THE BULLETIN OF SYMBOLIC LOGIC
Volume 29, Number 4, December 2023

INTERLEAVING LOGIC AND COUNTING

JOHAN VAN BENTHEM AND THOMAS ICARD

Abstract. Reasoning with quantifier expressions in natural language combines logical and arithmetical features, transcending strict divides between qualitative and quantitative. Our topic is this cooperation of styles as it occurs in common linguistic usage and its extension into the broader practice of natural language plus 'grassroots mathematics'.

We begin with a brief review of FO(#), first-order logic with counting operators and cardinality comparisons. This system is known to be of very high complexity, and drowns out finer aspects of the combination of logic and counting. We therefore move to a small fragment that can represent numerical syllogisms and basic reasoning about comparative size: monadic first-order logic with counting, MFO(#). We provide normal forms that allow for axiomatization, determine which arithmetical notions can be defined on finite and on infinite models, and conversely, we discuss which logical notions can be defined out of purely arithmetical ones, and what sort of (non-)classical logics can be induced.

Next, we investigate a series of strengthenings of MFO(#), again using normal form methods. The monadic second-order version is close, in a precise sense, to additive Presburger Arithmetic, while versions with the natural device of tuple counting take us to Diophantine equations, making the logic undecidable. We also define a system ML(#) that combines basic modal logic over binary accessibility relations with counting, needed to formulate ubiquitous reasoning patterns such as the Pigeonhole Principle. We prove decidability of ML(#), and provide a new kind of bisimulation matching the expressive power of the language.

As a complement to the fragment approach pursued here, we also discuss two other ways of lowering the complexity of FO(#) by changing the semantics of counting in natural ways. A first approach replaces cardinalities by abstract but well-motivated values of 'mass' or other mereological aggregating notions. A second approach keeps the cardinalities but generalizes the meaning of counting to work in models that allow dependencies between variables.

Finally, we return to our starting point in natural language, confronting the architecture of our formal systems with linguistic quantifier vocabulary and syntax, as well as with natural reasoning modules such as the monotonicity calculus. In addition to these encounters with formal semantics, we discuss the role of counting in semantic evaluation procedures for quantifier expressions and determine, for instance, which binary quantifiers are computable by finite 'semantic automata'. We conclude with some general thoughts on yet further entanglements of logic and counting in formal systems, on rethinking the qualitative/quantitative divide, and on connecting our analysis to empirical findings in cognitive science.

Received October 2, 2021.
2020 Mathematics Subject Classification. 03A05, 03B45, 03B65, 03B70.
Key words and phrases. logic, arithmetic, counting, qualitative, quantitative, cardinality.

1079-8986/23/2904-0002
DOI :10.1017/bsl.2023.30

§1. Introduction: inference and computing. Here is the archetypal logical inference with a basic quantifier:

From 'All *A* are *B*' and 'All *B* are *C*', conclude that 'All *A* are *C*'.

Next, here are two slightly modified premises in natural language.

'All *A except one* are *B* and all *B except two* are *C*'.

This time, one may need to think just a little bit more to conclude that

'All *A except at most three* are *C*'.

That extra bit of thought involves considering possible exceptions, or more generally: *counting*. In fact, the very term *quantifier* suggests quantities, and the semantics of quantifier expressions in logic and linguistics involves numbers by its emphasis on permutation invariance, which abstracts away from every feature of predicates except their size. This mix of logic and counting is not just about absolute numbers, it also extends to size comparisons. From

'Most *A* are *B*' and 'All *B* are *C*',

we may safely draw the conclusion that

'Most *A* are *C*'

and similar simple inference patterns govern explicitly comparative expressions such as 'More *A* than *B* are *C*'. But valid reasoning patterns with comparatives can also be more challenging, as in the following inference, which may require drawing a Venn diagram:

'More *A* than *B* are *C*', 'More *B* than *C* are *A*',
Therefore: 'More *A* than *C* are *B*'.

This has echoes of the mathematical *Triangle Inequality* underlying metric geometry.

Numerical comparisons in natural language can even occur between *proportions*, as happens in the relative sense of 'Many *A* are *B*', comparing the numbers of *B*s among the *A*s with the number of *B*s overall, defined more precisely in Section 3, and a running example later on.

Qualitative logical analyses are sometimes seen as replacing quantitative theories by 'more basic' qualitative ones, for instance, in the foundations of probability or in measurement theory. This can be illuminating, and success can be measured by representation theorems. And yet, historically, logic and quantitative reasoning, for instance, with probability, went together in the pioneering work of Bolzano and Boole. It is hard to say whether Boole's propositional logic is a qualitative basic form of binary arithmetic, or a way of making logical inference a form of counting. In a sense, it is both.

https://doi.org/10.1017/bsl.2023.30 Published online by Cambridge University Press

A divide arose only in the time of Frege, when logicism insisted that logical notions come first, and arithmetical ones are constructed out of these. To be sure, this reductionist program has yielded many fundamental notions and results, and we owe a lot of modern logic to its arrival. But in this paper, we will follow the linguistic practice that we started with, and treat logic and *counting*, taken as the realm of numerical comparisons and basic arithmetic, on a par.

In what follows we will take this linguistic practice in a broad sense, including ubiquitous forms of reasoning that might be called 'grassroots mathematics' rather than pure natural language inference. A typical example underlies the following pattern:

> 'Twenty farmers own at most 15 cows each'. Therefore:
> 'At least two farmers own the same number of cows'.

The reader may find it difficult to see how this would follow as a straightforward matter of overt logical or linguistic form. Instead, what is needed is the following:

> *Pigeonhole Principle* If one puts n objects into k boxes, with $n > k$, then at least one box must contain at least two objects.

Here k is the number of cows owned, which runs from 0 to 15, n the number of farmers. The Pigeonhole Principle occurs in elementary mathematics where it can have non-trivial consequences when applied imaginatively, but it is also of interest in cognitive science as a benchmark in reasoning ability including finding the right representation of problems [87]. In this paper, the principle will occur at various places as we determine its position in combined systems of logic and counting.

Where should we start with our investigation of logic and counting? It is well-known that combining a standard system like first-order logic with counting syntax and cardinality comparisons leads to a system $\mathsf{FO}(\#)$ of very high complexity. Therefore, for our purpose, this 'view from above' is not that illuminating, and after just a quick look at $\mathsf{FO}(\#)$ and its properties, we will start work 'from below', exploring very simple combinations of logic and counting, and only then move to more complex systems.

Our presentation follows mainstream practice in offering a sequence of formal systems of increasing expressive strength. We will prove many results about these systems that demonstrate their precise mixture of logic and counting. Toward the end of the paper, we return to the naturally occurring practice of mixed qualitative and quantitative reasoning that we started with here, linking up with Generalized Quantifier Theory for natural language, and touching on empirical issues in cognitive science. Finally, in a sequence of appendices, we broaden the context, and point out yet further entanglements of logic and counting that show the ubiquity of the phenomenon we are

https://doi.org/10.1017/bsl.2023.30 Published online by Cambridge University Press

after. True understanding of how logical systems work involves numbers and counting from manipulating syntax to proofs by formula induction, but also semantically, e.g., in the use of numerical invariants in Ehrenfeucht games.

There are several ways of looking at the topics and results presented in this paper. Simple combinations of logic and counting can often be seen as fragments of richer *logics of generalized quantifiers* [9, 100]. In this sense, we are looking at fine-structure of fragments of well-known systems from mathematical logic. Moreover, the interplay of logic and counting has long been studied in *computational logic* [98, 116]. Accordingly, themes and results from the literature in theoretical computer science will appear at many places in this paper. We have added an appendix with references to a wide, and hopefully representative, swath of the preceding literature, though a full overview is beyond our capacity.

Against this background, the technical main novelty of this paper is the series of simple combined systems that we define and study. However, a further contribution may be the more empirical perspective we are adding of *connections with natural language and cognition*. In addition to our technical results about logic and counting, we see this stance in between logic, computation, and cognition, as fruitful and worth pursuing.

In the next section, we first present a higher-end combination of logic and counting, as a first pass through our main themes. After that, we give more detail on the lower-end systems that will be the focus of our analysis in the core of the paper.

§2. First-order logic with counting.

Perhaps the obvious starting point is to consider a counting operator # on top of standard first-order logic, allowing us to count the number of objects satisfying a given formula. Where x is a first-order variable and φ a first-order formula, in a first-order model $\mathcal{M}$ with variable assignment s, the term $\#_x\varphi$ denotes the cardinality of the set of x's satisfying φ:

$$[\![\#_x\varphi]\!]^{\mathcal{M},s} = |\{d \in D : \mathcal{M}, s^x_d \models \varphi\}|.$$

Count terms thus denote cardinal numbers. What kinds of assertions would we want to make about cardinal numbers to formalize interesting reasoning about counting? Here we start with a basic and fundamental capacity, namely *comparison*. We inductively define count comparison formulas $\#_x\varphi \succsim \#_y\psi$, with the obvious interpretation according to which:

$$\mathcal{M}, s \models \#_x\varphi \succsim \#_y\psi \quad \text{iff} \quad [\![\#_x\varphi]\!]^{\mathcal{M},s} \geq [\![\#_y\psi]\!]^{\mathcal{M},s}.$$

Call this language $\mathcal{L}_\#$ and call the logical system $\mathsf{FO}(\#)$.

This system has been studied thoroughly. It is natural to construe $\mathsf{FO}(\#)$ as first-order logic with a generalized quantifier, sometimes known in the literature as the *Rescher quantifier* [56, 98] after a related extension considered in [109]; in the philosophical literature it has sometimes been called the *Frege quantifier* [2]. Other well-known quantifiers, such as the so-called *Härtig quantifier* and the *Chang quantifier*, are easily definable in $\mathsf{FO}(\#)$ (see, e.g., [100]). It will be convenient to abbreviate the Härtig quantifier $(\#_x\varphi \succsim \#_y\psi) \wedge (\#_y\psi \succsim \#_x\varphi)$ by $\#_x\varphi \approx \#_y\psi$; likewise, we abbreviate $(\#_x\varphi \succsim \#_y\psi) \wedge \neg(\#_y\psi \succsim \#_x\varphi)$ by $\#_x\varphi \succ \#_y\psi$.

Typical of extensions of first-order logic, we have the following:

Proposition 2.1. $\mathsf{FO}(\#)$ *fails to be compact and it lacks the Löwenheim–Skølem property down to any cardinality below* $\aleph_\omega$.

Proof. First, note that the infinity quantifier is easily definable in $\mathsf{FO}(\#)$:

$$\exists^\infty y.\varphi \equiv \exists y.\big(\varphi \wedge \#_x(\varphi[{}^x/_y] \wedge x \neq y) \succsim \#_y(\varphi)\big), \tag{2.1}$$

where the substitution $\varphi[{}^x/_y]$ is defined as usual. Then we can force the domain to have size at least $\aleph_k$ simply by stating, for instance, $\exists^\infty x.P_0(x) \wedge \bigwedge_{i\leq k}\big(\#_x P_{i+1}(x) \succ \#_x P_i(x)\big)$, for $k+1$ predicate symbols $P_0, \dots, P_k$.

Compactness also fails easily: abbreviating $\bigwedge_{i,j\leq n} x_i \neq x_j$ by $\mathsf{diff}(\mathbf{x})$, and using $\exists^{\geq n}x.P(x)$ to abbreviate $\exists x_1 \dots x_n.\big(\mathsf{diff}(\mathbf{x}) \wedge \bigwedge_{i\leq n} P(x_i)\big)$, the set

$$\{\neg\exists^\infty x.P(x)\} \cup \{\exists^{\geq n}x.P(x) : n < \omega\} \tag{2.2}$$

is unsatisfiable, but finitely satisfiable. ⊣

To see just how much stronger $\mathsf{FO}(\#)$ is than ordinary FO, note the following:

Fact 2.2. We can enforce in $\mathsf{FO}(\#)$ that a binary relation R is a well-order of order type ω.

Proof. Let σ be the statement that R is a serial, strict total order (i.e., serial, irreflexive, transitive, total), and conjoin σ with the statement $\forall x.\neg\exists^\infty y.R(y,x)$, saying that each element has only finitely many R-predecessors. ⊣

It follows that the validity problem for $\mathsf{FO}(\#)$ is not arithmetical; in fact it is Π^1_1-hard. If we do not allow embedding # comparisons, then we can also show that the satisfiability problem is in Σ^1_1: every comparison amounts to the existence of an injective function.

Fact 2.3. The set of validities of $\mathsf{FO}(\#)$ without embedded # terms is Π^1_1-complete.

https://doi.org/10.1017/bsl.2023.30 Published online by Cambridge University Press

However, for the general case the situation is much worse. Herre et al. [56] showed the following result for first-order logic with the Härtig quantifier:[1]

Theorem 2.4 [56]. *The set of validities of* $\mathsf{FO}(\#)$ *is neither in* Π^1_2 *nor in* Σ^1_2.

$\mathsf{FO}(\#)$ clearly brings a potent combination of logical expressive power and explicit count comparison. To what degree can we tease apart the separate contributions of logic and counting in this rich setting? Specifically, how much do # comparisons add to the counting repertoire native to first-order logic; and vice versa, how much logic could we already extract from counting alone? We begin with the second question.

2.1. From counting to logic. Let us restrict attention to a very small fragment of the language $\mathcal{L}_\#$ described above. Given some variables Var and predicate symbols Pred, we only allow two types of atomic formulas and one operation for building complex formulas. Let $\mathcal{L}^-_\#$ be generated by the grammar:

$$\varphi ::= P(x_1, \dots, x_n) \quad | \quad x \neq y \quad | \quad \#_x\varphi \succsim \#_y\varphi.$$

Aside from predication and variable inequality, we can only compare cardinalities.

A first observation is that Boolean implication can already be defined in $\mathcal{L}^-_\#$. Where x occurs free in neither φ nor ψ, we can take

$$\psi \to \varphi \equiv \#_x\varphi \succsim \#_x\psi. \tag{2.3}$$

Boolean negation can also be defined. Where $\mathsf{0}$ is an abbreviation for the formula $\#_x(x \neq x)$ (cf. Frege), and again x is a variable that does not occur free in φ, we can define

$$\neg\varphi \equiv \mathsf{0} \succsim \#_x\varphi. \tag{2.4}$$

With these we recover any other Boolean connective, as well as variable equality. In some respect, count comparison already incorporates Boolean structure, and familiar Boolean laws emerge as principles of count comparisons. For instance, the pattern $\varphi \to (\psi \to \varphi)$ is encoded simply as $\#_x(\#_x\varphi \succsim \#_x\psi) \succsim \#_x\varphi$.

Going further, first-order quantification is expressible in $\mathcal{L}^-_\#$:

$$\exists x.\varphi \equiv \#_x\varphi \succ \mathsf{0}. \tag{2.5}$$

This thus brings us back to full $\mathsf{FO}(\#)$, in which we can again define the infinity quantifier $\exists^\infty$ in (2.1), its dual $\forall^\infty$, and so on. From rather austere (atomic) primitives, count comparisons already encode a significant amount

[1] In fact, the hardness result in Theorem 2.4 holds even if we take the *negationless* fragment of $\mathsf{FO}(\#)$. Mayer [86] has given a computable reduction from the negationless fragment to the full fragment with negation.

https://doi.org/10.1017/bsl.2023.30 Published online by Cambridge University Press

of logic, provided of course that we allow iteration of comparisons within comparisons.

Remark 2.5 (Extended logical vocabulary). Counting can also define non-first-order quantifiers that are often considered logical in an extended sense. An example is the binary quantifier 'Most φ are ψ', which is definable as $\#_x(\varphi \wedge \psi) \succ \#_x(\varphi \wedge \neg\psi)$. But even closer to first-order logic, counting suggests different kinds of universal quantifiers, depending on how we extend the standard meaning on finite sets to infinite ones. One option is $\neg\exists x.\neg\varphi$, the dual of the existential quantifier defined in (2.5), which expresses exceptionless universal quantification. But there are also interesting weaker variants, such as $\#_x\varphi \approx \#_x\top \wedge \#_x\varphi \succ \#_x\neg\varphi$. This says that the set of objects satisfying φ has the size of the universe, while the possible exceptions have a smaller size. This is a version of the quantifier 'almost all' which has elegant mathematical properties and interesting measure-theoretic applications [122].

Remark 2.6 (Non-classical logics). In addition to options qua expressive power, counting also offers options for deductive power. The definitions (2.3)–(2.5) show that we can reconstruct classical logic from # comparisons. Is FO(#) in some way inherently classical, or could we instead naturally extract *non-classical* connectives?

One route would be to keep the same implication in (2.3), but to redefine negation in terms of an arbitrary predicate, say, $G(x)$. If we then let $\neg\varphi$ stand for the sentence $\#_x G(x) \succsim \#_x\varphi$, where again x is not free in φ, we lose one direction of the law of double negation, namely, $\neg\neg\varphi \to \varphi$, while the other direction remains valid. We retain the converse of contraposition, $(\varphi \to \psi) \to (\neg\psi \to \neg\varphi)$, while losing the contraposition formula, $(\neg\psi \to \neg\varphi) \to (\varphi \to \psi)$. The resulting logic has some intuitionistic flavor, which would be worth determining exactly.

A more dramatic route to non-classical logics would be to change the semantics of # terms altogether. We explore this route further in Section 8.

2.2. From logic to counting. While pure FO is also capable of encoding facts about counting and arithmetic, it is far less extensive. As already mentioned, first-order logic can define the simple counting quantifiers like $\exists^{\geq n}x$; however, first-order logic does so by means of *counting in the syntax*. That is, the formula expressing that there are at least n objects satisfying a given condition achieves this by concatenating n existential quantifiers and adding n conjuncts. Basic arithmetic principles like $\exists^{\geq m}x.\varphi \to \exists^{\geq n}x.\varphi$ for $m \geq n$, thus follow from elementary logical patterns like distribution of existential quantification over conjunction, applied the requisite number of times (e.g., $m - n$ times). This style of counting in the syntax also produces a case-by-case formulation of the Pigeonhole Principle:[2]

EXAMPLE 2.7. Suppose we have k monadic predicate symbols $P_1, \dots, P_k$ and let $n > k$. Then,

$$\big(\exists^{=n}x.\bigvee_{i\leq k} P_i(x) \wedge \forall x.\bigwedge_{i\neq j}\neg(P_i(x)\wedge P_j(x))\big) \to \bigvee_{i\leq k}\exists^{\geq 2}x.P_i(x) \tag{2.6}$$

says that if these k predicates together include n objects, then at least one must include at least two objects. This schema is of course valid for every choice of k and $n > k$.

We will see more examples of counting in the syntax with subsequent sections (see especially Remark 9.15 and Appendix E).

REMARK 2.8. The fact that FO can only count in the syntax reverberates in interesting ways when we consider *finite variable* fragments of FO. While the two-variable fragment is known to have the (bounded) finite-model property [89], which in turn establishes its decidability, this fragment with counting quantifiers $\exists^{\geq n}$ can easily enforce infinite models:

$$\forall x.\exists^{=1}y.R(x,y) \wedge \forall y.\exists^{\leq 1}x.R(x,y) \wedge \exists y.\forall x.\neg R(x,y).$$

Such a language is in fact decidable [45]: like the two-variable fragment without counting, its satisfiability problem is NEXPTIME-complete [103]. However, the complexity analysis of this system and its extensions [69] reveals arithmetical content that does not appear in analyses of the plain two-variable fragment, witness connections to integer programming (Section 3.2.1) and to semi-linear sets (Section 4.1).

2.3. Finite models. It is natural to consider a related system in the same language, but with interpretations restricted to *finite* models. Call such a system $\mathsf{FO}^{\phi}(\#)$. As $\mathcal{L}_{\#}$ extends the language of first-order logic, Trakhtenbrot's theorem tells us that the validities of $\mathsf{FO}^{\phi}(\#)$ are still not computably enumerable. Nonetheless, $\mathsf{FO}^{\phi}(\#)$ and variations on it have also been intensely studied in the literature on finite model theory. See, e.g., [98] or [116] for summaries of relevant work.

As an example of distinctive issues that come up in the finitary setting, one might ask about the *asymptotic probabilities* of formulas in $\mathsf{FO}^{\phi}(\#)$ over finite structures. It was shown in [48] that FO with the Härtig quantifier in fact possesses a zero–one law, just as pure FO does. As possession of a zero–one law is commonly interpreted as evidence that a logic cannot formalize

[2] Even more simply, the Pigeonhole Principle has a natural encoding in propositional logic where complexity theorists have been interested in lower bounds on the lengths of proofs for instances of the principle across different proof systems [26, 75].

https://doi.org/10.1017/bsl.2023.30 Published online by Cambridge University Press

any non-trivial counting, this can be taken as justification for our choice of comparison rather than equality as a primitive. Indeed, $\mathsf{FO}^{\phi}(\#)$ lacks a zero–one law; e.g., $\#_x P(x) \succsim \#_x \neg P(x)$ has asymptotic probability $1/2$. It is conjectured in [48] that (an extension of) $\mathsf{FO}^{\phi}(\#)$ nonetheless possesses a limit law, and that the limits are all rational numbers between 0 and 1.

For many purposes in finite model theory (e.g., descriptive complexity) authors have been motivated to consider proper *extensions* of our language $\mathcal{L}_{\#}$, a notable example being *fixed point logic* with counting [22]. Our purpose here is different: we aim to isolate weaker fragments of this language that might further reveal the subtle interplay between logic and counting, also pinpointing differences and commonalities between finitary and infinitary patterns in counting.

2.4. Fragments of $\mathcal{L}_{\#}$. While full first-order logic with counting may be a natural starting point for exploring our subject, the above observations invite the search for natural fragments and weaker variants of $\mathsf{FO}(\#)$. It may be desirable, for example, to identify *decidable* fragments of $\mathcal{L}_{\#}$. From this perspective it is noteworthy that some familiar ways of taming complexity are less effective here. For example, finite-variable fragments do not result in decidability: as shown by Grädel et al. [46], the two-variable fragment of $\mathsf{FO}(\#)$ is still undecidable (Π^1_1-complete, so we do observe a reduction in complexity, compared to Theorem 2.4). The two-variable fragment of $\mathsf{FO}^{\phi}(\#)$ is also undecidable. Evidently, a significant source of the complexity is the potent combination of counting and arbitrary quantificational-relational reasoning, witness Lemma 2.2. The undecidability proof in [46] for the two-variable fragment crucially involves counting successors along binary relations.

A more dramatic route would be to move to a much tamer syllogistic or propositional fragment [32, 91]. For instance, if we let $\mathcal{L}^0_{\#}$ be the language of propositional logic with count comparisons, the resulting system $\mathsf{PL}(\#)$ is easily shown to be decidable (e.g., it will follow immediately from our results below). This route at once eliminates relational reasoning and first-order quantification.

An alternative route is to put relational reasoning to the side, but still retain first-order quantification. The *monadic* fragment of $\mathcal{L}_{\#}$, which we will call $\mathcal{L}^1_{\#}$, does not allow counting along relations, but it otherwise preserves the counting content of $\mathsf{FO}(\#)$. Observe, for example, that our definition of the infinity quantifier in (2.1) and our reconstruction of logical connectives from count comparisons (Section 2.1) depend in no way on the arity of available predicates. We will thus use $\mathsf{MFO}(\#)$, monadic first-order logic with counting, as a base system to explore richer combinations (Section 3). In this context we will consider adding second-order quantification (system $\mathsf{MSO}(\#)$ in Section 4), as well as the ability to count not just individuals but *sequences* of individuals (systems $\mathsf{MFO}(\sharp)$ and $\mathsf{MSO}(\sharp)$ in Section 5).

https://doi.org/10.1017/bsl.2023.30 Published online by Cambridge University Press

Table 1. A hierarchy of counting languages and logics, covered in Sections 2-7. For each logical system $\mathsf{L}(\#)$ we also have a version $\mathsf{L}^{\phi}(\#)$, where we restrict to finite models. In these systems terms can only denote natural numbers

Language	Logical system	Typical expression
$\mathcal{L}_{\#}$	$\mathsf{FO}(\#)$	$\forall x.\#_y R(x, y) \succ \#_y\big(R(y, x) \wedge P(y)\big)$
$\mathcal{L}_{\sharp}^{2}$	$\mathsf{MSO}(\sharp)$	$\exists Y.\big(\sharp_x P(x) \approx \sharp_{x,u,v}(Y(x) \wedge Y(u) \wedge Y(v))\big)$
$\mathcal{L}_{\sharp}^{1}$	$\mathsf{MFO}(\sharp)$	$\sharp_{x,y}\big(P(x) \wedge P(y)\big) \succsim \sharp_{x,y,z}\big(Q(x) \wedge Q(y) \wedge Q(z)\big)$
$\mathcal{L}_{\#}^{2}$	$\mathsf{MSO}(\#)$	$\exists Y.\big(\#_x Y(x) \approx \#_x Q(x) \wedge \#_x P(x) \succ \#_x Y(x)\big)$
$\mathcal{L}_{\#}^{1}$	$\mathsf{MFO}(\#)$	$\exists y.\big(P(y) \wedge \#_x(P(x) \wedge x \neq y) \succ \#_x Q(x)\big)$
$\mathcal{L}_{\#}^{\mathsf{ml}}$	$\mathsf{ML}(\#)$	$\#(\#\neg p \succ \#p) \succ \#(\#p \succsim \#\neg p)$
$\mathcal{L}_{\#}^{0}$	$\mathsf{PL}(\#)$	$\#\neg p \succsim \#(p \vee q)$

Of course, counting along relations is also common and natural. We therefore explore a tractable *modal* fragment of $\mathcal{L}_{\#}$, which we call $\mathcal{L}_{\#}^{\mathsf{ml}}$, as a way of taming the interaction among counting, quantification, and relational reasoning. A summary appears in Table 1.

Following this work we consider a different route altogether, namely changing the semantics of $\mathcal{L}_{\#}$. Relaxing either the logical interpretation (relativizing to sets of 'admissible' variable assignments; cf. [96]) or the numerical content of the # terms again results in systems that retain much of the character of $\mathsf{FO}(\#)$, while gaining in tractability.

§3. Monadic first-order counting logic.

The system $\mathsf{MFO}(\#)$ of monadic first-order logic with identity and cardinality comparisons, though restricted in its expressive power, still captures a good deal of the natural reasoning mentioned in our Introduction. It is easy to see that *numerical syllogisms* can be represented, and so can simple comparative reasoning with quantifiers like 'most'. But $\mathsf{MFO}(\#)$ can also represent the earlier more complex inference

from	'More A than B are C'	$\big(\#_x\big(A(x) \wedge C(x)\big) \succ \#_x\big(B(x) \wedge C(x)\big)\big)$
and	'More B than C are A'	$\big(\#_x\big(B(x) \wedge A(x)\big) \succ \#_x\big(C(x) \wedge A(x)\big)\big)$
to	'More A than C are B'	$\big(\#_x\big(A(x) \wedge B(x)\big) \succ \#_x\big(C(x) \wedge B(x)\big)\big).$

The underlying Venn diagram-style reasoning will be analyzed more generally below.

Beyond the basic linguistic inference repertoire, $\mathsf{MFO}(\#)$ can also represent some of what we called 'grassroots mathematics'. Note, for instance, that Example 2.7 encoding the Pigeonhole Principle only involved monadic predicates (and in fact did not even need #-terms). In $\mathsf{MFO}(\#)$ we can also express a natural infinitary generalization:

https://doi.org/10.1017/bsl.2023.30 Published online by Cambridge University Press

$$\left(\exists^\infty x. \bigvee_{i \le k} P_i(x) \wedge \forall x. \bigwedge_{i \neq j} \neg(P_i(x) \wedge P_j(x))\right) \to \bigvee_{i \le k} \exists^\infty x. P_i(x), \quad (3.1)$$

stating that infinitely many objects in finitely many disjoint boxes ('pigeonholes') must result in at least one box having infinitely many objects.

We will now look more systematically at what this monadic counting logic can express. Suppose $\mathsf{Pred} = \{P_1, \dots, P_n\}$ is finite, and list the 2^n possible state-descriptions over Pred as $S_1, \dots, S_{2^n}$, so that each $S_i(x)$ is of the form $\bigwedge_{j \in J} P_j(x) \wedge \bigwedge_{j \notin J} \neg P_j(x)$. Call the extension of a state-description S_i in a model a *region*. In $\mathcal{L}^1_\#$ we can easily state count comparisons between regions. A count comparison, such as a statement $\#_x S_i(x) \succsim \#_x S_j(x)$, can be succinctly written with numerical variables replacing cardinalities: $\mathsf{s}_i \ge \mathsf{s}_j$. As the S_i are pairwise disjoint we can more generally encode constraints involving sums of (cardinalities of) regions by disjunctions of state-descriptions. For instance, a sentence like $\#_x \bigvee_i S_i(x) \succsim \#_x \bigvee_j S_j(x)$ encodes a typical linear inequality between sums of variables $\mathsf{s}_1, \dots, \mathsf{s}_{2^n}$:

$$\sum_i \mathsf{s}_i \ge \sum_j \mathsf{s}_j. \quad (3.2)$$

By closing under Booleans we can of course express equality and strict inequality versions of (3.2). When restricting to finite models call the resulting logical system $\mathsf{MFO}^\phi(\#)$. In this case 'solutions' to such (in)equations will always be natural numbers. However, if we allow models of arbitrary cardinality, then solutions may involve infinite cardinal numbers. This is the system that we call $\mathsf{MFO}(\#)$.

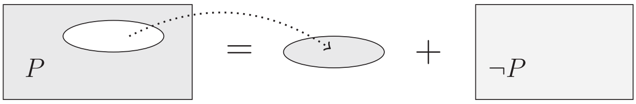
How much more can we express in $\mathsf{MFO}^\phi(\#)$ or $\mathsf{MFO}(\#)$ than the simple linear inequalities in (3.2)? We have already seen an instructive example in the formula (2.1) defining the infinity quantifier. The encoding of $\exists^\infty x. S(x)$ for a state description S is essentially an inequality statement $\mathsf{s} \ge \mathsf{s} + 1$. The use of individual variables here is an instance of a more general pattern, also relevant in the finite case. Indeed, everything we say in the present section will apply equally to $\mathsf{MFO}^\phi(\#)$ and $\mathsf{MFO}(\#)$.

As above, consider two non-overlapping sets $T_1 = \{S_i\}_i$, $T_2 = \{S_j\}_j$ of state-descriptions, whose respective cardinalities we will label $\{\mathsf{s}_i\}_i$ and $\{\mathsf{s}_j\}_j$. Then we can encode not only inequalities like those in (3.2), but also those such as

$$\sum_i \mathsf{s}_i = \sum_j \mathsf{s}_j + k, \quad (3.3)$$

$$\sum_i \mathsf{s}_i > \sum_j \mathsf{s}_j + k. \quad (3.4)$$

https://doi.org/10.1017/bsl.2023.30 Published online by Cambridge University Press



$$\exists \mathbf{y}.\big(\mathsf{diff}(\mathbf{y}) \wedge \textstyle\bigwedge_{i\leq k} P(y_i) \wedge \#_x(P(x) \wedge \bigwedge_{i\leq k} x \neq y_i) \approx \#_x(\bigvee_{i\leq k} x = y_i \vee \neg P(x))\big)$$

FIGURE 1. A visualization of the formula expressing that the number of P points (darkly shaded) is exactly $2k$ greater than the numbers non-P points (lightly shaded), where k is the size of the ‘extracted’ set of P points (i.e., the size of $\mathbf{y}$).

For instance, to express (3.3) we can assert the existence of k distinct objects $\mathbf{y}$ all of which satisfy one of the T_1, such that ‘removing’ these elements from the regions spanned by T_1 results in the same cardinality as the regions spanned by T_2:

$$\exists \mathbf{y}.\Big(\mathsf{diff}(\mathbf{y}) \wedge \bigwedge_{y\in\mathbf{y}} T_1(y) \wedge \#_x\big(\bigwedge_{y\in\mathbf{y}} x \neq y \wedge T_1(x)\big) \approx \#_x T_2(x)\Big).$$

Here $T_1(y)$ is shorthand for $\bigvee_i S_i(y)$, and similarly for $T_2(x)$.

Meanwhile (3.4) is expressed by replacing the equality with a strict inequality. In fact, with k variables $\mathbf{y}$ (in addition to the variable x used in the count comparisons) we can already encode (3.3) and (3.4) with a constant $2k$, simply by taking these variables $\mathbf{y}$ and ‘adding’ them to the regions spanned by the T_2 (see Figure 1 for visualization):

$$\exists \mathbf{y}.\Big(\mathsf{diff}(\mathbf{y})\wedge\bigwedge_{y\in\mathbf{y}} T_1(y)\wedge\#_x\big(\bigwedge_{y\in\mathbf{y}} x \neq y\wedge T_1(x)\big) \approx \#_x\big(\bigvee_{y\in\mathbf{y}} x = y \vee T_2(x)\big)\Big). \tag{3.5}$$

We are effectively stating that $|T_1| \geq k$, and that $|T_1| - k = |T_2| + k$; in other words, $|T_1| = |T_2| + 2k$. Again, the same argument extends to inequality statements.

3.1. Some core principles. Both systems, $\mathsf{MFO}^{\phi}(\#)$ and $\mathsf{MFO}(\#)$, are evidently invariant under automorphisms. In the monadic setting automorphisms are precisely the maps that permute elements within a region: all the points that satisfy a given state-description are indistinguishable. This means that if a property holds for one point in a region, it holds for every point in that region. This theme of *permutation invariance* is characteristic of counting, and it will return when we discuss generalized quantifiers in Section 9.

As demonstrated above, use of individual variables essentially allows manipulating regions—removing or adding points. We can correspondingly state a more general *invariance principle*. Fix some variables $\mathbf{y}$ and a fixed (finite) set $\mathbf{P}$ of predicate letters, and let $\alpha^{\mathbf{y}}(x)$ specify a state-description for x as well as which of the variables $\mathbf{y}$ are (un)equal to x. Then, for any formula φ (in predicates $\mathbf{P}$), if there is at least one x satisfying $\alpha^{\mathbf{y}}$ and φ, then *every* x satisfying $\alpha^{\mathbf{y}}$ also satisfies φ. Codified in a general invariance principle:

$$\exists x.\big(\alpha^{\mathbf{y}}(x) \wedge \varphi(x)\big) \to \#_x\big(\alpha^{\mathbf{y}}(x) \wedge \varphi(x)\big) \approx \#_x\big(\alpha^{\mathbf{y}}(x)\big). \tag{INV}$$

Since either none of the α's satisfy φ or all of them do, once we have specified α in a count formula, reference to φ becomes redundant. In fact, (INV) follows from an even stronger statement (that is, stronger provided we admit infinite models):

$$\#_x\big(\alpha^{\mathbf{y}}(x) \wedge \varphi(x)\big) \succ 0 \to \#_x\big(\alpha^{\mathbf{y}}(x) \wedge \neg\varphi(x)\big) \approx 0. \tag{INV2}$$

A related observation about terms $\#_x\varphi$ is that subformulas of φ that do not involve x do not contribute any fine-grained information to the term's denotation. If the free variables of ψ are not among the bound variables of $\#_x\varphi$, then the following is valid:

$$(\psi \wedge \#_x\varphi \approx \#_x\varphi[{}^{\top}\!/_{\psi}]) \vee (\neg\psi \wedge \#_x\varphi \approx \#_x\varphi[{}^{\bot}\!/_{\psi}]). \tag{SUB}$$

Here $\alpha[{}^{\beta}\!/_{\gamma}]$ is the result of substituting β for every occurrence of γ in α.

3.2. Normal forms. The principles recorded in (INV) and (SUB), together with basic propositional reasoning and a few other elementary principles (see Section 3.4 for the others), allow the derivation of a normal form result, which works uniformly for $\mathsf{MFO}^{\phi}(\#)$ and $\mathsf{MFO}(\#)$. As a first step, we can show that any formula is equivalent to one with no embedded #-terms or quantifiers within #-terms, as these terms can always be replaced by unembedded existential quantifiers. This is already a dramatic departure from full relational $\mathsf{FO}(\#)$, where embedding is non-trivial. (Recall that $\mathsf{FO}(\#)$ with no embedded count comparisons was Π^1_1-complete, in stark contrast to Theorem 2.4.)

Define *depth* $\mathsf{d}(\varphi)$ by recursion, with $\mathsf{d}(\alpha) = 0$ for α atomic, $\mathsf{d}(\varphi \wedge \psi) = \max(\mathsf{d}(\varphi), \mathsf{d}(\psi))$, $\mathsf{d}(\neg\varphi) = \mathsf{d}(\varphi)$, while $\mathsf{d}(\#_x\varphi \succsim \#_y\psi) = \max(\mathsf{d}(\varphi), \mathsf{d}(\psi)) + 1$ and $\mathsf{d}(\exists x.\varphi) = \mathsf{d}(\varphi) + 1$.

Generically, a monadic formula with free variables $\mathbf{y}, x$ can be written in disjunctive normal form $\bigvee_i \big(\alpha_i(\mathbf{y}) \wedge \alpha_i^{\mathbf{y}}(x) \wedge \varphi_i(\mathbf{y}, x)\big)$, where $\alpha_i(\mathbf{y})$ specifies state-descriptions for $\mathbf{y}$ and which of these variables are (un)equal, $\alpha_i^{\mathbf{y}}(x)$ is as in the previous subsection, and $\varphi_i(\mathbf{y}, x)$ is some other formula that may in general have positive depth. We want to show that any formula

$$\#_x \bigvee_{i \in I} \big(\alpha_i(\mathbf{y}) \wedge \alpha_i^{\mathbf{y}}(x) \wedge \varphi_i(\mathbf{y}, x)\big) \succsim \#_x \bigvee_{j \in J} \big(\alpha_j(\mathbf{y}) \wedge \alpha_j^{\mathbf{y}}(x) \wedge \varphi_j(\mathbf{y}, x)\big)$$

is equivalent to one with no embedded count comparisons or quantifiers. In other words this formula is equivalent to one of depth 1. First, by (SUB) we can take the subformulas $\alpha_i(\mathbf{y}), \alpha_j(\mathbf{y})$ outside the count comparisons, which leaves

$$\#_x \bigvee_{i \in I} \big(\alpha_i^{\mathbf{y}}(x) \wedge \varphi_i(\mathbf{y}, x)\big) \gtrsim \#_x \bigvee_{j \in J} \big(\alpha_j^{\mathbf{y}}(x) \wedge \varphi_j(\mathbf{y}, x)\big)$$

to analyze. Let κ_k range over formulas $\exists x.\big(\alpha_k^{\mathbf{y}}(x) \wedge \varphi_k(\mathbf{y}, x)\big)$ for $k \in I \cup J$. Then by appeal to (INV), we have the equivalent formula:

$$\bigvee_{K \subseteq I \cup J} \big(\bigwedge_{k \in K} \kappa_k \wedge \bigwedge_{k \notin K} \neg\kappa_k \wedge \#_x \bigvee_{i \in I \cap K} \alpha_i^{\mathbf{y}}(x) \gtrsim \#_x \bigvee_{j \in J \cap K} \alpha_j^{\mathbf{y}}(x)\big)$$

Note that we have traded one level of # embedding for one existential quantifier. Since $\alpha_i^{\mathbf{y}}, \alpha_j^{\mathbf{y}}$ are of depth 0, this concludes the argument for:

Lemma 3.1. *Every $\mathcal{L}_{\#}^1$ formula is equivalent to one in which every count comparison subformula has depth exactly* 1.

Using Lemma 3.1, the main result of this section is:

Theorem 3.2. *Every depth $k+1$ sentence is equivalent, in* $\mathsf{MFO}^{\phi}(\#)$ *as well as in* $\mathsf{MFO}(\#)$, *to a disjunction of conjunctions of sentences specifying $T_1 = T_2 + m$ or $T_1 > T_2 + m$, for T_1, T_2 sums of (cardinalities of) state-descriptions, and $m \leq 2k$.*

Proof. We show more generally that a formula of depth $k+1$ over predicates $\mathbf{P}$ with free variables $\mathbf{y} = y_1, \ldots, y_n$ is equivalent to a disjunction

$$\bigvee \big(\alpha(\mathbf{y}) \wedge (\sigma)_{\alpha(\mathbf{y})}\big), \tag{3.6}$$

where $\alpha(\mathbf{y})$ ranges over possible descriptions of $\mathbf{y}$, and σ is a complete description of the regions over $\mathbf{P}$, i.e., specifying $T_1 = T_2 + m$ or $T_1 > T_2 + m$ for all $m \leq 2(n+k)$. The notation $(\sigma)_{\alpha(\mathbf{y})}$ denotes a formula that specifies the description σ on the assumption of $\alpha(\mathbf{y})$. In other words, we claim that for each disjunct of (3.6), for all variable assignments s,

$$\mathcal{M}, s \vDash \alpha(\mathbf{y}) \wedge (\sigma)_{\alpha(\mathbf{y})} \Rightarrow \mathcal{M} \text{ satisfies the } 2(n+k) \text{ description } \sigma. \tag{3.7}$$

The statement in the theorem will be the special case of (3.6) with no free variables ($n = 0$).

Example 3.3. For an example of such a disjunct over one predicate letter P, see the formula inside the existential quantifier in Figure 1. This formula has k free variables and depth 1. Here $\alpha(\mathbf{y})$ is the formula $\mathsf{diff}(\mathbf{y}) \wedge \bigwedge_{i \leq k} P(y_i)$, while $(\sigma)_{\alpha(\mathbf{y})}$ is the count comparison. Note that

$(\sigma)_{\alpha(\mathbf{y})}$ has free variables and it ‘means’ that $|P| = |\neg P| + 2k$ provided $\alpha(\mathbf{y})$ holds. $\dashv$

To show that depth $k+1$ formulas are always equivalent to formulas (3.6) satisfying (3.7), we proceed by inducting on k, starting with the case of depth 1 formulas ($k = 0$) in free variables $\mathbf{y} = y_1, \dots, y_n$. The critical case is a count comparison:

$$\#_x \bigvee \alpha(\mathbf{y}, x) \succsim \#_x \bigvee \beta(\mathbf{y}, x).$$

As before, by (SUB) we can separate out the descriptions of $\mathbf{y}$ to obtain a formula

$$\bigvee \left(\gamma(\mathbf{y}) \wedge \#_x \bigvee \alpha^{\mathbf{y}}(x) \succsim \#_x \bigvee \beta^{\mathbf{y}}(x)\right), \tag{3.8}$$

where we have a $\gamma(\mathbf{y})$ disjunct exactly when $\gamma(\mathbf{y}) = \alpha(\mathbf{y}) = \beta(\mathbf{y})$; that is, all disjuncts inside the # terms must agree on the characterization of variables $\mathbf{y}$. It is then straightforward to check, by considering all cases, that the count comparison in each disjunct of (3.8), in context $\gamma(\mathbf{y})$, asserts $T_1 = T_2 + m$ or $T_1 > T_2 + m$ for $m \leq 2n$ (or a disjunction of such comparisons). So this fits the form in (3.6), and (3.7) is satisfied.

In general, the normal forms (3.6) for a fixed k and n are closed under Boolean combinations, so we only need to consider the case of depth $k+1$ and n variables. By Lemma 3.1 we can assume all count comparison subformulas have depth 1, so it suffices to consider an existential quantification, which by induction we assume is

$$\exists z. \bigvee \left(\alpha(\mathbf{y}, z) \wedge (\sigma)_{\alpha(\mathbf{y},z)}\right).$$

Such a formula will be equivalent to

$$\bigvee \exists z. \left(\alpha(\mathbf{y}, z) \wedge (\sigma)_{\alpha(\mathbf{y},z)}\right)$$

and indeed to

$$\bigvee \left(\alpha(\mathbf{y}) \wedge \exists z. \left(\alpha^{\mathbf{y}}(z) \wedge (\sigma)_{\alpha(\mathbf{y},z)}\right)\right). \tag{3.9}$$

It remains to be seen that (3.9) is of the form (3.6) with each disjunct satisfying (3.7). By the inductive assumption we know that for any s, if $\mathcal{M}, s \vDash \alpha(\mathbf{y}, z) \wedge (\sigma)_{\alpha(\mathbf{y},z)}$, then $\mathcal{M}$ satisfies the $2(n+k)$ description σ. But if $\mathcal{M}, s \vDash \alpha(\mathbf{y}) \wedge \exists z. \left(\alpha^{\mathbf{y}}(z) \wedge (\sigma)_{\alpha(\mathbf{y},z)}\right)$, then there is a z-variant s' of s such that $\mathcal{M}, s' \vDash \alpha(\mathbf{y}, z) \wedge (\sigma)_{\alpha(\mathbf{y},z)}$, which establishes the result.

3.2.1. Connection to integer programming. As with ordinary monadic first-order logic, putting a sentence into normal form may result in a significantly longer formula. The satisfiability problem for monadic first-order logic (as for the two-variable fragment) is NExpTime-complete [80], even though checking satisfiability of normal forms is in NP. As with monadic logic, checking satisfiability of a normal form in $\mathsf{MFO}^{\phi}(\#)$ is of relatively low complexity. In fact, it is of the same complexity. A set of (in)equalities of types (3.3) and (3.4) give us an *integer program*, whose solvability is known to be decidable in NP-time [18]. Meanwhile, the special case of integer programming in which all coefficients are 1 or 0—in other words, the special case of inequalities like those in (3.2)—was already included in Karp's [67] original list of NP-complete problems. With this lower bound we can conclude that the satisfiability problem for normal forms in $\mathsf{MFO}^{\phi}(\#)$ is NP-complete.

3.3. Questions of definability. Theorem 3.2 affords a refined understanding of the numerical relations that can be defined in $\mathsf{MFO}^{\phi}(\#)$, as well as $\mathsf{MFO}(\#)$. Where T is a set of state-descriptions, let $|T|_{\mathcal{M}}$ denote the sum of cardinalities of extensions in $\mathcal{M}$ of state-descriptions in T. We will say that $\mathcal{M} \sim_k \mathcal{M}'$ if for all T_1, T_2 and all $m \leq k$,

$$|T_1|_{\mathcal{M}} \geq |T_2|_{\mathcal{M}} + m \quad \text{iff} \quad |T_1|_{\mathcal{M}'} \geq |T_2|_{\mathcal{M}'} + m$$

Then, where $\mathcal{M} \equiv_k \mathcal{M}'$ signifies that $\mathcal{M}$ and $\mathcal{M}'$ agree on all sentences up to depth k, Theorem 3.2 immediately gives:

Corollary 3.4. *$\mathcal{M} \sim_{2k} \mathcal{M}'$ iff $\mathcal{M} \equiv_{k+1} \mathcal{M}'$.*

As an initial example, we can characterize precisely the binary logical quantifiers definable in $\mathsf{MFO}^{\phi}(\#)$ (see Section 9 for a proof, and for further discussion of generalized quantifiers):

Theorem 3.5. *The binary quantifiers definable in* $\mathsf{MFO}^{\phi}(\#)$ *correspond exactly to those expressible in the first-order theory of* $\langle \mathbb{N}; > \rangle$.

This includes many of the standard logical quantifiers: 'most', 'all', 'some', 'all but one', 'at least two', etc. The following gives an example of a statement that cannot be expressed.

Fact 3.6. *'There are twice as many Ps as Qs'* cannot be expressed in $\mathsf{MFO}^{\phi}(\#)$.

Proof. Supposing it could, such a sentence would have some depth $k + 1$. In light of Corollary 3.4, it suffices to show that, for any k, we can find $\mathcal{M}, \mathcal{M}'$ that disagree on the statement and yet $\mathcal{M} \sim_{2k} \mathcal{M}'$. Define a first model $\mathcal{M}$ with $9k$ elements, such that $|P^{\mathcal{M}}| = 6k$ while $|Q^{\mathcal{M}}| = 3k$. The statement clearly holds of $\mathcal{M}$. But now define $\mathcal{M}'$ with $9k + 1$ elements, such that $|P^{\mathcal{M}'}| = 6k + 1$ and again $|Q^{\mathcal{M}'}| = 3k$. The statement fails in $\mathcal{M}'$, yet $\mathcal{M} \sim_{2k} \mathcal{M}'$. ⊣

For a second example, consider a natural rendering of the natural language expression 'many', often taken to refer to a number above some contextual

https://doi.org/10.1017/bsl.2023.30 Published online by Cambridge University Press

threshold. On a more sophisticated, but not uncommon, reading (cf. [111, 128]), 'Many Qs are P' amounts to a comparison between the *proportion* of Ps among the Qs and the proportion of Ps overall, which we might symbolize as

$$\frac{\#_x\big(P(x) \wedge Q(x)\big)}{\#_x Q(x)} \succ \frac{\#_x P(x)}{\#_x \top}. \tag{3.10}$$

Fact 3.7. '*Many Qs are P*' cannot be expressed in $\mathsf{MFO}^\phi(\#)$.

Proof. Again, for any k, we must find two models $\mathcal{M}, \mathcal{M}'$ that disagree on the statement and yet $\mathcal{M} \sim_{2k} \mathcal{M}'$. It suffices to specify the cardinalities of four regions within the model: $p = |P \cap \overline{Q}|$, $q = |Q \cap \overline{P}|$, $r = |P \cap Q|$, $s = |\overline{P \cup Q}|$.

In both models let $r = k$, $q = 3k$, and $p = 4k$. In $\mathcal{M}$ let $s = 15k$, while in $\mathcal{M}'$ let $s = 11k$. In both cases $s > p + q + r + 2k$, and this is the crucial case to establish that $\mathcal{M} \sim_{2k} \mathcal{M}'$, and $\mathcal{M} \equiv_{k+1} \mathcal{M}'$. However, in $\mathcal{M}$ we have $\frac{r}{r+q} > \frac{p+r}{p+q+r+s}$, while in $\mathcal{M}'$ the inequality fails. ⊣

We will return to more analysis of natural language constructions in Section 9. Note that Corollary 3.4 can be used to derive undefinability results in $\mathsf{MFO}(\#)$ as well:

Fact 3.8. The successor function on infinite cardinals is not expressible in $\mathsf{MFO}(\#)$.

Proof. Every two models that agree on the *order* of cardinalities for infinite definable sets will stand in the relation $\sim_k$ for all k. ⊣

3.3.1. Interpolation failure. Another consequence of Theorem 3.2 is a particularly simple normal form result for the 'letterless' fragment of $\mathcal{L}^1_\#$, that is, the fragment with no predicate symbols, built up from atomic formulas $\top$ and $\bot$. In fact, the normal forms are identical to those for monadic first-order logic with the infinity quantifier [24]:

Lemma 3.9. *Every letterless sentence is equivalent in* $\mathsf{MFO}(\#)$ *to a disjunction of formulas having one of the following forms* $\exists^\infty x.\top$, $\forall^\infty x.\bot \wedge \exists^{\geq k}\top$, *or* $\exists^{=k}x.\top$.

For the restriction $\mathsf{MFO}^\phi(\#)$ to finite models, this simplifies even further to include only statements of the form $\exists^{>k}.\top$ and $\exists^{=k}x.\top$. As a consequence we can show:

Proposition 3.10. *Neither* $\mathsf{MFO}^\phi(\#)$ *nor* $\mathsf{MFO}(\#)$ *enjoys the interpolation property.*

Proof. Let $\varphi(P)$ be the formula:

$$\forall^\infty x.\bot \wedge \#_x(P(x)) \approx \#_x(\neg P(x)),$$

which is only true in finite models of even sizes. Let $\psi(Q)$ be the formula:

$$\exists x.\#_y(y \neq x \wedge Q(y)) \approx \#_y(y \neq x \wedge \neg Q(y)),$$

which in finite models requires the domain to be odd. Evidently $\varphi(P) \vDash \neg\psi(Q)$. Let χ be a purported interpolant: $\varphi(P) \vDash \chi \vDash \neg\psi(Q)$. As χ must be letterless, Lemma 3.9 implies that it must be a disjunction of sentences with one of the three specified forms. Furthermore, as it is entailed by $\varphi(P)$ we can assume that $\exists^\infty x.\top$ is not a disjunct. A straightforward case analysis shows that χ must either be true only in models up to some fixed size—in which case it cannot be entailed by $\varphi(P)$—or it is true in all finite models from some finite size onward—in which case it cannot entail $\neg\psi(Q)$. ⊣

A familiar way of extending a language to guarantee interpolation is to allow second-order quantification. We will turn to such an extension below in Section 4. But first, we analyze the reasoning content of our normal form analysis a bit further.

3.4. Questions of axiomatization. What is the calculus of valid reasoning suggested by our current systems? For both of our basic monadic systems, $\mathsf{MFO}^\phi(\#)$ and $\mathsf{MFO}(\#)$, we can locate a kind of separation between two components: (a) the general, more 'logical' principles that allow our normal form result (Theorem 3.2) and (b) more specific numerical reasoning for solving systems of inequalities. We discuss each component in turn for the system $\mathsf{MFO}^\phi(\#)$ which allows only finite domains. The general system $\mathsf{MFO}(\#)$ involves one more component for dealing with infinite sets that we will remark on at the end.

3.4.1. Step I. The *normal form principles* underlying our normal form result are as follows:

(a) general validities of propositional and first-order predicate logic,
(b) the two general principles (INV) and (SUB) highlighted earlier,
(c) the linear order properties of the relation $\succsim$.

Here the linearity in Principle (c), used in our case distinctions, is worth high-lighting:

$$\#_x\varphi \succsim \#_x\psi \vee \#_x\psi \succsim \#_x\varphi. \qquad \text{(COMP)}$$

The soundness of (COMP) in $\mathsf{MFO}(\#)$ depends on the axiom of choice. Indeed, (COMP) is equivalent to the axiom of choice [55]. Significantly, in the generalized semantics discussed in Section 8, Principles (a) and (b) will remain valid, while the strong reasoning principle (c) is naturally replaced by just the pre-order properties for $\succsim$.

3.4.2. Step II. As a result of the normal form analysis, we are left with a satisfiability problem for inequalities all of whose variables denote natural numbers. This system can be solved effectively, e.g., using the well-known *Fourier–Motzkin algorithm* [115, Section 12.2].

At this stage, we might say that we have solved the reasoning problem in the spirit of this paper, having used a simple combination of logic and counting. The above calculus uses logic to reduce a reasoning problem to a numerical one that is most elegantly solved on its own terms. This is precisely the sort of combination that we find natural and insightful.

REMARK 3.11. Even so, we could go further in *Step III*, and determine the exact arithmetical principles that drive the Fourier–Motzkin algorithm. Here is a sketch.

The algorithm works as follows. One picks a variable s as long as still possible, and then considers one of three cases.

(i) The variable s occurs only to the right in $\geq$-inequalities of the system. Then s can be dropped from all inequalities: setting its value to 0 will always suffice.

(ii) The variable s occurs only to the left in $\geq$-inequalities. Then all inequalities where s occurs can be dropped, since they can be made true at the end by choosing some suitably large value for s.

(iii) In case s occurs both to the left and to the right in inequalities, one groups the inequalities of the form $\mathsf{u} \geq \mathsf{s} + \mathsf{v}, \mathsf{w} > \mathsf{s} + \mathsf{z}$ and those of the forms $\mathsf{y} + \mathsf{s} \geq \mathsf{t}$, $\mathsf{r} + \mathsf{s} > \mathsf{x}$, and forms all sums as follows: $\mathsf{u} \geq \mathsf{s} + \mathsf{v}$ with $\mathsf{y} + \mathsf{s} \geq \mathsf{t}$ gives $\mathsf{u} + \mathsf{y} \geq \mathsf{v} + \mathsf{t}$; $\mathsf{u} \geq \mathsf{s} + \mathsf{v}$ with $\mathsf{r} + \mathsf{s} > \mathsf{x}$ gives $\mathsf{u} + \mathsf{r} > \mathsf{v} + \mathsf{x}$, and so on.

In the end, a set of variable-free statements about concrete natural numbers remains, which can be inspected immediately for truth or falsity.

Now each step of this algorithm can be checked for the principles that guarantee its soundness. Here are a few representative illustrations. All steps involve evident principles for inequalities, such as symmetry and associativity of addition, and monotonicity inferences such as the implication from $\mathsf{u} \geq \mathsf{v}$ to $\mathsf{u} + \mathsf{z} \geq \mathsf{v}$. Step (i) also involves the equality $\mathsf{v} = \mathsf{v} + 0$, while step (ii) involves $\mathsf{u} + \mathsf{v} \geq \mathsf{v}$. The key step (iii) involves principles like the equivalence of $\mathsf{z} + \mathsf{u} \geq \mathsf{v} + \mathsf{u}$ and $\mathsf{z} \geq \mathsf{v}$ and addition principles such as the implication from $\mathsf{u} \geq \mathsf{v}$ and $\mathsf{w} \geq \mathsf{z}$ to $\mathsf{u} + \mathsf{w} \geq \mathsf{v} + \mathsf{z}$. The final inspection step involves some simple principles for the successor function, if we think of numbers as encoded in a unary format.

The preceding observations amount to what one might call a 'mixed' axiomatization of the system $\mathsf{MFO}^{\phi}(\#)$, letting the logic do what it is good at: reducing assertions to normal form, and then letting the arithmetical component do what *it* is good at: solving equational problems involving numbers. By itself, a two-stage analysis with reductions to syntactic normal forms plus a separate combinatorial analysis of the latter is a common practice in logic, e.g., in quantifier elimination arguments. However, the above specific division of labor between logic and counting is a perfect fit with the methodological spirit of this paper, and with the general empirical reasoning practices that we started with. We will return to such combinations of logic and (explicit) arithmetic a bit more systematically in Section 6.

Even so, it is also natural to explore the road of greater purity, and ask for a purely logical axiomatization, or a purely numerical one. We consider each of these roads in turn.

Can the arithmetical steps in the Fourier–Motzkin algorithm be replaced by an illuminating *purely logical proof system* that goes beyond routine transcription? There is an interesting conceptual issue here. The variable-elimination step in the algorithm typically forms sums of single variables in its step (iii), and these sums have no direct interpretation in our logical systems: in particular, $|P| + |P|$ has no defining expression in our logical languages (Fact 3.6). There are ways of dealing with this problem, for instance, by adding special inference rules as is done in [32], which essentially axiomatizes the slightly smaller system $\mathsf{PL}(\#)$ (cf. the discussion in Appendix A). Such inference rules can be seen as expressing the admissibility of certain *model constructions* for the logic, such as taking disjoint unions. This makes sense in our case, since, while$|P| + |P|$ may not be definable in a given model $\mathcal{M}$, it does denote the extension of P in the disjoint union of $\mathcal{M}$ with itself. Even so, we are not aware of obvious model constructions matching the invariants needed for $\mathsf{MFO}(\#)$, and therefore leave this issue as an open problem.

Equally well, in terms of purity, one could go to the opposite side and ask for a *purely numerical calculus* for our systems. We could restrict ourselves to (the monadic fragment of) the small sublanguage $\mathcal{L}^-_\#$ consisting only of predication, variable inequality, and count comparison (Section 2.1). As all logical operators are definable there, such an axiomatization would be possible in principle. For example, the following numerical claims (suppressing individual variables as all are chosen fresh) capture the basic principles of propositional logic:

(1) $\#(\#\varphi \succsim \#\psi) \succsim \#\varphi$,
(2) $\#\big(\#(\#\chi \succsim \#\varphi) \succsim \#(\#\psi \succ \#\varphi)\big) \succsim \#\big(\#(\#\chi \succsim \#\psi) \succsim \#\varphi\big)$,
(3) $\#(\#\varphi \succsim \#\psi) \succsim \#\big(\#(0 \succsim \#\psi) \succsim \#(0 \succsim \varphi)\big)$.

However, what one wants is not transcription, but an independently motivated numerical system that generates the logic. As with the purely logical axiomatization, we leave providing an illuminating purely numerical axiomatization of our systems as an open problem.

https://doi.org/10.1017/bsl.2023.30 Published online by Cambridge University Press

3.4.3. Infinite cardinalities. In the general system $\mathsf{MFO}(\#)$, we must also deal with infinite cardinalities. This makes no difference to the principles producing our normal forms, but it changes the subsequent phase of solving equations. The key observation is that there is a simple expression distinguishing the infinite from the finite extensions, namely $\mathsf{s} \geq \mathsf{s} + 1$. We can thus completely separate reasoning about inequalities among finite variables from reasoning about the variables denoting infinite sets (cf. [32]). This is done systematically in 4.2 for the second-order version of $\mathsf{MFO}(\#)$, to which we now turn.

§4. Monadic second-order counting logic. From a logical point of view, a natural extension of $\mathcal{L}^1_\#$ is to allow quantification over predicates. Call the resulting language $\mathcal{L}^2_\#$, and the finitary and general systems $\mathsf{MSO}^\phi(\#)$ and $\mathsf{MSO}(\#)$, respectively. One immediate observation is that, while in MFO count comparisons $\#_x\varphi \gtrsim \#_x\psi$ are not definable from the Härtig quantifier $\#_x\varphi \approx \#_x\psi$ alone (see, e.g., [100, p. 470]), with second-order quantification this is straightforward:

$$\exists X.\big(\#_x\varphi \approx \#_x(\psi \vee X(x))\big).$$

How much more powerful will $\mathsf{MSO}^\phi(\#)$ and $\mathsf{MSO}(\#)$ be in comparison to $\mathsf{MFO}^\phi(\#)$ and $\mathsf{MFO}(\#)$? The question is of some interest, since it is known (at least since [1]) that adding second-order quantification to monadic first-order logic does not increase expressive power. At the same time, if we add quantification over *finite* sets to MFO this becomes equivalent to monadic logic with the infinity quantifier (see [126] for the case without equality, or Appendix B including equality).

The failure of interpolation (Proposition 3.10) shows that we could not expect a similar collapse when adding monadic second-order quantification to our counting extensions of MFO. We saw that $\mathsf{MFO}(\#)$ can already distinguish between finite and infinite, so in effect we automatically gain access to quantification over finite sets. In fact we gain much more.

Example 4.1. As a preview, within the finite setting, in contrast to $\mathsf{MFO}^\phi(\#)$ (Fact 3.6), in $\mathsf{MSO}^\phi(\#)$ the statement 'There are twice as many Ps as Qs' now becomes expressible:

$$\exists X.\big(\#_y(X(y) \wedge \neg Q(y)) \approx \#_y Q(y) \wedge \#_y P(y) \approx \#_y(X(y) \vee Q(y))\big).$$

This essentially asserts the existence of a set whose extension outside of Q is the same size as Q, and that P is the same size as the union of these two.

It turns out that Example 4.1 is just the tip of the iceberg. In addition to obviously guaranteeing interpolants, there is another sense in which these second-order systems, $\mathsf{MSO}^\phi(\#)$ and $\mathsf{MSO}(\#)$, 'fill in the gaps' of

$\mathsf{MFO}^{\phi}(\#)$ and $\mathsf{MFO}(\#)$. While the latter systems could enforce a certain type of inequality between sums, namely those in Equations (3.3) and (3.4), the second-order versions are capable of enforcing arbitrary linear constraints over cardinalities. We now proceed to make this more precise, first in the finitary case, then infinitary.

4.1. Finitary case. We saw above that normal forms in $\mathsf{MFO}^{\phi}(\#)$ correspond to (disjunctions of) sets of inequality constraints, a class whose solvability problem is already NP-complete. In the general setting of integer programming, there is a close correspondence between sets of linear inequalities and quantifier-free formulas of *Presburger Arithmetic*, that is, first-order logic with addition over the natural numbers (see, e.g., [97]). The sets of solutions to such inequalities (or equivalently, assignments satisfying Presburger formulas with free variables) are exactly the *semi-linear* sets [44], a generalization of the 'ultimately periodic' sets of numbers:

Definition 4.2. A set $\mathcal{V} \subseteq \mathbb{N}^n$ of n-ary vectors is called *linear* if there is a system of equations over variables $\mathsf{v}_1, \dots, \mathsf{v}_n, \mathsf{u}_1, \dots, \mathsf{u}_m$ and constants $b_1, \dots, b_n, a_{1,1}, \dots, a_{n,m}$,

$$\begin{pmatrix} \mathsf{v}_1 \\ \vdots \\ \mathsf{v}_n \end{pmatrix} = \begin{pmatrix} b_1 + a_{1,1}\mathsf{u}_1 + & \dots & +a_{1,m}\mathsf{u}_m \\ \vdots & & \vdots \\ b_n + a_{n,1}\mathsf{u}_1 + & \dots & +a_{n,m}\mathsf{u}_m \end{pmatrix} \tag{4.1}$$

such that $\mathbf{x} \in \mathcal{V}$ if and only if there exist values of $\mathsf{u}_1, \dots, \mathsf{u}_m$ for which $\mathbf{v} = \mathbf{x}$ is a solution to (4.1). We say $\mathcal{V} \subseteq \mathbb{N}^n$ is *semi-linear* if it is a finite union of linear sets.

Definition 4.3. Suppose $S_1, \dots, S_n$ are some state-descriptions over predicates $\mathbf{P}$, and that φ is an $\mathcal{L}^2_{\#}$ sentence in these same predicates $\mathbf{P}$. We say that φ *defines* a set $\mathcal{V} \subseteq \mathbb{N}^n$ just in case, for any model $\mathcal{M}$, we have

$$\mathcal{M} \vDash \varphi \quad \text{iff} \quad [|S_1|_{\mathcal{M}}, \dots, |S_n|_{\mathcal{M}}] \in \mathcal{V}. \tag{4.2}$$

Lemma 4.4. *Every semi-linear set is definable in* $\mathsf{MSO}^{\phi}(\#)$.

Proof. As $\mathcal{L}^2_{\#}$ closes under disjunction, it suffices to show that every linear set is definable. So we describe how to encode any linear set of the form in (4.1) by an $\mathcal{L}^2_{\#}$ sentence. In words:

(i) For all $i \leq n$, assert the existence of:
- Sets $Z_{i,j,1} \dots, Z_{i,j,a_{i,j}}$ (none if $a_{i,j} = 0$) for all $j \leq m$,
- Individuals $z_{i,1}, \dots, z_{i,b_i}$ (none if $b_i = 0$).

(ii) Add conjuncts for:
- $\#_x\big(Z_{i,j,p}(x) \wedge Z_{i,j',p'}(x)\big) \approx 0$, whenever $j \neq j'$ or $p \neq p'$,
- $\neg Z_{i,j,p}(z_{i,l})$, for all i, j, p, l,
- $\#_x Z_{i,j,p}(x) \approx \#_x Z_{i',j,p'}(x)$, for all j and all i, i', p, p',

(iii) Finally, conjoin these together with the claim that each state-description S_i has the same cardinality as the union of all $Z_{i,j,p}$, together with $z_{i,1}, \dots, z_{i,b_i}$:

$$\#_x S_i(x) \approx \#_x\Big(\bigvee_{\substack{j \leq m \\ p \leq a_{i,j}}} Z_{i,j,p}(x) \vee \bigvee_{l \leq b_i} x = z_{i,l}\Big).$$

For a given $i \leq n$ and $j \leq m$, the sets $Z_{i,j,p}$ correspond to $a_{i,j}$-many copies of the variable u_j in (4.1). The individual variables $z_{i,l}$ count the constant 'base' number b_i. The numerical equalities stated in (iii) guarantee that each 'variable' $|S_i|_{\mathcal{M}}$ has the right cardinality according to (4.1), under the conditions specified by (ii). By existentially quantifying all variables (i) the resulting formula defines the linear set in (4.1) in the sense of (4.2). ⊣

We now want to show the other direction, that all $\mathcal{L}^2_{\#}$ sentences in fact define semi-linear sets. Toward this result we first note that $\mathsf{MSO}(\#)$ possesses a prenex normal form.

Lemma 4.5. *Every $\mathcal{L}^2_{\#}$ sentence in predicates $\mathbf{P}$ is equivalent—in $\mathsf{MSO}(\#)$ as well as in $\mathsf{MSO}^{\phi}(\#)$—to one in prenex form, that is, of the form $Q_1 X_1, \dots, Q_n X_n.\varphi(\mathbf{P}, X_1, \dots, X_n)$, where $\varphi(\mathbf{P}, X_1, \dots, X_n)$ is a first-order $\mathcal{L}^1_{\#}$ sentence (treating $X_1, \dots, X_n$ as additional predicates) and $Q_1, \dots, Q_n$ are second-order quantifiers.*

Proof Sketch. The argument is as usual for prenex normal forms in first-order logic. As in case of $\mathsf{MFO}(\#)$, the soundness of (INV) allows us to extract any first-order or second-order quantifier from the scope of a #-term. The only case we need to consider is a first-order (universal) quantifier scoping directly over a second-order quantifier. The point is to convert the first-order universal quantifier into a universal second-order quantifier restricted to singleton sets. That is, a formula $\forall x.QY.\varphi$, with Q a second-order quantifier, will be equivalent to $\forall X.QY.\forall x.\big(\forall z(X(z) \leftrightarrow z = x) \rightarrow \varphi\big)$. ⊣

By Theorem 3.2 we know that $\varphi(\mathbf{P}, X_1, \dots, X_n)$ has a normal form involving expressions $T_1 = T_2 + m$ and $T_1 > T_2 + m$, where T_1, T_2 are cardinalities of state-descriptions over $\mathbf{P}$ and additional predicates $X_1, \dots, X_n$. Such formulas are thus easily seen to be semi-linear, indeed linear. As semi-linear sets are closed under Boolean combinations, and second-order quantifiers distribute over disjunction, the main goal is to show:

Lemma 4.6. *Let X be a predicate variable, and let $O_1, \dots, O_n$ be either predicate letters or predicate variables. Suppose that $\varphi(O_1, \dots, O_n, X)$ defines a linear set in state-descriptions $S_1, \dots, S_{2^{n+1}}$ over $O_1, \dots, O_n, X$. Then $\exists X.\varphi(O_1, \dots, O_n, X)$ defines a linear set in state-descriptions $S'_1, \dots, S'_{2^n}$ over just $O_1, \dots, O_n$.*

https://doi.org/10.1017/bsl.2023.30 Published online by Cambridge University Press

Proof. As $\varphi(O_1, \dots, O_n, X)$ is linear, we can assume it defines the solutions to

$$\begin{pmatrix} |S_1| \\ \vdots \\ |S_{2^{n+1}}| \end{pmatrix} = \begin{pmatrix} b_1 + a_{1,1}\mathsf{u}_1 + & \cdots & +a_{1,m}\mathsf{u}_m \\ \vdots & & \vdots \\ b_{2^{n+1}} + a_{2^{n+1},1}\mathsf{u}_1 + & \cdots & +a_{2^{n+1},m}\mathsf{u}_m \end{pmatrix}. \tag{4.3}$$

To show that $\exists X.\varphi(O_1, \dots, O_n, X)$, too, is linear, we define another linear set of equations by 'projecting out' the variable X.

Specifically, note for each state-description S'_k, both $S'_k \wedge X$ and $S'_k \wedge \neg X$ are (equivalent to) some state-descriptions, S_i and S_j, and in fact S'_k is equivalent to $S_i \vee S_j$. The new linear system in 2^n variables is then as follows for each $k \leq 2^n$:

$$|S'_k| = b_i + b_j + (a_{i,1} + a_{j,1})\mathsf{u}_1 + \cdots + (a_{i,m} + a_{j,m})\mathsf{u}_m. \tag{4.4}$$

It remains only to show that:

$$\begin{aligned} &\mathcal{M} \vDash \exists X.\varphi(O_1, \dots, O_n, X) \\ &\quad \Leftrightarrow [|S'_1|_{\mathcal{M}}, \dots, |S'_{2^n}|_{\mathcal{M}}] \text{ is a solution to Equation (20).} \end{aligned}$$

($\Rightarrow$): If $\mathcal{M} \vDash \exists X.\varphi(O_1, \dots, O_n, X)$ then for some subset A of the domain, we have $\mathcal{M}, s^X_A \vDash \varphi(O_1, \dots, O_n, X)$. Treating X now as a predicate constant, we have a model $\mathcal{M}'$ for which $X^{\mathcal{M}'} = A$, and by assumption this gives a solution $[|S_1|_{\mathcal{M}'}, \dots, |S_{2^{n+1}}|_{\mathcal{M}'}]$ to (4.3). But each state-description S'_k is equivalent to a disjunction $S_i \vee S_j$, whose cardinality is the sum $|S_i| + |S_j|$. Therefore $\mathcal{M}'$ will satisfy each of the constraints in (4.4). As the state-descriptions $S'_1, \dots, S'_{2^n}$ are independent of X, this means $[|S'_1|_{\mathcal{M}}, \dots, |S'_{2^n}|_{\mathcal{M}}]$ also gives a solution to (4.4).

($\Leftarrow$): Suppose $[|S'_1|_{\mathcal{M}}, \dots, |S'_{2^n}|_{\mathcal{M}}]$ gives a solution to (4.4) for some particular choices $\mathsf{u}_1, \dots, \mathsf{u}_m$. We need to find a set A such that $\mathcal{M}, s^X_A \vDash \varphi(O_1, \dots, O_n, X)$. Since the extensions of $S'_1, \dots, S'_{2^n}$ are all disjoint, to define A it suffices to identify subsets of each $[\![S'_k]\!]_{\mathcal{M}}$. As above, suppose S'_k is equivalent to $S_i \vee S_j$, so that S_i is equivalent to $S'_k \wedge X$ and S_j is equivalent to $S'_k \wedge \neg X$. Then let B_k be any subset of $[\![S'_k]\!]_{\mathcal{M}}$ of size $b_i + a_{i,1}\mathsf{u}_1 + \cdots + a_{i,m}\mathsf{u}_m$, such that the complement $[\![S'_k]\!]_{\mathcal{M}} - B_k$ has size $b_j + a_{j,1}\mathsf{u}_1 + \cdots + a_{j,m}\mathsf{u}_m$. This is always possible since $|S'_k|$ is simply the sum of these two numbers. Finally let $A = \bigcup_{k \leq 2^n} B_k$. Once again absorbing X into the language and defining $\mathcal{M}'$ to be just like $\mathcal{M}$ but with $X^{\mathcal{M}'} = A$, the tuple $[|S_1|_{\mathcal{M}'}, \dots, |S_{2^{n+1}}|_{\mathcal{M}'}]$ gives a solution to (4.3) with the same choices $\mathsf{u}_1, \dots, \mathsf{u}_m$. Hence, $\mathcal{M}' \vDash \varphi(O_1, \dots, O_n, X)$, from which it easily follows that $\mathcal{M}, s^X_A \vDash \varphi(O_1, \dots, O_n, X)$ and finally $\mathcal{M} \vDash \exists X.\varphi(O_1, \dots, O_n, X)$. ⊣

The foregoing thus establishes:

Theorem 4.7. *The numerical relations definable in* $\mathsf{MSO}^{\phi}(\#)$ *are the semi-linear sets. In other words,* $\mathsf{MSO}^{\phi}(\#)$ *expresses the same numerical relations as Presburger Arithmetic.*

Remark 4.8. In $\mathcal{L}^2_{\#}$ we allow arbitrary second-order quantification. However, we saw in Lemma 4.4 that we only needed an initial block of *existential* second-order quantifiers to encode any (semi-)linear set. The fact that every sentence in $\mathsf{MSO}^{\phi}(\#)$ defines a semi-linear set demonstrates a collapse of $\mathsf{MSO}^{\phi}(\#)$ into its purely existential fragment.

As in the first-order case, numerous undefinable results again follow. For example:

Corollary 4.9. *The expression 'many' is still not definable in* $\mathsf{MSO}^{\phi}(\#)$.

Proof. Adopting the notation from the proof of Fact 3.7, the constraint on state-descriptions, $\frac{r}{r+q} > \frac{p+r}{p+q+r+s}$, is not semi-linear. ⊣

Indeed, the theory of definability for Presburger Arithmetic carries over exactly to $\mathsf{MSO}^{\phi}(\#)$, thanks to Theorem 4.7. Moreover, since there is an algorithmic means of putting a formula of $\mathsf{MSO}^{\phi}(\#)$ into normal form and finding a suitable semi-linear form, decidability follows from the decidability of Presburger Arithmetic.

Corollary 4.10. $\mathsf{MSO}^{\phi}(\#)$ *is decidable.*

4.2. Infinitary case. Allowing second-order quantification does increase the expressive power of our initial system $\mathsf{MFO}^{\phi}(\#)$. While the latter essentially amounts to a proper fragment of Presburger Arithmetic, $\mathsf{MSO}^{\phi}(\#)$ gave us precisely Presburger Arithmetic. How does this look for the system $\mathsf{MSO}(\#)$ over models of arbitrary cardinality? One immediate difference is that, in contrast to $\mathsf{MFO}(\#)$ (Fact 3.8), the *successor* function on cardinal numbers can now be easily expressed:

$$\forall X.\big(\#_y P(y) \succ \#_y X(y) \rightarrow \#_y Q(y) \succsim \#_y X(y)\big).$$

This formula states that there is no cardinality strictly in between that of P and that of Q. How much more cardinal arithmetic does $\mathsf{MSO}(\#)$ encode?

As in the case of $\mathsf{MSO}^{\phi}(\#)$, we can calibrate this by appeal to additive first-order (now cardinal) arithmetic. Consider the elementary theory of the structure $\langle C_{\aleph_\omega}; +\rangle$, addition over the cardinals numbers less than $\aleph_\omega$. This is the theory of cardinal numbers in a first-order language with one binary function symbol, namely addition. We show in Appendix C that this theory admits quantifier elimination provided we augment the language with constants for the (definable in $\mathsf{MSO}(\#)$) functions and relations:

https://doi.org/10.1017/bsl.2023.30 Published online by Cambridge University Press

- $\{0\}$ and $\{\aleph_0\}$,
- s — the successor function,
- $>$ — the 'greater than' relation,
- $\equiv_k$ — equivalence modulo k for each $k > 1$.

Furthermore, we can derive a normal form result for this language:

Proposition 4.11. *Every first-order sentence is equivalent over the structure* $\langle C_{\aleph_\omega}; +\rangle$ *to a disjunction of conjunctions* $\delta \wedge \iota \wedge \phi$ *each specifying*:

- which (ordinary first-order) variables in the disjunct are finite or infinite (δ),
- a description of a linear set for the finite variables (ϕ),
- a description of a set of infinite cardinals using 0, s, and $>$ over $\aleph$-number indices, for the infinite variables (ι).

This can be understood as a kind of separation result. The finitary part, Presburger Arithmetic, simply describes ordinary addition. As for the infinitary part, observe that there is an isomorphism from $\langle \mathbb{N}; 0, s, >\rangle$ onto $\langle \{\aleph_k\}_{k\in\mathbb{N}}; \aleph_0, s, >\rangle$, sending k to $\aleph_k$. In other words, the additive structure of cardinals less than $\aleph_\omega$ amounts to a 'product' of $\langle \mathbb{N}; +\rangle$ and $\langle \mathbb{N}; >\rangle$.[3]

Our aim is to show that MSO(#) possesses the same normal forms as in Proposition 4.11. To see that any statement of the form $\delta \wedge \iota \wedge \phi$ can be expressed, note that δ merely requires distinguishing finite and infinite sets (recall Equation (2.1)), while definability of any linear set (specified by ϕ) was shown already in Lemma 4.4. Meanwhile, ι is a conjunction of formulas of types $\mathsf{v} = s^k(\mathsf{u})$, $\mathsf{v} > s^k(\mathsf{u})$, $\mathsf{v} = \aleph_k$, and $\mathsf{v} > \aleph_k$. We noted above that successor is expressible, and, for instance, we can assert that P has cardinality $\aleph_0$ simply by stating that P is infinite and there is no infinite set with smaller cardinality. Thus, any such statement is expressible.

To show that this exhausts what is definable in MSO(#), given Lemma 4.5, it remains to observe that the $\mathcal{L}^2_\#$-definable sets are closed under 'projection' by existentially quantifying one of the variables. Thus, suppose we have an $\mathcal{L}^2_\#$ formula $\varphi(\mathbf{O}, Y)$ with Y a predicate variable and $\mathbf{O} = O_1, \dots, O_n$ all either predicate variables or letters. We will assume $\varphi(\mathbf{O}, Y)$ has the form $\delta(\mathbf{O}, Y) \wedge \iota(\mathbf{O}, Y) \wedge \phi(\mathbf{O}, Y)$, analogously to the additive language: $\delta(\mathbf{O}, Y)$ describes which state descriptions over variables $O_1, \dots, O_n, Y$ are (in)finite, $\iota(\mathbf{O}, Y)$ characterizes the infinite state descriptions, while $\phi(\mathbf{O}, Y)$ describes a linear set. We need to analyze $\exists Y. \big(\delta(\mathbf{O}, Y) \wedge \iota(\mathbf{O}, Y) \wedge \phi(\mathbf{O}, Y)\big)$.

We can replace $\delta(\mathbf{O}, Y)$ with a formula $\delta'(\mathbf{O})$ specifying that a state description S over $O_1, \dots, O_n$ is finite iff $S \wedge Y$ and $S \wedge \neg Y$ were both finite according to $\delta(\mathbf{O}, Y)$. List the finite state descriptions according to $\delta(\mathbf{O}, Y)$

[3] A version of this result can be traced back at least to [94]. A general notion of product that subsumes this case is that of Feferman and Vaught [38].

as $S_1, \dots, S_k$. The subformula $\phi(\mathbf{O}, Y)$ defines a linear set over the possible (finite) cardinalities:

$$\begin{pmatrix} |S_1| \\ \vdots \\ |S_k| \end{pmatrix} = \begin{pmatrix} b_1 + a_{1,1}\mathsf{u}_1 + & \cdots & +a_{1,m}\mathsf{u}_m \\ \vdots & & \vdots \\ b_k + a_{k,1}\mathsf{u}_1 + & \cdots & +a_{k,m}\mathsf{u}_m \end{pmatrix}. \tag{4.5}$$

Suppose $S_i = S \wedge Y$ is finite but $S \wedge \neg Y$ is infinite. Then the constraint in (4.5) on $|S_i|$ is no constraint at all: since $|S|$ must be infinite, carving out a finite portion $S \wedge Y$ of any size will always be possible. So in this case we can simply drop the equation for S_i. Otherwise, if $S_i = S \wedge Y$ and $S_j = S \wedge \neg Y$ are both finite, then we can repeat the argument from Section 4.1, again combining these two equations into a single equation for $|S|$. The result is a set of equations in (cardinalities of) state descriptions over $\mathbf{O}$, all asserted finite in $\delta'(\mathbf{O})$.

The subformula $\iota(\mathbf{O}, Y)$ represents constraints of the form $\mathsf{v} = \mathsf{w}$ and $\mathsf{v} > \mathsf{w}$, where v and w are either 'infinite' state descriptions over $\mathbf{O}$, Y, k-fold successors of such state descriptions, or aleph-numbers. In view of the isomorphism between $\langle \mathbb{N}; 0, s, > \rangle$ and $\langle \{\aleph_k\}_{k \in \mathbb{N}}; \aleph_0, s, > \rangle$, we can construe these conjuncts as describing relations on natural numbers. As these relations are a special case of linear sets and can thus be encoded as in (4.5), we once again run the argument to 'merge' the equations for $S \wedge Y$ and $S \wedge \neg Y$ into a single equation for S, provided all of these are infinite. (If only one of $S \wedge Y$ and $S \wedge \neg Y$ is infinite, then that equation remains as before since S will have the same cardinality.) The resulting formula will in general involve addition. But as discussed further in Appendix C, since all variables are infinite we can eliminate all explicit sums, using equivalences such as $\mathsf{t} = \mathsf{v} + \mathsf{w} \Leftrightarrow (\mathsf{t} = \mathsf{v} \wedge \mathsf{v} \geq \mathsf{w}) \vee (\mathsf{t} = \mathsf{w} \wedge \mathsf{w} \geq \mathsf{v})$.

THEOREM 4.12. *The definable relations on cardinal numbers in* MSO(#) *are exactly the same as those definable in additive first-order logic.*

In effect, we have shown how to reduce a sentence $\varphi(\mathbf{P})$ in $\mathcal{L}^2_{\#}$ to an additive first-order formula $\alpha(\mathbf{x})$, with a variable x_i in $\mathbf{x}$ corresponding to each state-description over $\mathbf{P}$. Moreover, $\varphi(\mathbf{P})$ is satisfiable if and only if $\exists \mathbf{x}.\alpha(\mathbf{x})$ is true in $\langle C_{\aleph_\omega}; + \rangle$. Thus, from decidability of the elementary theory of $\langle C_{\aleph_\omega}; + \rangle$ (see Theorem C.3 in Appendix C) we obtain:

COROLLARY 4.13. MSO(#) *is decidable.*

§5. Counting sequences. We have so far considered a base monadic system, $\mathsf{MFO}^{\phi}(\#)$, and a second-order extension, $\mathsf{MSO}^{\phi}(\#)$, both of which are essentially restricted to reasoning about *sums* of numbers. The latter theme carries over to the setting of infinite models, with MFO(#) and MSO(#). These previous systems involve unary variable binding operators,

https://doi.org/10.1017/bsl.2023.30 Published online by Cambridge University Press

which count *sets* of objects. But it is also very natural from a logical point of view to count *sequences* of objects. Indeed, polyadic quantifiers are ubiquitous across natural language; cf. Section 9. We now consider such an extension, essentially moving from sets to products of sets. We would like to understand what additional arithmetical capacity this affords.

Let $\mathcal{L}^1_\sharp$ be the first-order monadic language with polyadic counting terms $\sharp_{\mathbf{x}}\varphi$, where $\mathbf{x} = x_1, \dots, x_k$ is a sequence of variables, which may appear in φ. Then:

$$\mathcal{M}, s \vDash \sharp_{\mathbf{x}}\varphi \succsim \sharp_{\mathbf{y}}\psi \text{ iff } |\{\mathbf{d} \in D^n : \mathcal{M}, s^{\mathbf{x}}_{\mathbf{d}} \vDash \varphi\}| \geq |\{\mathbf{d} \in D^m : \mathcal{M}, s^{\mathbf{y}}_{\mathbf{d}} \vDash \psi\}|.$$

Over finite models let us call the resulting system $\mathsf{MFO}^\phi(\sharp)$, and $\mathsf{MFO}(\sharp)$ for the general case.

It is known that polyadic counting over full first-order logic is more expressive than unary counting (i.e., our $\mathsf{FO}^\phi(\#)$; see, e.g., [98, Example 4.13]). In our monadic fragment this is particularly dramatic, as shown by the following example.

Example 5.1. Consider the earlier ‘Many Qs are P’, defined in (3.10) and repeated here:

$$\#_x\big(P(x) \wedge Q(x)\big) \times \#_x\top \succ \#_x P(x) \times \#_x Q(x).$$

We can express this as follows:

$$\sharp_{x,y}\big(P(x) \wedge Q(x)\big) \succ \sharp_{x,y}\big(P(x) \wedge Q(y)\big).$$

In a finite model, the term $\sharp_{x,y}\big(P(x) \wedge Q(x)\big)$ gives us the product of the model’s total cardinality and the region in which P and Q both hold, while the term on the right gives us the product of cardinalities for P and Q.

Evidently $\mathsf{MFO}^\phi(\sharp)$ incorporates some reasoning about *multiplication*. Another example:

Example 5.2. We can encode Pythagorean triples of cardinalities for state-descriptions S_1, S_2, S_3, i.e., the statement that $|S_1|^2 + |S_2|^2 = |S_3|^2$:

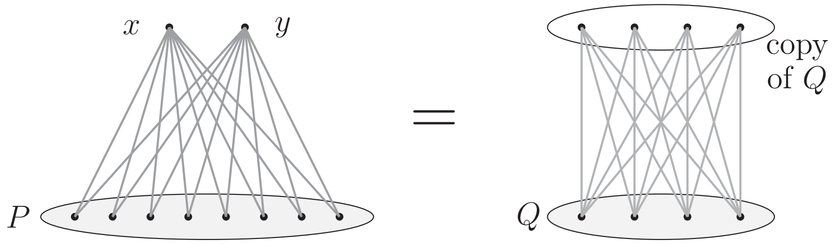
$$\sharp_{x,y}\big((S_1(x) \wedge S_1(y)) \vee (S_2(x) \wedge S_2(y))\big) \approx \sharp_{x,y}(S_3(x) \wedge S_3(y)).$$

The multiplication again comes from taking products, while the addition in this example arises from disjunction, just as in our initial system $\mathsf{MFO}^\phi(\#)$.

The next examples involves a different combination of multiplication and addition:

Example 5.3. This sentence expresses the constraint that $|P| \times 2 = |Q|^3 + 2$.

https://doi.org/10.1017/bsl.2023.30 Published online by Cambridge University Press



$$\exists x, y.\big(x \neq y \wedge \sharp_{u,v}(P(u) \wedge (v = x \vee v = y)) \approx \sharp_{u,v}(Q(u) \wedge Q(v))\big)$$

FIGURE 2. A visualization of the formula expressing that 2 times the number of P points is exactly the number of Q points squared, i.e., $|P| + |P| = |Q|^2$. The formula asserts that the number of lines on the left is equal to the number of lines on the right. This is a simplified version of Example 5.3 and of the more general construction in Lemma 5.4. In the case pictured, $|P| = 8$ and $|Q| = 4$.

$$\exists x, y.\Big(\neg(Q(x) \vee Q(y)) \wedge x \neq y \wedge \sharp_{u,v}\big(P(u) \wedge (v = x \vee v = y)\big) \approx \\ \sharp_{z,u,v}\big((Q(z) \wedge Q(u) \wedge Q(v)) \vee z = u = v = x \vee z = u = v = y\big)\Big).$$

Note the use of variables x, y for both $\sharp$ terms. In the first term, $\sharp_{u,v}\big(P(u) \wedge (v = x \vee v = y)\big)$, we simply want to multiply the cardinality of P by 2—the fact that Q holds of neither x nor y does not matter here. In the second $\sharp$ term we consider all triples of points satisfying Q, i.e., $|Q| \times |Q| \times |Q|$-many points, and we add two points x, y—here it is important that Q holds of neither, since this guarantees that we indeed add 2 to the product $|Q|^3$ in the second $\sharp$ term.

For a visualization, see Figure 2. All of these examples certainly go beyond what can be expressed in $\mathsf{MFO}^{\phi}(\#)$, and even $\mathsf{MSO}^{\phi}(\#)$. What is the full scope of $\mathsf{MFO}^{\phi}(\sharp)$?

5.1. Diophantine inequalities. To start our discussion, consider any polynomial inequality

$$m_1(\mathbf{v}) + \cdots + m_k(\mathbf{v}) \geq m'_1(\mathbf{v}) + \cdots + m'_j(\mathbf{v}), \tag{5.1}$$

where $m_1, \dots, m_k, m'_1, \dots, m'_j$ are all monomials in variables $\mathbf{v} = \mathsf{v}_1, \dots, \mathsf{v}_n$. Each monomial $m(\mathbf{v})$ is of the form $a\mathsf{v}_1^{e_1} \dots \mathsf{v}_n^{e_n}$, with $a, e_1, \dots, e_n$ all natural numbers and $a > 0$. We would like to show that sentences in $\mathsf{MFO}^{\phi}(\sharp)$ express all the Diophantine inequalities of type (5.1). The first result generalizes the observations above:

LEMMA 5.4. *Every Diophantine inequality can be expressed in* $\mathsf{MFO}^{\phi}(\sharp)$.

Proof. Let a^* be the sum of all the coefficients of $m_1, \dots, m_k, m'_1, \dots, m'_j$, and let e^* be the maximum over all the sums $\sum_{i \leq n} e_i$. Then our sentence will take the form:

$$\exists \mathbf{z}.\Big(\mathsf{diff}(\mathbf{z}) \wedge \sharp_{\mathbf{x}} \bigvee_{1 \leq i \leq k} \alpha_i \succsim \sharp_{\mathbf{x}} \bigvee_{1 \leq i \leq j} \beta_i\Big), \tag{5.2}$$

with $\mathbf{z} = z_1, \dots, z_{a^*}$ and $\mathbf{x} = x_0, x_1, \dots, x_{e^*}$. We need to ensure that each tuple of values for $\mathbf{x}$ satisfies at most one of the α_i formulas (and the same for the β_i formulas), and that each α_i contributes exactly $a \times |S_1|^{e_1} \times \cdots \times |S_n|^{e_n}$ to the overall sum, when $m_i(\mathbf{v}) = a\mathsf{v}_1^{e_1} \dots \mathsf{v}_n^{e_n}$. To that end let α_i be the conjunction of the following formulas:

(i) $S_1(x_1) \wedge \cdots \wedge S_1(x_{e_1})$, with a similar conjunct for each of $S_2, \dots, S_n$, predicating $e_2, \dots, e_n$ variables, respectively.
(ii) For any remaining x up to x_{e^*}, include a conjunct $x = x_1$.
(iii) As a final conjunct in the formula $(x_0 = z_{i_1} \vee \cdots \vee x_0 = z_{i_a})$, for a variables $z_{i_1}, \dots, z_{i_a}$ from among $z_1, \dots, z_{a^*}$, guaranteed unique to this disjunct α_i.

The last conjunct 5.1 ensures that each α_i contributes a multiplied by the number of tuples satisfying $|S_1|^{e_1} \times \cdots \times |S_n|^{e_n}$, since each such tuple appears with exactly a (unique) values of x_0. Defining the β_is analogously produces a formula whose models capture precisely the same solutions as (5.1), provided the sum of these numbers is at least a^*. There may of course be solutions that together add up to less than a^*, in which case (5.2) will fail; however, there will be at most finitely many. For each such solution $b_1, \dots, b_n$ we can simply disjoin (5.2) with the statement $|S_1| = b_1 \wedge \cdots \wedge |S_n| = b_n$, the latter being easily definable (even in $\mathsf{MFO}^{\phi}(\#)$). ⊣

Conjunctions of inequalities in (5.1) give us the well-studied class of Diophantine equations. The Matiyasevich–Robinson–Davis–Putnam (MRDP) theorem shows that there can be no decision procedure to determine whether a given Diophantine equation has a solution. So:

Proposition 5.5. *The satisfiability problem for* $\mathsf{MFO}^{\phi}(\sharp)$ *is undecidable.*

Moreover, while it is possible to enumerate the satisfiable formulas in an effective way, the valid sentences of $\mathsf{MFO}^{\phi}(\sharp)$—those whose negations define equations with no solutions—are not computably enumerable. Therefore:

Proposition 5.6. $\mathsf{MFO}^{\phi}(\sharp)$ *is not computably axiomatizable.*

5.2. Normal forms. In the direction of a normal form for $\mathsf{MFO}^{\phi}(\sharp)$, a first observation is that a version of the invariance principle (INV) from Section 3.2 holds in the present setting as well:

$$\exists \mathbf{x}\big(\alpha^{\mathbf{y}}(\mathbf{x}) \wedge \varphi(\mathbf{x})\big) \rightarrow \sharp_{\mathbf{x}}\big(\alpha^{\mathbf{y}}(\mathbf{x}) \wedge \varphi(\mathbf{x})\big) \approx \sharp_{\mathbf{x}}\alpha^{\mathbf{y}}(\mathbf{x}),$$

where now $\alpha^{\mathbf{y}}(\mathbf{x})$ is a complete description of the list $\mathbf{x}$ of variables (relative to $\mathbf{y}$). By an analogous argument we can then show that every formula in $\mathcal{L}^1_\sharp$ is equivalent to one with no embedded $\sharp$ comparisons. More generally, as in Theorem 3.2, we have:

THEOREM 5.7. *The definable sets of* $\mathsf{MFO}^\phi(\sharp)$ *are exactly those definable by quantifier-free formulas in first-order arithmetic* (*with addition and multiplication*).

PROOF SKETCH. We want to show more generally that every formula of $\mathcal{L}^1_\sharp$ in free variables $\mathbf{y}$ is equivalent to a disjunction

$$\bigvee \big(\alpha(\mathbf{y}) \wedge (\sigma)_{\alpha(\mathbf{y})}\big), \tag{5.3}$$

where $\alpha(\mathbf{y})$ ranges over possible descriptions of $\mathbf{y}$, and σ is a conjunction of (strict and weak) Diophantine inequalities (5.1). Specifically, each disjunct is such that, for all s:

$$\mathcal{M}, s \vDash \alpha(\mathbf{y}) \wedge (\sigma)_{\alpha(\mathbf{y})} \Rightarrow \mathcal{M} \text{ satisfies the description } \sigma. \tag{5.4}$$

As in the proof of Theorem 3.2, we show that every formula is equivalent to one of the form (5.3) satisfying (5.4) by induction on the quantifier depth of formulas. In the base case, with no quantifiers and just a single $\sharp$-comparison, our normal forms will be disjunctions of conjunctions $\alpha(\mathbf{y}) \wedge (\sigma)_{\alpha(\mathbf{y})}$ where $(\sigma)_{\alpha(\mathbf{y})}$ takes the form:

$$\bigwedge \Big(\sharp_{\mathbf{x}} \bigvee \alpha^{\mathbf{y}}_i(\mathbf{x}) \succsim \sharp_{\mathbf{x}} \bigvee \alpha^{\mathbf{y}}_j(\mathbf{x})\Big).$$

It is straightforward to check that each such disjunct corresponds to a set of Diophantine inequality constraints satisfying (5.4). The inductive case is just as in the proof of Theorem 3.2, reducing to the claim that normal forms in (5.3) are closed under existential quantification. ⊣

5.3. Second-order extensions. The jump from $\mathsf{MFO}^\phi(\#)$ to $\mathsf{MSO}^\phi(\#)$, incorporating second-order quantification, was relatively minor. Arithmetically speaking, it simply allowed 'filling out' the class of linear inequalities we were able to encode. Given that $\mathsf{MFO}^\phi(\sharp)$ already defines quantifier-free first-order arithmetic (Theorem 5.7), we might expect a more dramatic increase in expressive power when moving to the second-order version $\mathsf{MSO}^\phi(\sharp)$.

The proof of Lemma 5.4 was given for state-descriptions corresponding to the variables in (5.1). Suppose, however, that we replace some of these state-descriptions with second-order variables. It then becomes clear that we can consider arbitrary quantificational statements involving Diophantine inequalities. Here is an illustration, revisiting Example 5.3:

https://doi.org/10.1017/bsl.2023.30 Published online by Cambridge University Press

Example 5.8. We can encode the purely arithmetical statement $\forall n \exists m.(m \times 2 = n^3 + 2)$:

$$\forall X.\exists Y.\exists x, y.\Big(\neg(X(x) \vee X(y)) \wedge x \neq y \wedge \sharp_{u,v}\big(Y(u) \wedge (v = x \vee v = y)\big) \approx$$
$$\sharp_{z,u,v}\big((X(z) \wedge X(u) \wedge X(v)) \vee z = u = v = x \vee z = u = v = y\big)\Big).$$

This is the same expression as in Example 5.3, except that we quantify out the predicates P and Q. Given the quantification, this statement is of course unsatisfiable. Were the initial universal quantifier instead existential, this would be a valid statement.

More systematically, we can derive a normal form result in this language, such that every sentence is equivalent to a Boolean combination of arithmetical statements (involving arbitrary quantification over natural numbers), from which we obtain:

Theorem 5.9. $\mathsf{MSO}^{\phi}(\sharp)$ *is equivalent to full first-order arithmetic.*

It of course follows that the set of validities in $\mathsf{MSO}^{\phi}(\sharp)$ is non-arithmetical.

Remark 5.10. Note that the use of second-order quantification in $\mathsf{MSO}^{\phi}(\sharp)$ is quite different from that in $\mathsf{MSO}^{\phi}(\#)$. The power of addition afforded by the latter, relative to the base system $\mathsf{MFO}^{\phi}(\#)$, is already guaranteed by polyadic counting. As a typical example, $\mathsf{MSO}^{\phi}(\#)$ went beyond $\mathsf{MFO}^{\phi}(\#)$ by defining relations such as $|P| = |Q| + |Q|$; recall Fact 3.6 and Example 4.1. In $\mathsf{MFO}^{\phi}(\sharp)$ such an example is handled directly by multiplication, encoding $|P| = 2 \times |Q|$.

In a sense, quantification over predicates collapses in $\mathsf{MSO}^{\phi}(\#)$ (recall Remark 4.8), reflecting quantifier elimination in additive arithmetic. As Theorem 5.9 shows, this does not happen when counting sequences, reflecting failure of quantifier elimination in full first-order arithmetic.

Remark 5.11. Natural fragments of $\mathsf{MSO}^{\phi}(\sharp)$ arise when limiting second-order quantification in principled ways. For instance, by the MRDP theorem the purely existential fragment of $\mathsf{MSO}^{\phi}(\sharp)$ encodes precisely the computably enumerable sets. Closing this fragment under Booleans leads to the class Σ_1^* of definable sets (also known as n-c.e. sets), introduced by Putnam [106] in the context of formal learning theory.

5.4. Infinitary counting. For both systems, $\mathsf{MFO}^{\phi}(\sharp)$ and $\mathsf{MSO}^{\phi}(\sharp)$, we can consider their more general versions, $\mathsf{MFO}(\sharp)$ and $\mathsf{MSO}(\sharp)$, where we allow infinite models. It is shown in Appendix C that, similar to the purely additive case, cardinal arithmetic (say, up to $\aleph_\omega$) with addition and multiplication separates cleanly into the finitary and infinitary components, with the infinitary component effectively reducing to the first-order theory of $\langle \mathbb{N}; > \rangle$. A similar (but in fact simpler) analysis to that given for $\mathsf{MSO}(\#)$

https://doi.org/10.1017/bsl.2023.30 Published online by Cambridge University Press

shows that $\mathsf{MFO}(\sharp)$ defines exactly the quantifier-free definable relations on cardinals less than $\aleph_\omega$, while $\mathsf{MSO}(\sharp)$ coincides with full first-order cardinal arithmetic (with addition and multiplication).

§6. An alternative route: explicit arithmetical operators. The sequence of systems so far studied was motivated primarily by natural operations in logic, viz. second-order quantification and polyadicity. We were then able to calibrate the arithmetical content of these operations over our base monadic system $\mathsf{MFO}(\#)$. Another approach to extending $\mathsf{MFO}(\#)$ in the spirit of logic and counting would rather strengthen the counting component in natural ways, in particular, by allowing complex terms built directly out of arithmetical operations. Instead of comparisons involving terms like $\#_x\varphi$ we might allow comparing, for instance, sums of terms $\#_x\varphi + \#_y\psi$, and in general allow inequalities $\mathbf{t}_1 \gtrsim \mathbf{t}_2$ between complex terms. We can then study the consequences of different choices of complex term building operators. Most salient are of course addition and multiplication, and for these it turns out that, speaking abstractly, we would have arrived at the same systems.

6.1. Addition. Let $\mathsf{MFO}(\#,+)$ be the system that results by allowing arbitrary finite sums of basic # terms. In other words we allow terms of the form $\#_{x_1}\varphi_1 + \cdots + \#_{x_n}\varphi_n$. We already know that $\mathsf{MSO}(\#)$ can express all such inequalities. Conversely, the normal form result for $\mathsf{MSO}(\#)$ by means of linear inequalities shows that this system and $\mathsf{MFO}(\#,+)$ are in fact equally expressive when it comes to defining relations on cardinal numbers.

Note also that the numerical reasoning involved in Fourier–Motzkin (Remark 3.11) can be transcribed into this language without any further ado. For instance, we can encode the crucial step (iii) by a simple scheme:

$$\big(|S_1| \gtrsim |S_2| + |S_3| \wedge |S_4| + |S_3| \gtrsim |S_2|\big) \rightarrow |S_1| + |S_4| \gtrsim |S_2| + |S_2|.$$

Thus, similar to analogous work on (rational) linear programming [37], we could codify the steps of the algorithm into axioms of a formal system.

6.2. Multiplication. From an arithmetical point of view, it is natural to allow arbitrary finite *products* of basic #-terms as well. How would such a system relate to our systems for counting sequences, such as $\mathsf{MFO}^\phi(\sharp)$ or $\mathsf{MSO}^\phi(\sharp)$? Needless to say, if we had explicit multiplication and addition we would be able to encode all arithmetical relations, which would give the same expressive power as $\mathsf{MSO}^\phi(\sharp)$ (thanks to Theorem 5.9).

Similar to the case of $\mathsf{MSO}^\phi(\#)$, even without explicit addition we can simulate addition if we avail ourselves of second-order quantification. Indeed, let $\mathsf{MSO}^\phi(\#,\times)$ be the second-order monadic fragment with products of #-terms (in fact, binary products suffice). Echoing observations

https://doi.org/10.1017/bsl.2023.30 Published online by Cambridge University Press

dating back to Skølem [119], we can thereby encode arbitrary Diophantine inequalities. Indeed, consider any such

$$m_1(\mathbf{v}) + \cdots + m_k(\mathbf{v}) \geq m'_1(\mathbf{v}) + \cdots + m'_j(\mathbf{v}), \tag{6.1}$$

over variables $\mathbf{v}$ corresponding to state-descriptions over $\mathbf{P}$. To express (6.1) in $\mathsf{MSO}^{\phi}(\#, \times)$ we introduce $k + j$ predicate variables $X_1, \ldots, X_{k+j}$ and consider the statement that all X_i are disjoint but that $\#(X_1 \vee \cdots \vee X_k) \succsim \#(X_{k+1} \vee \cdots \vee X_{k+j})$. Each monomial $m_i(\mathbf{v})$ can clearly be expressed as a product of #-terms (possibly using first-order quantification) in the original variables $\mathbf{P}$, so we set each of these equal to the corresponding term $\#X_i$. To define the same set of solutions as (6.1) (in the sense of Definition 4.3, so that the formula includes no free predicate variables), we existentially quantify all the variables $X_1, \ldots, X_{k+j}$.

Given second-order quantification we can repeat the same analysis as with $\mathsf{MSO}^{\phi}(\sharp)$ to obtain not just the Diophantine sets, but again all arithmetical sets: the state-descriptions over $\mathbf{P}$ can themselves be quantified arbitrarily. Hence, this system $\mathsf{MSO}^{\phi}(\#, \times)$ is precisely equivalent to $\mathsf{MSO}^{\phi}(\sharp)$. Analogously to $\mathsf{MSO}^{\phi}(\#, \times)$, restricting to vectors of length 2 suffices. We leave more fine-grained analysis of fragments of these systems (e.g., the purely first-order fragment) for future explorations.

6.3. Other arithmetical operations. Aside from addition and multiplication, we could naturally consider a host of other common arithmetical functions and relations. As an example, unlike addition and multiplication, *exponentiation* does not trivialize in the infinitary setting. Indeed, whereas in the finitary setting exponentiation is definable from addition and multiplication (e.g., by Gödel's famous β function), the Generalized Continuum Hypothesis can already be stated succinctly in $\mathsf{MSO}(\#)$ with exponentiation:

$$\forall X, Y, Z. \big(|X| \approx \mathbf{2}^{|Y|} \wedge |Y| \succsim \aleph_0 \wedge |Z| \succ |Y| \rightarrow |Z| \succsim |X| \big),$$

where $\mathbf{2}$ abbreviates a set with cardinality two. It could be illuminating to study the properties of such a system across different models of set theory.

Another natural example is the relation of *divisibility*, which also arises in the study of natural language quantifiers and automata hierarchies (see Section 9.5 and in particular Proposition 9.17(b)). A non-trivial observation in the finite case (due to Julia Robinson) is that first-order logic with divisibility and the successor function already provide the full suite of arithmetically definable relations [112]. At the same time, the *existential fragment* of Presburger Arithmetic with divisibility is known to be decidable [84], which leaves open the possibility that some of our systems may too remain relatively well-behaved (recall Remark 4.8). We defer these and further explicit arithmetical excursions for another occasion.

https://doi.org/10.1017/bsl.2023.30 Published online by Cambridge University Press

Table 2. A hierarchy of monadic counting logics, covered in Sections 2–6. Where $\mathfrak{M}$ is a structure, $\mathcal{D}(\mathfrak{M})$ are the first-order definable relations over the domain of $\mathfrak{M}$, while $\mathcal{D}_{\mathsf{qf}}(\mathfrak{M})$ are the relations definable by quantifier-free formulas

Language	Logical system	Arithmetical content
$\mathcal{L}^2_\sharp$	$\mathsf{MSO}^\phi(\sharp)$	$\mathcal{D}(\langle\mathbb{N}; +, \times\rangle)$, i.e., arithmetical sets
	$\mathsf{MSO}(\sharp)$	$\mathcal{D}(\langle C_\kappa; +, \times\rangle)$
$\mathcal{L}^1_\sharp$	$\mathsf{MFO}^\phi(\sharp)$	$\mathcal{D}_{\mathsf{qf}}(\langle\mathbb{N}; +, \times\rangle)$
	$\mathsf{MFO}(\sharp)$	$\mathcal{D}_{\mathsf{qf}}(\langle C_\kappa; +, \times\rangle)$
$\mathcal{L}^2_\#$	$\mathsf{MSO}^\phi(\#)$	$\mathcal{D}(\langle\mathbb{N}; +\rangle)$, i.e., semi-linear sets
	$\mathsf{MSO}(\#)$	$\mathcal{D}(\langle C_\kappa; +\rangle)$
$\mathcal{L}^1_\#$	$\mathsf{MFO}^\phi(\#)$	Inequalities between sums $\sum_i \mathsf{v}_i + k$
	$\mathsf{MFO}(\#)$	Same, interpreted over cardinals
$\mathcal{L}^0_\#$	$\mathsf{PL}^\phi(\#)$	Inequalities between sums $\sum_i \mathsf{v}_i$
	$\mathsf{PL}(\#)$	Same, interpreted over cardinals

6.4. Interim summary. Table 2 lists the logical systems we have studied so far. In this monadic setting, we assess each system's ability to reason about counting by analyzing the arithmetical content of its family of definable relations. All of these systems speak about unary predicates and their Boolean combinations, but we have been most interested in the abstract relations over *cardinal numbers* that sentences in these systems can define, essentially taking cardinalities of state-descriptions (or more generally, non-overlapping predicates) as numerical variables. We have seen that the landscape here is quite rich, naturally calibrated by familiar first-order arithmetical languages. With this grasp of the pure monadic fragment, we now move on to consider well-behaved fragments employing *relational* reasoning.

§7. Modal logic of binary relations.

We started with adding counting operators to the full language of first-order logic, and found a system $\mathsf{FO}(\#)$ with very high complexity. We then moved our base level to monadic fragments, which were decidable and allowed us to see combinations of logic and counting at work in more controlled settings. Even so, many simple intuitive examples of reasoning with numerical aspects go further than this, and involve binary relations.

Example 7.1. The well-known Pigeonhole Principle says that, if we put n objects into $k < n$ boxes, then at least one box must contain two or more objects. For all particular values of k, n, this principle can be expressed in monadic first-order logic using unary predicates for boxes (recall (2.6)). But for a generic formulation, we need to go to binary relations, which admit

https://doi.org/10.1017/bsl.2023.30 Published online by Cambridge University Press

of the following elegant statement. Consider any binary relation R whose domain has a larger cardinality than its range. Then at least one object must have two or more predecessors in the relation. In formal notation, $\#_x \exists y.Rxy \succ \#_x \exists y.Ryx$ implies $\exists x.\exists^{\geq 2} y.Ryx$.

In this light, it makes sense to study count versions of fragments of $\mathsf{FO}(\#)$ that allow for some reference to binary relations, though without running into the high complexity noted earlier with the full language $\mathcal{L}_{\#}$. To this end, we will explore some count versions of *modal languages* in some detail, starting with a simplest case, and returning to further extensions suggested by the Pigeonhole Principle later. For the basic notions and results of modal logic needed in this section, we will refer to the literature at appropriate places.

7.1. Language and semantics. The language of *propositional modal logic with counting*, $\mathcal{L}_{\#}^{\mathsf{ml}}$, has a syntax defined inductively as follows:

$$\varphi \quad := \quad p \mid \neg\varphi \mid \varphi \wedge \psi \mid \#\varphi \succsim \#\psi \ .$$

The *depth* of formulas is defined recursively as for our earlier logics, with standard clauses for atoms p and Booleans, while $\mathsf{d}(\#\varphi \succsim \#\psi) = \max(\mathsf{d}(\varphi), \mathsf{d}(\psi)) + 1$.

The semantics of this language uses standard modal relational models $\mathfrak{M} = (W, R, V)$. At points in these models, we define truth of formulas, and term values in a mutual recursion. For a point s we write $R_s = \{t : Rst\}$ for its R-successors. Here are the two key clauses:

- $[\![\#\varphi]\!]^{\mathfrak{M},s} = |R_s \cap [\![\varphi]\!]^{\mathfrak{M}}|$,
- $\mathfrak{M}, s \models \#\varphi \succsim \#\psi$ iff $[\![\#\varphi]\!]^{\mathfrak{M},s} \geq [\![\#\psi]\!]^{\mathfrak{M},s}$.

Given this, we define an *existential modality* $\Diamond\varphi$ as $\#\varphi \succ \#\bot$, and using negation we can then also define its universal dual $\Box\varphi$. There is also some definability for the Booleans, as we saw with $\mathsf{MFO}(\#)$, but we will let this rest here. Call the resulting system $\mathsf{ML}(\#)$. As before, we denote the logic interpreted over finite models by $\mathsf{ML}^{\phi}(\#)$.

Remark 7.2. As was the case with the variety of quantifiers in $\mathsf{MFO}(\#)$, there are also further natural counting modalities such as ‘in most successors’, ‘in almost all successors’, but we will not study their logic separately here.

As for expressive power, iterated counting in this simple modal language can produce non-trivial assertions. The reader might consider the formula $\#(\#\neg p \succ \#p) \succ \#(\#p \succsim \#\neg p)$, and determine what this says numerically, for instance, on finite trees. An example model is depicted in Figure 3. One can also enforce infinity of some sets of successors. E.g., the modal formula $\Diamond q \wedge \#(p \wedge \neg q) \approx \#(p \vee q)$ requires an infinity of successors satisfying p.

https://doi.org/10.1017/bsl.2023.30 Published online by Cambridge University Press

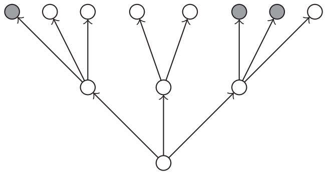


FIGURE 3. An example model in which $\#(\#\neg p \succ \#p) \succ \#(\#p \succsim \#\neg p)$ holds at the root point. The points where p holds are shaded. The number of successors with more non-p successors than p successors is greater than the number of successors with at least as many p successors as non-p successors.

7.2. Some basic model theory. Here are some invariance properties of our modal counting language that are useful for studying its expressive power.

A *generated submodel* of a modal model is a submodel that is closed under taking R-successors (see, e.g., [17]). Given the 'forward-looking' nature of the modal counting language along the order R, the following is a counterpart of the analogous invariance property for the basic modal language.

PROPOSITION 7.3. (a) *Formulas of $\mathcal{L}_{\#}^{\mathsf{ml}}$ are invariant for generated submodels.* (b) *Terms of $\mathcal{L}_{\#}^{\mathsf{ml}}$ have the same value in generated submodels.*

PROOF. A straightforward mutual induction on formulas and numerical terms. ⊣

The Finite Depth Property of modal logic also goes through, where 'finite depth' refers to the following cut-off versions of our models: $\mathfrak{M}|_n, s$ is the submodel of $\mathfrak{M}$ consisting of only those points that can be reached from s in at most n relational steps.

PROPOSITION 7.4. *For any model $\mathfrak{M}$ and $\mathcal{L}_{\#}^{\mathsf{ml}}$-formula φ, $\mathfrak{M}, s \models \varphi$ iff $\mathfrak{M}|_{\mathsf{d}(\varphi)}, s \models \varphi$.*

The following invariance property refers to the standard *tree unraveling* of arbitrary relational models yielding tree-like models in basic modal logic [17].

PROPOSITION 7.5. (a) *Formulas of $\mathcal{L}_{\#}^{\mathsf{ml}}$ are invariant for tree unraveling under the map taking finite branches to their end-points.* (b) *Terms of $\mathcal{L}_{\#}^{\mathsf{ml}}$ have the same value under tree unraveling at points related by this same map.*

PROOF. The proof is a straightforward induction on formulas and numerical terms, using the fact that the immediate successors of a branch in the tree are in one–one correspondence with the successors of the end-point in the original model. ⊣

Finally define *duplication* of a tree as making copies of all immediate successors of the root: each successor t splits into t_1, t_2 each heading a

https://doi.org/10.1017/bsl.2023.30 Published online by Cambridge University Press

disjoint copy of the original subtree at t. This construction can also be defined for models in general, and it can also be iterated going down the tree [13], but we will not use this generality here.

Proposition 7.6. *$\mathcal{L}_{\#}^{\mathsf{ml}}$-formulas at the root are invariant for tree duplication.*

Proof. The crucial cases are the numerical modal comparison statements $\#\varphi \succsim \#\psi$ of our language, and these are obviously closed under taking multiples. ⊣

These invariance properties put limits on expressive power. For instance, our counting logic $\mathsf{ML}(\#)$ does not contain the well-known system of *graded modal logic* that describes specific finite numbers of successors [41].

Corollary 7.7. *The graded modality 'in at most one successor' is not definable in* $\mathsf{ML}(\#)$, *as it is not invariant under tree duplication.*

In fact, $\mathsf{ML}(\#)$ and graded modal logic are incomparable in expressive power; see [30] for a more powerful system that subsumes both.

7.3. Bisimulation. Behind the above preservation facts lies a general notion of *bisimulation* for $\mathcal{L}_{\#}^{\mathsf{ml}}$. For convenience, we define this to be a standard modal bisimulation [17] satisfying a further requirement of cardinality comparison between sets satisfying a structural property matching modal definability.

Definition 7.8 (#-bisimulation). Let Z be a modal bisimulation between two points in two models $\mathfrak{M}$, $\mathfrak{N}$ satisfying the usual conditions of (a) atomic harmony for proposition letters at Z-connected points, and (b) the standard back-and-forth clauses for matching relational successors of Z-connected points.

Next, we define an auxiliary relation $\sim_Z$ between points in $\mathfrak{M}$ as follows: $x \sim_Z y$ iff for some $z \in \mathfrak{N}$: xZz and yZz. The relation $\sim_Z$ in the model $\mathfrak{N}$ is defined likewise. Now, Z is a *#-bisimulation* if the following comparative cardinality conditions hold.

(a) Whenever sZt and X, Y are $\sim_Z$-closed sets of successors of s with $X \succsim Y$ in our cardinality sense, then $Z[X] \cap R_t \succsim Z[Y] \cap R_t$.[4]

(b) The same requirement in the opposite direction.

See Figure 4. Note: in Clause (a), we mean that the sets X, Y are $\sim_Z$-closed with respect to successors of s, not necessarily in the whole model $\mathfrak{M}$, and likewise in Clause (b).

To understand what the map $Z[X] \cap R_t$ does, note that $R_t - Z[X] = Z[R_s - X]$, given the $\sim_Z$-closedness of X and the fact that Z is a modal bisimulation.

[4] Here, $Z[X]$ is the set $\{y \in \mathfrak{N} : xZy \text{ for some } x \in X.\}$.

https://doi.org/10.1017/bsl.2023.30 Published online by Cambridge University Press

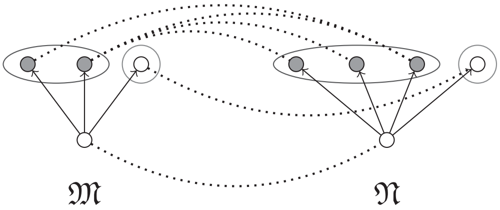


FIGURE 4. An ordinary modal bisimulation Z between $\mathfrak{M}$ and $\mathfrak{N}$ is depicted by the dotted lines. In both of these models the root point has four $\sim_Z$-closed sets of successors: the empty set, the whole set, and the two encircled sets. To be a #-bisimulation (Definition 7.8), the same ordering of these sets by cardinality must hold in each, as it does here.

PROPOSITION 7.9. *Formulas of $\mathcal{L}_{\#}^{\mathsf{ml}}$ are invariant for #-bisimulation.*

PROOF. The only non-routine part of the inductive argument is checking that #-bisimulations preserve truth values of atomic formulas $\#\varphi \succsim \#\psi$ both ways for points s, t with sZt.

To see this, first note that the set of all φ-successors of a point s in a model $\mathfrak{M}$ satisfies the closure condition for $\sim_Z$ (using the inductive assumption on bisimulation invariance for the formula φ), and the same is true for the set of ψ-successors. We apply the comparison clause for our #-bisimulation to these sets X, Y and get that $Z[X] \cap R_t \succsim Z[Y] \cap R_t$ in $\mathfrak{N}$.

Next, we show that $Z[X]$ is the set of successors of t satisfying φ. By definition, each point in $Z[X]$ is Z-connected to some point in X, and so it satisfies φ by the inductive hypothesis. Moreover, each point in $R_t - Z[X]$ was Z-connected to some point in $R_s - X$, and again by the inductive hypothesis, it then fails to satisfy φ. The same reasoning works for Y and ψ. It follows that $\#\varphi \succsim \#\psi$ is true at t in $\mathfrak{N}$.

Given the symmetry in the above comparative clause for a #-bisimulation, the argument also works in the opposite direction. ⊣

Bisimulation invariance can be used to show that certain notions are not definable.

EXAMPLE 7.10. Infinity of a set of successors is not definable in $\mathsf{ML}(\#)$. Consider two models: one with a root and one successor, the other with a root with infinitely many successors. All proposition letters are true at all points. Connecting the two roots while also connecting all successors across the models is easily seen to be a bisimulation in the above sense.

As in general modal logic, converse results require additional conditions. We formulate two versions, starting with a Hennessy–Milner result for ‘image-finite’ models where each point has only finitely many relational successors.

Proposition 7.11. *On points in two image-finite modal relational models, the relation E of $\mathcal{L}^{\mathsf{ml}}_{\#}$-equivalence is a #-bisimulation.*

Proof. By a standard argument from the modal literature, since $\mathcal{L}^{\mathsf{ml}}_{\#}$ contains the basic modal language, we have that E is an ordinary modal bisimulation.

Now for the set-comparison clause. Start with sEt. We first show that any $\sim_E$-closed set X of successors of s is definable among the successors of s. This follows by a well-known model-theoretic definability argument if we can show that this set is closed under $\mathsf{ML}(\#)$-equivalence in the finite set of successors of s. But the latter fact can be seen as follows. Suppose that in $\mathfrak{M}$, $x \in X$ is $\mathcal{L}^{\mathsf{ml}}_{\#}$-equivalent to x' in R_s. By the ordinary forth clause for a modal bisimulation, x is E-related to some u in R_t in $\mathfrak{N}$. But then x', too, is E-related to u, and by the assumed $\sim_E$-closure, x' must be in X.

Now assume that $|X| \geq |Y|$ is true in $\mathfrak{M}$ at s. Given the preceding observation, this shows in the truth of some formula $\#\alpha \succsim \#\beta$ at s where α defines X and β defines Y. Given the definition of E, this formula will also be true at t in $\mathfrak{N}$, and then it suffices to note, using the above definability in $\mathfrak{M}$ plus the inductive hypothesis, that the set of α-successors of t is just $E[X] \cap R_t$, and likewise for the β-successors. ⊣

Still, the common assumption of image-finiteness runs counter to the fact that $\mathsf{ML}(\#)$ can also compare infinite cardinalities among successors. To reach a perfect correspondence we can employ another device, passing to an *infinitary* modal language, allowing conjunctions and disjunctions of arbitrary sets of formulas. Call this language $\mathcal{L}^{\infty\mathsf{ml}}_{\#}$.

Theorem 7.12. *The following are equivalent*: (a) *There exists a* $\mathsf{ML}(\#)$-*bisimulation connecting* $\mathfrak{M}, s$ *to* $\mathfrak{N}, t$. (b) $\mathfrak{M}, s$ *and* $\mathfrak{N}, t$ *satisfy the same formulas of* $\mathcal{L}^{\infty\mathsf{ml}}_{\#}$.

Proof. The inductive proof of the invariance assertion from (a) to (b) is essentially as before. From (b) to (a), we can use the earlier reasoning for image-finite models almost literally, noting now that in specific models, we only have *sets of* successors, which are 'small' w.r.t. the class of formulas of $\mathcal{L}^{\infty\mathsf{ml}}_{\#}$, and then using the closure of the latter language under arbitrary set conjunctions and disjunctions. ⊣

Remark 7.13. The bisimulation analysis presented here puts the crucial count comparisons of #-languages in the back-and-forth clauses by brute force. A more refined notion of bisimulation might directly relate the *counting procedures* that underlie the comparative cardinality judgments in the two models. We leave this here as a further desideratum.

For a first study of standard modal themes like bisimulation and frame correspondence for $\mathsf{ML}(\#)$ and its extensions, see [42].

7.4. Normal forms for $\mathsf{ML}(\#)$**.** The modal counting language admits syntactic normal forms that combine standard normal forms for modal logic with the numerical equational normal forms that we found for $\mathsf{MFO}(\#)$. The idea is to start with the earlier state descriptions, and then describe inductively, for successors at increasing distance, which types occur with which multiplicity. In what follows, we fix a finite vocabulary of proposition letters.

Definition 7.14. A *0-type* is a complete conjunction of literals. An $(n+1)$*-type* is a conjunction of a 0-type and a complete set of inequalities describing a linear order on the count terms $\#T$ that describe all unions of n-types.

The inductive step in this definition makes sense because it is easy to show inductively that the set of n-types is finite for each n. To understand this definition, note that modal types record inductively which types of lower rank are present and absent on successors of given points. $\mathsf{ML}(\#)$ merely enriches this to more precise numerical information.

Fact 7.15. Each formula of $\mathsf{ML}(\#)$ with count depth k is equivalent to a disjunction of k-types.

Proof. For $k = 0$, this is the disjunctive normal form of propositional logic. For the case $k+1$, a formula of this depth is equivalent to a Boolean combination of proposition letters and formulas $\#\varphi \succsim \#\psi$ with φ, ψ of depth at most k. By the inductive hypothesis, these formulas are equivalent to disjunctions of k-types. So, the whole formula is equivalent to a disjunction of conjunctions of such statements, where negations of comparison atoms can be replaced by strict inequalities. Thus, a certain number of comparisons between regions is already given, and all we need to do is to replace this formula by the disjunction of all completions to fill in comparisons between all regions, which is possible by the linearity of $\succsim$. ⊣

Here are two points of comparison with the earlier normal forms for monadic counting logic. First, we cannot compress our modal normal forms further to depth 1 as we did for $\mathsf{ML}(\#)$ in Section 3. Their simplicity is rather in that each level of counting refers to points farther away in the modal accessibility structure, so, intuitively, nested count information refers to different positions. Next, normal forms for the monadic counting language are ‘loose’ in that they do not necessarily contain complete information about all regions in the model. The difference is slight, since, by the linearity of the cardinality order, loose forms can be expanded to disjunctions of complete forms. (We implicitly invoked this fact already when stating Corollary 3.4 in Section 3.) In the modal case, we could also allow loose forms, but we chose the complete version because of the following point.

https://doi.org/10.1017/bsl.2023.30 Published online by Cambridge University Press

Remark 7.16. In modal logic, normal form results are often proved semantically, showing that a formula φ of depth k is equivalent to the disjunction of all k-types occurring in pointed models where φ holds. This semantic argument involves a finite restriction of the notion of modal bisimulation to k back-and-forth steps, and a similar argument can also be given with the more complex notion of bisimulation identified above.

Normal forms are related to *Scott sentences* in infinitary languages, describing, up to suitable ordinal depths, what syntactic types in the language are realized in a given model [117]. Modal Scott sentences in $\mathcal{L}_{\#}^{\infty\mathsf{ml}}$ also include information on numbers of occurrences of types, and can define given pointed models up to bisimulation.

Proposition 7.17. $\mathsf{ML}(\#)$ *is decidable.*

Proof. We show that SAT is decidable for normal forms. At depth 0, this is trivial. At depth $k+1$, we proceed by means of the following pseudo-algorithm.

Working outside in, we check, successively, that (a) the given atomic description for the root point is satisfiable, (b) the system of inequalities for the types of level k occurring among the successors of the root is numerically satisfiable, say by the Fourier–Motzkin algorithm allowing infinite cardinalities as described in Section 4.2, and finally, (c) for each non-zero term that occurs in the solution in point (b), i.e., each relevant type of level k, we test for satisfiability again of these simpler types.

This simple decision procedure is correct because stages (b) and (c) of the procedure are largely separate. If we can satisfy one of the types in stage (c) at all, then, by copying and taking disjoint subtrees, we can satisfy it at any desired number of successors for the root as described by the inequalities of stage (b). That no truth values are disturbed in this procedure is precisely the content of the earlier observations about invariance of modal counting formulas for generated submodels. ⊣

Remark 7.18. The preceding analysis is constructive, and it also contains information about the *reasoning system* for $\mathsf{ML}(\#)$. We will not spell this out in further detail here.

Note also that, by results of Demri and Lugiez [30] on a strictly larger modal fragment (their Theorem 1), Proposition 7.17 can in fact be sharpened to decidability in polynomial space.

7.5. Language extensions. Our modal language is still quite weak in some respects. For instance, unlike the system $\mathsf{MFO}(\#)$, it cannot talk about specific finite numbers of relevant objects: in our case, successors of the

https://doi.org/10.1017/bsl.2023.30 Published online by Cambridge University Press

current point. In addition, we still lack the resources to express further features of the earlier-mentioned Pigeonhole Principle

$$\#_x\exists y.R(y,x) \succ \#_x\exists y.R(x,y) \to \exists x.\exists^{\geq 2}y.R(y,x). \tag{7.1}$$

A modal rendering of the FO(#)-style syntax with explicit variables in this principle requires (a) numerical *graded* modalities, (b) both forward $\Diamond^{\rightarrow}$ and *backward* $\Diamond^{\leftarrow}$ modalities along the ordering R, and also, (c) the notion of counting involved is not local to some current point, but involves a *global* operator $\#_g$, referring to the whole domain. With these modal devices, the Pigeonhole Principle will come to look like this:

$$\#_g\Diamond^{\rightarrow}\top \succ \#_g\Diamond^{\leftarrow}\top \to E\Diamond^{\leftarrow,\geq 2}\top, \tag{7.2}$$

where the 'existential modality' E is defined as having global count greater than 0.

We briefly discuss these extensions in turn. Adding graded modalities to ML(#) seems natural, so as to give the system the same expressive power as MFO(#) over sets of successors. In fact, Demri and Lugiez [30] present a generalized graded modal logic with Presburger-like constraints on types of successors which extends our system ML(#). Given our analysis of additive arithmetic in terms of second-order counting logics, there may be connections here with the second-order version of ML(#) mentioned below in Example 9.9.

Next, adding backward modalities for the converse of the accessibility relation leads to a *tense-logical* version of ML(#). While such an extension seems straightforward, several earlier notions would need non-trivial adaptation. Moreover, the typical valid tense-logical principles $\varphi \to \Box^{\rightarrow}\Diamond^{\leftarrow}\varphi, \varphi \to \Box^{\leftarrow}\Diamond^{\rightarrow}\varphi$ relating the two directions of R suggest a more systematic analysis of the connections between counting in the forward and backward directions.

Also of interest is adding global counting operators, which, as noted above, can define the usual global existential modality over the whole domain. Extending the well-known fact that standard modal S5 provides an alternative notation for monadic first-order logic without identity, we could also consider global counting as a device by itself, yielding another presumably simple modal counterpart to the system MFO(#).

Of course, beyond MFO(#), the other systems considered in earlier sections, too, suggest modal extensions. For instance, an analogue to the notion of *multiple counting* in MFO(♯) might involve 'multi-dimensional' modal counting logics [85]. Perhaps more importantly in representing natural patterns of reasoning, one can add *second*-order quantifiers over sets, on the analogy of the earlier system MSO(#). This would result in a second-order version of ML(#) comparable to basic modal logic with quantifiers over propositions [40]. In fact, if we add quantification over proposition letters to ML(#) with *global* counting, then this gives us an alternative, modal notation for sentences of MSO(#), thanks to Theorem 3.2 and Lemma 4.5.[5] By the same argument as in Section 4.2, such a system will be decidable (Corollary 4.13). One concrete use for it will be found in Section 9.2, when discussing quantifier constructions in natural language.

§8. Generalizing the counting semantics. The systems we studied in Sections 3–7 all dealt with *syntactic* fine-structure and tractable fragments of the natural, but excessively rich and complex system FO(#). However, there is another, complementary means of recovering from intractability; that is to change the *semantics* (cf. [14]). In the present setting, at least two possibilities suggest themselves, each with their own motivation: (1) broaden the interpretation of #-terms, so that they may denote elements of a more general class of algebraic structures, and (2) generalize the logical semantics in ways known to reduce complexity, e.g., by allowing variation in the space of allowable variable assignments [96]. We discuss each in turn, with an emphasis on the generalized-value approach. In this more exploratory section, we will not provide the same level of detail as in our earlier presentation.

8.1. Beyond counting. We can break our interpretation of count terms $\#_x\varphi$ into two steps. In a model $\mathcal{M}$ with domain D and variable assignment s, we first consider the set $[\![\varphi]\!]_x^{\mathcal{M},s} = \{d \in D : \mathcal{M}, s_d^x \models \varphi\}$. In the second step we map subsets $S \subseteq D$ of the domain to *numbers*. Thus we have a map $f : \wp(D) \to \{\kappa : |D| \geq \kappa\}$, with $S \mapsto |S|$. Ultimately we set

$$[\![\#_x\varphi]\!]^{\mathcal{M},s} = f([\![\varphi]\!]_x^{\mathcal{M},s}). \tag{8.1}$$

We now want to consider generalizing Equation (8.1) by allowing a broader class of functions $f : \wp(D) \to \mathbb{P}$, where $\mathbb{P} = (P; \geq)$ may be some other poset than a set of cardinal numbers.

8.2. Probability and proportionality. As a first example, let $\mathbb{P} = ([0,1]; \geq)$ be the real unit interval. (The rational interval $([0,1] \cap \mathbb{Q}; \geq)$ would also suffice for much of what we will say.) Over finite models a natural map to consider is the function $f : \wp(D) \to [0,1]$ sending S to the ratio $|S|/|D|$. It is straightforward to verify that the valid reasoning principles in the systems MFO$^{\phi}$(#) and MSO$^{\phi}$(#) will remain unchanged. The basic propositional and modal systems, PL$^{\phi}$(#) and ML$^{\phi}$(#), can also be given such a proportionality interpretation whereby $[\![\#\varphi]\!]^{\mathfrak{M},s}$ is the proportion of (successor) points where φ holds.

[5] Strictly speaking we also need to add the statement $\#\varphi \approx 1$, expressing that there is exactly one φ point.

https://doi.org/10.1017/bsl.2023.30 Published online by Cambridge University Press

On this interpretation, terms $\#_x\varphi$ (or $\#\varphi$) can be construed as specifying the *probability* that φ is satisfied, a connection between elementary logics of counting and probability made explicit in [59]. The probability measures obtained in this way are all *regular* in that they assign every non-empty set non-zero probability (cf. [33]).

What happens when we move to polyadic systems $\mathsf{MFO}(\sharp)$ and $\mathsf{MSO}(\sharp)$? Some of our work has natural analogues here. For instance, recall again our analysis of 'Many Qs are P' (Equation (3.10)). Interpreted as a probabilistic statement about a measure μ, this says, $\mu(P \mid Q) > \mu(P)$, i.e., that Q 'confirms' P [108]. However, the two interpretations—counting and proportion—no longer agree on general logical principles when we allow poladicity:

Example 8.1. Consider the formula

$$\exists y, z.\big(y \neq z \wedge \sharp_x(x = y \vee x = z) \approx \sharp_{x,w}(P(x) \wedge P(w))\big).$$

This is not satisfiable under our counting interpretation since it would require $|P| = \sqrt{2}$. By contrast, on the proportionality interpretation: it merely requires that $2 \times |D| = |P|^2$.

This echoes a broader theme that reasoning about conditional probabilities already amounts to general reasoning about real fields [63, 93].

8.3. Mass, weight, and abstract values. Probability and proportionality are still clear quantitative numerical measures of sizes and ratios. However, generalizing beyond these, our logics also support more qualitative interpretations as calculi of 'weight' or 'mass' [83]. In the above two-step setup, we can think of terms $\#_x\varphi$ as denoting a collective entity in some intuitive sense, say, like in the semantics of *plural* expressions and mass terms in natural language. The values assigned to these might lie in some qualitative mereological algebra. The minimum needed for interpreting our #-languages is then some pre-order on this mereological algebra, while further structure may come in the form of, not addition or multiplication, but *fusion*, and perhaps other mereological primitives [79].

We will not pursue this more general perspective here, which deserves a separate development of its own. Instead, we only note two changes from our earlier logical analysis.

8.4. Non-classical logics. Recall that in Section 2.1, we pointed at different logics, classical or non-classical, to come out of the counting component of our systems. In the current generalized setting, ways of inducing logical operations multiply. For instance, as long as $f(D) \geq f(\varnothing)$ and $f(\varnothing) \not\geq f(D)$, we will recover at least the classical Booleans in the same way as before via (2.3) and (2.4). However, if we merely drop the requirement that $f(D) \geq f(\varnothing)$, we can already invalidate 'paradoxes of material implication' such as

https://doi.org/10.1017/bsl.2023.30 Published online by Cambridge University Press

$\varphi \to (\psi \to \varphi)$, while still validating some principles typical of relevant logics (see [110]), such as $\neg\neg\varphi \leftrightarrow \varphi$. We leave further exploration of this way of inducing logical systems to future work.

8.5. Embedding into multisorted FO**.** Our second logical observation is more technical. With a generalized semantics, some of our earlier conclusions about system behavior and complexity need to be reconsidered. Provided we only stipulate *first-order* conditions on the partial order $\mathbb{P}$ and on the map $f : \wp(D) \to \mathbb{P}$—both natural requirements in abstract mereological semantics—we can show that the set of valid principles for $\mathsf{FO}(\#)$ becomes *computably enumerable*. The same style of analysis also applies to the extended system $\mathsf{FO}(\sharp)$ which allows counting tuples.

THEOREM 8.2. *Over generalized models, satisfiability in* $\mathsf{FO}(\sharp)$ *can be translated faithfully into satisfiability in an associated three-sorted first-order language.*

PROOF SKETCH. The idea here is as follows. The above generalized semantics works on three-sorted structures with a domain D of objects, a domain P of collectives or 'predicates' for the denotations of #-terms, and a value domain V, with a binary relation E between objects and predicates, a function $f : P \to V$ sending predicates to values, and a binary relation $\geq$ on the value domain. We can state what is needed for this to work in first-order terms, on the analogy of Henkin models for second-order logic: (a) an *Extensionality* principle stating that predicates standing in the E relation to the same objects are the same, and (b) a set of *Comprehension* principles making sure that the domain of predicates is closed under definitions in our language with finitely many parameters.

With this in place, we can translate our #-languages into this three-sorted first-order language. In particular, the Tr-translation of an expression $\#_x\varphi \succsim \#_x\psi$ will read

$$\exists p, q. \forall x\big(E(x,p) \leftrightarrow \mathsf{Tr}(\varphi)(x)\big) \wedge \forall x.\big(E(x,q) \leftrightarrow \mathsf{Tr}(\psi)(x)\big) \\ \wedge f(p) \geq f(q).$$

This translation is easily extended to multiple count operators, where the domain of available predicates now includes predicates of arbitrary finite arities, whose natural closure properties can still be described in first-order style.

Now it is straightforward to show that a formula φ of our #-language is satisfiable in abstract value semantics iff its translation $\mathsf{Tr}(\varphi)$ is satisfiable in a three-sorted model for the above effectively axiomatized first-order theory consisting of Extensionality and Comprehension. ⊣

Incidentally, the same strategy can also bring down the complexity of second-order versions of our #-languages, as we let second-order quantifiers

https://doi.org/10.1017/bsl.2023.30 Published online by Cambridge University Press

range, Henkin-style, over the special set of available predicates in the above three-sorted models.

The shift to a first-order perspective has noteworthy repercussions for the meta-properties of #-logics. Consider the failure of *Compactness* observed earlier (Proposition 2.1): this property will hold now, because of the first-order reduction outlined above.

Example 8.3. It is of interest to see how this works with the standard counterexample to Compactness. This is the finitely satisfiable set $\{\neg\exists^{\infty}x.\top\} \cup \{\exists^{\geq n}x.\top : n < \omega\}$. This set is not satisfiable in our standard cardinality semantics, but it is satisfiable in generalized semantics. Concretely, we take a language with only the identity predicate, dropping unary predicates for convenience. Now consider finite models $\mathcal{M}_n$ of all cardinalities n and take an ultrapower over these with respect to a free ultrafilter. In the resulting model, all first-order properties of our finite models still hold, and we can say concretely how the generalized function f works. The value domain will be an uncountable linear order consisting of one copy of the standard natural numbers followed by copies of the integers with, at the end, one copy of the negative integers. On finite subsets, f gives sizes in the standard natural numbers, and on cofinite sets, it will give values in the final copy of the negative integers, counting down from the infinite largest element.

8.6. Generalized dependence semantics. There is also a more logic-oriented approach to lowering the complexity of our initial system $\mathsf{FO}(\#)$. In so-called *generalized assignment semantics* for first-order logic, models come with a range of admissible or available assignments, where gaps in the full space of all functions from variables to objects encode *dependencies* or correlations between variables [5, 96]. The main truth condition is now that $\mathcal{M}, s \models \exists x\varphi$ iff there exists some admissible assignment t in the model which is equal to s except for the value of x, and such that $\mathcal{M}, t \models \varphi$.

It is known that the set of validities on generalized assignment models is decidable [96], while additional first-order principles impose further existential closure conditions on the admissible assignments, such as Church–Rosser style confluence properties that support the encoding of undecidable tiling problems, thus elucidating the assumptions underlying the undecidability of FO on standard models. Moreover, generalized assignment models support various decidable language extensions, such as polyadic tuple quantifiers [14], and explicit atoms expressing functional dependence of variables y on sets of variables X [5].

To extend the semantics of $\mathsf{FO}(\#)$ to generalized assignment models, we need a stipulation as to how we are going to *count* in this setting. Various options may be considered given the richer environment of available vs. arbitrary assignments, but here is one that seems natural. At an assignment s in a model $\mathcal{M}$,

https://doi.org/10.1017/bsl.2023.30 Published online by Cambridge University Press

> $\#^X_Y\varphi$ denotes the cardinality of the set of all tuples of values taken by the set Y in those assignments in $\mathcal{M}$ that (a) agree with s on their X-values, and (b) make φ true.

This counts ranges of values for some variables conditional on the current values of other fixed variables, leaving open the status of yet other variables occurring in the formula φ.

In terms of this count notion, for instance, the existential quantifier (and the dependence modalities of Baltag and van Benthem [5]) can easily be defined in the style that was introduced in Section 2.1. Moreover, functional dependence of y on X can be written as $\#^X_y\top(y) \approx \#^{\{x\}}_x\top(x)$. But we can also express other notions of correlation, up to forms of *independence*. For instance, $\#^X_y\top(y) \approx \#^{\emptyset}_y\top(y)$ says that the local values of X leave a value range for y whose cardinality is equal to the total value range for y in the model.

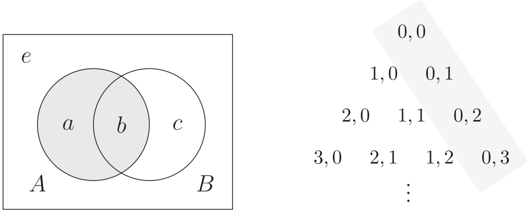
We submit that this combination of generalized assignment semantics and count terms is interesting, but exploring its natural open problems is beyond the scope of this paper.

§9. Generalized quantifiers and natural language. The preceding section concludes our analysis of elementary combinations of logic and counting in terms of a standard hierarchy of designed formal systems. Let us now return to the setting of our Introduction, and take a look at how these issues manifest in natural language, the vehicle for our broader daily practices of logical reasoning and counting.

An obvious source for such a comparison is Generalized Quantifier Theory [8, 11, 100], an area where logic and counting have always coexisted, even though the field often places the emphasis on logic in the formal syntax, while the counting aspect resides in the semantics. We will develop this interface with more empirical practice in some detail, and show how quantification in natural language and the theory developed around it connect in interesting ways with the earlier systems. As it happens, new questions will arise both ways.

A point of notation: Throughout this section we will be using letters $A, B, C, \ldots$ for subsets of a domain D. Following the semantic literature, we will use the same letters interchangeably as predicate symbols in a formal language, provided no confusion can arise.

9.1. Quantifier expressions in logical semantics. There is a wide range of quantifier words and quantificational constructions in natural language. The quantifier vocabulary includes first-order expressions such as ‘all’, ‘some’, ‘no’, but also numerals like ‘one’, ‘two’, or combinations like ‘all except two’. But there are also higher-order expressions such as ‘most’, and expressions



(a) Effect of Conservativity, Extension, and Invariance

(b) The tree of numbers (a, b) with 'all' highlighted in red

'all'	$a = 0$
'some'	$b \neq 0$
'at least two'	$b \neq 0 \wedge b \neq s0$
'most'	$b > a$
'even number of'	$\exists x.b = x + x$
'many'	$b \times e > a \times c$

(c) Arithmetical expressions for some binary quantifiers

FIGURE 5. (a) Assuming Conservativity, Extension, and Invariance, we need only be concerned with $a = A - B$ and $b = A \cap B$. (b) The tree of numbers consists of all pairs (a, b). Highlighted are pairs in the quantifier 'all'. (c) Examples of quantifiers and their arithmetic expressions. Note that, in addition to requiring multiplication, the quantifier 'many' violates Extension.

whose meaning is highly context-dependent such as 'many', 'enough', and so on. Moreover, the quantifier vocabulary includes comparative expressions such as 'More A than B are C' or 'As many A as B are C', or even 'Twice as many A as B are C'.

For a basic pattern, one usually takes the binary format QAB of $\langle 1, 1\rangle$ quantifiers, with Q a quantifier expression and A, B unary predicates denoting sets of objects. It is generally assumed that quantifiers in natural languages satisfy some universal constraints, such as,

Conservativity QAB holds iff $QA(A \cap B)$ holds.

In what follows, in line with the literature, we will assume Conservativity throughout, though most of our results could be given more general formulations. Other widely assumed constraints hold in many cases. An important one is *Extension* saying that the relation QAB does not depend on the total universe of objects inside which the sets A, B are located. Finally, true quantifier expressions are purely numerical in the sense of satisfying the following constraint, which also played a key role in Sections 3–5:

Permutation Invariance QAB holds iff $Q\pi[A]\pi[B]$ for any permutation $\pi : D \to D$.

The total effect of these three conditions ties quantifiers closely to counting. To specify the meaning of a quantifier expression Q, it suffices to list its acceptance behavior on all pairs of numbers (a, b) where $a = |A - B|, b = |A \cap B|$ for some A, B such that QAB holds (see Figure 5a). Accordingly, Generalized Quantifier Theory studies quantifiers equally well in numerical terms as in logical ones. A typical tool in Generalized Quantifier Theory for visualizing this double perspective is the so-called *tree of numbers* for representing quantifiers graphically in terms of the pairs (a, b) representing cardinalities $(|A - B|, |A \cap B|)$ (Figure 5b). Here a quantifier can be seen as

https://doi.org/10.1017/bsl.2023.30 Published online by Cambridge University Press

a subset of the tree. Further special properties of quantifiers then show up as geometrical patterns in the tree. For instance, if Q is upward monotonic in its right-hand argument, Q will be closed when moving upward along an upward diagonal line from any point where it holds. Upward monotonicity in the left-hand argument shows as acceptance of the sub-quadrant generated by points (a, b) where Q holds. These geometric descriptions lead to simple characterizations of all possible monotonic quantifiers, or all first-order definable quantifiers [11].

Remark 9.1. From our perspective, the tree of numbers approach is interesting for its twists. It arises by passing from logical syntax to numerical content of quantifier expressions, but with that in place, it again geometrizes that numerical content, something that one might see as a further move to qualitative geometric logic.

9.2. Linguistic vocabulary and #-logics. For a start, the logical system $\mathsf{MFO}(\#)$ shows various analogies with the preceding style of analysis. First in terms of general constraints, it enjoys the following well-known logical property.

Fact 9.2. $\mathcal{L}^1_\#$ is closed under *relativization* to definable subdomains.

Proof. The crucial step of defining a relativization map $\varphi \mapsto (\varphi)^A$ is given by the transformation of $\#_x(\varphi) \succsim \#_y(\psi)$ into $\#_x(A(x) \wedge (\varphi)^A) \succsim \#_y(A(y) \wedge (\psi)^A)$. $\dashv$

This property allows us to explore valid reasoning principles in $\mathsf{MFO}(\#)$ that assume Conservativity and Extension, which did not yet come to the fore in our earlier analysis in Section 3. An illustration is the following principle of *Quantity*:

$$\big((\varphi)^A(B) \wedge \#(A - B) \approx \#(C - D) \wedge \#(A \cap B) \approx \#(C \cap D)\big) \rightarrow (\varphi)^C(D).$$

Here $(\varphi)^A(B)$ makes a pure cardinality statement about B inside the set A (a strong form of Conservativity) and the conclusion is that this statement would hold for any set C with the same numerical behavior with respect to D. This expresses a form of Permutation Invariance in terms of validities in $\mathsf{MFO}(\#)$.

As for inference patterns for specific quantifiers, our Introduction highlighted numerical syllogisms with first-order quantifiers, numerals, exceptive quantifiers ‘all except at most k’, and comparative quantifiers such as ‘most’ or ‘more...than...’. Numerical syllogisms are often analyzed in practice using Venn Diagrams with number information written into the zones, as in Figure 5a. This representation was the intuitive basis for the normal forms for the system $\mathsf{MFO}(\#)$ (recall Figure 1) which encodes all of the above reasoning.

These informal comparisons can be made precise by means of definability results. Here is an illustration using the model-theoretic definability analysis presented in Section 3.3.

Theorem 3.5. *The binary quantifiers definable in* $\mathsf{MFO}^{\phi}(\#)$ *correspond exactly to those expressible in the first-order theory of* $\langle \mathbb{N}; > \rangle$.

Proof. Consider the elementary theory of $\langle \mathbb{N}; > \rangle$, the natural numbers with the binary relation 'greater than'. For simplicity, assume we also have (the definable) function symbols 0 and s. This logic then has quantifier elimination: every formula $\alpha(x, y)$ is equivalent to a Boolean combination of equalities and inequalities between terms of the form $s^{n_1}(0)$, $s^{n_2}(x)$, or $s^{n_3}(y)$. The key to the theorem is that these are also exactly the normal forms for the binary quantifiers definable in $\mathsf{MFO}(\#)$.

To make this more precise, we first characterize exactly the type $\langle 1, 1 \rangle$ quantifiers that can be defined by formulas $\varphi^A(A, B)$. By relativization (Fact 9.2) we only have two relevant state descriptions: $A \cap B$ and $A - B$. By Theorem 3.2, $\varphi^A(A, B)$ is equivalent to a disjunction of m-inequalities with constant numbers m involving $\#(A - B)$ and $\#(A \cap B)$. We can assume that such inequalities involve no sums, as they would denote the size of the whole domain, and we can then eliminate these. Of the remaining cases, the statement $k \succsim \#(A - B) + \#(A \cap B)$ can be rewritten as a large disjunction over all ways of dividing k or less between $\#(A - B)$ and $\#(A \cap B)$. Likewise, $\#(A - B) + \#(A \cap B) \succsim k$ is rewritten as a large disjunction over all pairs adding up to k. The remaining cases are handled similarly. The result is a Boolean compound of inequalities that is expressible in the first-order theory of $\langle \mathbb{N}; > \rangle$.

Next, we map the language of formulas $\varphi^A(A, B)$ to the arithmetical language by employing two distinguished variables x and y, corresponding to the $\mathcal{L}^1_\#$-terms $\#(A \cap B)$ and $\#(A - B)$, respectively. By the foregoing it is easy to see that every expression $\varphi^A(A, B)$ corresponds to an arithmetic formula $\alpha(x, y)$ in the two free variables x, y. In the other direction, each arithmetical formula $\alpha(x, y)$ of the form produced by quantifier elimination is easily seen to be expressed by an appropriate $\mathcal{L}^1_\#$ formula. ⊣

Thus, the first-order quantifiers of natural language are definable. But of course $\mathsf{MFO}(\#)$ can define non-first-order quantifiers too, such as 'Most *A* are *B*'. The normal form format of Section 3 cuts across standard first-order/second-order boundaries.

Remark 9.3. The binary quantifiers definable in $\mathsf{MFO}(\#)$ can be classified algebraically in terms of our normal forms, but a more geometrical perspective is provided by the earlier mentioned tree of numbers. This was a discrete version of the usual representations of solution sets for systems of linear inequalities (cf. [115, p. 85] for a systematic treatment). In this special

https://doi.org/10.1017/bsl.2023.30 Published online by Cambridge University Press

case with just two numbers a, b, the inequalities occurring in our normal forms reduce to the following types:

(i) $a = k$, (ii) $a > k$, (iii) $a + b = k$, (iv) $a + b > k$, (v) $a = b + k$, (vi) $a > b + k$, plus all versions of these with a, b interchanged.

To see why all these forms can occur, note that terms T_i in normal forms are disjunctions of state descriptions, with the empty disjunction allowed. On the other hand, we can suppress some possible forms such as $k > a$, since these are finite disjunctions of equalities.

Now, in the tree of numbers, these types correspond with simple geometrical patterns. (i) describes a right- or left-sloping *diagonal line*, (ii) an infinite downward *triangle*, (iii) a horizontal line, (iv) a *trapezoid* below a horizontal line, (v) a vertical line, and (vi) a *slice*: a left- or rightward half triangle. Analyzing up to finite disjunctions, we look at the intersections produced by these. Here, as in earlier definability arguments, we focus on what happens beyond some finite tree level, as a finite number of points above that level can be dealt with by adding explicit definitions in terms of intersecting lines. Next, shapes can be simplified further in term of finite unions: a horizontal line is a finite set of points, a trapezoid is a finite union of triangles, and a full triangle is a union of two slices. We are left with the following basic shapes (above a certain tree level): *diagonal lines*, *vertical lines*, and *slices*. Intersections of these can produce finite unions of (a) single points, (b) diagonal lines from some point onward, (c) horizontal lines from some point onward, and (d) slices from some point onward. For instance, intersecting two slices with different orientations produces an infinite 'band' extending vertically downward, but this is a finite union of vertical lines.

In all, we are left with finite disjunctions of the following MFO(#)-definable types of quantifiers: (a) 'Exactly k A are B and exactly m A are not-B', (b) 'There are at least k A and exactly m of these are B', 'There are at least k A and exactly m of these are not-B', (c) 'There are at least k A and among these, there are equally many B and not-B', and (d) 'There are at least k A with at least m B among these, and fewer B than non-B' (for a left-looking slice), and vice versa for the other case.

This geometric analysis extends that for first-order quantifiers in [11], where the only basic shapes needed are diagonal lines and triangles.

Remark 9.4. It should be noted that the preceding analysis is about quantifiers on finite domains only. While this restriction is often assumed in the semantics of natural language, a generalization to *infinite models* would be of interest. For an extension of the tree of numbers representation to infinite cardinalities, cf. [28].

Clearly, the infinitely many quantifiers definable in the above manner are not all realized in natural language, though they can drive an interesting search for examples and non-examples. For instance, the simple pattern

https://doi.org/10.1017/bsl.2023.30 Published online by Cambridge University Press

$b > a + 2$ seems to defy a simple unforced linguistic description, say, in terms of 'most' and 'except'. Of course, with enough words, one can always paraphrase what this says in artificial ways (cf. (1)), but we are interested here in the quantifiers that have actually been *lexicalized* in natural languages (roughly in the sense of being 'morphosyntactically simple'; see, e.g., [68]).

On the other hand, there are also realistic quantifiers in natural language that our base system cannot express.

Corollary 9.5. $\mathsf{MFO}(\#)$ *cannot express the quantifier 'An even number of As are B' or the proportionality quantifiers 'At least* $1/n$ *of the As are B' for* $n > 2$.

These additional quantifiers require the resources of our second-order system $\mathsf{MSO}(\#)$, which overshoots considerably compared to natural language, as it can define all Presburger definable logical quantifiers. The following is a direction consequence of Theorem 4.7:

Corollary 9.6. *The binary quantifiers definable in* $\mathsf{MSO}(\#)$ *are exactly those expressible in the first-order theory of* $\langle \mathbb{N}; + \rangle$.

Here, the over-generation of the logic continues. Say, $2a = b$ says that the number of As that are B equals twice the number of As that are not, or rephrased: two thirds of the As are Bs. This is intelligible, but not part of natural basic quantifier vocabulary.

Finally, our move to the richer system $\mathsf{MFO}(\sharp)$ and Diophantine arithmetic raises even further issues. Here is what we found earlier.

Corollary 9.7. *The quantifiers definable in* $\mathsf{MFO}(\sharp)$ *are exactly those expressible in the quantifier-free theory of* $\langle \mathbb{N}; +, \times \rangle$, *while the second-order system* $\mathsf{MSO}(\sharp)$ *adds those defined using first-order quantifiers over numbers.*

It has been suggested in [11] that the arithmetical content of linguistic quantifiers is essentially restricted to addition. In that case, multiplication would be irrelevant to understanding the linguistic quantifier repertoire. However, our current analysis throws doubts on this picture. The natural meaning of 'many' involved multiplication, and natural language does have resources for comparing proportions. Moreover, it does form *pairs of objects* in basic syntax, witness naturally occurring relational phrases such as 'who married whom'. The resulting counting of pairs or longer tuples suggests connections with our multiple count logic $\mathsf{MFO}(\sharp)$. However, the formulas that we used to define multiplication have somewhat artificial variable binding patterns that need not occur in natural language. The multiplicative content of natural quantifier expressions remains to be determined.

Finally, while the preceding discussion was about basic *quantifier vocabulary*, natural language also has more complex *quantifier constructions*. Well-known constructions of logical interest are 'cumulative' and 'branching

https://doi.org/10.1017/bsl.2023.30 Published online by Cambridge University Press

quantifiers' (see [100]). A particular construction worth highlighting here is the role of particles qualifying meanings of quantifier combinations. Consider a sentence like:

$$\text{`Every family has a different problem'.} \tag{9.1}$$

This is not just a simple $\forall\exists$ combination, demanding the existence of some choice function from families to problems. The particle 'different' requires that choice function to be one-to-one, more like our cardinality comparison statements. However, there is a crucial difference. In this case, the one-to-one function must lie *inside a given relation*, in our concrete sentence: the relation *having*. This seems a case where natural language poses a challenge.

Remark 9.8. Linguists have been well aware of these and related phenomena, and have advanced relatively complex machinery to handle the full range of attested patterns. See, e.g., [19, 20] for two recent dynamic accounts.

Notably, such constructions come up in the context of probabilistic reasoning as well [54], in a way that reverberates elsewhere in natural language, witness expressions for probability and likelihood [62].

We suspect that this notion of 'guarded injection' is not even definable in the strong counting logic $\mathsf{FO}(\#)$. However, for finite cardinalities there is a connection with the weaker logics considered in this paper: in this case, a modal system.

The *Hall marriage theorem* in graph theory [51] says that there is an injection from a set A into a set B contained in a relation $R \subseteq A \times B$ iff for each subset C of A, $|R[C]| \geq |C|$. But this can be used to give a simple definition of 'different' sentences like (9.1) in our modal logic with global counting and one second-order quantification over sets:

Example 9.9. Let F be the unary predicate for family, G for problems, and suppose $\Diamond$ moves along the relation 'x is had by y'. Then the required definition is

$$\forall X\big(X \subseteq F \to \#(G \wedge \Diamond X) \succsim \#X\big),$$

where $X \subseteq F$ is shorthand for $\#\bot \succsim \#(X \wedge \neg F)$.

This concludes our brief comparison of quantifier expressions in natural language with the expressive resources of our #-logics. Clearly, this is not so much a matter of proving theorems as of exploring empirical fit. The hierarchy in our system design may suggest patterns in the architecture of natural language, while, precisely when the fit is not evident, common constructions in natural language may pose non-trivial questions concerning logical systems. We have just provided some illustrations here, a deeper investigation of linguistic versus logical architecture would require another paper.

https://doi.org/10.1017/bsl.2023.30 Published online by Cambridge University Press

9.3. Varieties of monotonicity reasoning. Next we move from quantifier vocabulary to inference patterns in natural language. *Monotonicity inferences* arise when occurrences of a predicate in positive syntactic position are replaced 'upward' by occurrences of a predicate with a larger denotation, or when in negative position, 'downward' by a predicate with a smaller extension [11, 64, 114]. Monotonicity inference works all across natural language for many kinds of quantifiers, but just as well for other numerical expressions, witness a valid inference like 'If more A than B are C, and all A are E, then more E than B are C'.

Monotonicity with *inclusion premises* is also a valid inference form in logical systems, and in particular, in the ones studied here. Let us mark syntactic positions as follows in formulas of $\mathsf{MFO}(\#)$. An atomic formula $P(x)$ occurs positively in $P(x)$ itself, positive and negative occurrences keep their polarity in conjunctions and disjunctions, their polarity switches under negations, and finally, in atoms $\#_x\varphi \gtrsim \#_x\psi$, occurrences in ψ switch polarity, while those in φ keep their polarity. It is easy to show the following:

PROPOSITION 9.10. *Positive occurrences in formulas of* $\mathsf{MFO}(\#)$ *support valid upward monotonicity inferences, negative occurrences downward monotonicity inferences.*

REMARK 9.11. It seems likely that $\mathsf{MFO}(\#)$ also satisfies a *Lyndon Theorem* to the effect that semantic monotonicity amounts to positive definability up to logical equivalence (see [12, 65]). Our normal forms contain all information necessary for a constructive proof of this result. However, we leave this as an open problem.

The syntax of $\mathsf{MFO}(\#)$ in fact suggests two kinds of monotonicity reasoning: the usual one with inclusion premises, but also one with *cardinality premises*, in forms such as

$$\varphi(B) \text{ and } \#A \gtrsim \#B \text{ imply } \varphi(A).$$

As it happens, the inductive clauses for positive and negative occurrences work here as before, the crucial failure is the atomic clause, as premises $Bx, A \gtrsim B$ obviously do not imply Ax. Clearly, numerical monotonicity implies its set-theoretic variant, but the converse can fail. The quantifier 'Some B are C' is upward set-monotonic in its argument B, but obviously not numerically monotonic in B, since the larger set A may be disjoint from B and C.

REMARK 9.12. Numerical monotonicity as stated here has some interesting features as a mixture of logic and counting. As a special case, if $\varphi(A)$ is true and we replace A by a predicate B of the same cardinality, then $\varphi(B)$ is true. This very strong insensitivity property intuitively separates φ into some purely numerical assertion about A plus an assertion that is not about A at

https://doi.org/10.1017/bsl.2023.30 Published online by Cambridge University Press

all. This may be provable as a preservation theorem for formulas in first-order logic, and for $\mathsf{MFO}(\#)$, a complete characterization of numerically monotonic formulas may be provable through our normal forms. However, we end with one small observation.

Consider binary quantifiers Q definable in the logic $\mathsf{MFO}(\#)$. In some cases, the two kinds of monotonicity are close. For instance, if $Q\,AB$ is upward set-monotone in the argument B, then it is also upward cardinality-monotone in the following sense, restricted to the set A. If $Q\,AB$ and *at least as many A are C as B*, then $Q\,AC$. The crucial property here is *Permutation Invariance*: given A, the quantifier Q is fixed by the set of all sizes of subsets B which it accepts, and set-monotonicity plus permutation invariance imply that these sizes are upward closed. The restriction to comparing inside A is necessary here, since cardinal monotonicity w.r.t. B for arbitrary larger C not inside A can easily fail. The same failure occurs with upward set-monotonicity in the left-hand argument A, where a larger set C may change the context of evaluation. Even so, permutation invariance does support a valid second-order inference pattern for left-upward set-monotonic quantifiers Q: if $Q\,AB$ and *there are at least as many C as A*, then $\exists C' \approx C.\, Q\,C'B$, where C' can be taken to be any set equinumerous to C that contains A, so that left-upward monotonicity applies to it.

Cardinality monotonicity resembles monotonicity in numerical terms, where a variable x occurs positively in x, retains its polarity across addition, multiplication, and the left-hand side of inequalities $\mathbf{t}_1 \geq \mathbf{t}_2$, while switching polarity on the right-hand side of these inequalities. Making this work using our normal forms takes some care though, since their numerical terms T_i do not refer to sizes of predicates, as in the above, but of state descriptions. A unified perspective on monotonicity in logical and arithmetical syntax has been proposed in [64]. As for concrete examples, van Benthem and Liu [15] note several different versions of set-based and size-based monotonicity inference that hold for the natural language expression 'Many A are B' that involve increasing or decreasing the size of relevant zones in the Venn diagram for A, B.

REMARK 9.13 (Natural logic). Monotonicity reasoning in natural language is an engine of 'natural logic' [11, 90, 114]: efficient forms of surface reasoning based on simple fragments and proof systems. Our #-logics are more expressive than most of the calculi studied in this literature, and it would be of interest to locate natural logic fragments inside them (see, for example, [71, 91, 92, 104, 105]).

9.4. Dynamic modalities. Monotonicity inference can also be viewed dynamically in terms of *model change*. One such change is internal to a current model: one merely changes the denotation of some predicate A to a larger (or smaller) set X of objects, turning the current $\mathcal{M}$ into a new model $\mathcal{M}[A := X]$. Other operations on models arise with different intuitive takes on what upward monotonicity inference is about. It could also mean that we add *new objects* to the current model that satisfy the predicate A, in which case the relevant relation between models is extension. And this perspective can even be generalized. On the earlier analogy with monotonicity in numerical terms, since the latter stand for zones of the model in our normal forms, the replacement for, say, $x := x + 1$ applies to *regions* defined by state-descriptions, rather than single predicates.

In recent years, model change has been studied by adding dynamic modalities to logical languages, cf. the recent study [16]. A standard example is $[!\,\varphi]\psi$ which says that ψ is true after we *relativize* the current model to the submodel of all objects satisfying φ. This fits the earlier discussion of Conservativity and Extension for quantifiers. Next, upward inclusion monotonicity in our first sense suggests a modality $[+A]\psi$ which holds when ψ is true in all models arising from the current one by increasing the denotation of A. Downward monotonicity may then refer to decreasing the denotation of A, or more drastically, to removing objects from the current model. The dynamic modality $[-\,\varphi]\psi$ for the latter model change says that each removal from the current model of an object satisfying φ results in a model satisfying ψ.

Proposition 9.14. $\mathsf{MFO}(\#)$ *is closed under the dynamic modalities* $[!\,\varphi]$ *and* $[-\,A]$. $\mathsf{MSO}(\#)$ *is closed under the modality* $[+A]$.

Proof. The case of relativization can be dealt with by providing axioms that recursively analyze the possible syntactic shapes of the formula ψ. The proof for the deletion modality is by inspection of normal forms in a manner similar to that used in [16, Theorem 6.1]. $\mathsf{MFO}(\#)$ is not closed under the predicate extension modality, since it can define having an even number of points with some property, but it does have a straightforward definition in the second-order $\mathsf{MSO}(\#)$. ⊣

Similar closure results can be obtained in our monadic #-logics for dynamic modalities describing the effects of adding an object to the current model.

Remark 9.15. Another source for model change occurred with the discussion of counting in modal languages in Section 7. Instead of adding explicit numerical information like in graded modal languages, one can also count by 'setting aside' objects and then perhaps replacing them, removing or adding objects to a current model. For instance, having at least $k + 1$ successors with property p is definable using the deletion modality as $[-\,\top] \dots (k \text{ times}) \dots [-\,\top]\Diamond p$ and thus *counting in the syntax*. There is a link here with Remark 7.13 about possible finer notions of modal bisimulation that analyze counting procedures. A typical way of comparing sizes between

https://doi.org/10.1017/bsl.2023.30 Published online by Cambridge University Press

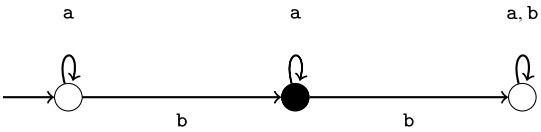


FIGURE 6. Acyclic finite automaton recognizing ‘exactly one’. The machine begins in the left-most state, and the middle is the only accepting state.

two sets picks an object in one set plus an object from the other set, and then puts these two objects aside, iterating the process. But keeping track of effects of removals of matched objects is exactly what MLSR-style bisimulations do (see [16]).

Technical topics like dynamic modalities may seem far from natural language. But the distance is not that great. Natural language contain many verbs of *fact change* that fit this setting. Indeed, [123] contains samples of logical reasoning in the ancient Chinese tradition that involve monotonicity inferences with dynamic verbs such as ‘increase’.

9.5. Semantic automata. Our final topic comes again from Generalized Quantifier Theory, and it brings one more entanglement of logic and counting. There is a natural way of classifying quantifiers in terms of the associated *verification procedures* and determining their complexity in the Automata Hierarchy [11]. The word ‘count’ is of course polysemous between a verbal use (the act of counting) and a nominal use (the total counted), and here the focus is on the former, dynamic aspects of counting.

Semantic automata read strings of symbols a, b standing for types of relevant objects encountered when traversing a finite domain (Figure 5a). That is, each element of $A - B$ corresponds to an occurrence of a in the string, while each element of $A \cap B$ corresponds to an occurrence of b. The automaton reads the string and accepts precisely when the pair (a, b) is in the quantifier. These automata, and the complexity jumps predicted by them for quantifier denotations, have also been studied as models for the mixture of quantifier reasoning in the brain and cognitive sciences (see [124] for an overview).

EXAMPLE 9.16. The acyclic finite automaton in Figure 6 recognizes the quantifier ‘exactly one’. It accepts any pair $(a, 1)$ with $a \geq 0$, and no other pairs. That is, there should be exactly one element in $A \cap B$; more or fewer should lead to non-acceptance.

Moreover, familiar operations on quantifiers, such as *iteration*, correspond systematically to natural operations on standard classes of automata [121]. We list some known results on the subject:

https://doi.org/10.1017/bsl.2023.30 Published online by Cambridge University Press

Proposition 9.17.

(a) *The first-order definable binary quantifiers are exactly those that are recognized by acyclic finite automata* [11].
(b) *Finite automata with non-trivial cycles can recognize 'An even number of A are B' and related periodic quantifiers. In fact, finite automata recognize precisely the quantifiers definable in first-order logic with divisibility* [95].
(c) *The binary quantifier 'most' and related proportionality quantifiers are not computable by finite automata, but they are computable by pushdown store automata. In fact, pushdown automata recognize precisely the quantifiers definable in additive Presburger Arithmetic, i.e., the semi-linear sets* [11].

We thus see numerous deep connections with our earlier systems. Most obviously, we saw the semi-linear sets in our analysis of $\mathsf{MSO}^{\phi}(\#)$ (Theorem 4.7). Proposition 9.17 adds a further computational dimension to this characterization: the quantifiers definable in $\mathsf{MSO}^{\phi}(\#)$ are precisely those that can be verified by pushdown automata. The counting procedures required for verifying claims of $\mathsf{MSO}^{\phi}(\#)$ are those that can be carried out with a pushdown store.

Identifying such a computational analogue for our other systems could also be illuminating. For instance, our initial system, $\mathsf{MFO}^{\phi}(\#)$, misses some quantifiers definable even by finite automata—'an even number of' being an illustrative example (Corollary 9.5)—while capturing some quantifiers that demand unbounded memory such as 'most' or 'exactly half'. It also makes sense to interrogate the other direction. What systems combining logic and counting would capture the quantifiers recognizable by intermediate classes such as *counter automata*, or even weaker classes like those recognizing *subregular languages* (cf. [47])? We leave such questions for further analysis, but end here with a final observation tying together several of our earlier themes, including *permutation invariance*.

As we have seen as multiple points (Sections 3.1, 9.1, and 9.3), the theme of permutation invariance is paramount in the analysis of logic and counting. Given this assumption for quantifiers, the corresponding formal languages will also be closed under permutations. For instance, if `ababa` appears in the quantifier language, so will `aaabb`. This is a relatively exceptional property for sets of strings: the permutation closures of languages accepted by finite automata and by pushdown automata actually coincide—as it happens, they characterize the semi-linear sets [99]. It is therefore of interest to understand the permutation closed (or 'commutative') languages in their own right. Such languages have been studied since the beginning of formal language theory (e.g., [35]). Here our question is the following: restricting to permutation closed languages, which semi-linear sets are also accepted

https://doi.org/10.1017/bsl.2023.30 Published online by Cambridge University Press

by *finite* automata? This would give us a way of calibrating the counting capacity of finite-state machines, relative to semi-linear sets.

With an alphabet of size two, recall that *linear* sets (Definition 4.2) are the solutions (for $\mathsf{v}_1, \mathsf{v}_2$) to equations given by constants $b_1, b_2, a_{1,1}, \dots, a_{1,m}, a_{2,1}, \dots, a_{2,m}$:

$$\begin{pmatrix} \mathsf{v}_1 \\ \mathsf{v}_2 \end{pmatrix} = \begin{pmatrix} b_1 + a_{1,1}\mathsf{u}_1 + \cdots + a_{1,m}\mathsf{u}_m \\ b_2 + a_{2,1}\mathsf{u}_1 + \cdots + a_{2,m}\mathsf{u}_m \end{pmatrix} \tag{9.2}$$

for some choices of $\mathsf{u}_1, \dots, \mathsf{u}_m$; the semi-linear sets are finite unions of linear sets.

Definition 9.18. Let us call a set *rectilinear* if it is of the form (9.2), but for all $i \leq m$ either $a_{1,i} = 0$ or $a_{2,i} = 0$ (or both). A set is *semi-rectilinear* if it is a finite union of rectilinear sets.

It may be helpful to explain this notion in the earlier geometrical setting of the tree of numbers in Remark 9.3. Linear forms in general can define both diagonal and horizontal lines, as well as more complex patterns such as triangles and slices. But there is a crucial difference. In order to produce a diagonal line, only one coordinate needs to be incremented, using a period $(0, i)$ or $(i, 0)$ with $i \neq 0$, but producing a horizontal line requires a simultaneous increment $(1, 1)$. This coordination is typically beyond the recognizing capacity of finite state machines. On the other hand, finite state machines are capable of performing counting tasks such as keeping track of cycles in the numbers of $\mathtt{a}$ (or of $\mathtt{b}$) read. This parity check can define quantifiers like ‘an even number of’ which were beyond $\mathsf{MFO}(\#)$. The geometric meaning of these cycles shows in automorphisms between tree positions accepted by the quantifier whose precise nature is explained in the proof of the following result, which is our main offering in this section.

Theorem 9.19. *The binary quantifiers recognized by finite semantic automata are precisely those whose associated arithmetical definitions are semi-rectilinear.*

This theorem follows from results of Kanazawa [66] (see also [34]), but for completeness we offer a full proof in Appendix D. Needless to say, this is just the beginning of a study of counting procedures and their relation to semantic meanings, as a natural complement to the logic and counting entanglements studied in this paper.

§10. Cognitive questions.

We encountered in the previous section some examples of interleaving logic and counting in natural language. This entanglement is very much on display in psychology and neuroscience as well. As pointed out by Carey [23], children first learn explicit numerical terms as examples of quantifiers, and work such as [7] has shown a strong

https://doi.org/10.1017/bsl.2023.30 Published online by Cambridge University Press

correlation in development between comprehension of number terms and comprehension of (logical) quantifiers.[6] Early learning about basic logical and numerical constructs is evidently intertwined, and as we have argued this continues even through more mature 'grassroots mathematics' and ordinary reasoning practices.

But how, more specifically, might the logical systems we have studied here relate to cognition? The fundamental primitives we have assumed in all of our logical systems are numerical comparisons such as $\#\varphi \succ \#\psi$ or $\#\varphi \approx \#\psi$. The ability to make such comparisons is present across a wide range of species, and appears to be available in human infants from birth (see [29, 39]). Unsurprisingly, 'more' emerges as one of the first quantificational phrases children learn, alongside plurals and 'a'/'some' [23]. There is also evidence for basic operations like addition and subtraction in preverbal infants [39], and in adults, researchers have even uncovered distinct brain areas for encoding addition and for making numerical comparisons [29]. This all raises the question of how, computationally speaking, numerical comparisons are made.

A prominent theme throughout the empirical literature is the distinction between reasoning about *individuals* and their properties, and reasoning about *collections* or *ensembles* and their properties. To solve a concrete task such as determining whether there are more As than Bs there are at least three conceivable families of strategies:

(1) Match each B one-to-one with an A and check whether there are any As left over.
(2) Explicitly count the numbers $\#A$ and $\#B$ and compare those numbers.
(3) Perceptually approximate $\#A$ and $\#B$ and compare those approximations.

(1) and (2) both require enumerating through the relevant objects in an explicit way—much like the semantic automata discussed in the previous section—while (3) bypasses any explicit enumeration or counting procedure, relying instead on fast, parallel perceptual processing (such as when we visually estimate the number of balls in a bin). Such an *approximate number system* (ANS) is in fact ubiquitous and phylogenetically ancient [29].

Much experimental work has gone into distinguishing hypotheses like these in specific instances [23]. A striking example investigates the psychological representation of quantifier expressions in natural language [72, 73, 81, 102]. Consider, for instance, verifying a sentence like 'Most of the dots are blue' (see Figure 7). Any of these strategies, (1), (2), or (3), could in

[6] The psychologist Piaget famously argued that children's understanding of number was built out of logical primitives (thus, another version of 'logicism'). Subsequent research has revealed a more subtle entanglement. with numerical primitives arising much earlier. See [23, 29] for discussion.

https://doi.org/10.1017/bsl.2023.30 Published online by Cambridge University Press

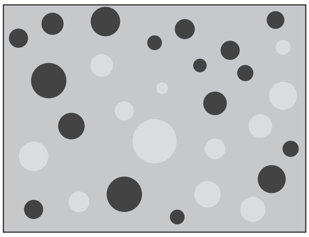


FIGURE 7. A display of dots, where experimental participants might be asked to determine whether, 'Most of the dots are blue' or 'There are more blue dots than yellow dots' (see, e.g., [72, 102]) (In the color version of this figure, the dark dots are blue and the lightly shaded dots are yellow.).

principle be used, where A is 'blue dots' and B is something like 'non-blue dots' (though see [81]). Pietroski et al. [102] present convincing evidence that people in fact employ a strategy more like (3), with the counts $\#A$ and $\#B$ likely determined by the ANS. Queries involving 'more' can also invoke the ANS, though the method people use appears distinct from that for 'most' [72]. In further work, Knowlton et al. [73] show that different English expressions for *universal* quantification in fact elicit different representations altogether: while 'all' and 'every' prompt representations of ensembles and their cardinalities, 'each' seems to elicit an individual-level procedural strategy more like semantic automata.

Relating these tasks to our logical systems, consider a first-order term $\#_x\varphi$.[7] We think of φ as describing the constraints that determine what is to be counted. The availability of any of these strategies, (1), (2), or (3), depends on the extent to which the mind can 'filter' by φ.[8] [9] For instance, successful application of the approximate number system (3) depends on specific perceptual qualities such as spatial or temporal contiguity [29], while application of (1) depends on how easy it is to match pairs one-to-one without repetition.

A logical property that is distinctive of our monadic first-order system MFO(#) and its extensions is that we allow a kind of 'quantifying in' to terms like $\#_x\varphi$ (recall, e.g., Figure 1). Consider a query such as

$$\text{'There are at least 2 more blue dots than yellow dots'}, \qquad (10.1)$$

[7] Recall that, given (SUB), we need only consider subformulas in φ that mention x.

[8] As a special case, there has been interest in understanding which organisms can reason with the number *zero* (i.e., $\#_x x \neq x$). Recent work suggests that this is within range for crows [70].

[9] Relevant is also the amount of memory required in principle to implement each of these strategies. For instance, it is possible to implement (1) with less memory than (2), provided A and B are represented as lists. Thanks to an anonymous reviewer for calling attention to this dimension.

https://doi.org/10.1017/bsl.2023.30 Published online by Cambridge University Press

i.e., $\#B \gtrsim \#Y + 2$. In $\mathsf{MFO}(\#)$ this is encoded naturally as

$$\exists y_1, y_2. y_1 \neq y_2 \wedge B(y_1) \wedge B(y_2) \wedge \#_x(B(x) \wedge x \neq y_1 \wedge x \neq y_2) \gtrsim \#_x Y(x),$$

whereby we ‘remove’ two blue dots and then compare. Perhaps even more natural is the second-order version in $\mathsf{MSO}(\#)$ (with appropriate abbreviations as introduced earlier):

$$\exists Z. |Z| \approx \mathbf{2} \wedge Z \subseteq B \wedge \#_x\big(B(x) \wedge \neg Z(x)\big) \gtrsim \#_x Y(x). \tag{10.2}$$

This essentially asks us to locate a subset of two blue dots and *subtract* those from the total number of blue dots before comparing. This type of predicate subtraction is consistent with observed patterns (e.g., [81]), and while (33) does not yet specify a precise procedure, it seems an interesting question whether verification of sentences like (32) would induce representations anything like (33). Exceptive phrases, such as ‘No one dared attempt the bonus question, except for a few of the best students’, also seem to call for a means of ‘removing’ subparts of a predicate (see, e.g., [100, Chapter 8]).

Moving beyond $\mathsf{MFO}(\#)$ and $\mathsf{MSO}(\#)$, what evidence is there for fundamental numerical representations involving polyadicity or multiplication? Of course, our running example of ‘many’ (like its antonym ‘few’) is exceedingly common, also appearing early in development, though there is still significant debate about how these expressions should be analyzed [111],[10] and how closely they should be unified with their *mass* counterparts like ‘much’ and ‘little’ ([113]; cf. our discussion in Section 8.3).

More direct evidence about polyadicity and multiplication comes from the surprising finding that 11-month infants can already compare proportions, for instance, preferring a ratio of 50/100 to one of 100/500 [31]. Such phenomena appear consistent with a representation involving counts of pairs, perhaps like our $\mathsf{MFO}(\sharp)$, though it has also been suggested that the ANS can directly represent and compare rational numbers (see [25]), which might look more like the probabilistic interpretation of our #-terms described in Section 8.2. Teasing apart these different possibilities presents an exciting opportunity to interface between experimental inquiry and more theoretical explorations.

As one last example of contacts between empirical cognitive science and the themes of this paper, let us return once again to the Pigeonhole Principle. In an experimental study of patterns resembling our opening example, repeated here:

Premise: 20 farmers own at most 15 cows each.
Conclusion: At least 2 farmers own the exact same number of cows.

[10] For example, it has been suggested, based on examples like, ‘his sins were many; his virtues were few’ [60], that ‘many’ should be understood grammatically not as a quantifier at all, but as an adjectival modifier.

https://doi.org/10.1017/bsl.2023.30 Published online by Cambridge University Press

Mercier et al. [87] found at most 30% of participants realized that the conclusion definitely follows. The proposed explanation for this is that, to apply the Pigeonhole Principle we need to construe the numbers less than 15 as themselves forming categories, viz. 'the property of having exactly k cows' for $k \leq 15$. Thus, while each instance of the first-order encoding (2.6) of the Pigeonhole Principle may be clearly valid, realizing that the P_i need to stand for these numerical predicates requires a further step of interpretation.

Although the relational encodings of the Pigeonhole Principle—(7.1) and its modal variant (7.2)—enjoy an elegant generality lacking in the monadic formulation, the interpretive step from stimulus to representation is even more formidable here. The relation Rxy, meaning 'x has y-many cows', is not one that most people are accustomed to thinking about. The premise of (7.2) then becomes something like, 'there are more cow-owners than numbers-of-cows-owned', which again may not come so naturally or immediately to people.

It is but a short way from reasoning puzzles and 'grassroots mathematics' to even more subtle and abstract applications of such principles in more advanced topics. The Pigeonhole principle itself manifests throughout mathematics, often in surprising ways. For instance, it is used in a simple proof of the Erdős–Szekeres theorem in graph theory [118], and the infinitary version of the principle (recall Equation (3.1)) for the case of $k = 2$ appears in proofs of the well-known Bolzano–Weierstrass theorem.[11] While the principle itself is straightforward enough, just as in the experiments by Mercier et al. [87], the difficulty is often in choosing the relevant predicates so as to see that it applies in the first place.

Once we turn to infinitary patterns in logic and counting, a whole additional array of cognitive questions arise. Chief among these is the question of how our initial conceptions of numbers and counting can be extended to accommodate basic infinitary reasoning.

Some researchers have suggested that the individual developmental stages in mastering the modern concept of infinity actually mirror the *historical* development of the concept (see [88], echoing a broader theme familiar from [101]). From Galileo's bewilderment that infinite sets could be matched one-to-one with their proper subsets (and thus that, in our terminology, $\mathsf{s} = \mathsf{s} + 1$ could be satisfiable), to Bolzano's explicit introduction of *infinity* as a potential feature of any set that we can describe (thus giving clear meaning to our notation $\#\varphi$ when the φs are unbounded), and eventually to 'Cantor's paradise', children undergo a surprisingly similar sequence of transitions [88]. It is intriguing to consider whether any of the systems studied here might correspond to intermediate 'way-stations' in this development, capturing

[11] In short: dividing an interval containing an infinite sequence into two subintervals will guarantee infinitely many points inside at least one of these subintervals.

https://doi.org/10.1017/bsl.2023.30 Published online by Cambridge University Press

only a suitably restricted range of more intuitive infinitary patterns. Because our monadic and modal systems involve at most addition and multiplication, the infinitary patterns in these systems are less complex than their finitary counterparts. Whether this type of logical complexity could be brought to match intuitive cognitive complexity is worth investigating further.

This concludes our brief tour of just a few salient points of contact with empirical issues in the cognitive sciences. A deeper foray into such contacts would undoubtedly reveal many further connections and opportunities.

§11. Conclusion. This paper has presented a number of contributions to studying the interplay of *logic and counting*, viewed as a basic phenomenon in human reasoning in its own right. In fact, we encountered three perspectives on what it means to combine logic and counting. The main perspective adopted here is one of consilience and synergistic co-existence. As a complement to the related bodies of research in the theory of generalized quantifiers [9, 100] and in computational logic [98, 116], we explored a hierarchy of progressively richer formal systems exemplifying this perspective (summarized in Table 1). A common theme running through all of these systems is the separation between logical reasoning patterns needed to derive meaningful *normal forms*, and the varieties of numerical reasoning suggested by those normal forms. The latter spanned from (fragments of) additive arithmetic to Diophantine inequalities and full elementary arithmetic, also encompassing basic counting along binary relations. In each case infinitary reasoning could be cleanly separated and, at least for the systems we considered here, revealed as a simplified version of the corresponding finitary patterns. Finally, we probed natural generalizations of these systems, obtained either by broadening the possible interpretations of numerical terms or by relaxing the logical semantics.

Parallel to this formal development, we explored entanglements between logic and counting in natural language and thought. Quantifier vocabulary alone provides a kind of microcosm illustrating many of our broader motifs, with rich logical, linguistic, psychological, and computational dimensions, all highlighting novel mixtures of logic and counting. We also touched on ontogenetically and phylogenetically more basic examples of ‘number sense’, in addition to more sophisticated reasoning patterns on the cusp of mature mathematics, the famous Pigeonhole Principle being a paradigmatic instance.

Throughout these explorations the individual contributions of logic and of counting, while often still distinctly identifiable, nonetheless resist disentanglement. Take a system like $\mathsf{MFO}(\#)$, the starting point of our analysis. The count term $\#_x\varphi$ is assumed to denote a cardinal number, but *under a logical description* specified by φ. Meanwhile, a characteristically quantitative principle—permutation invariance—begets qualitative principles in the logical language such as (INV) and (SUB), which in turn allow for derivation of explicitly numerical normal forms that support familiar numerical algorithms. As Hilbert [57] once put it, 'a partly simultaneous development of the laws of logic and arithmetic is requisite' (p. 347).

Similar patterns permeate our discussions around extensions of $\mathsf{MFO}(\#)$, and of the various empirical phenomena in language and cognition. Monotonicity inference, to take a typical example, operates at a level that abstracts away from logical or arithmetical details, for instance treating number lines and predicate hierarchies on a par.

The other two perspectives on logic and counting—less emphasized in the present treatment but historically at least as prominent—reflect an aspiration toward methodological purity. We briefly considered how much of logic could be extracted from 'pure' counting. As we saw, classical logic emerges from remarkably austere numerical primitives, and non-classical systems can also be elicited. For instance, in place of the 'true' universal quantifier $\neg\exists x.\neg\varphi$ we could entertain variants like $\#_x\varphi \approx \top \wedge \#_x\varphi \succ \#_x\neg\varphi$, which states that *almost all* objects satisfy φ, except for a few that 'do not count'. In the other direction we considered some of the counting principles already implicit in (first-order) logical systems. The recurrent theme of *counting in the syntax* is typical in this connection (developed further in Appendix E).

Even with the above exploration in place, the three angles on logic and counting pursued in this paper do not exhaust the rich and ubiquitous entanglement of logic and counting. To mention just one more instance, there are also natural and illuminating *computational* perspectives. We briefly explored one of these, in the form of a procedural semantics for logical expressions afforded by *semantic automata* (Section 9.5), that allow us to calibrate the counting content of meanings for quantifier expressions. But also more globally, we can measure the numerical content of an entire logical system in terms of the *computational complexity* of its satisfiability problem. Indeed, there is a precise sense in which any NP-hard logical system—for instance, ordinary propositional logic—can be said to solve arbitrary integer programs, via a simple (viz. polynomial) SAT reduction. In a similar vein, any Σ^1_1-hard system—even one that is not overtly quantitative such as first-order dynamic logic [53]—implicitly answers arbitrary arithmetical queries. This angle affords a relatively coarse-grained means of calibrating logical and numerical reasoning, and we have even seen in the present article how it would collapse expressively and intuitively distinct systems (e.g., $\mathsf{MFO}(\#)$ and $\mathsf{MSO}(\#)$). But entanglements via computational complexity can go even deeper, as seen in the methods of *proof complexity* where logical encodings of numerical principles like Pigeonhole take center stage [26, 75]. Research programs like this only reinforce the view of consilience and co-existence as a natural habitat.

https://doi.org/10.1017/bsl.2023.30 Published online by Cambridge University Press

In closing, it is important to acknowledge that reductive aspirations and methodological purity often originate from motivations that are not themselves logical or mathematical. The program of logicism, for instance, has been concerned with philosophical puzzles about the epistemology and metaphysics of 'number' (e.g., [50]). Measurement theorists, meanwhile, have maintained that only 'qualitative (that is, nonnumerical) empirical laws' have objective significance, with numerical representations merely 'a matter of convention', chosen for 'computational convenience' [76, pp. 12–13]. Whatever one's stance on these and other philosophical and methodological issues, we hope to have shown that the important borders and thresholds in understanding reasoning are not those between qualitative and quantitative reasoning, but between simple and complex combinations of logic and counting. Whatever we might lose in foundational purity by pursuing this path, we may gain a better understanding of human reasoning abilities in return.

Appendices. In these appendices we present some additional material that broadens the context for the main results of this paper. Appendix A is a survey of relevant literature. Appendices B–D present the details on some results mentioned in the main text, concerning infinity quantifiers and monadic second-order logic, infinitary addition and multiplication, and semantic automata, respectively. Finally, Appendix E highlights an intriguing interface of logic and counting that we have largely ignored in this paper, namely, the historical tradition of results on the entanglement of the very syntax of logical systems and systems of arithmetic.

A. Related work on logic and counting. As we have mentioned, there is a vast amount of important research on mixtures of logic and counting. Here we discuss logical systems in the literature that bear a close relationship to the hierarchy of systems studied here (summarized in Tables 1 and 2).

A.1. Logics with generalized quantifiers. An expansive literature has explored adding generalized quantifiers to first-order logic (as well as other languages, including monadic first-order logic). The system $\mathsf{FO}(\#)$ has been studied explicitly in that literature [2, 56, 100], and of course it is closely related to both the Härtig quantifier, $\#_x\varphi \approx \#_x\psi$, and the strict version $\#_x\varphi \succ \#_x\psi$ introduced explicitly by Lindström [82]. Earlier, Rescher [109] had considered a unary version, namely, $\#_x\varphi \succ \#_x\neg\varphi$.

Work on the *monadic* fragment of FO with generalized quantifiers dates back at least to Slomson [120], who studied the Chang quantifier, $\#_x\varphi \approx \#_x\top$, in this context. We refer to [100] for many other results and references in the area related to these particular generalized quantifiers, both for FO and for MFO.

https://doi.org/10.1017/bsl.2023.30 Published online by Cambridge University Press

A.2. Computational logic. Perhaps the largest body of work related to our systems comes from computation logic. A significant strand focuses on *extensions* of $\mathsf{FO}(\#)$ and even of $\mathsf{FO}(\sharp)$, but interpreted over finite models (e.g., [22, 48, 77, 116], among many others). As discussed in Remark 2.8, much is known about *finite variable* fragments with counting quantifiers as well, though here most of the results are negative [46, 69, 98]. Back-and-forth games, similar in spirit to our #-bisimulations (Definition 7.8), have also been explored (see, e.g., [22, 98]).

A.3. Syllogistic and propositional counting logic. A number of weak fragment of $\mathsf{MFO}(\#)$ and even of $\mathsf{PL}(\#)$ have been studied as *extended syllogistic systems*. For example, a whole series of papers charts the territory of small systems including 'more than', 'most', 'at least k', and related operators [36, 71, 78, 91, 92, 104, 105]. Pratt-Hartmann [104] in particular explores $\mathsf{FO}(\#)$ with *one* free variable, which is seen to be decidable. He also notes a natural probabilistic interpretation of the system. Locating precisely where these systems fit inside of our logics would be worthwhile. Notably, many of them enjoy quite low complexity.

Recent work by Ding et al. [32] essentially deals with what we call $\mathsf{PL}(\#)$, interpreted over (possibly) infinite models. As highlighted in Table 2, the main difference between $\mathsf{PL}(\#)$ and sentences in $\mathsf{MFO}(\#)$ is the ability of the latter to express inequalities with numerical bounds. An important instance is $\mathsf{s} \geq \mathsf{s} + 1$, showing that $\mathsf{MFO}(\#)$, unlike $\mathsf{PL}(\#)$, can characterize the infinite predicates. However, the higher expressive power of numerical bounds also marks an important distinction in the valid principles. For instance, the main principle in one of the axiomatizations from [32] employs a type of *polarization rule* [21, 74]. Adapted to our setting, provided the predicate P occurs nowhere in φ or ψ, the rule would say:

$$\text{From } \#_x\big(\varphi \wedge P(x)\big) \approx \#_x\big(\varphi \wedge \neg P(x)\big) \to \psi, \text{ infer } \psi. \quad (\textit{Polarization})$$

Polarization is not admissible even in our basic system $\mathsf{MFO}(\#)$. It implies, amongst other things, that consistent formulas can also be made true while duplicating the size of all regions. This is true for sets of inequalities without numerical bounds, but not for the ones expressible in $\mathsf{MFO}(\#)$. As discussed in Section 3.4, it remains to be seen whether a more intricate polarization rule for $\mathsf{MFO}(\#)$ would support a 'purely logical' axiomatization.

A.4. Probability logic. We mentioned a connection with probability logic in Section 8.2, namely, the systems $\mathsf{PL}^{\phi}(\#)$, $\mathsf{ML}^{\phi}(\#)$, $\mathsf{MFO}^{\phi}(\#)$, and $\mathsf{MSO}^{\phi}(\#)$ can all be interpreted probabilistically without any further ado, viz. proportionality. Under that interpretation, $\mathsf{PL}^{\phi}(\#)$ is indistinguishable from the propositional probability logic considered in [59], which is equivalent to the system studied earlier by Gärdenfors [43], provided the latter is restricted to *regular* probability measures, i.e., those assigning all

https://doi.org/10.1017/bsl.2023.30 Published online by Cambridge University Press

non-empty sets strictly positive probability. $\mathsf{MSO}^{\phi}(\#)$ is easily seen to be equally expressive as the probability logic with linear inequalities studied by Fagin et al. [37], again under the assumption of regularity. For discussion of regularity in probability logic, see [33].

A very strong probability logic was studied in [4] and [52], allowing inequalities between sums and products of terms $\pi_{\mathbf{x}}\varphi$ (cf. Section 6). While our polyadic terms $\sharp_{\mathbf{x}}\varphi$ in $\mathcal{L}^1_\sharp$ and $\mathcal{L}^2_\sharp$ are interpreted as cardinalities of Cartesian products, these terms $\pi_{\mathbf{x}}\varphi$ are interpreted directly as products of probabilities, which in general leads to a different set of principles (cf. Example 8.1). Quantifiers over term variables are also allowed. Unsurprisingly, these languages are highly undecidable, although decidable fragments can be found, e.g., by allowing only monadic predicates and eliminating variable equality [52].

A.5. Graded modal logic. In the areas of modal and description logics, a number of authors (since [41]) have considered graded modal logics involving unary modalities like $\Diamond^{\geq k}$. We mentioned that $\mathsf{ML}(\#)$ cannot express these modalities (Corollary 7.7), but of course the reverse is also true: the binary modality $\succsim$ is beyond the expressive capacity of graded modal logic. A broad study, with connections to generalized quantifiers, appears in [58]. More recently, some researchers have probed the precise counting capacity of such systems, employing notions of count-bisimulations as well (see, e.g., [3]). There has also been study of related logical systems that are expressively equivalent to, but more complex than, graded modal logic [10], as well as natural expressive extensions that remain of relatively low complexity [30]. Emerging connections between graded modal logic and classes of *graph neural networks* [6] promise yet further dimensions to our subject.

§B. The infinity quantifier and monadic second-order logic. Let MFO^∞ be monadic first-order logic with an infinity quantifier (simply the language $\mathcal{L}^1_\#$ without #-formulas but with $\exists^\infty$ added), and let WMSO be weak monadic second-order logic (quantification only over finite sets). It turns out MFO^∞ and WMSO are expressively equivalent. A version of this result without equality is due to Väänänen [126], and here we describe the result with equality. To translate MFO^∞ into WMSO the only interesting case is $(\exists^\infty y.\varphi)^* = \forall X.\exists y.\big(\neg X(y) \wedge (\varphi)^*\big)$. In the other direction, MFO^∞ possesses a normal form result [24, Theorem 3.15] whereby every sentence is equivalent to a disjunction of existentially quantified formulas of the form:

$$\begin{aligned}\mathsf{diff}(\mathbf{x}) \wedge \bigwedge \tau(x_i) \wedge \forall z.\Big(\mathsf{diff}(\mathbf{x}, z) \to \bigvee \sigma(z)\Big)\\ \wedge \bigwedge \exists^\infty y.\rho(y) \wedge \forall^\infty y. \bigvee \upsilon(y).\end{aligned}$$

Supposing that X is one of our monadic predicates, assuming it can only take on finite sets as values, the above is equivalent to one of the forms:

$$\alpha(\mathbf{x}) \wedge \forall z.\big(\mathsf{diff}(\mathbf{x}, z) \rightarrow (X(z) \rightarrow \psi(z)) \wedge (\neg X(z) \rightarrow \chi(z))\big)$$
$$\wedge \bigwedge \exists^{\infty} y.(\neg X(z) \wedge \rho(y)) \wedge \forall^{\infty} y.(\neg X(y) \rightarrow \varphi(y)).$$

Because $\exists X$ commutes with $\exists \mathbf{y}$ and disjunction, we need only consider what happens when appending $\exists X$ to this formula. This is evidently equivalent to another formula with no occurrences of X at all:

$$\alpha'(\mathbf{x}) \wedge \forall z.\big(\mathsf{diff}(\mathbf{x}, z) \rightarrow (\psi(x) \vee \chi(x))\big)$$
$$\wedge \bigwedge \exists^{\infty} y.(\rho(y) \wedge \chi(y)) \wedge \forall^{\infty} y.(\varphi(y) \wedge \chi(y)).$$

This concludes the argument for the other direction.

§C. Cardinal arithmetic: quantifier elimination and separation. Consider the elementary theory of the structure $\mathcal{C} = \langle C_{\aleph_\omega}; +\rangle$, that is, the first-order theory of addition on cardinal numbers less than $\aleph_\omega$. As in ordinary Presburger Arithmetic, $\{0\}$, s, $\equiv_n$ and $>$ are all definable in this structure, where s is the function that takes a cardinal number to the next largest cardinal number, and $\equiv_n$ is congruence mod n, for $1 < n < \omega$. Note that $\{\aleph_0\}$ is also definable. Assume we have all of these constants, functions, and relations in the signature, so we are considering $\mathcal{C}^+ = \langle C_{\aleph_\omega}; 0, \aleph_0, s, \{\equiv_n\}_{1<n<\omega}, >, +\rangle$.

We first derive a normal form for the quantifier-free fragment. By propositional reasoning we assume a disjunction of conjunctions of atomic formulas:

$$\mathsf{t} = \mathsf{u},$$
$$\mathsf{t} \equiv_m \mathsf{u},$$
$$\mathsf{t} > \mathsf{u},$$

and also by propositional reasoning we can assume that every disjunct includes a conjunct $x < \aleph_0$ or $x \geq \aleph_0$, for every variable x appearing in the disjunct. This allows us to separate the atomic formulas into those involving 'finite' terms and those involving 'infinite' terms: the successor function of course takes (in)finite to (in)finite cardinals, and infinite terms absorb finite terms in sums. Furthermore, if either t or u contains an infinite term, then we can assume without loss that both t and u contain only infinite terms, since otherwise all three types of atomic formulas trivialize. In other words, we have obtained a normal form characterized by disjunctions of conjunctions which include statements about which variables are finite/infinite, a set of statements describing the finite terms, and a set of statements describing the infinite terms.

The finite component can, as usual, be further regimented so that the three types of atomic statements involve sums of terms of the form $s^k(0)$ and $s^k(x)$ for $k \geq 0$ and x a variable. This is because of the law $s(x + y) = x + s(y)$. As usual, models of these conjunctions are effectively solutions to linear programs.

For the infinite component, successor in fact distributes over addition, that is, $s(x + y) = s(x) + s(y)$, which allows a similar regimentation. More regimentation is possible. First note that $\equiv_n$ can everywhere be replaced by $=$. But we can also eliminate all sums. For instance, $\mathsf{t} = \mathsf{u} + \mathsf{v}$ is equivalent to the disjunction $(\mathsf{t} = \mathsf{u} \wedge \mathsf{u} \geq \mathsf{v}) \vee (\mathsf{t} = \mathsf{v} \wedge \mathsf{v} > \mathsf{u})$. The same reduction works for strict inequalities.

Thus, the component describing the infinite terms simply contains conjuncts of the form $x = s^k(y)$, $x > s^k(y)$, $x = \aleph_k$, and $x > \aleph_k$, for $k \geq 0$. There is a trivial isomorphism from $\langle \mathbb{N}; 0, s, > \rangle$ onto $\langle \{\aleph_k\}_{k \in \mathbb{N}}; \aleph_0, s, > \rangle$ sending k to $\aleph_k$. This shows that the definable subsets of infinite cardinals coincides with the definable sets of indices in $\mathbb{N}$, viz. the finite and co-finite sets. This of course also easily establishes the decidability of determining whether a quantifier-free formula in the original language is satisfiable. Summarizing:

PROPOSITION C.1. *Every first-order quantifier-free formula is equivalent over the structure* $\mathcal{C}^+ = \langle C_{\aleph_\omega}; 0, \aleph_0, s, \{\equiv_n\}_{1<n<\omega}, >, + \rangle$ *to a disjunction of conjunctions, specifying*:

(1) which variables in that disjunct are finite or infinite,
(2) for the finite component a description of a linear set, and
(3) for the infinite component a description of a set of infinite cardinals using $0, s, >$ over the aleph-number indices.

COROLLARY C.2. *The quantifier-free theory of* $\mathcal{C}^+$ *is decidable.*

What about the full first-order theory of $\mathcal{C}$? As in ordinary Presburger Arithmetic, this theory does not admit quantifier elimination. But the theory of $\mathcal{C}^+$, in the augmented language, does. Consider a formula $\exists x.\theta$, where θ is in normal form (Proposition C.1), i.e., θ is a conjunction $\delta \wedge \iota \wedge \phi$, where δ is a description of which variables denote infinite sets, ι describes the infinite terms, and ϕ describes the finite terms. In our normal form x does not appear in both ι and ϕ, so $\exists x.\theta$ simplifies to either $\exists x.\iota$ or $\exists x.\phi$, where ι and ϕ are assumed to involve only infinite or finite terms, respectively. In the latter case we can perform the quantifier elimination as usual in additive arithmetic, reducing $\exists x.\phi$ to a quantifier free statement using $0, s, >, +$, and the congruence relations $\equiv_m$.

In the former case we want to show that we can reduce $\exists x.\iota$ to a quantifier-free form using only $\aleph_0$, s, and $>$. In fact, this proceeds exactly as the quantifier elimination procedure for $\langle \mathbb{N}; 0, s, > \rangle$: the isomorphism

https://doi.org/10.1017/bsl.2023.30 Published online by Cambridge University Press

between the latter structure and $\langle\{\aleph_k\}_{k\in\mathbb{N}};\aleph_0,s,>\rangle$ shows they have the same quantificational theory as well.

Having shown quantifier elimination for $\mathcal{C}^+$, this establishes:

THEOREM C.3. *The first-order theory of $\mathcal{C}$ is decidable.*

We now show essentially the same result for full first-order arithmetic over cardinals. That is, let $\langle C_{\aleph_\omega};+,\times\rangle$ be the structure of cardinal numbers less than $\aleph_\omega$ under addition and multiplication. The first-order theory of this structure is of course undecidable, but it is easy to see that this is only due to the substructure $\langle\mathbb{N};+,\times\rangle$. As before, this substructure is definable in the sense that a term t denotes a natural number if and only if $\mathsf{t}+1>\mathsf{t}$. Indeed, by the same argument as above, any formula will be equivalent to a disjunction of conjunctions $\delta\wedge\iota\wedge\phi$, where δ specifies which terms are (in)finite, ι involves the infinite terms, and ϕ the finite terms.

The ϕ component will be an arbitrary arithmetical formula, where quantifier elimination of course fails. But the ι component does allow for quantifier elimination. That is, we can consider the elementary theory of $\langle\{\aleph_k\}_{k\in\mathbb{N}};+,\times\rangle$. The crucial step is the same as in the purely additive case: every equality statement $\mathsf{t}=\mathsf{u}+\mathsf{v}$ or $\mathsf{t}=\mathsf{u}\times\mathsf{v}$ is equivalent over this structure to the disjunction $(\mathsf{t}=\mathsf{u}\wedge\mathsf{u}\geq\mathsf{v})\vee(\mathsf{t}=\mathsf{v}\wedge\mathsf{v}>\mathsf{u})$ (and similarly for strict inequalities between complex terms), implying that we can systematically eliminate both addition and multiplication. Thus, quantifier elimination for the language augmented with constant $\aleph_0$ and successor s again follows from the fact that $\langle\mathbb{N};0,s,>\rangle$ admits it.

THEOREM C.4. *Every formula in the language of first-order arithmetic is equivalent over $\langle C_{\aleph_\omega};+,\times\rangle$ to a disjunction of conjunctions involving a finite and an infinite component. Moreover, the set of 'infinitary formulas' (all of whose terms are declared infinite) possesses quantifier elimination and they define precisely the same relations over cardinals as the pure language of equality and strict inequality.*

§D. Finite automata and quantifier recognition procedures. *Finite automata* are particularly simple counting devices, and in what follows, we will determine what binary logical quantifiers this device can recognize. We first recall the main definitions and statement of the result from Section 9.5. Linear sets are the solutions to equations

$$\begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} b_1 + i_{1,1}\mathsf{x}_1 + \cdots + i_{1,m}\mathsf{x}_m \\ b_2 + i_{2,1}\mathsf{x}_1 + \cdots + i_{2,m}\mathsf{x}_m \end{pmatrix}, \tag{D.1}$$

while rectilinear sets are those in which $i_{1,k}=0$ or $i_{2,k}=0$ (or both) for all all $k\leq m$. Finally, a set is *semi-rectilinear* if it is a finite union of rectilinear sets. For the purpose of this appendix we will notate the linear sets by

https://doi.org/10.1017/bsl.2023.30 Published online by Cambridge University Press

$(v_1, v_2) + \mathsf{x}_1.(i_{1,1}, i_{2,1}) + \cdots + \mathsf{x}_m.(i_{1,m}, i_{2,m})$. So in this notation rectilinear sets can be seen as defined by forms $(v_1, v_2) + \mathsf{x}_k.(i_k, 0) + \cdots + \mathsf{x}_j.(0, i_j)$, which first lists the periods of type $(i, 0)$ and then those of type $(0, i)$. Our result states:

Theorem D.1. *The following are equivalent for permutation-closed languages L :*

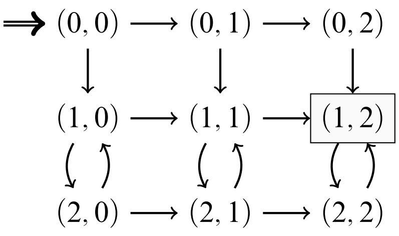
(a) *L is regular.*
(b) *The set of occurrence vectors for strings in L is semi-rectilinear.*

Proof. The idea of the proof is to associate semi-rectilinear forms with finite automata. In showing how this works, we shall be using geometrical representations in a number of places which are like the tree of numbers for generalized quantifiers (Section 9.1), except for a rotation to the grid $\mathbb{N} \times \mathbb{N}$ which fits our purposes better. In fact, the terminology 'rectilinear' was motivated by shapes in this grid. Also, we shall be using several well-known useful properties of finite automata, such as the closure under unions of the languages recognized, the fact that nondeterministic finite automata have the same recognizing power as deterministic ones, or the fact that the recognizing power of deterministic finite automata is not changed when we allow 0, 1, or more transitions for a symbol read in some states. ⊣

From (b) to (a). It suffices to show the implication for rectilinear forms, since the permutation-closed regular languages are closed under taking unions.

There are a few special cases here that are easily shown to be regular, namely, a single vector (v_1, v_2), or such a vector plus one period $(i, 0)$ or $(0, i)$ with $i \neq 0$. Before starting the main proof, here is a warm-up example.

Example D.2. The rectilinear form $(1, 2) + \mathsf{x}.(2, 0)$ matches the permutation-closed regular language of strings with an odd number of symbols a and two occurrences of symbol b. The following finite automaton recognizes just these strings.



Horizontal arrows are for b-moves, vertical arrows for a-moves, rightmost states allow no b-moves, the starting state is $(0, 0)$, and the only accepting

https://doi.org/10.1017/bsl.2023.30 Published online by Cambridge University Press

state is $(1, 2)$. Here are two illustrations. (a) It is easy to see that a state (i, j) can only be visited after having seen j occurrences of b plus a number of occurrences for a that equals i plus some multiple of 2 (reflecting the available cyclic detours). (b) A correct string such as $\mathtt{a}^5\mathtt{ba}^3\mathtt{ba}^5$ can be recognized by first cycling through $(1, 0)$ and $(2, 0)$ ending in $(1, 0)$, then moving to $(1, 1)$, then cycling through $(1, 1)$ and $(2, 1)$ ending in $(2, 1)$, then moving to $(2, 2)$, and finally cycling through $(2, 2)$ and $(1, 2)$ ending in $(1, 2)$. The general principle should be clear. Taken together, (a) and (b) show that the automaton recognizes the given language.

Incidentally, the automaton is not unique. The preceding reasoning would yield the same conclusion if we had allowed cycling between the top and middle layers of the state transition diagram.

Next, consider a general rectilinear form

$$F = (v_1, v_2) + \mathsf{x}_k.(i_k, 0) + \cdots + \mathsf{x}_j.(0, i_j).$$

Let N_1 be the sum of v_1 plus the maximum of all numbers i_k occurring to the left in periods of F, while N_2 is defined likewise using the right-hand side of the pairs occurring in F. Now we define a non-deterministic partial finite automaton $\mathcal{S}$:

- States are all pairs (u, v) with $u \leq N_1, v \leq N_2$.
- The only recognizing state is (v_1, v_2).
- The transition function is defined as follows, with two types of moves:
  I. from (x, y) via reading a to $(x + 1, y)$, if $(x + 1, y)$ is a state,and analogously for reading b,
  II. an a-move from state $(x + i - 1, y)$ to (x, y), if the period $(i, 0)$ occurs in F. Likewise for b-moves and periods $(0, j)$.

We say that an automaton $\mathcal{S}$ is *permutation invariant* if, whenever reading a string X can drive $\mathcal{S}$ from state S to state T, any permuted version of X can also drive $\mathcal{S}$ from state S to state T. The following can be shown by direct inspection of the above-defined transitions.

FACT D.3. The automaton $\mathcal{S}$ is permutation invariant.

PROOF. It suffices to show that a and b transitions can be interchanged at an input state without changing the output state. This is easily established by considering the various combinations of Type I. transitions and Type II. transitions. ⊣

LEMMA D.4. *The following assertions are equivalent*:

(i) *String X is recognized by the above-defined automaton $\mathcal{S}$.*
(ii) *The occurrence numbers for* a, b *in X are in the set defined by the rectilinear form F.*

PROOF. *From* (i) *to* (ii). Suppose that a string X drives $\mathcal{S}$ from the starting state to the accepting state (v_1, v_2). We prove the following stronger invariance statement by induction on the length of finite strings:

CLAIM. *If string X drives $\mathcal{S}$ to state (x, y), then the occurrence numbers in X are generated by (x, y) plus a (possibly empty) finite sum of periods occurring in the rectilinear form F.*

PROOF OF CLAIM. The claim is clear for the empty string at the starting state $(0, 0)$. (Here we use the fact that our automaton $\mathcal{S}$ as defined above has no ε-moves except the identity.)

The inductive step is by inspecting possible transitions. We discuss $\mathtt{a}$-moves only, $\mathtt{b}$-moves are similar. (a) Suppose that $X\mathtt{a}$ drives the initial state of $\mathcal{S}$ to (x, y), and then moves to $(x + 1, y)$ by reading the final $\mathtt{a}$. By the inductive hypothesis about X, the occurrence numbers match the stated description at the state (x, y). But then the occurrence numbers for $X\mathtt{a}$ satisfy that same description with respect to $(x + 1, y)$. (b) Now suppose that $X\mathtt{a}$ first reaches $(x + i - 1, y)$ in $\mathcal{S}$, and then moves to (x, y) by reading the final $\mathtt{a}$. By the inductive hypothesis, the occurrence numbers in X match the stated description at $(x + i - 1, y)$. But then, since by the definition of $\mathcal{S}$ there is a period $(i, 0)$ in F allowing a cyclic move, the occurrence numbers for $X\mathtt{a}$ satisfy the stated description at the state (x, y). ⊣

In particular, once the accepting state is reached, the string must have a pair of occurrence numbers in the given rectilinear set.

From (ii) to (i). Let string X have occurrence numbers in the given rectilinear set, with particular values for the period variables x. By the permutation-invariance of the automaton $\mathcal{S}$, the string X will be recognized iff the following permuted version is recognized: 'first v_1 symbols $\mathtt{a}$, then v_2 symbols $\mathtt{b}$ (i), then the remaining symbols $\mathtt{a}$ followed by the remaining $\mathtt{b}$ (ii)'. Part (i) of this sequence takes us to the recognizing state (v_1, v_2). The symbols in the final Part (ii) can be discounted by making the appropriate looping moves corresponding to admissible periods, always returning toward (v_1, v_2).

From (a) to (b). Consider any permutation-closed regular language $\mathcal{L}$. First, we produce a suitable automaton to work with in the rest of the proof.

FACT D.5. $\mathcal{L}$ is recognized by a permutation-invariant deterministic finite automaton $\mathcal{S}$.

PROOF. Consider the standard Nerode construction for regular languages, where two strings are called equivalent if they send the same continuations to accepting states. A recognizing deterministic finite automaton for the language has the equivalence classes for its states, and a transition

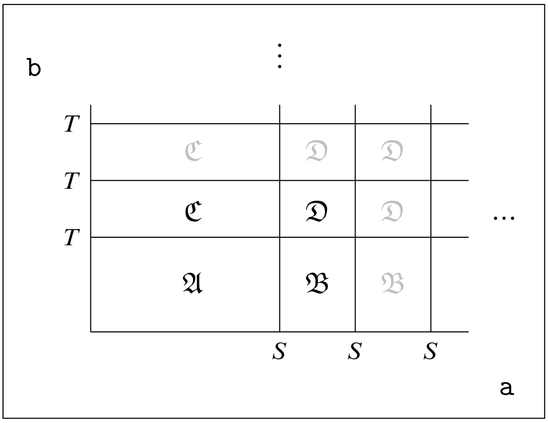
https://doi.org/10.1017/bsl.2023.30 Published online by Cambridge University Press

function plus accepting states defined in an obvious manner. Now, it suffices to note the simple fact that, if the regular language we start with is itself permutation-closed, then the Nerode automaton is permutation-invariant in the earlier sense. ⊣

The permutation invariance allows us to define, for each pair of numbers (i, j), a unique state S_{ij} that $\mathcal{S}$ will reach from its starting state when presented with any string with these occurrence numbers. We call (i, j) accepting iff S_{ij} is. While not strictly necessary for what follows, it is helpful to think of our two structures abstractly as two bimodal relational models: $\mathcal{S}$ and its 'grid unraveling' $\mathbb{N} \times \mathbb{N}$ which carries two commuting functions 'moving one step up' and 'moving one step right'. Then the following connection arises:

FACT D.6. S_{ij} is a modal p-morphism from the grid $\mathbb{N} \times \mathbb{N}$ to the automaton $\mathcal{S}$.

We can therefore consider the grid model $\mathbb{N} \times \mathbb{N}$ as an automaton that is equivalent to $\mathcal{S}$ in an obvious sense, and analyze its geometrical shape.



Explanation of the grid automaton. The two symbols $\mathtt{a}, \mathtt{b}$ represent the functions in this grid model. The state S is the first recurring state as we start reading symbols $\mathtt{a}$ only from the starting state. Each interval from S to S on the bottom row is then the same. And the same is true for their matching intervals on horizontal rows higher up, as these arise from applying the function $\mathtt{b}$ the same number of times to identical states. In particular, the rectangles toward the right in the area $\mathfrak{B}$ are all the same. The same analysis works for the first recurring state T on the left w.r.t. the $\mathfrak{C}$ area. Next, the area $\mathfrak{A}$ can have arbitrary state content, but it is finite, since non-recurring state sequences are bounded in length by the size of the given automaton

$\mathcal{S}$. Finally, the rectangle $\mathfrak{D}$ is very special. All its corner points must be the same (given their origins from the S and the T intervals), and $\mathfrak{D}$ will then repeat to fill the whole remaining quadrant of $\mathbb{N} \times \mathbb{N}$ with identical copies of itself.

Now consider any recognizing state U in $\mathcal{S}$. Its occurrences in the above grid can be described as follows, area by area in the diagram. The typical features of rectilinear forms now emerge. In area $\mathfrak{A}$: a finite disjunction of descriptions of single vectors. In area $\mathfrak{B}$: a finite disjunction of occurrences of U in the first rectangle, plus periods $\mathsf{x}.(k,0)$ where k is the length of the first interval from S to S. For area $\mathfrak{B}$ the enumeration is analogous with a period $\mathsf{x}.(0,l)$ for moving upward. Finally, for area $\mathfrak{D}$, all occurrences of U in its quadrant are described by a finite disjunction of their occurrences in the first generating rectangle while allowing both periods $\mathsf{x}.(k,0)$ and $\mathsf{x}.(0,l)$. In particular, no 'oblique' periods $\mathsf{x}.(i,j)$ (like the period $\mathsf{x}.(1,1)$ used in defining the non-regular quantifier 'most') are needed for this enumeration.

The preceding descriptions, taken disjunctively over all occurrences of accepting states in the grid, show that the permutation-closed language recognized by the given automaton $\mathcal{S}$ has a semi-rectilinear description. $\dashv$

The earlier-mentioned characterization of first-order quantifiers [11] is a special case, where the crucial area $\mathfrak{D}$ collapses to one state whose behavior then extends downward. As for generalizations, the result probably also holds for arbitrary *finite alphabets*, given the affinities of our treatment with the graph-theoretic analysis of permutation-closed regular languages over arbitrary alphabets in [61] (cf. [34]). See in addition Kanazawa [66], who also gives an arithmetical description of the permutation-invariant languages recognized by pushdown automata.

Here are a few questions raised by our results and proof method. In terms of other formats, what is the structure of the special *regular expressions* that describe permutation-invariant finite automata, and what algebraic laws govern their manipulation? Rectilinear forms amount to a flattening of nested iterations to just one level, which is reminiscent of the flattening of nested count terms in the normal forms for $\mathsf{MFO}(\#)$. Also, could the modal perspective in the above proof yield further insights? In particular, the use of the grid $\mathbb{N} \times \mathbb{N}$ might be significant, in that its decoration with a finite set of states is a form of a *tiling*, while modal logics of tiling problems have high complexity. Next, connecting back to our counting logics, another natural question is this. Are the above results reflected in *arithmetical definability* results for finite-state quantifiers, whether in terms of the inequalities in normal forms for $\mathsf{MFO}(\#)$ or directly in the first-order language of Presburger Arithmetic? Finally, our counting logics typically allow for *infinite cardinalities*. Can the above automata analysis be extended to infinite cardinalities, perhaps using Büchi automata for infinite strings?

https://doi.org/10.1017/bsl.2023.30 Published online by Cambridge University Press

§E. Logical syntax and counting. In addition to the mixtures of logic and counting discussed in this paper, here is one more perspective, with a long history. Working with a logical system presupposes an understanding of its *syntax*. But syntax is a combinatorial entity, and syntactic manipulations are very close to computing. We saw hints of this whenever we encountered *counting in the syntax* (e.g., Example 2.7 and Remark 9.15). But the connection goes much deeper. Counting and arithmetic start as soon as we introduce a logical system, even in defining the set of well-formed expressions of the language, not to mention in our specifications for what counts as a legal *proof derivation*. This potentially 'vicious circle' was already emphasized by Hilbert [57] toward the very beginning of modern logic: 'In the usual exposition of the laws of logic certain fundamental concepts of arithmetic are already employed, for example the concept of the aggregate, in part also the concept of number' (p. 347).

Subsequently work revealed a deep and precise sense in which syntax and counting are indeed two sides of the same coin. For instance, echoing related ideas from Tarski, Hermes, Löb, and others, Quine [107] showed that the first-order theory of the natural numbers (i.e., 'true arithmetic') is in fact bi-interpretable with the first-order theory of *concatenation* of strings (i.e., the theory of semigroups). That is, the theory of $+$ and $\times$ over the natural numbers is essentially the same as the theory of a concatenation operator $\smile$ over strings.

To see the intuition for this, and also to connect this theme with other themes in the present work, consider the laws of concatenation over an alphabet of size one, consisting just of a. Let ε be the empty string. It is easy to check that the following principles are all valid.

(1) $\neg x \smile a = \varepsilon$.
(2) $x \smile a = y \smile a \to x = y$.
(3) $x \smile \varepsilon = x$.
(4) $x \smile (y \smile a) = (x \smile y) \smile a$.
(5) Induction: $\varphi(\varepsilon) \to \forall x(\varphi(x) \to \varphi(x \smile a)) \to \varphi(x)$.

As it happens, interpreting a as 1, ε as 0, and $\smile$ as $+$, these principles completely axiomatize Presburger Arithmetic (they are precisely what you need to run the argument for quantifier elimination), the system we have met so often in this paper under different guises. Intuitively, the laws of addition are just the laws of concatenation for unary notations. What Quine showed is that, perhaps more surprisingly, the correspondence extends to full arithmetic as long as we have at least one more symbol. Similar results have also been shown for second-order number theory and second-order theories of strings (e.g., [27]).

More recently, Grzegorczyk [49] has demonstrated that a very weak theory of concatenation can even replace axiomatic theories of arithmetic in the

https://doi.org/10.1017/bsl.2023.30 Published online by Cambridge University Press

celebrated proof that 'sufficiently strong' theories are both undecidable and incomplete. Remarkably, this allows Gödel-style arguments but with no detour through arithmetization of syntax (and thus no use of the Chinese remainder theorem, and so on). Later on, Visser [127] proved that Grzegorczyk's theory of concatenation is in fact *essentially undecidable* (in the sense of Tarski et al. [125]) by showing it is mutually interpretable with Robinson's Arithmetic. These papers and the ensuing literature contain a wealth of further results on this rich topic, adding yet another dimension to the interplay between logic and counting.

Acknowledgements. For very helpful feedback we would like to thank Xiaoxuan Fu, David Gonzalez, Erich Grädel, Makoto Kanazawa, Phokion Kolaitis, Thomas Mayer, Paul Pietroski, Stanislav Speranski, Rineke Verbrugge, Zhiguang Zhao, and audiences at the Nordic Online Logic Seminar, the UC Berkeley Logic Colloquium, the Stanford University Logic Seminar, and the Tsinghua University Logic Seminar. We are also grateful to the editors and referees at the *Bulletin of Symbolic Logic* for productive comments and suggestions.

REFERENCES

[1] W. Ackermann, ***Solvable Cases of the Decision Problem***, Studies in Logic and the Foundations of Mathematics, North-Holland, Amsterdam, 1954.

[2] G. A. Antonelli, *Numerical abstraction via the Frege quantifier*. ***Notre Dame Journal of Formal Logic***, vol. 51 (2010), no. 2, pp. 161–179.

[3] F. Baader and F. De Bortoli, *On the expressive power of description logics with cardinality constraints on finite and infinite sets.*, ***Frontiers of Combining Systems*** (A. Herzig and A. Popescu, editors), Springer, Cham, 2019, pp. 203–219.

[4] F. Bacchus, ***Representing and Reasoning with Probabilistic Knowledge***, MIT Press, Cambridge, 1990.

[5] A. Baltag and J. van Benthem, *A simple logic of functional dependence*. ***Journal of Philosophical Logic***, vol. 50 (2021), pp. 939–1005.

[6] P. Barceló, E. V. Kostylev, M. Monet, J. Pérez, J. Reutter, and J. P. Silva, *The logical expressiveness of graph neural networks*, ***Proceedings of the International Conference on Learning Representations (ICLR)***, 2020.

[7] D. Barner, K. Chow, and S.-J. Yang, *Finding one's meaning: A test of the relation between quantifiers and integers in language development*. ***Cognitive Psychology***, vol. 58 (2009), no. 2, pp. 195–219.

[8] J. Barwise and R. Cooper, *Generalized quantifiers and natural language*. ***Linguistics and Philosophy***, vol. 4 (1981), no. 2, pp. 159–219.

[9] J. Barwise and S. Feferman, ***Model-Theoretic Logics***, Association for Symbolic Logic, 1985.

[10] B. Bednarczyk, S. Demri, R. Fervari, and A. Mansutti, *Modal logics with composition on finite forests: Expressivity and complexity*, ***Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science***, Association for Computing Machinery, New York, 2020, pp. 167–180.

[11] J. van Benthem, ***Essays in Logical Semantics***, Reidel, Dordrecht, 1986.

https://doi.org/10.1017/bsl.2023.30 Published online by Cambridge University Press

[12] ———, ***Language in Action: Categories, Lambdas, and Dynamic Logic***, Studies in Logic, vol. 130, Elsevier, Amsterdam, 1991.

[13] ———, *Program constructions that are safe for bisimulation*. ***Studia Logica***, vol. 60 (1998), pp. 311–330.

[14] ———, *Guards, bounds, and generalized semantics*. ***Journal of Logic, Language, and Information***, vol. 14 (2005), no. 3, pp. 263–279.

[15] J. van Benthem and F. Liu, *New logical perspectives on monotonicity*, ***Monotonicity in Logic and Language*** (D. Deng, F. Liu, M. Liu, and D. Westerståhl, editors), Springer, 2020.

[16] J. van Benthem, K. Mierzewski, and F. Zaffora Blando, *The modal logic of stepwise removal*. ***The Review of Symbolic Logic***, vol. 15 (2022), no. 1, pp. 36–63.

[17] P. Blackburn, M. de Rijke, and Y. Venema, ***Modal Logic***, Cambridge University Press, New York, 2001.

[18] I. Borosh and L. B. Treybig, *Bounds on positive integral solutions of linear Diophantine equations*. ***Proceedings of the American Mathematical Society***, vol. 55 (1976), no. 2, pp. 299–304.

[19] A. Brasoveanu, *Sentence-internal different as quantifier-internal anaphora*. ***Linguistics and Philosophy***. 34(2011), 93–168.

[20] D. Bumford, *Incremental quantification and the dynamics of pair-list phenomena*. ***Semantics and Pragmatics***, vol. 8 (2015), no. 9, pp. 1–70.

[21] J. P. Burgess, *Axiomatizing the logic of comparative probability*. ***Notre Dame Journal of Formal Logic***, vol. 51 (2010), no. 1, pp. 119–126.

[22] J.-Y. Cai, M. Fürer, and N. Immerman, *An optimal lower bound on the number of variables for graph identification*. ***Combinatorica***, vol. 12 (1992), pp. 389–410.

[23] S. Carey, ***The Origin of Concepts***, Oxford University Press, Oxford, 2009.

[24] F. Carreiro, A. Facchini, Y. Venema, and F. Zanasi, *Model theory of monadic predicate logic with the infinity quantifier*. ***Archive for Mathematical Logic***, vol. 61 (2022), pp. 465–502.

[25] S. Clarke and J. Beck, *The number sense represents (rational) numbers*. ***Behavioral and Brain Sciences***, vol. 44 (2021), p. e178.

[26] S. A. Cook and R. A. Reckhow, *The relative efficiency of propositional proof systems*. ***Journal of Symbolic Logic***, vol. 44 (1979), no. 1, pp. 36–50.

[27] J. Corcoran, W. Frank, and M. Maloney, *String theory*. ***The Journal of Symbolic Logic***, vol. 39 (1974), no. 4, pp. 625–637.

[28] K. van Deemter, *Generalized quantifiers: Finite versus infinite*, ***Generalized Quantifiers in Natural Language*** (J. van Benthem and A. ter Meulen, editors), Foris, Dordrecht, 1984, pp. 145–160.

[29] S. Dehaene, ***The Number Sense***, Oxford University Press, Oxford, 2011.

[30] S. Demri and D. Lugiez, *Complexity of modal logics with Presburger constraints*. ***Journal of Applied Logic***, vol. 8 (2010), no. 3, pp. 233–252.

[31] S. Denison and F. Xu, *The origins of probabilistic inference in human infants*. ***Cognition***, vol. 130 (2014), no. 3, pp. 335–347.

[32] Y. Ding, M. Harrison-Trainor, and W. H. Holliday, *The logic of comparative cardinality*. ***The Journal of Symbolic Logic***, vol. 83 (2020), no. 3, pp. 972–1005.

[33] Y. Ding, W. H. Holliday, and T. F. Icard, *Regularity for relative likelihood*, Peking University, University of California, Berkeley, and Stanford University, manuscript, 2021.

[34] A. Ehrenfeucht, D. Haussler, and G. Rozenberg, *On regularity of context-free languages*. ***Theoretical Computer Science***, vol. 27 (1983), no. 3, pp. 311–332.

[35] S. Eilenberg and M.-P. Schützenberger, *Rational sets in commutative monoids*. ***Journal of Algebra***, vol. 13 (1969), no. 2, pp. 173–191.

https://doi.org/10.1017/bsl.2023.30 Published online by Cambridge University Press

[36] J. Endrullis and L. S. Moss, *Syllogistic logic with "most"*. ***Mathematical Structures in Computer Science***, vol. 29 (2019), no. 6, pp. 763–782.

[37] R. Fagin, J. Y. Halpern, and N. Megiddo, *A logic for reasoning about probabilities*. ***Information and Computation***, vol. 87 (1990), pp. 78–128.

[38] S. Feferman and R. Vaught, *The first-order properties of products of algebraic systems*. ***Fundamenta Mathematicae***, vol. 47 (1959), pp. 57–103.

[39] L. Feigenson, S. Dehaene, and E. Spelke, *Core systems of number*. ***Trends in Cognitive Sciences***, vol. 8 (2003), no. 7, pp. 307–314.

[40] K. Fine, *Propositional quantifiers in modal logic*. ***Theoria***, vol. 36 (1970), pp. 336–346.

[41] ———, *In so many possible worlds*. ***Notre Dame Journal of Formal Logic***, vol. 13 (1972), no. 4, 516–520.

[42] X. Fu and Z. Zhao, *Modal logic with counting: Definability, semilinear sets and correspondence theory*, unpublished manuscript, China University of Political Science and Law, Beijing and School of Mathematics and Statistics, Taishan University, 2023.

[43] P. Gärdenfors, *Qualitative probability as an intensional logic*. ***Journal of Philosophical Logic***, vol. 4 (1975), no. 2, pp. 171–185.

[44] S. Ginsburg and E. H. Spanier, *Semigroups, Presburger formulas, and languages*. ***Pacific Journal of Mathematics***, vol. 16 (1966), no. 2, pp. 285–296.

[45] E. Grädel, M. Otto, and E. Rosen, *Two-variable logic with counting is decidable*, ***Proceedings of Twelfth Annual IEEE Symposium on Logic in Computer Science (LICS '97)***, IEEE Computer Society, Warsaw, 1997, pp. 306–317.

[46] ———, *Undecidability results on two-variable logics*. ***Archive for Mathematical Logic***, vol. 38 (1999), pp. 313–353.

[47] T. Graf, *A subregular bound on the complexity of lexical quantifiers*, ***Proceedings of the 22nd Amsterdam Colloquium*** (J. J. Schlöder, D. McHugh, and F. Roelofsen, editors), 2019, pp. 455–464.

[48] S. Grumbach and C. Tollu, *On the expressive power of counting*. ***Theoretical Computer Science***, vol. 149 (1995), pp. 67–99.

[49] A. Grzegorczyk, *Undecidability without arithmetization*. ***Studia Logica***, vol. 79 (2005), pp. 163–230.

[50] B. Hale and C. Wright, ***The Reason's Proper Study: Essays towards a Neo-Fregean Philosophy of Mathematics***, Oxford University Press, Oxford, 2001.

[51] P. Hall, *On representatives of subsets*. ***Journal of the London Mathematical Society***, vol. 10 (1935), no. 1, pp. 26–30.

[52] J. Y. Halpern, *An analysis of first-order logics of probability*. ***Artificial Intelligence***, vol. 46 (1990), pp. 311–350.

[53] D. Harel, *Recurring dominoes: Making the highly undecidable highly understandable*. ***Annals of Discrete Mathematics***, vol. 24 (1985), pp. 51–72.

[54] M. Harrison-Trainor, W. H. Holliday, and T. F. Icard, *Inferring probability comparisons*. ***Mathematical Social Sciences***, vol. 91 (2018), pp. 61–70.

[55] F. Hartogs, *Über das Problem der Wohlordnung*. ***Mathematische Annalen***, 76 (1915), 438–443.

[56] H. Herre, M. Krynicki, A. Pinus, and J. Väänänen, *The Härtig quantifier: A survey*. ***The Journal of Symbolic Logic***, vol. 56 (1991), no. 4, 1153–1183.

[57] D. Hilbert, *On the foundations of logic and arithmetic*. ***The Monist***, vol. 15 (1905), no. 3, pp. 338–352.

[58] W. van der Hoek and M. deRijke, *Generalized quantifier and modal logic*. ***Journal of Logic, Language, and Information***, vol. 2 (1993), pp. 19–58.

[59] W. van der Hoek, *Qualitative modalities*. ***International Journal of Uncertainty, Fuzziness, and Knowledge-Based Systems***, vol. 4 (1996), no. 1, pp. 45–59.

https://doi.org/10.1017/bsl.2023.30 Published online by Cambridge University Press

[60] J. Hoeksema, *Plurality and conjunction*, ***Studies in Model-Theoretic Semantics*** (A. ter Meulen, editor), Foris, Dordrecht, 1983, pp. 63–83.

[61] S. Hoffmann, *Commutative regular languages—Properties and state complexity*, ***Algebraic Informatics*** (M. Ćirić, M. Droste, and J.-É. Pin, editors), Springer, 2019, pp. 151–163.

[62] W. H. Holliday and T. F. Icard, *Axiomatization in the meaning sciences*, ***The Science of Meaning*** (D. Ball, and B. Rabern, editors), Oxford University Press, Oxford, 2018.

[63] D. Ibeling, T. Icard, K. Mierzewski, and M. Mossé, *Probing the quantitative–qualitative divide in probabilistic reasoning*. ***Annals of Pure and Applied Logic*** (2023), p. 103339, forthcoming.

[64] T. F. Icard and L. S. Moss, *Recent progress on monotonicity*. ***Linguistic Issues in Language Technology***, vol. 9 (2014), no. 7, 167–194.

[65] T. F. Icard, L. S. Moss, and W. Tune, *A monotonicity calculus and its completeness*, ***Proceedings of the 15th Meeting on the Mathematics of Language*** (M. Kanazawa, P. de Groote, and M. Sadrzadeh, editors), Association for Computational Linguistics, London, 2017, pp. 75–87.

[66] M. Kanazawa, *Monadic quantifiers recognized by deterministic pushdown automata*, ***Proceedings of the 19th Amsterdam Colloquium*** (M. Aloni, M. Franke, and F. Roelofsen, editors), 2013, pp. 139–146.

[67] R. M. Karp, *Reducibility among combinatorial problems*, ***Complexity of Computer Computations*** (R. E. Miller, J. W. Thatcher, and J. D. Bohlinger, editors), Springer, Boston, 1972, pp. 85–103.

[68] E. Keenan and D. Paperno, *Overview*, ***Handbook of Quantifiers in Natural Language***, Studies in Linguistics and Philosophy, vol. 90, Springer, 2012, pp. 941–950.

[69] E. Kieroński, I. Pratt-Hartmann, and L. Tendera, *Two-variable logics with counting and semantic constraints*. ***ACM SIGLOG News***, vol. 5 (2018), no. 3, pp. 22–43.

[70] M. E. Kirschhock, H. M. Ditz, and A. Nieder, *Behavioral and neuronal representation of numerosity zero in the crow*. ***Journal of Neuroscience***, vol. 41 (2021), no. 22, pp. 4889–4896.

[71] C. Kisby, S. A. Blanco, A. Kruckman, and L. S. Moss, *Logics for sizes with union or intersection*. ***Proceedings of the AAAI Conference on Artificial Intelligence***, vol. 34, 2020, no. 3, pp. 2870–2876.

[72] T. Knowlton, T. Hunter, D. Odic, A. Wellwood, J. Halberda, P. Pietroski, and J. Lidz, *Linguistic meanings as cognitive instructions*. ***Annals of the New York Academy of Sciences***, vol. 1500 (2021a), no. 1, pp. 134–144.

[73] T. Knowlton, P. Pietroski, J. Halberda, and J. Lidz, *The mental representation of universal quantifiers*. ***Linguistics and Philosophy***, vol. 45 (2021b), pp. 911–941.

[74] C. H. Kraft, J. W. Pratt, and A. Seidenberg, *Intuitive probability on finite sets*. ***The Annals of Mathematical Statistics***, vol. 30 (1959), no. 2, pp. 408–419.

[75] J. Krajíček, ***Proof Complexity***, Cambridge University Press, Cambridge, 2019.

[76] D. H. Krantz, R. D. Luce, P. Suppes, and A. Tversky, ***Foundations of Measurement , vol. 1***, Academic Press, New York, 1971.

[77] D. Kuske and N. Schweikardt, *First-order logic with counting: At least, weak hanf normal forms always exist and can be computed!*, ***Proceedings of the 32nd Annual ACM/IEEE Symposium on Logic in Computer Science***, IEEE, 2017, pp. 1–12.

[78] T. Lai, J. Endrullis, and L. S. Moss, *Majority digraphs*. ***Proceedings of the American Mathematical Society***, vol. 144 (2016), no. 9, pp. 3701–3715.

[79] S. Leśniewski, *O podstawach matematyki*. ***Przegląd Filozoficzny***, vol. 30 (1927), pp. 164–206.

[80] H. R. Lewis, *Complexity results for classes of quantificational formulas*. ***Journal of Computer and System Sciences***, vol. 23 (1980), no. 3, pp. 317–353.

https://doi.org/10.1017/bsl.2023.30 Published online by Cambridge University Press

[81] J. Lidz, P. Pietroski, J. Halberda, and T. Hunter, *Interface transparency and the psychosemantics of most*. ***Natural Language Semantics***, vol. 19 (2011), pp. 227–256.

[82] P. Lindström, *First order predicate logic with generalized quantifiers*. ***Theoria***, vol. 32 (1966), no. 3, pp. 186–195.

[83] G. Link, ***Algebraic Semantics in Language and Philosophy***, Cambridge University Press, Cambridge, 1998.

[84] L. Lipshitz, *The Diophantine problem for addition and divisibility*. ***Transactions of the American Mathematical Society***, vol. 235 (1978), pp. 271–283.

[85] M. Marx and Y. Venema, ***Multi-Dimensional Modal Logic***, Springer, Dordrecht, 1997.

[86] T. Mayer, *An investigation of the negationless fragment of the Rescher–Härtig quantifier*. Bachelor's thesis in Mathematics, Stanford University, 2023.

[87] H. Mercier, G. Politzer, and D. Sperber, *What causes failure to apply the pigeonhole principle in simple reasoning problems?* ***Thinking & Reasoning***, vol. 23 (2017), no. 2, pp. 184–189.

[88] L. E. Moreno and G. Waldegg, *The conceptual evolution of actual mathematical infinity*. ***Educational Studies in Mathematics***, vol. 22 (1991), no. 3, pp. 211–231.

[89] M. Mortimer, *On languages with two variables*. ***Mathematical Logic Quarterly***, vol. 21 (1975), no. 1, pp. 135–140.

[90] L. S. Moss, *Natural logic*, ***Handbook of Contemporary Semantic Theory***, second ed., Wiley-Blackwell, Oxford, 2015, pp. 646–681.

[91] ———, *Syllogistic logic with cardinality comparisons*, ***J. Michael Dunn on Information Based Logics*** (K. Bimbó, editor), Springer, Cham, 2016, pp. 391–415.

[92] L. S. Moss and S. Topal, *Syllogistic logic with cardinality comparisons, on infinite sets*. ***The Review of Symbolic Logic***, vol. 13 (2020), no. 1, pp. 1–22.

[93] M. Mossé, D. Ibeling, and T. Icard, *Is causal reasoning harder than probabilistic reasoning?* ***The Review of Symbolic Logic*** (2022), pp. 1–26, forthcoming.

[94] A. Mostowski and A. Tarski, *Arithmetical classes and types of well-ordered systems*. ***Bulletin of the American Mathematical Society***, vol. 55 (1949), p. 65.

[95] M. Mostowski, *Computational semantics for monadic quantifiers*. ***Journal of Applied Non-Classical Logics***, vol. 8 (1998), pp. 107–121.

[96] I. Németi, *Fine-structure analysis of first-order logic*, ***Arrow Logic and Multidimensional Logic*** (M. Marx, M. Masuch, and L. Pólos, editors), CSLI Publications, Stanford, 1996, pp. 221–247.

[97] D. C. Oppen, *A $2^{2^{2^{pn}}}$ upper bound on the complexity of Presburger Arithmetic*. ***Journal of Computer and System Sciences***, vol. 16 (1978), no. 3, pp. 323–332.

[98] M. Otto, ***Bounded Variable Logics and Counting***, Springer, New York, 1997.

[99] R. Parikh, *On context-free languages*. ***Journal of the ACM***, vol. 13 (1966), no. 4, pp. 570–581.

[100] S. Peters and D. Westerståhl, ***Quantifiers in Language and Logic***, Oxford University Press, Oxford, 2006.

[101] J. Piaget and R. Garcia, ***Psychogenèse et Histoire des Sciences***, Flammarion, Paris, 1983.

[102] P. Pietroski, J. Lidz, T. Hunter, and J. Halberda, *The meaning of "most": Semantics, numerosity and psychology*. ***Mind & Language***, vol. 24 (2009), no. 5, pp. 554–585.

[103] I. Pratt-Hartmann, *Complexity of the two-variable fragment with counting quantifiers*. ***Journal of Logic, Language and Information***, vol. 14 (2005), no. 3, pp. 369–395.

[104] ———, *On the computational complexity of the numerically definite syllogistic and related logics*, this JOURNAL, vol. **14** (2008), no. 1, pp. 1–28.

[105] ———, *No syllogisms for the numerical syllogistic*, ***Languages: From Formal to Natural***, Lecture Notes in Computer Science, vol. 5533, Springer, Berlin–Heidelberg, 2009, pp. 129–203.

[106] H. PUTNAM, *Trial and error predicates and the solution to a problem of Mostowski*. ***The Journal of Symbolic Logic***, vol. 30 (1965), no. 1, pp. 49–57.

[107] W. V. QUINE, *Concatenation as a basis for arithmetic*. ***The Journal of Symbolic Logic***, vol. 11 (1946), no. 4, pp. 105–114.

[108] H. REICHENBACH, ***The Direction of Time***, University of California Press, Berkeley, 1956.

[109] N. RESCHER, *Plurality quantification*. ***The Journal of Symbolic Logic***, vol. 27 (1962), pp. 373–374.

[110] G. RESTALL, ***An Introduction to Substructural Logics***, Routledge, London and New York, 2000.

[111] J. RETT, *The semantics of many, much, few, and little*. ***Language and Linguistics Compass***, vol. 12 (2018), no. 1, p. e12269.

[112] J. ROBINSON, *Definability and decision problems in arithmetic*. ***The Journal of Symbolic Logic***, vol. 14 (1949), no. 2, pp. 98–114.

[113] S. ROTHSTEIN, *Counting and the mass/count distinction*. ***Journal of Semantics***, vol. 27 (2010), no. 3, pp. 343–397.

[114] V. SÁNCHEZ-VALENCIA, *Studies on natural logic and categorial grammar*. PhD thesis, Universiteit van Amsterdam, 1991.

[115] A. SCHRIJVER, ***Theory of Linear and Integer Programming***, Wiley, Chichester, 1998.

[116] N. SCHWEIKARDT, *Arithmetic, first-order logic, and counting quantifiers*. ***ACM Transactions on Compututational Logic***, vol. 6 (2005), no. 3, pp. 634–671.

[117] D. SCOTT, *Logic with denumerably long formulas and finite strings of quantifiers*, ***The Theory of Models*** (J. ADDITION, L. HENKIN, and A. TARSKI, editors), North-Holland, Amsterdam, 1965, pp. 329–341.

[118] A. SEIDENBERG, *A simple proof of a theorem of Erdős and Szekeres*. ***Journal of the London Mathematical Society***, vol. s1–34 (1959), no. 3, p. 352.

[119] T. SKØLEM, ***Diophantische Gleichungen***, Ergebnisse der Mathematik und ihrer Grenzgebiete, Springer, Berlin, 1938.

[120] A. SLOMSON, *The monadic fragment of predicate calculus with the Chang quantifier and equality*, ***Proceedings of the Summer School in Logic Leeds, 1967*** (M. H. LÖB, editor), Springer, Berlin–Heidelberg, 1968, pp. 279–301.

[121] S. STEINERT-THRELKELD and T. F. ICARD, *Iterating semantic automata*. ***Linguistics and Philosophy***, vol. 36 (2013), no. 2, pp. 151–173.

[122] C. STEINHORN, *Borel structures for first-order and extended logics*, ***Harvey Friedman's Research on the Foundations of Mathematics*** (L. HARRINGTON, M. MORLEY, A. SVÊDROV, and S. SIMPSON, editors), Studies in Logic and the Foundations of Mathematics, vol. 117, Elsevier, 1985, pp. 161–178.

[123] Z. SUN and F. LIU, *The inference pattern Mou in Mohist logic—A montonicity reasoning view*. ***Roczniki Filozoficzne***, vol. 68 (2020), pp. 257–270.

[124] J. SZYMANIK, ***Quantifiers and Cognition: Logical and Computational Perspectives***, Springer, Berlin, 2016.

[125] A. TARSKI, A. MOSTOWSKI, and R. M. ROBINSON, ***Undecidable Theories***, North-Holland, Amsterdam, 1953.

[126] J. VÄÄNÄNEN, *Remarks on generalized quantifiers and second-order logics*, ***Set Theory and Hierarchy Theory**, **vol. 14***, Prace Naukowe Instytutu Matematyki Politechniki Wroclawskiej, Wroclaw, 1977, pp. 117–123.

https://doi.org/10.1017/bsl.2023.30 Published online by Cambridge University Press

[127] A. Visser, *Growing commas: A study of sequentiality and concatenation*. ***Notre Dame Journal of Formal Logic***, vol. 50 (2009), no. 1, pp. 61–85.

[128] D. Westerståhl, *Logical constants in quantifier languages*. ***Linguistics and Philosophy***, vol. 8 (1985), pp. 387–413.

INSTITUTE FOR LOGIC, LANGUAGE, AND COMPUTATION
UNIVERSITEIT VAN AMSTERDAM
SCIENCE PARK 107, 1098 XG AMSTERDAM
NETHERLANDS
E-mail: johan@stanford.edu

DEPARTMENT OF PHILOSOPHY, STANFORD UNIVERSITY
450 JANE STANFORD WAY, MAIN QUAD, BUILDING 90
STANFORD, CA 94305, USA
E-mail: icard@stanford.edu

https://doi.org/10.1017/bsl.2023.30 Published online by Cambridge University Press