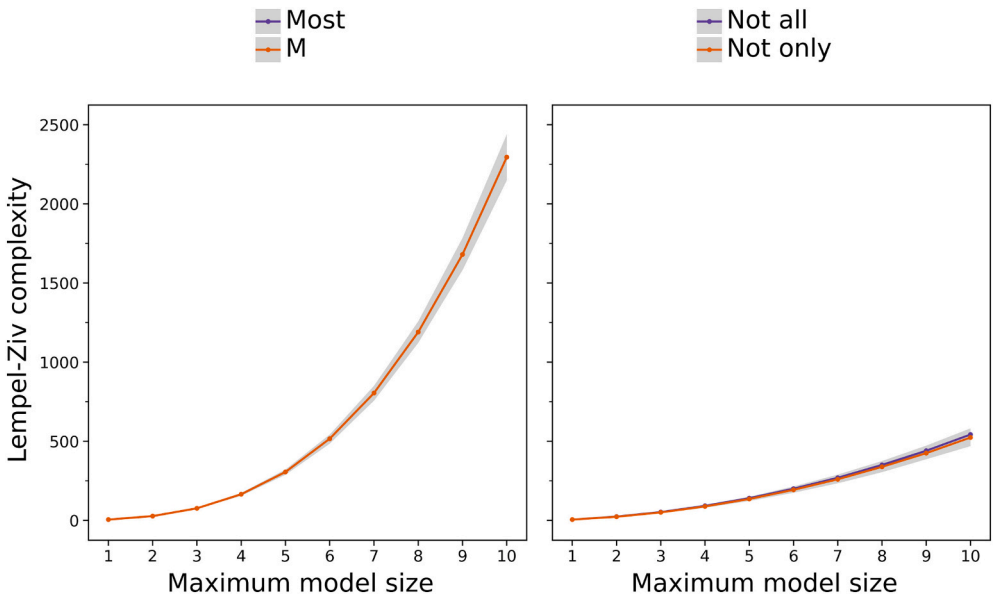

Fig. 5. Quantifier pairs to test the property of conservativity. Complexity values for `most` and `M`, and for `not all` and `not only`. Mean complexity values with 95% confidence interval over all 12 lexicographical model sequences.

While Hunter and Lidz (2013) found that conservative quantifiers were easier to learn for children, this effect was not found in a replication of their study by Spenader and de Villiers (2019). There is also work in linguistics that suggests that, contrary to the properties of monotonicity and quantity, conservativity might not arise from semantic pressures on the lexicon (such as simplicity and learnability) but follows, instead, from an interaction between syntax and semantics (Fox, 2002; Romoli, 2015; Sportiche, 2005). On this view, even if a non-conservative quantifier existed, the truth-conditions of sentences containing it would "appear" conservative. If that holds, one should not expect to find a difference in complexity. Future work should continue to explore parallels and differences between conservativity and the other universals.

Our results for monotonicity show that Lempel-Ziv complexity can indeed capture differences in complexity between quantifiers in the minimal pair methodology. Overall, the results of this minimal quantifier pair experiment are not decisive. A general drawback of the minimal pair methodology is its limited scale and its inability to scale up in a principled and automated manner. A limitation of Lempel-Ziv complexity is that its outcomes are sensitive to the order in which the quantifier models are placed in the (binary) meaning sequence of the quantifier. For monotonicity, all 12 lexicographical orders showed qualitatively similar results, while for quantity and conservativity, not all 12 lexicographical orders showed the same pattern. To overcome these limitations, we run a second simulation study, Experiment 2, in which we generate a large collection of both natural and artificial quantifiers in a principled manner, using a language of quantifier expressions defined by a logical grammar, and compute their complexities in terms of their minimal expression lengths in the grammar.

4. Experiment 2: language of minimal expression length

To further investigate the relation between the complexity of quantifiers and whether they adhere to universal properties, we implement a large scale study in which we use a logical grammar to generate a large collection of quantifiers, of which we measure both their minimal expression lengths and their Lempel-Ziv complexities and whether they adhere to the universal properties. Instead of directly comparing the complexities between individual quantifiers, such as in Experiment 1, here we look at the aggregate differences over the entire collection of quantifiers that we consider.

We generate the collection of quantifiers and determine their minimal expression lengths using similar methods as Steinert-Threlkeld (2020, 2021) use to study the complexity of quantifiers in the trade-off between simplicity and informativeness.[20]

4.1. Methods

We use the following methods to generate a large body of generalized quantifiers, in order to study their complexities in relation to the universal properties being present or not.

4.1.1. Grammar and language

To study the universal properties of quantifiers we need to look both at quantifiers that do and that do not have these properties. We use a principled way of generating a large space of generalized quantifiers with and without these properties. In particular, we use a logical grammar to generate quantifier expressions, over which the meaning of a quantifier[21] can be computed.

We define a simple yet expressive grammar that consists of basic building blocks and standard rules for how to combine them. In particular, the grammar is defined by the collection of operators presented in Table 1. These consist of standard set-theoretic operators ($\cup, \cap, \setminus, |\cdot|, \subseteq$), integer operators ($=, >$), and boolean operators ($\wedge, \vee, \neg$). In order to investigate the property of *quantity*, we need an operator that is sensitive to the ordering over objects. In fact, whenever we say set, we mean a tuple of a set and an ordering $<$ over the elements in the set. We include such an index-like operator, namely operator iota, $\iota(\cdot,\cdot)$, that, given a set and an index position, returns a singleton with the object at that index position, given the ordering over the set.

[20] We note that the concurrent and complementary Katzir, Lan, and Peled (2020) applies *the Minimum Description Length Principle* to learning quantifiers. Their representational format, however, is based on semantic automata (Van Benthem, 1986) and not on expression length in a logical grammar.

[21] The meaning of a quantifier is defined by the collection of models in which a quantifier expression is true. This is also called the *extension* of a quantifier.

Table 1
The collection of operators used to generate quantifier expressions. Note that the sets are accompanied by an ordering, which is of relevance for operator ι.

operator	type	gloss
$\cup$	SET $\times$ SET $\rightarrow$ SET	Union
$\cap$	SET $\times$ SET $\rightarrow$ SET	Intersection
$\setminus$	SET $\times$ SET $\rightarrow$ SET	Setminus
$\iota(\cdot,\cdot)$	INT $\times$ SET $\rightarrow$ SINGLETON SET	"Object at index"
$\lvert\cdot\rvert$	SET $\rightarrow$ INT	Cardinality
$\subseteq$	SET $\times$ SET $\rightarrow$ BOOL	Subset equal
$=$	INT $\times$ INT $\rightarrow$ BOOL	Integer equality
$>$	INT $\times$ INT $\rightarrow$ BOOL	Integer larger than
$\neg$	BOOL $\rightarrow$ BOOL	Negation
$\wedge$	BOOL $\times$ BOOL $\rightarrow$ BOOL	And
$\vee$	BOOL $\times$ BOOL $\rightarrow$ BOOL	Or

In addition to the operators, the grammar has two variables $\{A, B\}$—where A and B are placeholders for sets (with an ordering)—and a collection of constants $\{0, 1, \ldots, s\}$,[22] which represent integers. This grammar defines a formal language with quantifier expressions, which are the expressions that can be formed by using the given variables, constants, and operators, (adhering to the type restrictions of the operators) and that return a Boolean value. For example $A \subseteq B$ and $2 > |A \cap B|$ are expressions in this language.

We call the language defined by the operators in Table 1 language $\mathscr{L}_{+\iota}$. The number of quantifier expressions in this language grows exponentially in the maximum expression length ℓ. To manage this exponential blow-up we set the maximum expression length in language $\mathscr{L}_{+\iota}$ to 5. To be able to push the maximum expression length a bit further, in addition, we also consider the language that results from excluding the ι operator from the operators in Table 1, which we call language $\mathscr{L}_{-\iota}$. Excluding the ι operator reduces the number of different quantifier expressions per expression length, allowing us to generate all expressions up to length 7. For language $\mathscr{L}_{-\iota}$ we can only investigate the properties of monotonicity and conservativity, because all of its quantifiers are quantitative by definition, due to the exclusion of the index operator ι.

We note here that our measure of expression length depends on the choice of grammar and especially the primitives therein (Feldman, 2000; Goodman, 1955; Goodman et al., 2008; Piantadosi, Tenenbaum, & Goodman, 2016). While this dependence is unavoidable, we note several motivations for our particular choice of primitives.[23] Piantadosi et al. (2016) explicitly studied which choices of primitives best explain human concept learning. While no single grammar was best, the basic logical operators that we use here ($\wedge$, $\vee$, $\neg$) were shown there to be among the best performing set. Crucially, some logically equivalent systems (e.g., those with a single connective like NAND or NOR) were shown to perform worse at predicting human learning curves. We take this to show that the choice of logical connectives in our grammar is a particularly natural and motivated one. While less work studies explicitly set-theoretic concepts in the framework of a logical grammar, the tight formal connection between the set-theoretic operators union, intersection, and complement ($\cup$, $\cap$, $\setminus$) and the included logical connectives conjunction, disjunction, and negation ($\wedge$, $\vee$, $\neg$) makes these a natural choice for set-theoretic primitives. That being said, future work should (a) verify the robustness of our results to various choices of primitives and (b) study human quantifier learning as a basis for choosing such primitives, by analogy with the work by Piantadosi et al. (2016).

4.1.2. Minimal expression length

Quantifier expressions in these languages are not unique. The quantifier `at most one` can, for instance, be defined both by $(2 > |A \cap B|)$ and by $\neg(|A \cap B| > 1)$. The meanings of these expressions are equivalent: they are true in exactly the same models. We define the length of an expression by the number of operators in it. So the length of expression $(2 > |A \cap B|)$ is 3 and the length of expression $\neg(|A \cap B| > 1)$ is 4. The minimal expression length of a quantifier in this language is the length of the shortest expression for this quantifier.

We generate the collection of quantifier expressions of minimal expression length by the following procedure. We first generate all expressions of length 1, one by one, by going through the list of operators. For each expression we compute its meaning for all models from size 1 to size s. We compare this meaning to the meanings of expressions that we stored so far. If the meaning is not yet present, we add this expression and its meaning to our collection. If the meaning was already present, this means we already included an equivalent expression of equal or shorter length. Then we do not add it and continue with the next quantifier expression in line. When finished with all possible quantifiers of length 1, we continue with quantifiers of length 2 and repeat the procedure up to length ℓ. This way, we generated all 24,632 semantically unique quantifiers expressions, up to and including length 5, for language $\mathscr{L}_{+\iota}$, and all 22,287 semantically unique quantifier expressions, up to and including length 7, for language $\mathscr{L}_{-\iota}$. By virtue of this procedure, we know that all quantifier expressions in our collection are of minimal expression length: their meanings cannot be expressed by a shorter combination of operators in our grammar. Moving forward, we use the names language $\mathscr{L}_{+\iota}$ and language $\mathscr{L}_{-\iota}$ to denote only the collections of quantifier expressions of minimal expression length.

Due to the fact that we can only compute the meaning of quantifier expressions over a finite sequence of quantifier models, i.e., up to a maximum model size s, not all quantifier expressions of minimal expression length will be included in these languages. Some of the quantifier expressions that have an equivalent meaning to a shorter quantifier expression when evaluated in models up to size s, might not be equivalent when considering all possible models of arbitrarily large size. This means that there might be some model of size $s' > s$ for which the meaning of that quantifier expression is not equivalent to the meaning of the shorter expression. So while all quantifier expressions included in our collection are guaranteed to be of minimal expression length, irrespective of the maximum model size s that is considered, it is not guaranteed to include *all* quantifier expressions of minimal expression length when considering all possible models (of arbitrarily large size).

4.1.3. Encoding quantifier meanings as binary sequences

To compute and compare the meanings of the quantifier expressions, we generate binary representations of those meanings. We use the same procedure for this as described in Section 3.1.2: we first encode each model as a sequence of symbols and enumerate the models in the lexicographical order over their symbol representations. Then, given a quantifier expression, we put a `1` in the place of each model when the quantifier expression is true in that model, and a `0` when the quantifier expression is false in that model.

For these encodings we are assuming the property of *extensionality* (universe independence), i.e., that the subarea $M \setminus (A \cup B)$ does not matter for the meaning of a quantifier (Peters and Westerståhl, 2006). This means that instead of looking at the four subareas of a quantifier model, we now only need to look at three subareas and give them a label, say $A \cap B \mapsto d$, $A \setminus B \mapsto e$, and $B \setminus A \mapsto f$. This reduces the number of different quantifier models that are relevant for the meaning of a quantifier, and thereby reduces the length of the binary encoding of the quantifier meaning, which is now of length $|(d, e, f)|^s$ (instead of length $|(d, e, f, g)|^s$ in Section 3.1.2) for a given model size s.

In addition to minimal expression length, we also compute the Lempel-Ziv complexity of each quantifier, for comparison to the minimal expression length results in the current experiment and to the Lempel-Ziv complexity results in Experiment 1. As we explain in Section

[22] Where s is the maximum model size that is considered, which, to limit computational blow-up, we set to 8.
[23] Thanks to an anonymous referee for pushing us to clarify here.

3.1.2, there is not one unique lexicographical ordering of the models. In the case of three model areas, there are six such orderings, because the three model area labels $\{d, e, f\}$ can be ordered in $3! = 6$ ways (in contrast to the $4! = 24$ mappings in the case of four model areas, such as in Experiment 1). This result in six different sequences of the quantifier models, of which three are unique for our purposes.[24] For robustness, we look at all three uniquely different lexicographical orderings over $\{d, e, f\}$ and we report the mean Lempel-Ziv complexity over these three orderings in Section 4.2.2 and we report the complexities for each individual ordering in Appendix B.[25]

4.2. Results

Using the described procedures we generated a collection of 24,632 quantifiers for language $\mathscr{L}_{+\iota}$ and 22,287 quantifiers for language $\mathscr{L}_{-\iota}$. For each quantifier we computed whether they have the property of monotonicity, quantity, and conservativity, and we computed their complexity scores, both for minimal expression length (ML) and Lempel-Ziv complexity (LZ). To facilitate the comparison of the results for ML and LZ, we standardized the complexity data by computing their z-scores.[26] The code that we used for generating these data and the data themselves can be found at `https://github.com/ivdpol/QuantifierComplexity`.

We report two measures of descriptive statistics: (1) the average complexity of quantifiers with versus quantifiers without the universal property, and (2) the percentage of quantifiers with the universal property per minimal expression length.[27] In addition to considering the descriptive statistics, we performed logistic regressions for each universal property individually and for all properties taken together, with the universal property as dependent variable and complexity as the independent variable. The logistic regression model quantifies the relation between complexity and universal property, by estimating the probability that a quantifier satisfies a universal given the complexity of the quantifier. In the case of minimal expression length this means that the logistic regression model quantifies the relation between minimal expression length and universal property that is apparent from Fig. 6. The regression coefficient β of the (standardized) complexity indicates that a change of 1 standard deviation in complexity is associated with a change of β in log odds of a quantifier satisfying a universal.

We compared the coefficient value of complexity to the coefficient value of a random baseline, which we generated by randomly shuffling the actual complexity values over the different quantifiers and doing logistic regression over those randomly shuffled values. The coefficient value of the random baseline can be seen as quantifying the relation between a random property of the quantifiers and whether a quantifier satisfies a semantic universal. We compared the coefficient of the random baseline with the coefficient of complexity to see if complexity indeed has a different relation to the universal than a random property.

We used bootstrap resampling to compute a distribution over the regression coefficient of complexity and of the random baseline.[28] Note that we are not performing a significance test of a sample here, but rather aim to quantify the uncertainty of a difference in a whole population of simulated results. To achieve this, bootstrapping samples from that population and comparing that to randomly labeled bootstrap samples (the random baseline) yields an estimate of the uncertainty around the regression coefficient and whether this distribution of coefficients exceeds the distribution of coefficients observed by chance alone. We generated the distribution over the coefficient of the random baseline by randomly shuffling the actual complexity values over the different quantifiers, each time before taking a sample. We computed the coefficient of the original complexity data and of the random baseline in pairs, over the same random samples of quantifiers. To compare the coefficient of complexity to the coefficient of the random baseline, we plotted the distribution of the difference, per sample, between the coefficient of complexity and the coefficient of the random baseline. We report the mean of this distribution of the coefficient difference as a measure for the relation between complexity and universal property: if the mean is negative, that indicates that quantifiers satisfying the universal are indeed simpler.

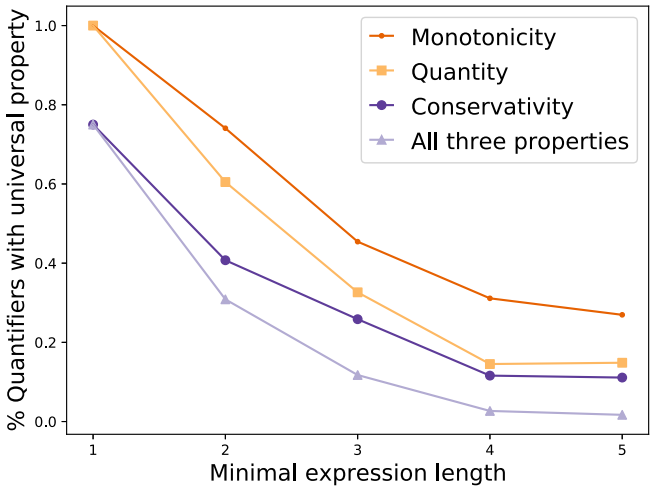

Fig. 6. Percentage of universal property per minimal expression length. For language $\mathscr{L}_{+\iota}$.

4.2.1. Minimal expression length (ML)

First, we present the results for the minimal expression length (ML) scores. Next, in Section 4.2.2, we present the results for the Lempel-Ziv complexities (LZ scores).

4.2.1.1. Language $\mathscr{L}_{+\iota}$(ML). Here we present the results for the collection of quantifiers generated by the grammar defined by the operators in Table 1, which we call language $\mathscr{L}_{+\iota}$. This collection contains 24,632 semantically unique quantifier expressions, with an expression length (i.e., the number of operators in a quantifier) ranging from 1 to 5. Because this grammar contains the index-sensitive operator ι, the collection contains quantifiers that are order dependent, like the quantifier `the first three`. This allows for investigating the property of quantity because the collection contains both quantifiers that do and that do not satisfy this property.

Descriptive Statistics (ML, $\mathscr{L}_{+\iota}$). The descriptive statistics show a negative relation between minimal expression length (ML) and universal property. Quantifiers that have all three properties have a lower average ML, i.e., they are less complex, than quantifiers that do not have all three properties (that have either two, one, or none of the properties). In addition, also for each individual universal property the average ML of quantifiers with that property is consistently lower than the average ML

[24] Of these six sequences three are the reverse of one of the other three. Since the Lempel-Ziv complexity measure C_{LZ} takes the average between the complexity of a sequence and the complexity of the reverse of that sequence (see Section 3.1.1), this leaves three uniquely different lexicographical model sequences over which we can compute this measure.

[25] Note that these orderings are relevant only for the Lempel-Ziv complexity and that they do not influence the minimal expression length.

[26] A z-score is obtained by subtracting the mean and dividing by the standard deviation for each value of the variable. This results in a standardized variable, which is a variable rescaled to have a mean of 0 and a standard deviation of 1.

[27] We only report this percentage for ML and not for LZ because the latter is not a discrete measure and can therefore not be used to make a meaningful contingency table by which this percentage can be computed.

[28] We use a sample size of 5000 quantifiers and we repeat the process for 20,000 random samples.

Table 2
Average (standardized) ML scores of quantifiers with (YES) versus without (NO) universal property and the proportion (%) of quantifiers with that universal property. *All* stands for quantifiers that have all three properties. The category "*all, NO*" stands for quantifiers that lack at least one property, i.e., quantifiers that do not have all three properties. For language $\mathscr{L}_{+\iota}$.

	YES	NO	%
monotonicity	−0.12	0.05	0.28
quantity	−0.15	0.03	0.15
conservativity	−0.16	0.02	0.12
all	−0.79	0.02	0.02

of quantifiers without that property. See Table 2 for an overview of average (standardized) ML scores $\mathscr{L}_{+\iota}$.

Furthermore, both for quantifiers with all three properties and for each universal property individually, the percentage of quantifiers with the universal property (per a given ML score) shows a negative relation to ML: the lower the minimal expression length, the higher the proportion of quantifiers with a universal property. See Fig. 6 for an overview of the percentage of universal property per minimal expression length for language $\mathscr{L}_{+\iota}$.

Logistic Regression (ML, $\mathscr{L}_{+\iota}$). In line with the descriptive statistics, the regression results for language $\mathscr{L}_{+\iota}$ show a negative relation between ML complexity and universal property, i.e., a positive relation between simplicity and universal property, both when looking at all three properties combined and for each property individually. For each of the four cases (monotonicity, quantity, conservativity, and all properties combined) the mean of the coefficient of the original (standardized) ML data has a negative sign, while the mean of the baseline (the randomly shuffled, standardized ML scores) is around zero. See the top panels of Fig. 7 for density plots of the coefficient value of the regressions over all 20,000 samples. The distribution of the difference (computed per sample) between the coefficient of the original (standardized) ML data and the randomly shuffled (standardized) ML scores has a 95% confidence interval that is entirely below zero (Fig. 7, bottom panels). The values of the mean and the 95% confidence interval of the coefficient difference between original and randomly shuffled data are as follows. All three properties: -0.43 (95% CI [−0.72, −0.19]); monotonicity: -0.15 (95% CI [−0.24, −0.06]); quantity: -0.15 (95% CI [−0.26, −0.04]); conservativity: -0.15 (95% CI [−0.28, −0.03]).

4.2.1.2. Language $\mathscr{L}_{-\iota}$(ML). To push the maximum expression length of the language a bit further,[29] we generated a collection of quantifiers by a slightly altered grammar, namely the grammar that results from excluding the ι operator from the operators in Table 1. Excluding the ι operator substantially limits the number of semantically unique expressions that the grammar produces, which results in a collection of quantifier expressions (computationally feasible to generate) with a larger maximal expression length, namely 7, instead of 5. This collection contains 22,287 semantically unique quantifier expressions, which we call language $\mathscr{L}_{-\iota}$. Because this grammar does not contain any index-sensitive operator, all of the quantifiers in this collection are order independent. This means that we can only investigate the properties of monotonicity and conservativity for this language, and not the property of quantity, because all of the quantifiers in this collection satisfy the property of quantity by definition.

Descriptive Statistics (ML, $\mathscr{L}_{-\iota}$). The descriptive statistics for ML in language $\mathscr{L}_{-\iota}$ show a similar qualitative result as for language $\mathscr{L}_{+\iota}$. They show a negative relation between minimal expression length (ML) and universal property. The quantifiers that have both properties have a lower average ML, i.e., they are less complex, than the quantifiers that do not have both properties (that have either one or none of the properties). In addition, also for each individual universal property the average ML of quantifiers with that property is consistently lower, i.e., less complex, than the average ML of quantifiers without that property. In fact, for the individual properties the average differences in complexity show a stronger result then for language $\mathscr{L}_{+\iota}$: the differences between quantifiers with versus without monotonicity or conservativity are larger. See Table 3 for an overview of average ML scores for language $\mathscr{L}_{-\iota}$.

The percentage of quantifiers with the universal property (per a given ML score) shows the same general trend in relation to ML as for language $\mathscr{L}_{+\iota}$. It shows a negative trend in the relation between the percentage of universal property and ML: the lower the minimal expression length, the higher the proportion of quantifiers with a universal property. Compared to language $\mathscr{L}_{+\iota}$ this downward pattern is less clearly pronounced, especially when looking at the properties individually, but note that the differences in average complexity for the individual properties are actually larger than for language $\mathscr{L}_{+\iota}$. When looking at quantifiers that have both properties, the pattern is very similar to that for language $\mathscr{L}_{+\iota}$. See Fig. 8 for an overview of the percentage of universal property per minimal expression length for language $\mathscr{L}_{-\iota}$.

Logistic Regression (ML, $\mathscr{L}_{-\iota}$). The regression results for ML in language $\mathscr{L}_{-\iota}$ show the same qualitative pattern as for language $\mathscr{L}_{+\iota}$. They show a negative relation between ML complexity and universal property, i.e., a positive relation between simplicity and universal property, both when looking at both properties combined and for each individual property. See the top panels of Fig. 9 for density plots of the coefficient value of the regressions over all 20,000 samples. For each of the three cases (monotonicity, conservativity, and both properties combined) the mean of the coefficient of the original (standardized) ML data has a negative sign, while the mean of the baseline (the randomly shuffled, standardized ML scores) is around zero. The regression results for language $\mathscr{L}_{-\iota}$ show a stronger effect than for language $\mathscr{L}_{+\iota}$. For language $\mathscr{L}_{-\iota}$ the distribution of the difference (computed per sample) between the coefficient of the original (standardized) ML data and the randomly shuffled (standardized) ML scores is entirely below zero for all of the samples (Fig. 9, bottom panels). The values of the mean and the 95% confidence interval of the coefficient difference between original and randomly shuffled data are as follows. Both properties: -0.47 (95% CI [−0.61, −0.33]); monotonicity: -0.41 (95% CI [−0.50, −0.32]); conservativity: -0.27 (95% CI [−0.38, −0.16]).

4.2.1.3. Summary Minimal Expression Length (ML). For all three semantic universals, these results show that, in general, quantifiers satisfying the universal properties have a lower minimal expression length, i.e., are simpler, than those that do not.

4.2.2. Lempel-Ziv complexity (LZ)

In addition to minimal description length, we also measured the Lempel-Ziv complexity (LZ) of the quantifiers, similarly as in Experiment 1. As explained in Section 4.1.3, we compute the LZ scores of the quantifiers over the three different lexicographical orderings of the quantifier models, which we refer to as LZ_0, LZ_1, and LZ_2 scores. The LZ results for each of the orderings are very similar. In this section we report the mean values over LZ_0, LZ_1, and LZ_2. For the sake of readability we refer to these mean LZ scores simply by LZ. The individual results for the LZ_0, LZ_1, and LZ_2 scores can be found in Appendix B.

4.2.2.1. Language $\mathscr{L}_{+\iota}$(LZ). First, we present the LZ results for language $\mathscr{L}_{+\iota}$, which is a collection of 24,632 quantifier expressions with an expression length ranging from 1 to 5. Next, in Section 4.2.2.2, we present the LZ results for language $\mathscr{L}_{+\iota}$, containing quantifier expressions with an expression length ranging from 1 to 7.

Descriptive Statistics (LZ, $\mathscr{L}_{+\iota}$). The descriptive results for LZ in

[29] This is the maximum value of the minimal expression lengths of the quantifiers in the language.

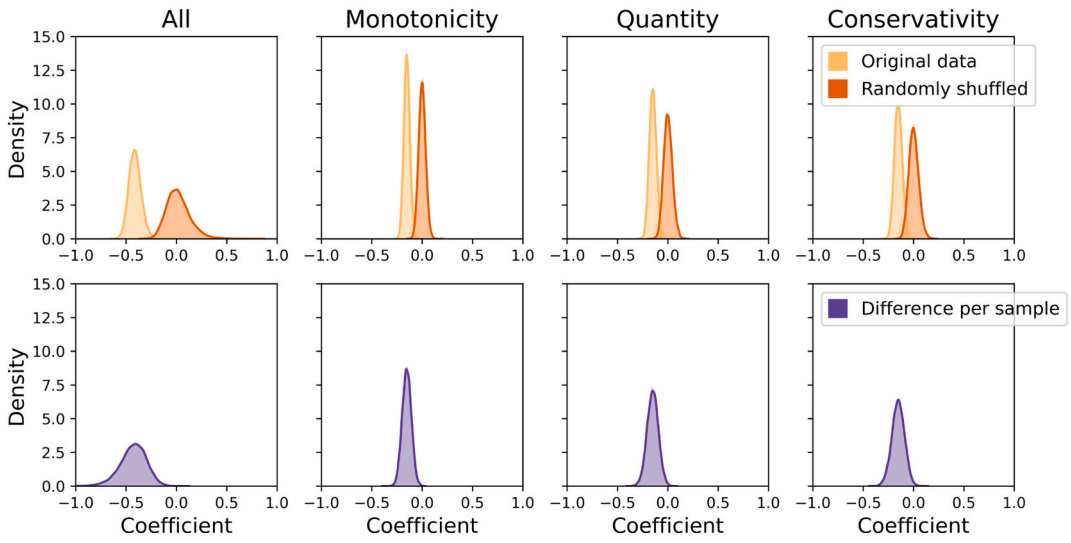

Fig. 7. Bootstrapped logistic regression results for standardized ML scores for language $\mathscr{L}_{+\iota}$.

Table 3
Average standardized ML scores of quantifiers with (YES) versus without (NO) universal property and the proportion (%) of quantifiers with that universal property. *Both* stands for quantifiers that have both properties. The category "both, NO" stands for quantifiers that lack at least one property (i.e., that do not have both properties). For language $\mathscr{L}_{-\iota}$.

	YES	NO	%
monotonicity	−0.28	0.14	0.33
conservativity	−0.27	0.05	0.14
both	−0.60	0.05	0.08

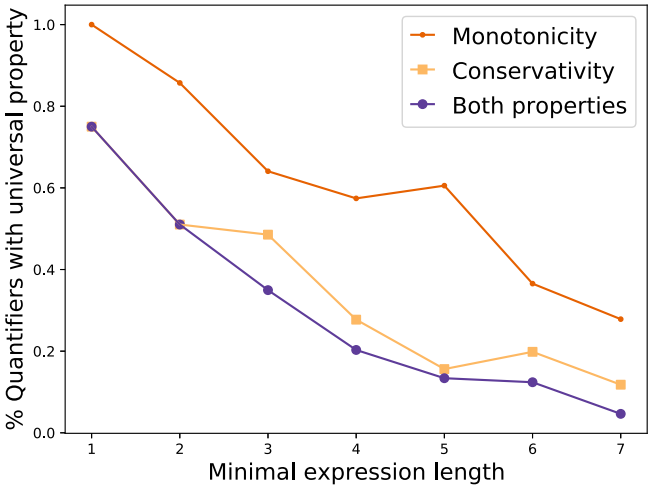

Fig. 8. Percentage of universal property per minimal expression length. For language $\mathscr{L}_{-\iota}$.

language $\mathscr{L}_{+\iota}$ show a different pattern than those for the ML scores. The relation between Lempel-Ziv complexity and universal property differs greatly per individual property. Similar to the ML scores, the LZ scores show a negative relation between LZ and monotonicity. Monotone quantifiers have a lower average LZ complexity, i.e., they are less complex, than quantifiers that are non-monotone. Conservative quantifiers, on the other hand, show the opposite relation, they have a higher average LZ complexity than non-conservative quantifiers. The LZ scores show no relation between quantity and LZ complexity: quantitative quantifiers have a similar average LZ complexity than non-quantitative quantifiers. Finally, quantifiers that have all three properties have a higher average LZ complexity than quantifiers that do not have all three properties (that have either two, one, or none of the properties). See Table 4 for an overview of average LZ scores for language $\mathscr{L}_{+\iota}$.

Logistic Regression (LZ, $\mathscr{L}_{+\iota}$). The regression results for LZ in language $\mathscr{L}_{+\iota}$ show a similar mixed pattern as the descriptive statistics. Like for the ML scores, the regression results for LZ show a negative relation between LZ complexity and monotonicity. The regression results for conservativity, on the other hand, show a positive relation between LZ complexity and monotonicity. The regression results for quantity show no relation between complexity and quantity. The regression results for all three properties taken together shows a weakly positive relation. See the top panels of Fig. 10 for density plots of the coefficient value of the regressions over all 20,000 samples. The values of the mean and the 95% confidence interval of the coefficient difference (computed per sample) between original and randomly shuffled data are as follows (Fig. 10, bottom panels). All three properties: 0.16 (95% CI [−0.12, 0.45]); monotonicity: -0.32 (95% CI [−0.41, −0.22]); quantity: 0.02 (95% CI [−0.10, 0.14]); conservativity: 0.18 (95% CI [0.06, 0.31]).

4.2.2.2. Language $\mathscr{L}_{-\iota}$ (LZ). Here we present the results for language $\mathscr{L}_{-\iota}$, which is a collection of 22,287 quantifiers with an expression length ranging from 1 to 7. Note that this is the language defined by the grammar that results from excluding the ι operator from the operators in Table 1.

Descriptive Statistics (LZ, $\mathscr{L}_{-\iota}$). The descriptive statistics for LZ in language $\mathscr{L}_{-\iota}$ partly show a similar pattern to those for language $\mathscr{L}_{+\iota}$. The LZ scores show no relation between LZ and monotonicity, instead of the negative relation between LZ and monotonicity for language $\mathscr{L}_{+\iota}$. In language $\mathscr{L}_{-\iota}$, monotone quantifiers have a very similar LZ complexity, than quantifiers that are non-monotone. Conservative quantifiers, on the other hand, have a higher average LZ complexity than non-conservative quantifiers, even more strongly so than for language $\mathscr{L}_{+\iota}$. The average LZ complexity of quantifiers that have both properties is higher than that of quantifiers that do not have both properties (that have either one or none of the properties). See Table 5 for an overview of average LZ scores for language $\mathscr{L}_{-\iota}$.

Logistic Regression (LZ, $\mathscr{L}_{-\iota}$). The logistic regression results for language $\mathscr{L}_{-\iota}$ show a similar qualitative pattern as the descriptive statistics. They show a weakly negative relation between LZ and monotonicity, and a positive relation for conservativity and for both properties taken together. See the top panels of Fig. 11 for density plots of the coefficient value of the regressions over all 20,000 samples. The values of the mean and the 95% confidence interval of the coefficient difference (computed per sample) between original and randomly

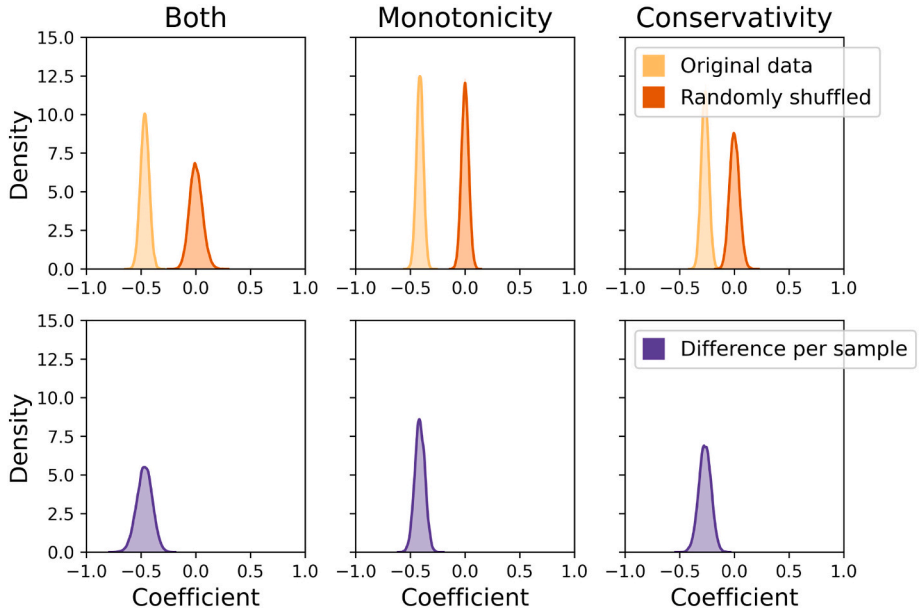

Fig. 9. Bootstrapped logistic regression results for standardized ML scores for language $\mathscr{L}_{-t}$.

Table 4
Average standardized LZ scores of quantifiers with (YES) versus without (NO) universal property and the proportion (%) of quantifiers with that universal property. *All* stands for quantifiers that have all three properties. The category "all, NO" stands for quantifiers that lack at least one property (i.e. which do not have all three). For language $\mathscr{L}_{+t}$.

	YES	NO	%
monotonicity	−0.22	0.08	0.28
quantity	0.02	0.00	0.15
conservativity	0.16	−0.02	0.12
all	0.16	0.00	0.02

shuffled data are as follows (Fig. 11, bottom panels). Both properties: 0.23 (95% CI [0.08, 0.38]); monotonicity: -0.08 (95% CI [−0.16, 0.01]); conservativity: 0.41 (95% CI [0.29, 0.52]).

4.2.2.3. Summary Lempel-Ziv Complexity (LZ). The results for Lempel-Ziv complexity show a different picture for each universal property. In general, they show a negative relation between LZ and monotonicity, a positive relation between LZ and conservativity, and no relationship between LZ and quantity.

4.3. Discussion of Experiment 2

We used a simple yet expressive logical grammar to generate two large collections of logically possible quantifiers, we measured their

Table 5
Average standardized LZ scores of quantifiers with (YES) versus without (NO) universal property and the proportion (%) of quantifiers with that universal property. *Both* stands for quantifiers that have both properties. The category "both, NO" stands for quantifiers that lack at least one property (i.e., that do not have both properties). For language $\mathscr{L}_{-t}$.

	YES	NO	%
monotonicity	−0.05	0.03	0.33
conservativity	0.34	−0.06	0.14
both	0.22	−0.02	0.08

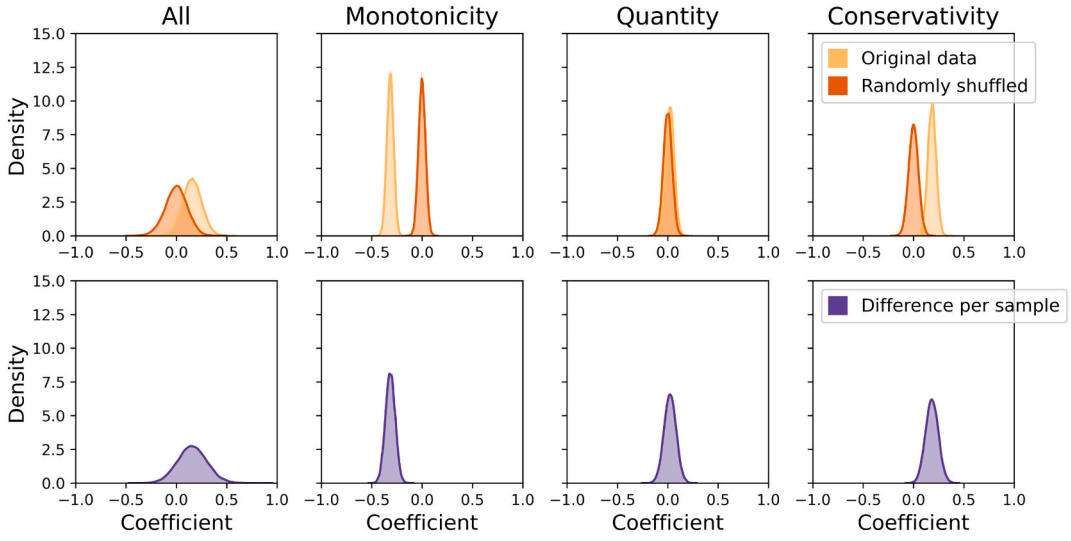

Fig. 10. Bootstrapped logistic regression results for standardized LZ scores for language $\mathscr{L}_{+t}$.

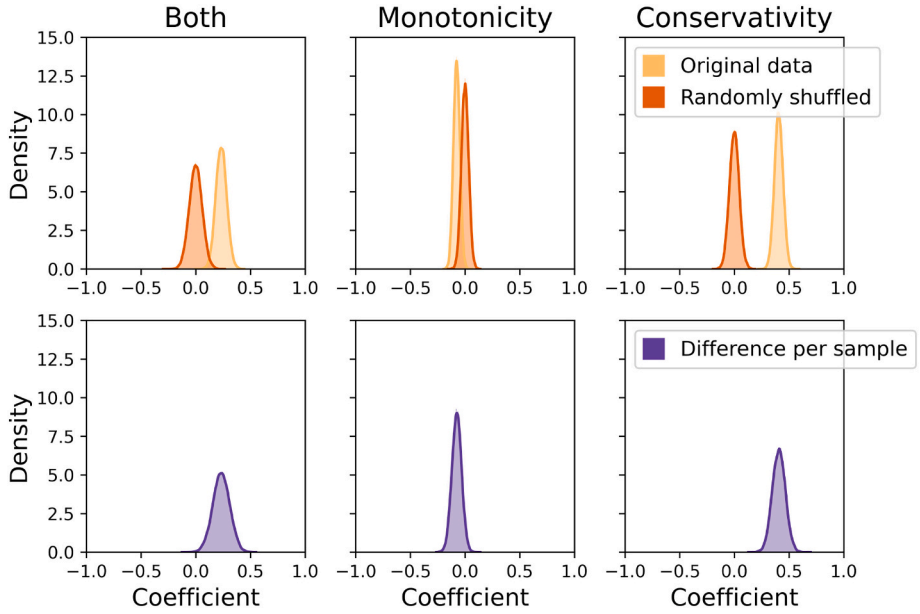

Fig. 11. Bootstrapped logistic regression results for standardized LZ scores for language $\mathscr{L}_{-i}$.

complexities both in terms of their minimal expression lengths and their Lempel-Ziv complexities, and identified whether they have the properties of monotonicity, quantity, and conservativity. We investigated whether quantifiers that satisfy semantic universals are simpler than those that do not. With respect to minimal expression length we found for each of these universal properties that quantifier expressions that satisfy them are simpler: they have a shorter minimal expression length. This suggests the following explanation for semantic universals in the domain of quantifiers: meanings satisfying semantic universals are simpler.

With respect to Lempel-Ziv complexity we found diverging results. Similar to minimal expression length, we found that monotone quantifiers are simpler than non-monotone quantifiers: they have a lower Lempel-Ziv complexity. Conservative quantifiers, on the other hand, were found to have a higher Lempel-Ziv complexity than non-conservative quantifiers and we found no difference in Lempel-Ziv complexity between quantitative and non-quantitative quantifiers.

These results are somewhat explainable in terms of the nature of the two measures. At a high level, Lempel-Ziv complexity searches for regularity in the distribution of truth values across the ordered list of models, i.e., a regularity in the distribution of in which models a given quantifier is true or false, given an enumeration of the models. Monotonicity provides such regularity, since models that stand in the super- or submodel relation will be enumerated in a predictable way.[30] By contrast, quantity says that the same truth value gets assigned to all models which arise via shuffling the underlying ordering of objects in a model; these will occupy myriad different positions in the overall ordering of the model sequence, and so this constraint does not introduce much regularity. Conservativity says that particular pairs of models must agree on the truth value; this "smaller-scale" constraint (on pairs of models instead of on the whole sequence of models) will also not induce much regularity in the overall ordering of the model sequence.

While both Lempel-Ziv complexity and minimal expression length can be motivated by intuitions about compression, the two measures are searching for structure in different places. Both measures can be seen as capturing regularities or structure in the meaning of quantifiers, i.e., in the mapping between the space of possible models and the truth values of a quantifier in those models. While Lempel-Ziv complexity looks at a binary representation of this mapping, minimal expression length uses a language to represent the mapping from models to truth values *directly*, instead of first turning this mapping into a binary sequence. Conceptually, one can think of Kolmogorov complexity—and thus its approximation by Lempel-Ziv complexity—as measuring the length of the shortest program to generate the binary sequence representing the mapping from models to truth values, in comparison, one could think of minimal expression length as measuring the length of the shortest program to generate the mapping from models to truth values directly (without first encoding that mapping into a binary sequence). Because these two approaches take different inputs, they also have different primitives in their respective "programming languages".[31]

Finally, the results of our experiment are not a priori obvious, given that the grammar can generate quantifiers that have a relatively low expression length while they do not satisfy one or more of the universal properties (see Fig. 6 and Fig. 8). So while not *every* quantifier with a short minimal expression length satisfies the universal properties, our results show that quantifiers that satisfy semantic universals do, *overall*, have a shorter minimal expression length.

5. Discussion

We performed two simulation studies to investigate the relation between the complexity of quantifiers and whether they adhere to universal properties. In Experiment 1, we measured the Lempel-Ziv complexity of quantifiers in a small-scale study using a minimal pair methodology. In Experiment 2, we used a logical grammar to perform a large-scale study in which we measured both the minimal expression length and the Lempel-Ziv complexity of quantifiers. Using Lempel-Ziv complexity to measure the complexity of quantifiers and using the minimal expression lengths of quantifiers to investigate semantic universals are, to our knowledge, novel applications of these frameworks.

[30] For instance, in models of size 3, the model with $A = \{0,1\}$ and $B = \{2\}$ gets assigned sequence *eef*, and the model with $A = \{0,1\}$ and $B = \{0,2\}$ (a supermodel of the former) gets sequence *def*, according to the model encoding described in Section 3.1.2 and Section 4.1.3. The position of the latter sequence in the lexicographical order of models is predictable from the former.

[31] See Grünwald, Myung, and Pitt (2005) for more on the relationship between description length and Kolmogorov complexity. We thank an anonymous reviewer for helpful discussion here.

How these two complexity measures relate to each other in this setting and whether they measure the same aspects of the complexity of quantifiers is an open question that has not been studied before (see Section 4.3 for more discussion of their relation in this context).

While Lempel-Ziv complexity and minimal expression length both measure a form of the complexity of the meaning, i.e., the semantics, of quantifiers, interestingly we found diverging results between them. The Lempel-Ziv results in Experiment 2 (partly) diverge from the Lempel-Ziv results in Experiment 1 and from the minimal expression length results in Experiment 2. The results for monotonicity show the same pattern between Experiment 1 and 2, and between the different measures and the different languages within Experiment 2. For conservativity, in Experiment 2 we found that conservative quantifiers were more complex in terms of Lempel-Ziv complexity, while we found no difference in Lempel-Ziv complexity for conservativity in Experiment 1, and we found that conservative quantifiers were simpler in terms of minimal expression length in Experiment 2. For quantitative quantifiers we found no difference in Lempel-Ziv complexity in Experiment 2, while we found a non-robust tendency towards being more simple in terms of Lempel-Ziv complexity in Experiment 1, and we found that quantitative quantifiers were robustly simpler in terms of minimal expression length in Experiment 2.

The fact that we found differences between Experiment 1 and 2 is relatively unsurprising, given that both the experimental setup and the scale on which they operate are entirely different. What is more surprising, however, is the differences between the two different complexity measures within Experiment 2 itself, since they both operate on exactly the same quantifiers in the same experimental setup. When we look at the differences between the results for the two languages that we considered, the complexity results for minimal expression length for language $\mathscr{L}_{-\iota}$—which is the collection of quantifiers with a higher maximum value of the expression length—are of the same pattern and with a larger effect size (larger differences in complexity between quantifiers with versus without universal properties) than for language $\mathscr{L}_{+\iota}$. This indicates that the distribution of minimal expression length (with respect to universal property) scales robustly over larger expressions. Interestingly, for Lempel-Ziv complexity the effect sizes (for monotonicity) were actually smaller and less clear for the language $\mathscr{L}_{-\iota}$, indicating a less robust scaling of the results over larger expressions.

It is hard to say why exactly we find these differences between the two different complexity measures. The fact that we found different result for Lempel-Ziv complexity between Experiment 1 and Experiment 2 suggests that Lempel-Ziv complexity might not scale robustly. In principle, it could be the case that Lempel-Ziv complexity gives different results for smaller quantifier expressions than for larger quantifier expressions, and that when significantly scaling the length of the quantifiers—which is quite challenging due to exponential increase in the required computational resources—its results move closer to those of minimal expression length. Our measure of minimal expression length stems from the language of thought framework, which has a long history of providing robust explanations for human concept learning (Feldman, 2000; Goodman et al., 2008; Piantadosi et al., 2016), and in the current study it provided robust results in the explanation of semantic universals. While Lempel-Ziv complexity and other forms of approximate Kolmogorov complexity have also proven to be useful in several domains of cognitive science (Chater and Vitányi, 2003; Feldman, 2016; Planton et al., 2021), the Lempel-Ziv complexity of quantifiers shows diverging results over different experimental scales and settings. While future work should investigate whether Lempel-Ziv complexity could provide more robust results when further scaling up our experimental setting, these results are also broadly consistent with those of Planton et al. (2021), who found that a measure of complexity based on the language of thought approach out-performed Lempel-Ziv complexity in predicting behavior in a sequential memory task.

Two factors of interest for such an investigation are the following. First, as mentioned in Section 3.3, a limitation of Lempel-Ziv complexity as a complexity measure for quantifiers is that it is sensitive to the order in which the quantifier models are placed in the (binary) meaning sequence of the quantifier expression. To keep the computations of the quantifier meanings feasible for such a large collection of quantifiers, in Experiment 2 we looked at three subareas of the model instead of four (assuming the property of extensionality, i.e., excluding the subarea $M \setminus (A \cup B)$ from our model representations). Ideally, future work includes pushing our computations in Experiment 2 further to look at models with four subareas instead of three and compare those to the current results. Second, as mentioned in Section 4.1.2, our procedure guarantees that all quantifier expressions in the collections are of minimal expression length, but they are not the complete collections of *all* quantifier expressions of minimal expression length (up to length ℓ) when considering all possible models of arbitrarily large size. It is infeasible to guarantee the completeness of any collection of quantifier expressions of minimal expression length. Instead, future work could include pushing the maximum model size a bit further and comparing results over series of different model sizes, to investigate whether the results scale robustly over larger model sizes.

To arbitrate which of these measures is cognitively the most relevant one in the explanation of semantic universals of quantifiers, ideally they should be related to human performance on processing quantifiers with versus without universal properties. A challenge here is that, due to the ubiquity of these properties in natural language quantifiers, such empirical investigations are only possible in artificial learning experiments in which subjects need to learn novel words (see, e.g., Hunter & Lidz, 2013; Maldonado & Culbertson, 2021). Existing work has shown that children are sensitive to the syntactic environment or category of novel words when learning superlatives, using determinerhood versus adjectivehood to distinguish between quantity-based meanings and quality-based meanings of the novel superlative (Wellwood, Gagliardi, & Lidz, 2016). Our measures of complexity are not sensitive to the syntactic distribution of natural language expressions but are defined solely in terms of their meaning; future work should build models which incorporates both. On the other end of the spectrum, Chemla, Buccola, and Dautriche (2019) has shown that monotone quantifiers are easier to learn than connected (roughly: the conjunction of monotone) quantifiers, both of which are easier than entirely non-monotone quantifiers. To the extent that these learning results pattern with our complexity results, this provides further support for the cognitive reality of our measures. One further complication that needs to be taken into account when analyzing artificial word learning studies is that the results of such experiments might be influenced by the subjects' bias towards the properties that are prevalent in their native language. When the novel quantifiers are more difficult to process for subjects, this could be simply due to such a bias towards what subjects are used to, instead of to the properties themselves being cognitively simpler or more complex to process.

The descriptive statistics showed that there are quite a few expressions in language $\mathscr{L}_{+\iota}$ with the relatively short expression length of 2, that do not satisfy one or more universal properties (see Fig. 6 and Fig. 8). The majority of these expressions include the ι operator—which takes as input and integer i and a set P and returns a singleton with the i-th object in P—and all but one of these expressions include an integer constant. For example, for $i \in \{1, \ldots, 8\}$ the expression $A \subseteq \iota(i, B)$ is an expression in language $\mathscr{L}_{+\iota}$ that has length 2 and that does not satisfy monotonicity, quantity, or conservativity. To our knowledge, there is no quantifier attested in natural language that expresses this meaning, which could be described by "either there is no A or there is exactly one A, which is the i-th B." The prevalence of the ι operator in these expressions suggests that the ι operator might be a less basic operator. Future work could include refining the definition of expression length by assigning different weights to the operators, and possibly assigning extra weight to the ι operator.

We used the framework of generalized quantifiers because it is a well-defined and well-studied framework for representing the (literal)

meaning of quantifiers. Our aim in defining the grammar that we used to build a large collection of quantifier expressions, was to keep it as basic as possible, while at the same time capturing a significant part of natural language quantifiers and in addition also going beyond natural language (see also the discussion in Section 4.1.1). Where possible, we avoided complex operators that are combinations of more basic operators, thereby not including single operators for, i.e., "is an empty set" or "is of an even number." Since there are multiple collections of basic set-theoretical and logical operators that are definable in terms of each other—i.e., that in the infinite case define the same collection of expressions—there is not just one unique grammar that satisfies these objectives. Future work includes investigating such alternative grammars and comparing the results.

Relatedly, one may object that a certain circularity appears here: perhaps our set-theoretic formalisms for expressing the meanings of quantifiers look the way that they do because the relations denoted by quantifiers require them.[32] There are minimally three things to say in response. First, this further motivates work on vindicating the choice of primitives in the grammar (see Section 4.1.1 for discussion). Second, Kemp (2012) found that a grammar built on set-theoretical operators—more specifically, a grammar based on predicate logic—can account for concept learning across a variety of different conceptual domains. Their results indicate that grammars without set-theoretic operators—namely, grammars based on propositional logic—are not suited to account for learning across the same breadth of conceptual domains. This suggests that the explanatory function of our set-theoretic formalisms go beyond the meaning of quantifiers. Third, and more fundamentally, we find it much more probable that instead of circularity there is a common cause situation. We know that there is a pre-verbal representation of sets of objects over and above individual objects (Ariely, 2001; Whitney & Yamanashi Leib, 2018). It seems very plausible that basic cognitive operations for manipulating set representations were recombined into meanings for quantifiers (this is the idea behind the language of thought approach in general; Goodman, Tenenbaum, & Gerstenberg, 2015) and also were explicitly formalized in set theory.

Our results focus only on the universal properties that have been identified in the literature thus far. Another possible, and more general, constraint on quantifiers to analyze would be: *all (quantified) determiners express quantifiers whose meaning depends both on set A and set B (in the generalized quantifier model* $\langle M, A, B, <\rangle$), i.e., that quantifier meanings where either set A or set B is irrelevant to the truth value of the quantifier are not lexicalised (as mono-morphemic words).[33] A counting argument can show that such meanings are exceedingly rare: while there are 2^{4^n} quantifiers on models of size n, only 2^{2^n} of those do not depend on set A and another 2^{2^n} do not depend on set B.[34] That being said, future work should explore whether the explanations offered in this paper also explain this property, or whether its source may differ.

Both the descriptive statistics and the logistic regression results for minimal expression length show a robust difference in complexity between quantifiers with versus without the universal properties. This suggests that a bias for simplicity might indeed be an explanatory factor for these semantic universals. At the same time, a bias towards simplicity is likely not the only force at play in shaping the semantic properties of quantifiers. Other likely candidates that could play a role in either pushing towards or away from these properties are cultural evolution (Carcassi, Steinert-Threlkeld, & Szymanik, 2019) and communicative needs (Steinert-Threlkeld, 2020, 2021). On this front, these latter two works study quantifier universals in a very similar setting to the present one: generating expressions from a grammar and measuring complexity via minimal expression length. There are three major differences in that work: (i) they analyze *sets* of quantifiers, while (ii) also measuring communicative cost of an entire set of quantifiers, and (iii) use different dependent variables. The present paper shows that minimal expression length *on its own* and *at the level of individual quantifiers* can be correlated with the presence of the universals. This shows that many of the other pressures (e.g., the need for a set of quantifiers to cover a large space of communicative needs) may not be necessary for explaining the semantic universals in question. More generally, in what way a simplicity bias shapes (the learning of) semantic systems in various domains precisely, how it functions within the trade-off between informativeness and simplicity at the level of a whole language and how that influences the lexicon is an ongoing debate (Carr et al., 2020; Chaabouni, Kharitonov, Dupoux, & Baroni, 2021; Denić, Steinert-Threlkeld, & Szymanik, 2022; Enguehard & Spector, 2021; Galdo, Sloutsky, & Turner, 2021; Steinert-Threlkeld, 2020; Zaslavsky, Maldonado, & Culbertson, 2021).

6. Summary and conclusions

We investigated the complexity of quantifiers in relation to semantic universals. We studied whether a bias towards simplicity can explain the semantic universals of monotonicity, quantity, and conservativity.

We analyzed the minimal expression length of a large collection of quantifiers and found for all three universals that quantifiers satisfying them are simpler: they have a shorter minimal expression length. We found monotone quantifiers to be consistently simpler than non-monotone quantifiers for two different measures of complexity and in two different experimental setups. For quantity and conservativity we found different results between the small-scale and the large-scale setting, and between the two measures of complexity: minimal expression length and approximate Kolmogorov complexity, as measured by Lempel-Ziv complexity. These differences motivate future work on independently validating these measures of complexity: to the extent that minimal expression length as a measure of the simplicity of quantifiers can be validated on independent grounds (by, e.g., the kinds of learning results discussed in Section 5), these results would provide support for the notion that universals arise partially due to simplicity.

We also found preliminary evidence showing that the complexity and learnability of quantifiers pattern together. A natural follow-up experiment would involve investigating the learnability over a large collection of quantifiers for further comparison between the simplicity and learnability of quantifiers in the context of semantic universals. Similar methods as used by Steinert-Threlkeld and Szymanik (2020) could be used to investigate the learnability of the collection of quantifiers that we considered here.

CRediT authorship contribution statement

Iris van de Pol: Conceptualization, Methodology, Software, Formal analysis, Data curation, Writing – original draft. **Paul Lodder:** Methodology, Software. **Leendert van Maanen:** Methodology, Writing – review & editing, Supervision. **Shane Steinert-Threlkeld:** Conceptualization, Methodology, Writing – review & editing, Supervision. **Jakub**

[32] We thank an anonymous reviewer for suggesting this.

[33] We are grateful to an anonymous referee for suggesting this.

[34] The number of quantifiers can be counted as follows. Let m be the number of subareas of a model, and let n be the number of objects in a model. Then there are m^n many different models (because each object can be in one of the m different model areas). A quantifier is defined by a distribution of truth values over the models: in each model a quantifier is either true or false. Then there are 2^{m^n} many different quantifiers. A similar procedure can be used to count the number of quantifiers whose meaning does not depend on set A. The meaning of such a quantifier is solely determined by whether an object is in set B or not, thereby reducing the number of relevant model areas to two (namely B and $M \setminus B$). Therefore, there are 2^{2^n} quantifiers that do not depend on A, and, vice versa, similarly for B. For the models in Experiment 1, where we represent quantifiers over models with four subareas, i.e., not assuming the property of extensionality, there are 2^{4^n} quantifiers of size n. For the models we consider in Experiment 2, where we represent quantifiers over models with three subareas, i.e., assuming the property of extensionality, there are 2^{3^n} quantifiers of size n. The same argument holds in both cases.

Szymanik: Conceptualization, Methodology, Writing – review & editing, Supervision, Funding acquisition.

Declaration of Competing Interest

The authors declare that they have no conflict of interest.

Acknowledgements

We thank three anonymous reviewers for their useful feedback. We thank Fausto Carcassi, Nima Motamed, the CoSaQ research group, and the Meaning, Logic, and Cognition seminar at the Institute for Logic, Language and Computation, the Computational Cognitive Science group at the Donders Centre for Cognition, Thomas Icard, Michael Hahn, Steven Piantadosi, and the CoCoLab and CoLaLa research groups at Stanford University and UC Berkeley, respectively, for interesting discussions and useful feedback. Iris van de Pol was supported by Gravitation Grant 024.001.006 of the Language in Interaction Consortium from the Netherlands Organization for Scientific Research. Jakub Szymanik received funding from the European Research Council under the European Union's Seventh Framework Programme (FP/2007–2013) / ERC Grant Agreement n. STG 716230 CoSaQ. The computational work was carried out on the Dutch national e-infrastructure with the support of SURF Cooperative.

Appendix A. Experiment 1

This appendix contains the individual complexity plots (for all 12 lexicographical model sequences) for all minimal quantifier pairs in Experiment 1.

A.1. Monotonicity

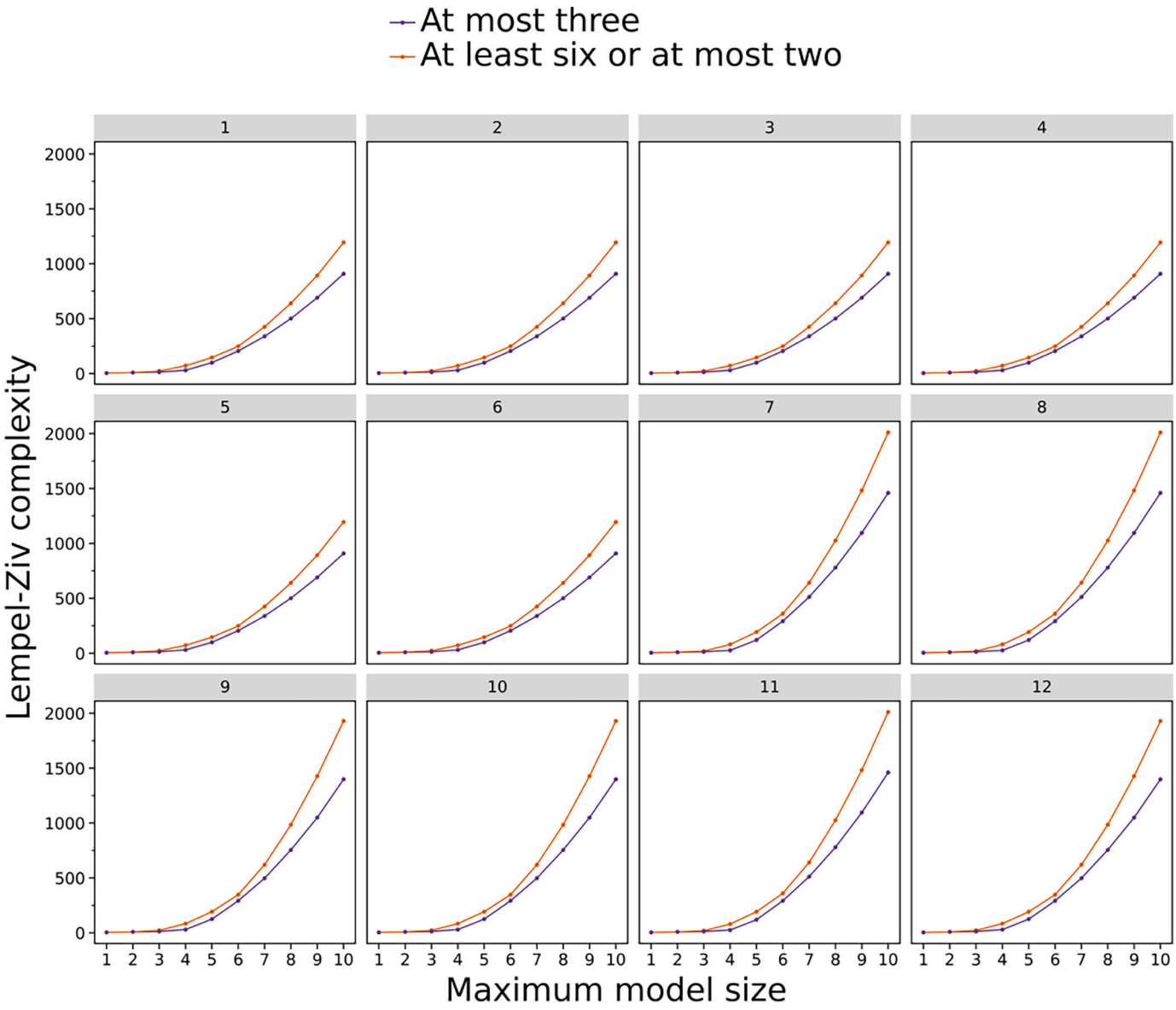

Fig. A.12. Complexity values for `at most three` and `at least six or at most two`, for all 12 lexicographical model sequences.

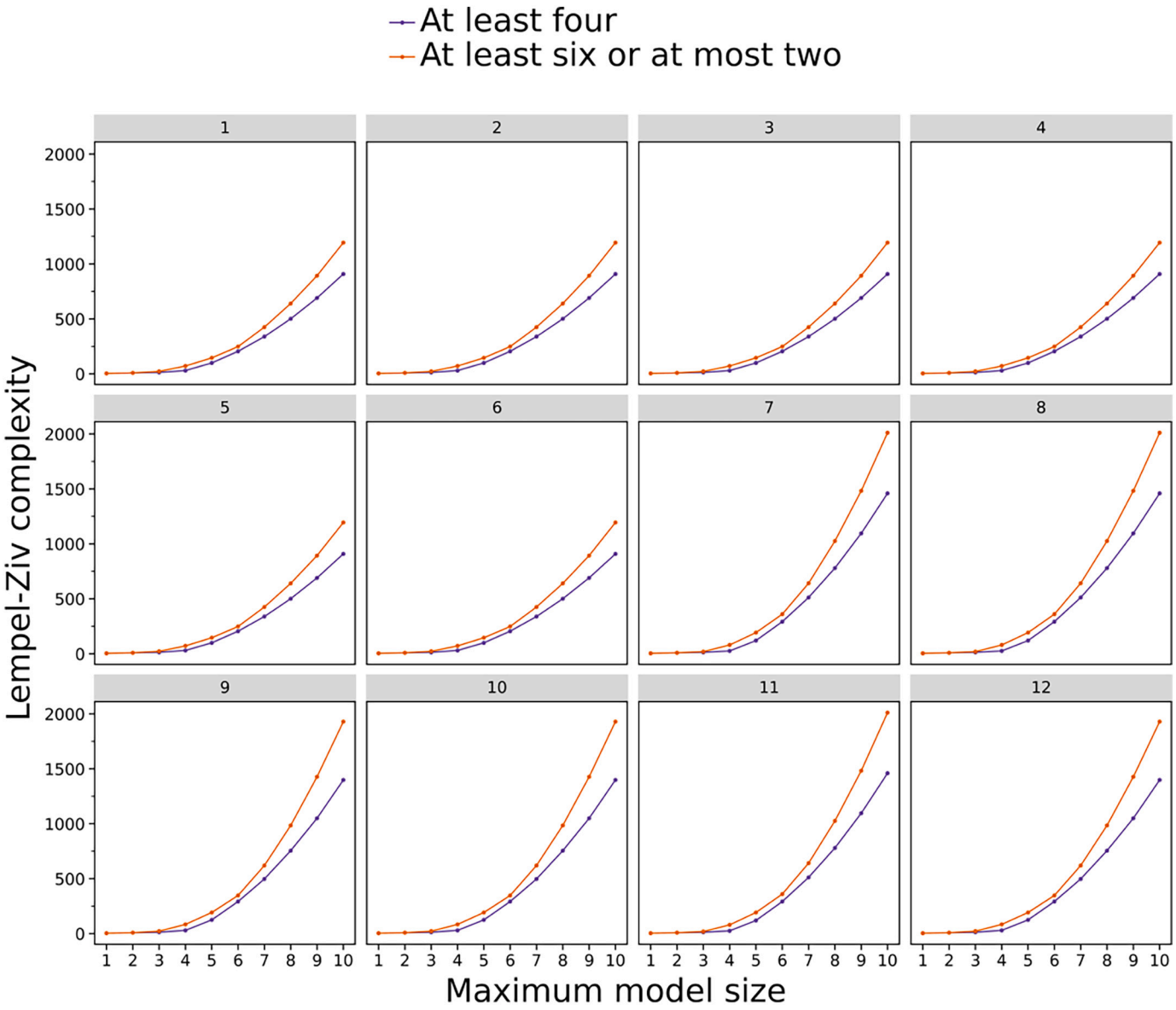

Fig. A.13. Complexity values for `at least four` and `at least six or at most two`, for all 12 lexicographical model sequences.

A.2. Quantity

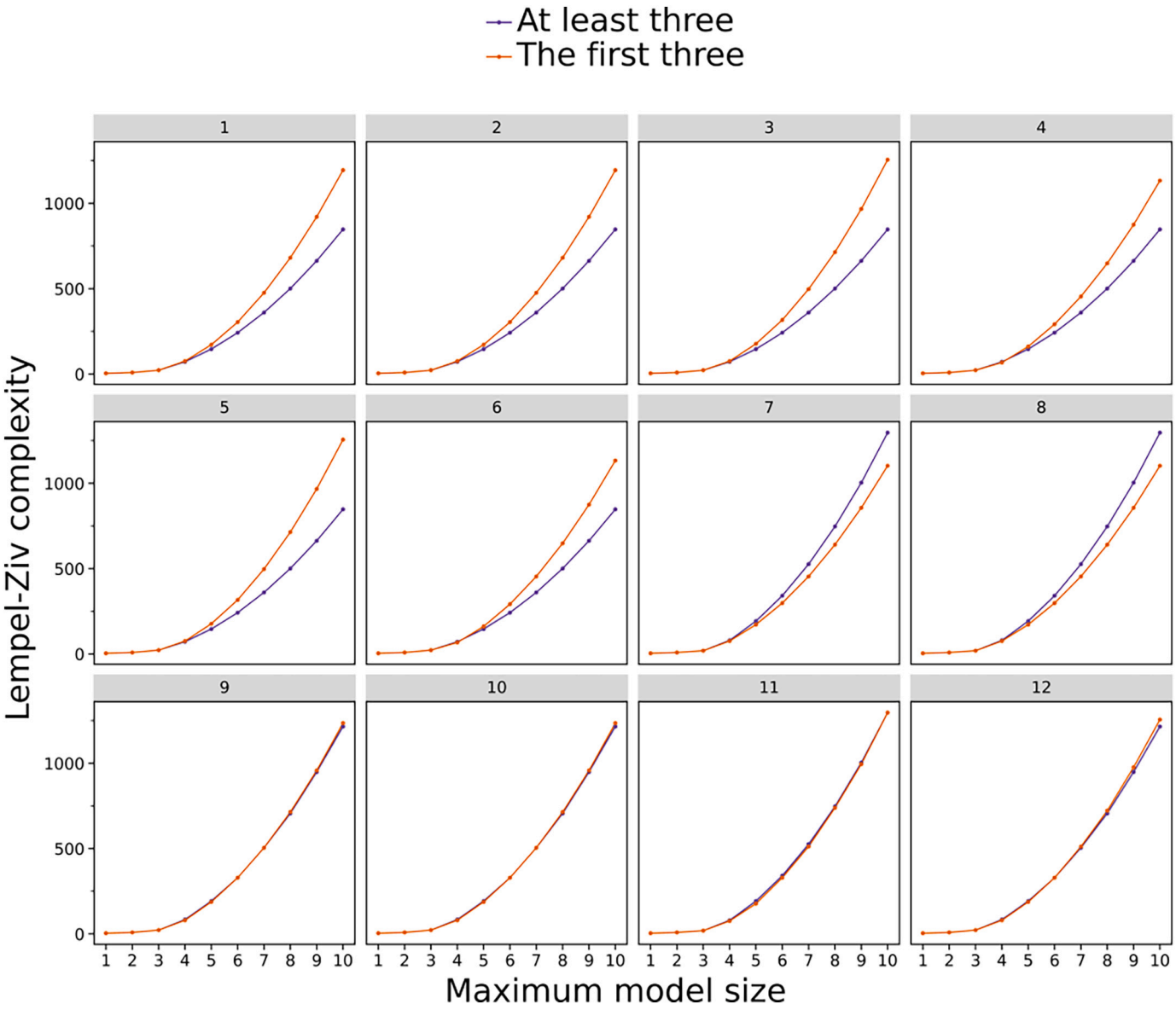

Fig. A.14. Complexity values for `at least three` and `the first three`, for all 12 lexicographical model sequences.

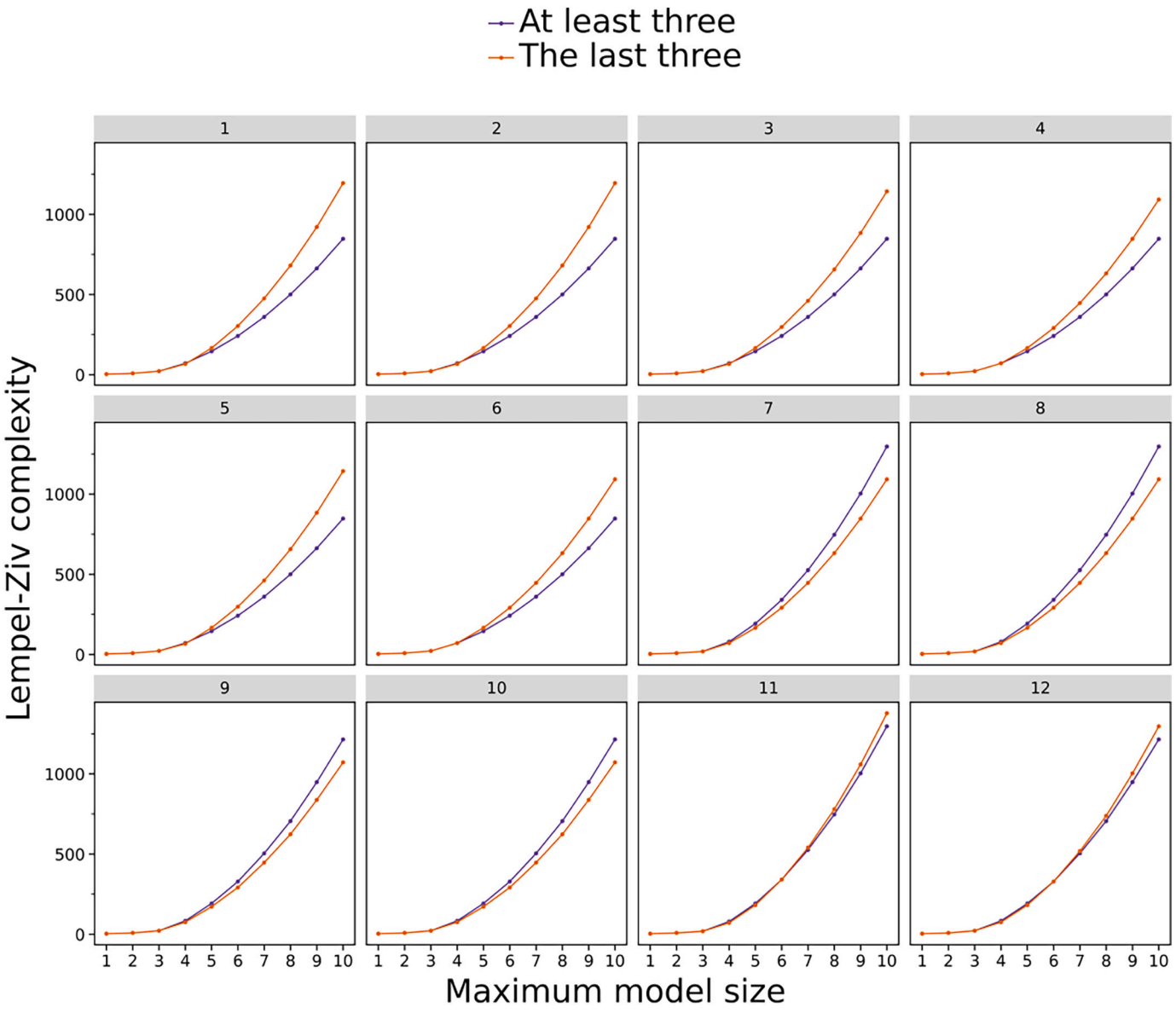

Fig. A.15. Complexity values for `at least three` and `the last three`, for all 12 lexicographical model sequences.

A.3. Conservativity

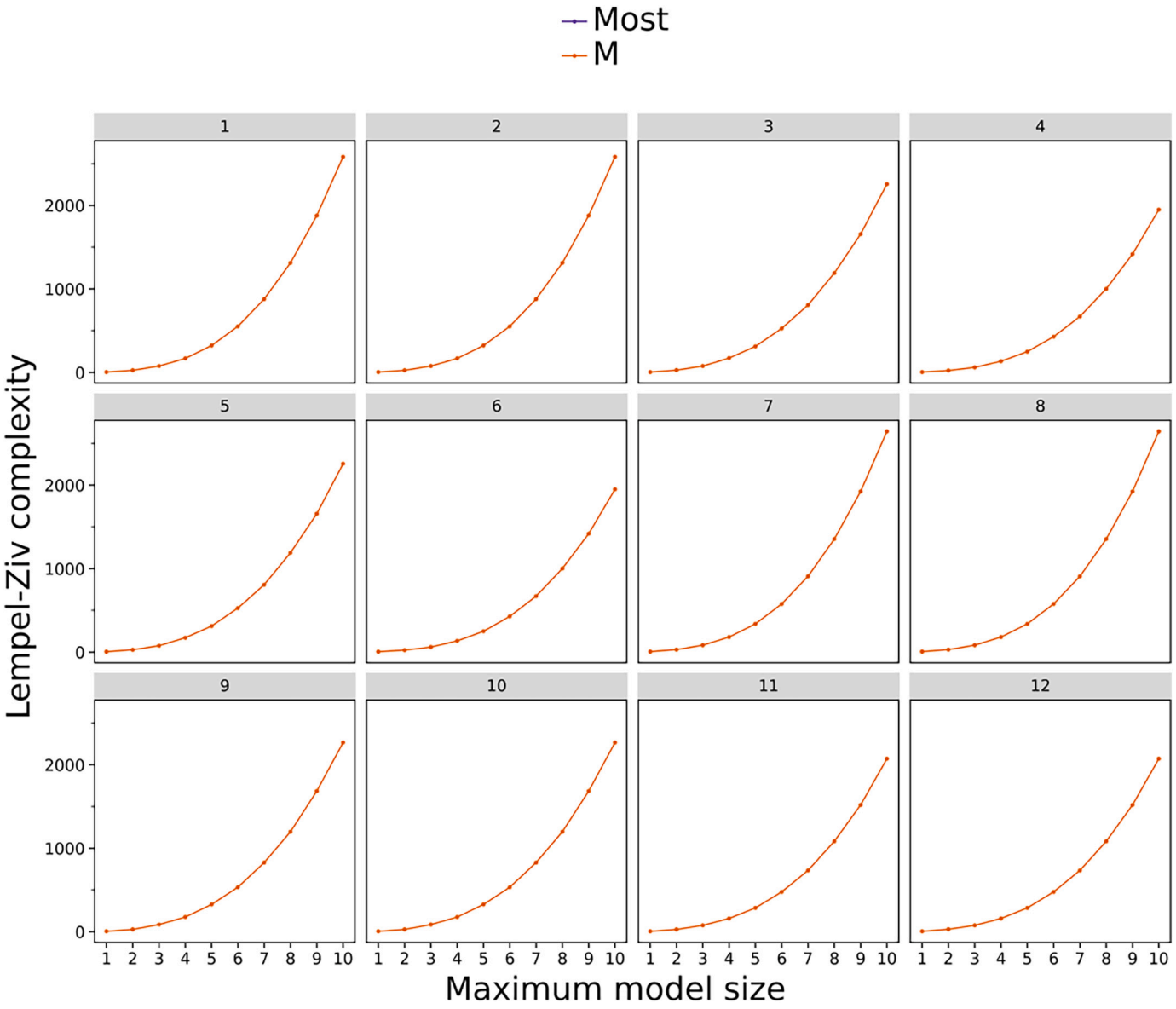

Fig. A.16. Complexity values for `most` and `M`, for all 12 lexicographical model sequences.

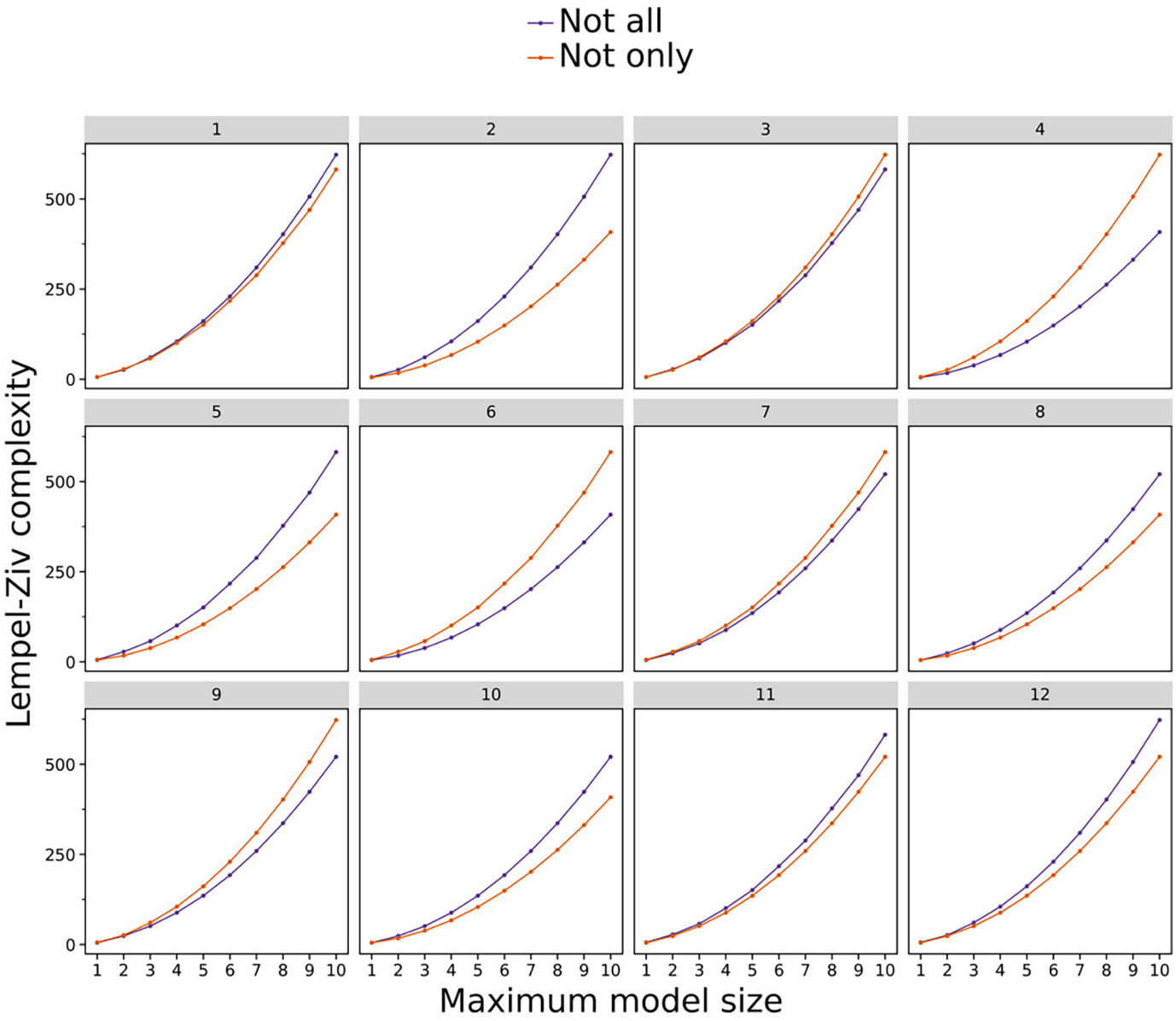

Fig. A.17. Complexity values for `not all` and `not only`, for all 12 lexicographical model sequences.

Appendix B. Experiment 2

This appendix contains the descriptive statistics and logistic regression results for the Lempel-Ziv complexities over the three lexicographical model sequences in Experiment 2. We refer to these complexity scores over the three lexicographical model sequences by LZ_0, LZ_1, and LZ_2.

B.1. Descriptive Statistics for Language $\mathscr{L}_{+\imath}$

B.1.1. LZ_0 scores, $\mathscr{L}_{+\imath}$

Table B.6
Average standardized LZ_0 scores of quantifiers with (YES) versus without (NO) universal property and the proportion (%) of quantifiers with that universal property. *All* stands for quantifiers that have all three properties. The category *"all, NO"* stands for quantifiers that lack at least one property (i.e., that do not have all three properties). For language $\mathscr{L}_{+\imath}$.

	YES	NO	%
monotonicity	−0.19	0.08	0.28
quantity	0.03	0.00	0.15
conservativity	0.18	−0.02	0.12
all	0.16	0.00	0.02

B.1.2. LZ_1 scores, $\mathscr{L}_{+\iota}$

Table B.7
Average standardized LZ_1 scores of quantifiers with (YES) versus without (NO) universal property and the proportion (%) of quantifiers with that universal property. *All* stands for quantifiers that have all three properties. The category *"all, NO"* stands for quantifiers that lack at least one property (i.e., that do not have all three properties). For language $\mathscr{L}_{+\iota}$.

	YES	NO	%
monotonicity	−0.23	0.09	0.28
quantity	0.00	0.00	0.15
conservativity	0.19	−0.02	0.12
all	0.16	0.00	0.02

B.1.3. LZ_2 scores, $\mathscr{L}_{+\iota}$

Table B.8
Average standardized LZ_2 scores of quantifiers with (YES) versus without (NO) universal property and the proportion (%) of quantifiers with that universal property. *All* stands for quantifiers that have all three properties. The category *"all, NO"* stands for quantifiers that lack at least one property (i.e., that do not have all three properties). For language $\mathscr{L}_{+\iota}$.

	YES	NO	%
monotonicity	−0.23	0.09	0.28
quantity	0.03	0.00	0.15
conservativity	0.12	−0.02	0.12
all	0.15	0.00	0.02

B.2. Logistic Regression for Language $\mathscr{L}_{+\iota}$

B.2.1. LZ_0 scores, $\mathscr{L}_{+\iota}$

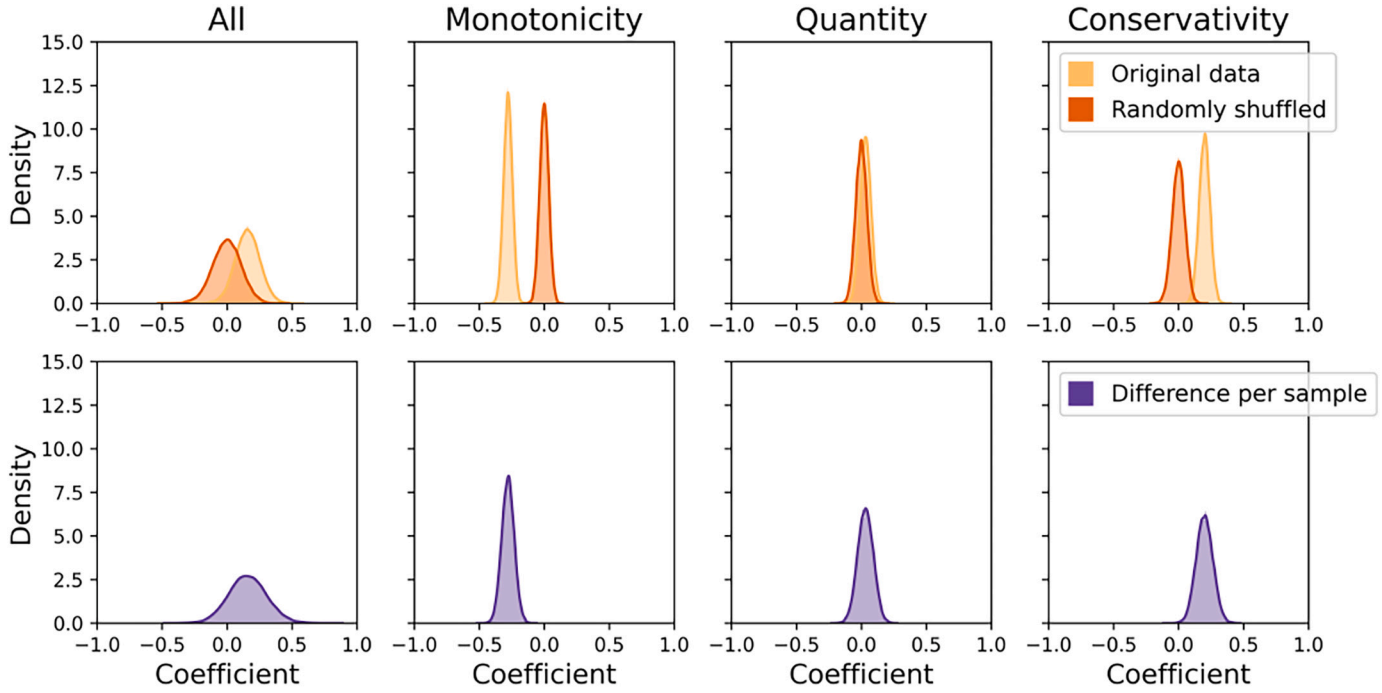

Fig. B.18. Bootstrapped logistic regression results for standardized LZ_0 scores for language $\mathscr{L}_{+\iota}$. The values of the mean and the 95% confidence interval of the coefficient difference (computed per sample) between original and randomly shuffled data (bottom panels) are as follows. All three properties: 0.16 (95% CI [−0.12, 0.45]); monotonicity: -0.28 (95% CI [−0.37, −0.18]); quantity: 0.03 (95% CI [−0.09, 0.15]); conservativity: 0.20 (95% CI [0.08, 0.33]).

B.2.2. LZ_1 scores, $\mathscr{L}_{+1}$

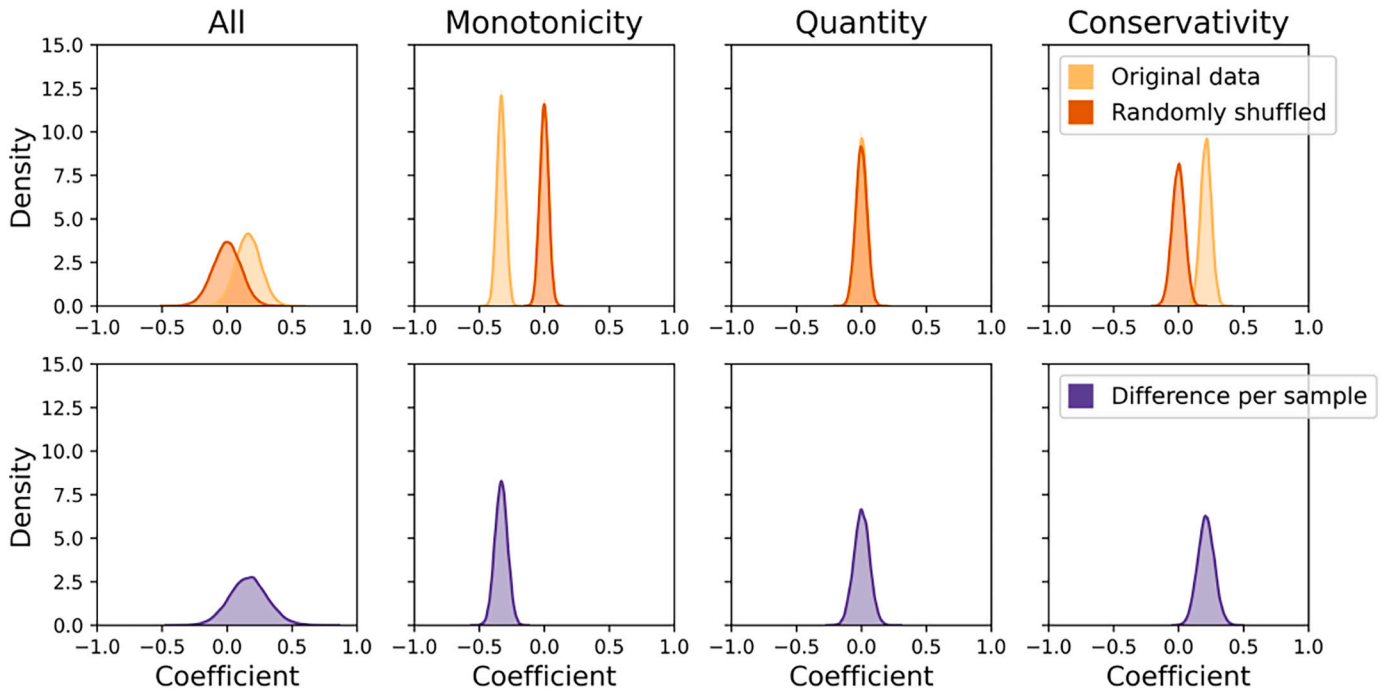

Fig. B.19. Bootstrapped logistic regression results for standardized LZ_1 scores for language $\mathscr{L}_{+1}$. The values of the mean and the 95% confidence interval of the coefficient difference (computed per sample) between original and randomly shuffled data (bottom panels) are as follows. All three properties: 0.17 (95% CI [−0.12, 0.46]); monotonicity: -0.33 (95% CI [−0.43, −0.24]); quantity: 0.00 (95% CI [−0.12, 0.12]); conservativity: 0.21 (95% CI [0.09, 0.34]).

B.2.3. LZ_2 scores, $\mathscr{L}_{+1}$

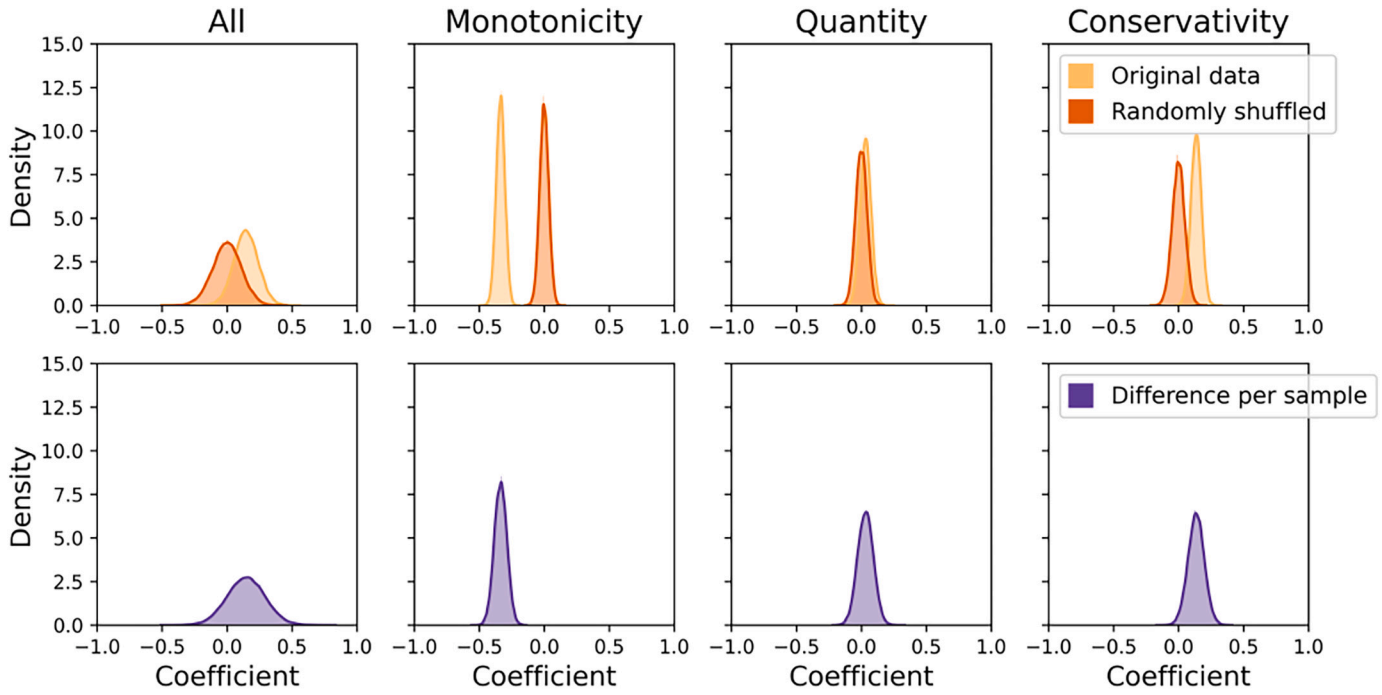

Fig. B.20. Bootstrapped logistic regression results for standardized LZ_2 scores for language $\mathscr{L}_{+1}$. The values of the mean and the 95% confidence interval of the coefficient difference (computed per sample) between original and randomly shuffled data (bottom panels) are as follows. All three properties: 0.15 (95% CI [−0.14, 0.43]); monotonicity: -0.34 (95% CI [−0.43, −0.24]); quantity: 0.03 (95% CI [−0.09, 0.15]); conservativity: 0.14 (95% CI [0.01, 0.26]).

B.3. Descriptive Statistics for Language $\mathscr{L}_{-\iota}$

B.3.1. LZ_0 scores, $\mathscr{L}_{-\iota}$

Table B.9
Average standardized LZ_0 scores of quantifiers with (YES) versus without (NO) universal property and the proportion (%) of quantifiers with that universal property. *Both* stands for quantifiers that have both properties. The category *"both, NO"* stands for quantifiers that lack at least one property (i.e., that do not have both properties). For language $\mathscr{L}_{-\iota}$.

	YES	NO	%
monotonicity	−0.05	0.02	0.33
conservativity	0.34	−0.06	0.14
both	0.22	−0.02	0.08

B.3.2. LZ_1 scores, $\mathscr{L}_{-\iota}$

Table B.10
Average standardized LZ_1 scores of quantifiers with (YES) versus without (NO) universal property and the proportion (%) of quantifiers with that universal property. *Both* stands for quantifiers that have both properties. The category *"both, NO"* stands for quantifiers that lack at least one property (i.e., that do not have both properties). For language $\mathscr{L}_{-\iota}$.

	YES	NO	%
monotonicity	−0.05	0.03	0.33
conservativity	0.37	−0.06	0.14
both	0.24	−0.02	0.08

B.3.3. LZ_2 scores, $\mathscr{L}_{-\iota}$

Table B.11
Average standardized LZ_2 scores of quantifiers with (YES) versus without (NO) universal property and the proportion (%) of quantifiers with that universal property. *Both* stands for quantifiers that have both properties. The category *"both, NO"* stands for quantifiers that lack at least one property (i.e., that do not have both properties). For language $\mathscr{L}_{-\iota}$.

	YES	NO	%
monotonicity	−0.05	0.03	0.33
conservativity	0.33	−0.06	0.14
both	0.19	−0.02	0.08

B.4. Logistic Regression for Language $\mathscr{L}_{-1}$

B.4.1. LZ_0 scores, $\mathscr{L}_{-1}$

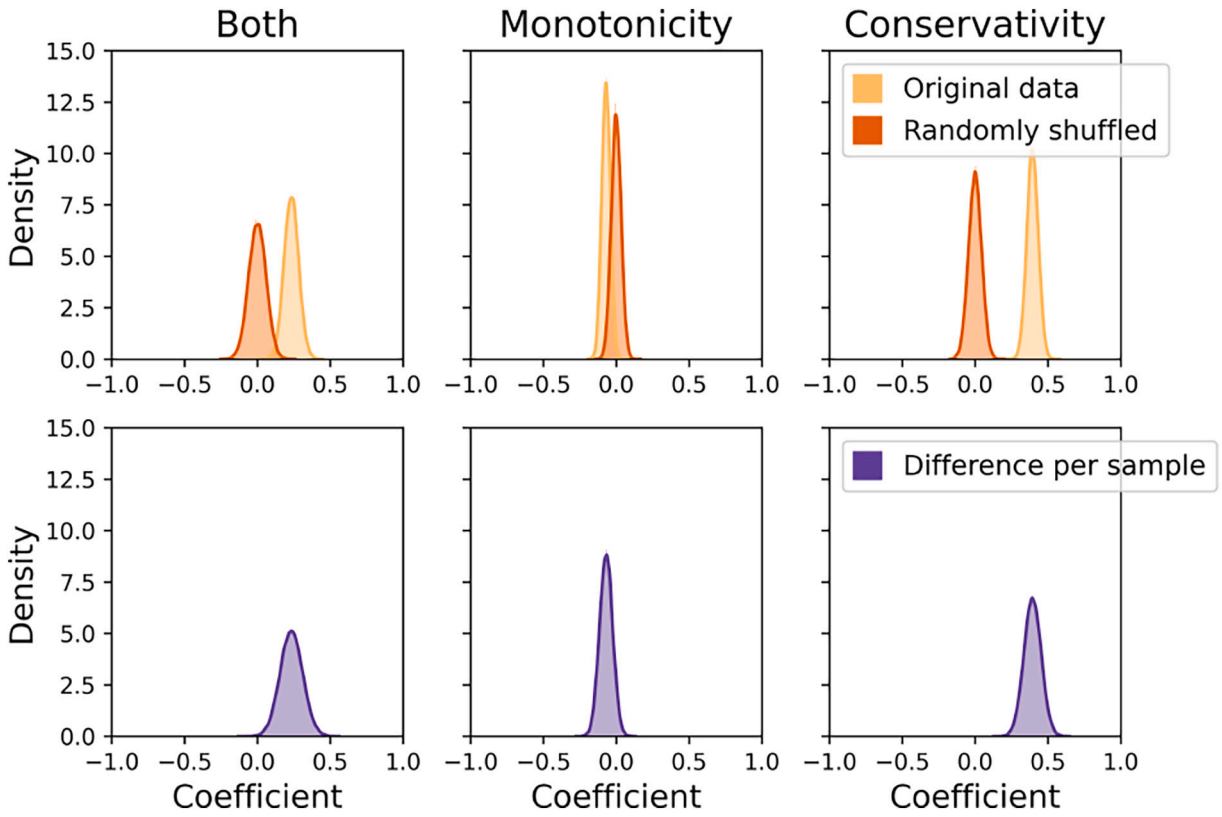

Fig. B.21. Bootstrapped logistic regression results for standardized LZ_0 scores for language $\mathscr{L}_{-1}$. The values of the mean and the 95% confidence interval of the coefficient difference (computed per sample) between original and randomly shuffled data (bottom panels) are as follows. Both properties: 0.23 (95% CI [0.08, 0.39]); monotonicity: -0.07 (95% CI [−0.16, 0.02]); conservativity: 0.40 (95% CI [0.28, 0.51]).

B.4.2. LZ_1 scores, $\mathscr{L}_{-i}$

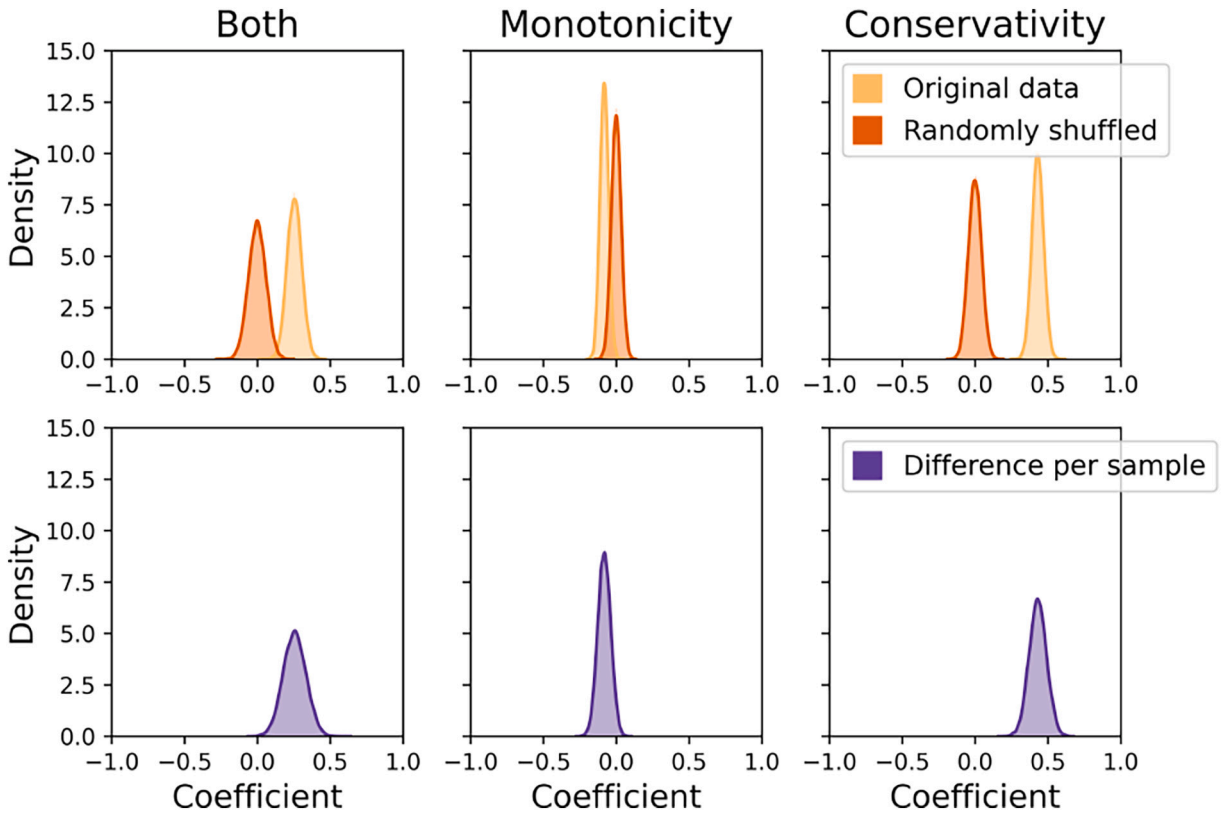

Fig. B.22. Bootstrapped logistic regression results for standardized LZ_1 scores for language $\mathscr{L}_{-i}$. The values of the mean and the 95% confidence interval of the coefficient difference (computed per sample) between original and randomly shuffled data (bottom panels) are as follows. Both properties: 0.26 (95% CI [0.10, 0.41]); monotonicity: -0.08 (95% CI [−0.17, 0.01]); conservativity: 0.43 (95% CI [0.31, 0.55]).

B.4.3. LZ_2 scores, $\mathscr{L}_{-i}$

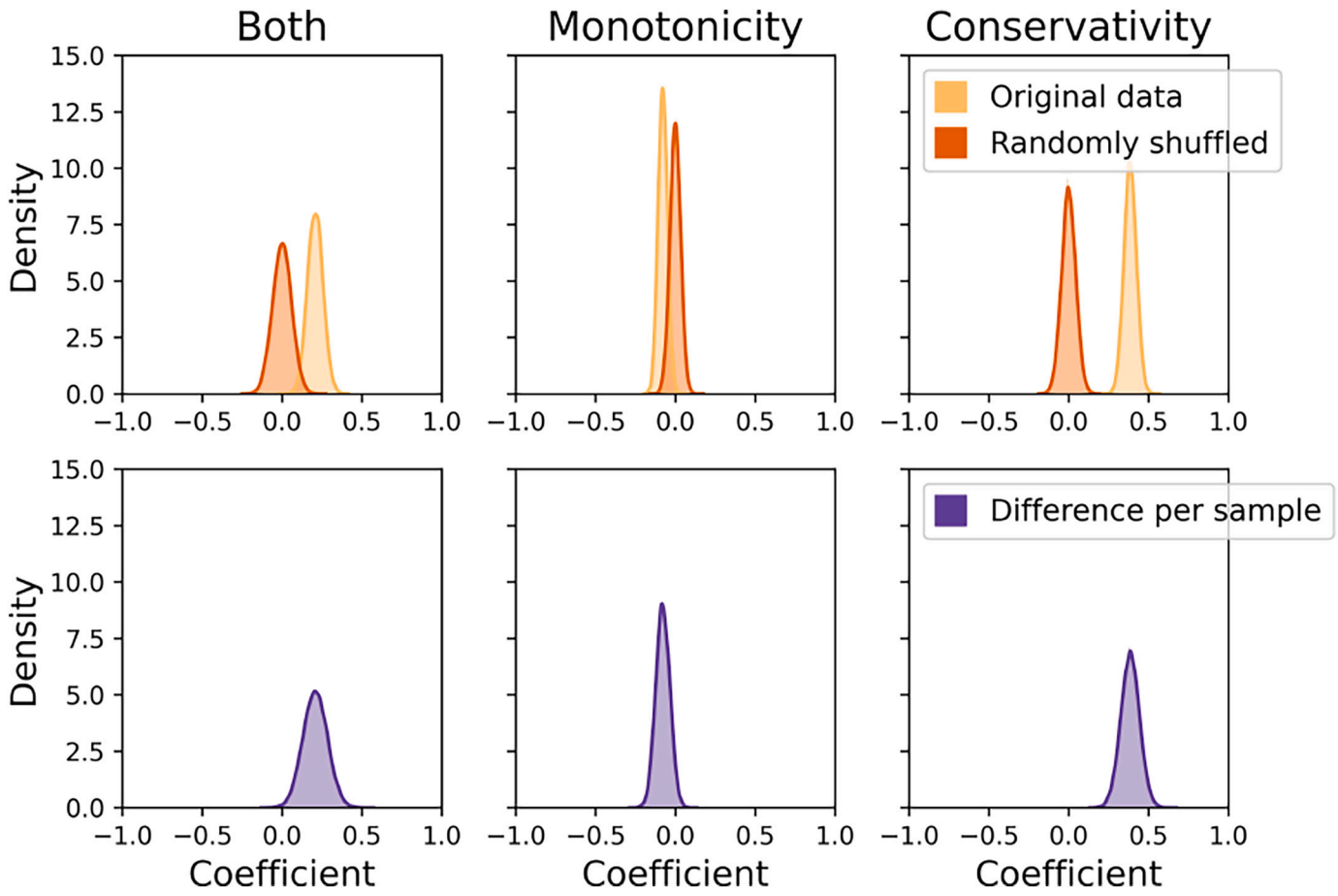

Fig. B.23. Bootstrapped logistic regression results for standardized LZ_2 scores for language $\mathscr{L}_{-i}$. The values of the mean and the 95% confidence interval of the coefficient difference (computed per sample) between original and randomly shuffled data (bottom panels) are as follows. Both properties: 0.21 (95% CI [0.06, 0.36]); monotonicity: -0.08 (95% CI [−0.16, 0.01]); conservativity: 0.39 (95% CI [0.27, 0.50]).

Appendix C. Supplementary Material

The code that we used for generating the data for Experiment 1 and those data themselves can be found both on GitHub at https://github.com/ivdpol/quantifier-LZ-complexity and on the Open Science Framework at https://osf.io/nh9tw/ (Van de Pol, Lodder, Van Maanen, Steinert-Threlkeld, & Szymanik, 2022). The code that we used for generating the data for Experiment 2 and those data themselves can be found both on GitHub at https://github.com/ivdpol/QuantifierComplexity and on the Open Science Framework at https://osf.io/nh9tw/ (Van de Pol et al., 2022).

References

Ariely, D. (2001). Seeing sets: Representation by statistical properties. *Psychological Science, 12*(2), 157–162.

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

van Benthem, J. (1984). Questions about quantifiers. *The Journal of Symbolic Logic, 49*(2), 443–466.

van Benthem, J. (1986). *Essays in logical semantics*. Springer.

Carcassi, F., Steinert-Threlkeld, S., & Szymanik, J. (2019). The emergence of monotone quantifiers via iterated learning. *Proceedings of CogSci 2019*.

Carr, J. W., Smith, K., Culbertson, J., & Kirby, S. (2020). Simplicity and informativeness in semantic category systems. *Cognition, 202*, Article 104289.

Chaabouni, R., Kharitonov, E., Dupoux, E., & Baroni, M. (2021). Communicating artificial neural networks develop efficient color-naming systems. *Proceedings of the National Academy of Sciences, 118*(12).

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

Chemla, E., Buccola, B., & Dautriche, I. (2019). Connecting content and logical words. *Journal of Semantics, 36*(3), 531–547.

Croft, W. (1990). *Typology and universals*. Cambridge: Cambridge University Press.

Denić, M., Steinert-Threlkeld, S., & Szymanik, J. (2022). Indefinite pronouns optimize the simplicity/informativeness trade-off. *Cognitive Science, 46*(5), e13142.

Dingle, K., Camargo, C. Q., & Louis, A. A. (2018). Input–output maps are strongly biased towards simple outputs. *Nature Communications, 9*(1), 761.

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

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

Feldman, J. (2016). The simplicity principle in perception and cognition. *Wiley Interdisciplinary Reviews: Cognitive Science, 7*(5), 330–340.

von Fintel, K., & Matthewson, L. (2008). Universals in semantics. *The Linguistic Review, 25*(1–2), 139–201.

Fox, D. (2002). Antecedent-contained deletion and the copy theory of movement. *Linguistic Inquiry, 33*(1), 63–96.

Galdo, M., Sloutsky, V., & Turner, B. (2021). *The quest for simplicity in human learning*.

Gauvrit, N., Zenil, H., Delahaye, J.-P., & Soler-Toscano, F. (2014). Algorithmic complexity for short binary strings applied to psychology: A primer. *Behavior Research Methods, 46*(3), 732–744.

Gibson, E., Futrell, R., Piantadosi, S. T., Dautriche, I., Mahowald, K., Bergen, L., & Levy, R. (2019). How efficiency shapes human language. *Trends in Cognitive Sciences, 23*(5), 389–407.

Goddard, C., & Wierzbicka, A. (2002). *Meaning and universal grammar: Theory and empirical findings* (vol. 1). John Benjamins Publishing.

Goodman, N. (1955). *Fact, fiction, & forecast*. Harvard University Press.

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.

Goodman, N. D., Tenenbaum, J. B., & Gerstenberg, T. (2015). *Concepts in a probabilistic language of thought*. MIT Press.
Greenberg, J. H. (1966). *Language universals*. The Hague: Mouton.
Grünwald, P. D., Myung, J. I., & Pitt, M. A. (Eds.). (2005). *Advances in minimum description length: Theory and applications. Neural information processing series*. A Bradford Book.
Higginbotham, J., & May, R. (1981). Questions, quantifiers and crossing. *The Linguistic Review, 1*, 41–80.
Hsu, A. S., Chater, N., & Vitányi, P. (2013). Language learning from positive evidence, reconsidered: A simplicity-based approach. *Topics in Cognitive Science, 5*(1), 35–55.
Hunter, T., & Lidz, J. (2013). Conservativity and learnability of determiners. *Journal of Semantics, 30*(3), 315–334.
Katzir, R., Lan, N., & Peled, N. (2020). A note on the representation and learning of quantificational determiners. *Proceedings of Sinn und Bedeutung, 24*(1), 392–410.
Keenan, E. L. (1981). A boolean approach to semantics. In J. A. Groenendijk, T. M. Janssen, & M. B. Stokhof (Eds.), *Formal methods in the study of language: Part 2* (pp. 343–379). Amsterdam: Mathematisch Centrum.
Keenan, E. L., & Stavi, J. (1986). A semantic characterization of natural language determiners. *Linguistics and Philosophy, 9*(3), 253–326.
Kemp, C. (2012). Exploring the conceptual universe. *Psychological Review, 119*(4), 685.
Kontinen, J., & Szymanik, J. (2008). A remark on collective quantification. *Journal of Logic, Language and Information, 17*(2), 131–140.
Langley, P., & Stromsten, S. (2000). Learning context-free grammars with a simplicity bias. In R. López de Mántaras, & E. Plaza (Eds.), *Machine learning: ECML 2000* (pp. 220–228). Berlin, Heidelberg: Springer Berlin Heidelberg.
Lempel, A., & Ziv, J. (1976). On the complexity of finite sequences. *IEEE Transactions on Information Theory, 22*(1), 75–81.
Lesne, A., Blanc, J.-L., & Pezard, L. (2009). Entropy estimation of very short symbolic sequences. *Physical Review E, 79*(4), 1–10, 046208.
Li, M., & Vitányi, P. (2008). *An introduction to Kolmogorov complexity and its applications*. Springer.
Maldonado, M., & Culbertson, J. (2021). Person of interest: Experimental investigations into the learnability of person systems. *Linguistic Inquiry*, 1–71.
McMillan, C. T., Clark, R., Moore, P., Devita, C., & Grossman, M. (2005). *Neural basis for generalized quantifier comprehension. 43(12)* pp. 1729–1737).
Mostowski, M. (1998). Computational semantics for monadic quantifiers. *Journal of Applied Non-Classical Logics, 8*, 107–121.
Peters, S., & Westerståhl, D. (2006). *Quantifiers in language and logic*. Oxford: Clarendon Press.
Piantadosi, S. T., Tenenbaum, J. B., & Goodman, N. D. (2013). *Modeling the acquisition of quantifier semantics: A case study in function word learnability*.
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.
Planton, S., van Kerkoerle, T., Abbih, L., Maheu, M., Meyniel, F., Sigman, M., Wang, L., Figueira, S., Romano, S., & Dehaene, S. (2021). A theory of memory for binary sequences: Evidence for a mental compression algorithm in humans. *PLoS Computational Biology, 17*(1), Article e1008598.
van de Pol, I., Lodder, P., van Maanen, L., Steinert-Threlkeld, S., & Szymanik, J. (2021). Quantifiers satisfying semantic universals are simpler. In *Proceedings of CogSci 2021*.
van de Pol, I., Lodder, P., van Maanen, L., Steinert-Threlkeld, S., & Szymanik, J. (2022). Supplementary material for: Quantifiers satisfying semantic universals have shorter minimal description length. https://osf.io/nh9tw.
van de Pol, I., Steinert-Threlkeld, S., & Szymanik, J. (2019). Complexity and learnability in the explanation of semantic universals of quantifiers. In *Proceedings of CogSci 2019*.
Ristad, E. S. (1993). *The language complexity game. Artificial intelligence*. MIT Press.
Romoli, J. (2015). A structural account of conservativity. *Semantics-Syntax Interface, 2*(1), 28–57.
Spenader, J., & de Villiers, J. (2019). Are conservative quantifiers easier to learn? Evidence from novel quantifier experiments. In J. J. Schlöder, D. McHugh, & F. Roelofsen (Eds.), *Proceedings of the 22nd Amsterdam Colloquium*.
Sportiche, D. (2005). Division of labor between merge and move: Strict locality of selection and apparent reconstruction paradoxes. In *, vol. 378. Proceedings of the workshop divisions of linguistic labor, the La Bretesche workshop* (pp. 80–126).
Steinert-Threlkeld, S. (2020). Quantifiers in natural language optimize the simplicity/informativeness trade-off. In J. J. Schlöder, D. McHugh, & F. Roelofsen (Eds.), *Proceedings of the 22nd Amsterdam colloquium* (pp. 513–522).
Steinert-Threlkeld, S. (2021). Quantifiers in natural language: Efficient communication and degrees of semantic universals. *Entropy, 23*(10), 1335.
Steinert-Threlkeld, S., & Szymanik, J. (2019). Learnability and semantic universals. *Semantics and Pragmatics, 12*, 4.
Steinert-Threlkeld, S., & Szymanik, J. (2020). Ease of learning explains semantic universals. *Cognition, 195*, Article 104076.
Szabolcsi, A. (2010). *Quantification. Research surveys in linguistics*. Cambridge: Cambridge University Press.
Szymanik, J. (2016). *Quantifiers and cognition. Logical and computational perspectives. Studies in linguistics and philosophy*. Springer.
Szymanik, J., & Zajenkowski, M. (2010). Comprehension of simple quantifiers: Empirical evaluation of a computational model. *Cognitive Science, 34*(3), 521–532.
Tiede, H.-J. (1999). Identifiability in the limit of context-free generalized quantifiers. *Journal of Language and Computation, 1*(1), 93–102.
Vitányi, P. M. (2013). Similarity and denoising. *Philosophical Transactions of the Royal Society A, 371*(1984).
Wellwood, A., Gagliardi, A., & Lidz, J. (2016). Syntactic and lexical inference in the acquisition of novel superlatives. *Language Learning and Development, 12*(3), 262–279.
Whitney, D., & Yamanashi Leib, A. (2018). Ensemble perception. *Annual Review of Psychology, 69*(1), 105–129.
Zaslavsky, N., Maldonado, M., & Culbertson, J. (2021). *Let's talk (efficiently) about us: Person systems achieve near-optimal compression* (In *Proceedings of CogSci 2021)*.
Ziv, J., & Lempel, A. (1978). Compression of individual sequences via variable-rate coding. *IEEE Transactions on Information Theory, 24*(5), 530–536.
Zuber, R., & Keenan, E. L. (2019). A note on conservativity. *Journal of Semantics, 36*(4), 573–582.