Recursive numeral systems are highly regular and easy to process

Anonymous ACL submission

Abstract

Previous work has argued that recursive numeral systems optimise the trade-off between lexicon size and average morphosyntatic complexity (Denić and Szymanik, 2024). However, showing that *only* natural-language-like systems optimise this tradeoff has proven elusive, and the existing solution has relied on ad-hoc constraints to rule out unnatural systems (Yang and Regier, 2025). Here, we argue that this issue arises because the proposed trade-off has neglected *regularity*, a crucial aspect of complexity central to human grammars in general. Drawing on the Minimum Description Length (MDL) approach, we propose that recursive numeral systems are better viewed as efficient with regard to their regularity and processing complexity. We show that our MDL-based measures of regularity and processing complexity better capture the key differences between attested, natural systems and unattested but possible ones, including "optimal" recursive numeral systems from previous work, and that the ad-hoc constraints from previous literature naturally follow from regularity. Our approach highlights the need to incorporate regularity across sets of forms in studies that attempt to measure and explain optimality in language.

1 Introduction

One key question in the study of language is why languages are the way they are; what cognitive and communicative constraints shape linguistic systems? Recent research suggests that linguistic systems are shaped by efficient communication, such that natural systems optimise the trade-off between simplicity and informativeness in various domains (Gibson et al., 2019), from kinship terms (Kemp and Regier, 2012), colour terms (Regier et al., 2015; Zaslavsky et al., 2019), grammatical categories (Mollica et al., 2021), indefinite pronouns (Denić et al., 2022), boolean connectives (Uegaki, 2022) to spatial demonstratives (Chen et al., 2023). For example, colour-naming systems tend to either partition the space of colour referents into few large, connected categories (e.g., *dark* vs. *light* colours), or into many small specific categories (e.g., *red, green, blue, yellow, brown*, ...); the former systems are simple but do not allow precise communication about colour, the terms in the latter systems are highly informative but the systems are complex (Regier et al., 2015).

Xu et al. (2020) investigated this trade-off in the domain of numeral systems. They found that while numeral systems with limited precision (e.g., *a few* vs. *many*) followed the same trade-off, recursive numeral systems like the English decimal system (*one*, *two*, ... *ninety*, *ninety-one*, ...) are maximally informative in that any integer can be precisely expressed. Since informativeness is fixed, Denić and Szymanik (2024) (henceforth, D&S) argued that the relevant trade-off shaping natural recursive numeral systems is instead between lexicon size and the average length of the numerals. However, as reviewed in Section 2, the optimal numeral systems under this trade-off included not only human-like systems, but also unnatural, highly irregular ones. Yang and Regier (2025) (Y&R) attempted to resolve this with ad-hoc constraints that artificially restrict the space of possible systems to be human-like.

In this paper, we argue instead that the key characteristic of human-like recursive numeral systems is how regular they are (*regularity*), and how easy it is to process them (*processing complexity*). We leverage a precise formal definition of regularity from formal language and complexity theory to demarcate natural language systems from other theoretically possible ones. Our contributions are:

- We introduce regularity and processing complexity as the key characteristics of recursive numeral systems (Section 3),
- We demonstrate that human-like systems are

significantly more regular and easier to process than other theoretically possible systems (Section 4),

- We show that our measures capture the difference between natural systems and unattested ones regardless of assumptions about the prior over numbers (Section 5).

The code and data for our analyses are available at https://anonymous.4open.science/r/regularity-processing-complexity-recursive-numerals-322B.

2 Prior work on efficiency of recursive numeral systems

D&S proposed that recursive numeral systems optimise the trade-off between *lexicon size* and *average morphosyntactic complexity*. Lexicon size is the number of meanings encoded by single morphemes (e.g. in Mandarin this includes the numerals from 1 *yī* 一 to 10 十 *shí*). Average morphosyntactic complexity is the weighted average of the count of number and arithmetic combinator morphemes $(+, -, *)$ across all numerals. For example, the *numeral* (i.e. the form) of *number* 43 is formed in Mandarin as 四十三, *sì shí sān*, "four [times] ten [plus] three", $4 * 10 + 3$. This numeral has morphosyntactic complexity of 5, counting the digit and null morphemes $*$ and $+$.

To show that natural recursive systems optimise this trade-off, D&S compared the numeral systems of natural languages to theoretically possible systems over the number range $1 - 99$. They generated these possible recursive numeral systems using Hurford's grammar (Hurford, 1975, 2007):

$$\begin{aligned} Num &\rightarrow D \mid Phrase \mid Phrase \pm Num \\ Phrase &\rightarrow M \mid Num * M \end{aligned} \tag{1}$$

where D and M are sets of digits and multipliers.

Hurford's grammar allows one to represent a recursive numeral system succinctly by only a (D, M) pair. D&S used a genetic algorithm (cf. Appendix A, Algorithm 1) to obtain the *Pareto frontier*, the theoretical bound beyond which no possible language can improve on lexicon size without sacrificing average morphosyntactic complexity (or vice-versa), and showed that human recursive numeral systems lie on or close to the frontier.

However, unlike human numeral systems, the numerals in their optimal artificial systems did not consistently employ the same multiplier across numerals (cf. Table 1). Systems like this are widespread along D&S's frontier, and were taken to be as "efficient" as natural ones, but they are evidently not human-like. This suggests that optimisation for lexicon size and average morphosyntactic complexity alone may not accurately capture what makes natural recursive numeral systems different from other possible ones.

Number	Numeral (Mandarin)	Numeral (unnatural)
18	$10 + 8$	$25 - 7$
19	$10 + 9$	$25 - 6$
20	$2 * 10$	$5 * 4$
21	$2 * 10 + 1$	$3 * 7$
22	$2 * 10 + 2$	$25 - 3$
23	$2 * 10 + 3$	$25 - 2$
24	$2 * 10 + 4$	$6 * 4$
25	$2 * 10 + 5$	25

Table 1: Numerals in Mandarin (second column) and in one of the optimal artificial systems of D&S (third column) for the number range $18 - 25$. Notice that Mandarin is consistently base-10, whereas the unnatural system has alternating bases (i.e., 4, 7 and 25).

Y&R attempted to solve these issues by introducing ad-hoc constraints over the space of possible numeral systems, resolving the unorderliness issue of D&S by restricting the numeral system space to be more human-like. They added an extra rule for suppletives, and four additional constraints, which specify which multipliers to use for which number ranges, the maximum number that can be added or subtracted from $Phrase$ within a given number range, as well as exception rules. They further require that the digits in D run sequentially from 1 to a number between $1 - 20$. They also only allowed for the smallest multiplier to be used to construct numbers smaller than or equal to the next largest multplier in M (Other examples can be found in Yang and Regier, 2025).

Crucially, Y&R still maintain that lexicon size and morphosyntactic complexity are the relevant measures. In this sense, D&S's optimal languages are still considered as efficient as Y&R ones, even though they cannot be generated under the grammar constraints used by Y&R.

3 (Ir)regularity and processing complexity

We find the resolution proposed by Y&R unsatisfying, since the source of these additional constraints itself stands in need of explanation; ideally, these characteristics could be accounted for as byprod-

uct of other better-motivated pressures on natural recursive numeral systems. Importantly, neither of the existing measures captures a crucial aspect of human systems, namely *regularity*. An "unorderly" system, such as one with alternating multipliers, is intuitively not human-like precisely because the numerals are irregular, and Y&R constraints yield more human-like systems precisely because they are explicit constraints enforcing regularity. We propose that recursive numeral systems are better thought of as optimising the trade-off between irregularity and processing complexity, and provide measures of each below.

3.1 Irregularity

One way to measure regularity is in terms of *compressibility*. Regular systems are highly compressible, such that one could write a small set of general rules for the construction of numerals. For instance, recursive numerals in Karo Batak (Woollams, 1996) are extremely regular. For numerals from 1 to 99, it only requires the rules $Num \rightarrow D$ (for 1-9), and $Num \rightarrow D * 10 + D$ (for 10-99). English numerals are slightly less regular, i.e. featuring suppletive numerals (11, 12) that require separate rules. We can thus measure the (ir)regularity of a system by finding the length of the most general/shortest grammar for numerals. However, as inferring the shortest recursive grammar is challenging (Charikar et al., 2005), we simplify the problem by restricting ourselves to the number range 1-99, following D&S and Y&R, and treating the system as a finite language. We inferred a minimal partial deterministic finite automaton (DFA) that generates all and only valid numerals to represent a numeral grammar using Mihov and Schulz's (2019) algorithm.[1] Intuitively, a regular system has a lot of repeated parts, so the number of states and transitions of the DFA will be small (cf. Figure 1).

Our measure of irregularity is thus the representational complexity of the numeral system, i.e., the size of the partial DFA. Adapting the scheme of Brighton (2003), the irregularity $L(G)$ of a recursive numeral grammar G, in bits, is:

$$\underbrace{|Z|\big(2 \cdot log_2|S| + log_2|\Sigma|\big)}_{\text{(A)}} + \underbrace{log_2|S|}_{\text{(B)}} + \underbrace{|S|}_{\text{(C)}} \quad (2)$$

where Z is the set of transitions, S is the set of states, and Σ is the set of symbols (i.e., morphemes). (A) gives the number of bits required to encode the transition table: Each transition requires $2 \cdot log_2|S|$ bits to identify the states at the start and end of the transition, and $log_2|\Sigma|$ to identify the transition symbol (morpheme). (B) and (C) are the numbers of bits for identifying the single initial state and accepting states (one bit for each state, indicating whether it is an accepting state or not), respectively.

3.2 Processing complexity

The DFA can be thought of as a description language used to encode the data (a set of numerals). The more complex the description language (the larger the DFA), the easier it is to generate or parse a numeral. Drawing on the Minimum Description Length principle (Li et al., 2008), Brighton (2003) shows that this quantity could be calculated from the average information required to determine the unique parsing path through the DFA for a given form-meaning pairing. In our case, this pairing simply corresponds to a numeral (the form) as one morpheme refers to only one meaning (e.g., *one* in *twenty one* refers only to '1'). For instance, in the DFA shown in Figure 1, parsing $2 * 10$ (20) requires going through 4 transitions connecting the states λ (initial state), D, $D*$ and $D * 10$, and one would need $\underbrace{log_2(9)}_{\lambda} + \underbrace{log_2(1) + 1}_{D} + \underbrace{log_2(1)}_{D*} + \underbrace{log_2(1) + 1}_{D*10}$ bits to encode or generate this numeral as a path through the DFA, with each accepting state requiring an extra bit to encode the accepting status. Averaging over these bits for all numbers gives the *processing complexity* of generating or parsing a set of numerals using a given DFA. Again adapting Brighton's (2003) scheme, the processing complexity $L(N \mid G)$ of recursive numeral grammar G for a set of numbers N is given by:

$$\sum_{i=1}^{|N|}\Big\{P(n_i) \cdot \sum_{j=1}^{|S_i|}\big(log_2|Z_{ij}| + F(s_{ij})\big)\Big\} \quad (3)$$

where $P(n_i)$ is the prior/weight for $n_i \in N$, S_i is the list of states through which the parsing proceeds for the numeral for n_i, Z_{ij} is the set of outward transitions from the jth state in the parsing of the numeral for n_i, and F is an indicator function evaluating to 1 if $s_{ij} \in S_i$ is an accepting state. D&S and Y&R assume that the prior is $P(n_i) \propto n_i^{-2}$, taken to reflect the communicative need of numbers (Dehaene and Mehler, 1992; Piantadosi, 2016), i.e.

[1] We use the Python package `automata-lib` (https://pypi.org/project/automata-lib/) for inference.

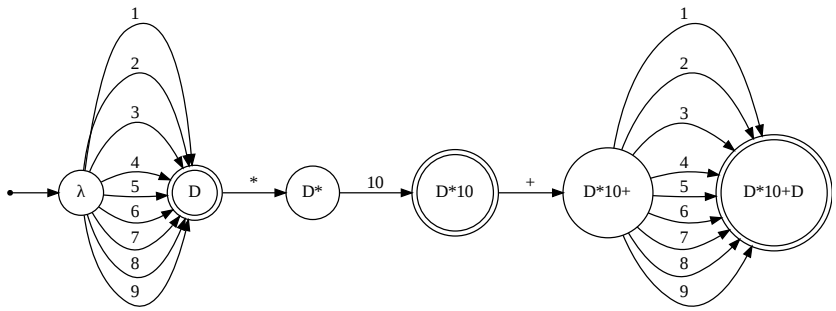

Figure 1: The minimal partial DFA representing Karo Batak (highly regular) numerals in the range 1 to 99. Circles are states. Double circles are accepting states. Arrows are transitions. λ denotes the initial state. A given numeral is parsed or generated by traversing the automaton and emitting the symbols (morphemes) associated with the transitions. Accepting states mark valid potential termination points of the parsing or generation. Note that, here, we name each state in accordance with the rules to generate (partial) numerals when the state is reached. For example, the number 10 in Karo Batak, expressed by $1 * 10$, corresponds to the path from λ through the transition for 1 to the state D, then the transition for $*$ to the state $D*$, then the transition for 10, ending at the state $D * 10$.

there is high communicative need / frequent use of small numbers, and the need to express higher numbers declines rapidly. We follow this assumption in Section 4 but challenge it in Section 5.

4 Applying the measures

We now apply our complexity measures to natural recursive numeral systems and compare them to other theoretically possible ones.

4.1 Natural vs. baseline numeral systems

We first ask if natural recursive numerals are particularly regular and of low processing complexity, relative to the entire space of possible systems. We compute our measures on two datasets: (1) the recursive numeral systems from 128 natural languages annotated at the morpheme level from D&S,[2] and (2) a sample of 10,000 baseline artificial systems generated with Hurford's grammar. To obtain the baseline sample, we adapt the implementation of the genetic algorithm in D&S. When constructing artificial numeral systems, in cases where a grammar allows a number to be expressed as several possible numerals, D&S always chose the shortest numeral for each number (e.g. given $D = \{1, 2\}, M = \{5, 10, 12\}$, the possible numerals for 12 include the shortest 12, along with $1 * 12$, $10 + 2$, $1 * 10 + 2$ and $2 * 5 + 2$). This is partly the source of unorderliness: favouring the shortest possible numeral will result in a system that does not reuse structures in smaller numerals in bigger ones (e.g., using 12 instead of $10 + 2$ when the numeral for 11 is $10 + 1$). To alleviate this, we allow numerals of any length and any combination of digits, multipliers and combinators ($+, *, -$). For computational tractability, however, we limit the max search depth to 5, meaning that a numeral has at most 5 number morphemes and 4 arithmetic combinators, and use only digits and multipliers that occur in at least one natural language. We draw languages in 100 batches of 100 language types. We determine the type for each batch as follows. We randomly select 3 to 12 digits and 1 to 3 multipliers. We always allow $+$ and $*$ as combinators, with a 20% chance of also having the subtraction combinator $-$, which is less common among the world's languages (Comrie, 2022). We resample if the resulting set of digits, multipliers and combinators cannot express all numbers between 1 and 99. Otherwise, we use the selected digits, multipliers and combinators to generate 100 sets of numerals, each covering the numbers 1-99, where those samples can differ in how a given number is expressed (cf. the multiple possible numerals for 12 above).

Figure 2 plots irregularity against processing complexity for natural systems and these baseline systems. As expected, natural systems exhibit significantly more regularity and less processing complexity than the baselines. In fact, none of the sampled systems are simultaneously more regular and less complex than natural systems.

Next, we ask whether our measures successfully capture the key properties of natural systems that distinguish them from other possible systems, rather than simply recapitulating the previous measures. We compute our measures on two more

[2] https://osf.io/cmuq4

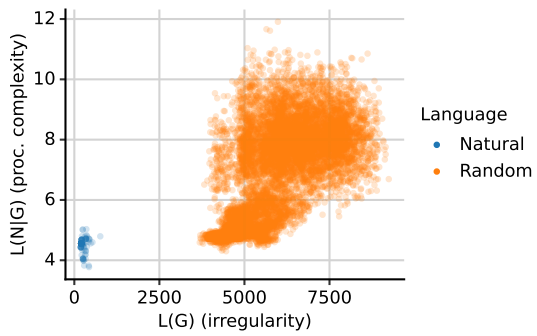

Figure 2: Irregularity (x-axis) and processing complexity (y-axis) of natural languages (blue) and 10,000 randomly generated baseline artificial languages (orange).

datasets: (1) optimal artificial systems evolved through the genetic algorithm from D&S, and (2) optimal artificial systems evolved through the genetic algorithm with the constraints from Y&R.

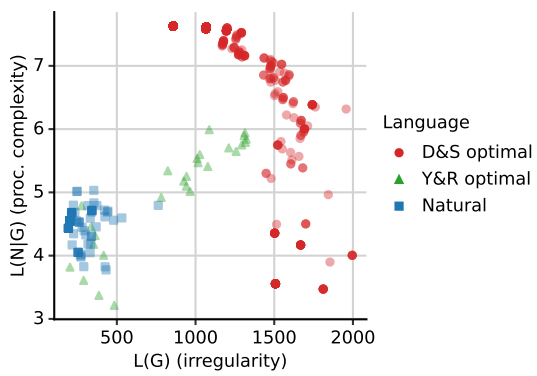

Figure 3: Irregularity (x-axis) and processing complexity (y-axis) of natural languages (blue, square) and optimal languages in D&S (red, circle) and Y&R (green, triangle)

Figure 3 shows that the different systems exhibit varying degrees of regularity and processing complexity, with natural systems typically being more efficient than the optimal systems according to the metrics and grammars used by D&R and Y&R. Interestingly, among Y&R systems, which already have some degrees of regularity built in, there are some that are clearly less efficient than natural languages (the middle cluster), and crucially a few that appear to be *more* efficient than natural languages (bottom left corner). Upon inspection, the less efficient systems use a large set of digits and large mutlipliers, e.g., digits up to 20 and M = {21, 42, 56, 71, 85}, which results in higher irregularity (cf. $|\Sigma|$ in (2)), and processing complexity due to more transitions when constructing numerals in higher ranges. The four highly efficient Y&R numeral systems are those with digits and multiplier in the range 1 to 5. In other words, these have an extremely small lexicon size with digits in the smallest ranges. This allows them to reuse smaller numerals to build up larger ones, thereby increasing regularity at little cost of processing complexity (e.g., 5 expressed as $2*2+1$; 6 as $(2+1)*2$; 7 as $(2+1)*2+1$; 8 as $2*2*2$; 9 as $2*2*2+1$; 10 as $(2*2+1)*2$). Larger numerals in these systems end up with high processing complexity (e.g., 46 is expressed as $(((2*2+1)*2+1)*2+1)*2$), but contribute little to the average processing complexity because the power law prior places little weight on larger numbers.

4.2 Controlling for digits, multipliers, combinators, and numeral lengths

Most of the D&S and Y&R optimal systems in Figure 3 do not necessarily feature the same *combinations* of digits and multipliers as natural languages. One might wonder, then, whether the apparent regularity and processing simplicity of natural recursive numeral systems could still result from the optimisation for the trade-off between lexicon size and average morphosyntactic complexity, in conjunction with the specific (D, M)s that natural systems employ. Indeed, the Y&R inefficient systems are the ones with unnatural (D, M)s, e.g., $M = \{21, 42, 56, 71, 85\}$. One response is to say that the choice of human-like (D, M) pairs themselves would still demand an explanation, which regularity and processing simplicity could provide. Here, however, we seek an even stronger piece of evidence. We aim to show that natural recursive numeral systems are (near-)optimally regular and of low processing complexity *even* among their *local neighbourhood*. Specifically, each natural numeral system's local neighbourhood is the set of possible systems that share its digits D, multipliers M, arithmetic combinators C and the lengths of numerals L_{Num}. For example, every system in Drehu's local neighbourhood will have numerals that make use of $D = \{1, 2, 3, 4\}$, $M = \{5, 10, 15, 20\}$, $C = \{*, +\}$ and match Drehu's numerals in length. To illustrate, Drehu 24 is expressed as $1*20+4$, so its alternative numerals are $\{4*5+4, 5+15+4, 15+5+4, 10+10+4\}$. Note that, because of this, every system in a local neighbourhood will also share the same lexicon size and average morphosyntactic complexity, effectively controlling for these measures.

Since exhaustively constructing the local neighbourhoods is computationally prohibitive, we implement a greedy algorithm to approximate only the most (and least) efficient numeral systems. For each natural system, we consider adding alternative numerals starting with the largest numbers and going to lower range of numbers; from the set of such perturbations, we weed out all systems apart from the ones on the Pareto frontier, and repeat the process until we obtain full systems covering the number range 1–99. The full procedure is explained and given as pseudocode alongside our parameter choices in Algorithm 2, Appendix B. The expectation is that natural systems should consistently be as efficient or more efficient than the local frontier from this algorithm.

We obtain the frontiers for 37 unique local neighbourhoods inferred from 127 out of the 128 numeral systems of natural languages.[3] We find that, while the spaces are (unsurprisingly) small, *all* 121 out of 127 systems are either as efficient or more efficient than their local frontiers. While for some of these, the spaces are so small that there are only a couple of possible systems (cf. the space for Karo Batak-like systems in the leftmost panel of Figure 4, which has only two possible systems under our constraints: best and worst), there are larger spaces with clearly more room for variation, and we found that natural systems are consistently on the local frontiers (Panels 2-4 in Figure 4 illustrate typical results; consult the full plots on our (anonymised) GitHub repository). Overall, this suggests that optimisation for lexicon size, average morphosyntactic complexity, along with (D, M, C, L_{Num}) combinations, do not fully explain the degree of regularity and processing complexity that natural systems exhibit.

5 The role of the prior

Previous work has argued that recursive numeral systems are subjected to a power law prior distribution $P(n) \propto n^{-2}$, which reflects communicative need over numbers (Dehaene and Mehler, 1992; Xu et al., 2020; Denić and Szymanik, 2024; Yang and Regier, 2025). This presents a very strong bias towards the smallest numbers, as when calculating either average morphosyntactic or processing complexity most numerals will have minimal influence. More specifically, when considering the numeral range $1-99$, more than 75% of the probability mass is placed on numbers 1 and 2 alone, and more than 90% is placed on numbers $1-6$. This is why, for example, the optimal system in Section 4.1 with incredibly complex large numerals (e.g. a system where the number 46 expressed as the numeral $(((2*2+1)*2+1)*2+1)*2)$ is considered easier process than natural systems: these numerals contribute negligibly to the final result when considering the power law prior.

However, the purpose of recursive numeral systems is to be able to precisely express the higher-range numbers that are heavily penalised by the power law prior. If the larger numbers' influence was so negligible, one would expect recursive numeral systems to be less widespread, as exact-restricted systems like Kayardild, which has a single word for any number greater than 4 (*muthaa*), would seem as effective and much easier to learn.

Therefore, there might be reason to believe that these sorts of systems are not influenced by a need distribution which emphasises small numbers so heavily. Instead, a uniform prior over numbers may accommodate this theory the best. To investigate the role of the prior, we obtain systems that are optimal in terms of lexicon size and average morphosyntactic complexity, in the same vein as D&S and Y&R, except under a uniform prior. We then compare these new optimal systems to human systems in terms of irregularity and (uniformly weighted) processing complexity. As Figure 5 shows,[4] while natural systems still exhibit high regularity, the optimal artificial systems lie much further right, with Y&R systems being slightly more regular than D&S, but still far from human systems. Notably, the four Y&R systems in Section 4.1 are no longer considered more efficient than the human ones under the uniform prior.

Surprisingly, many D&S optimal systems end up being more efficient than human systems in terms of processing complexity under this prior. This is likely because, unlike human and Y&R systems, D&S do not constrain the lexicon size and always choose the shortest numerals possible. Thus, when the lexicon size is large, the processing complexity is low (Figure 6). For example, a D&S system with all number morphemes from 1 to 99 will express every number with a single morpheme, and will have a single accepting state and 99 transitions to

[3] We exclude Kayah Li's neighbourhood due to limited computational resources. cf. Section B, Appendix A.

[4] We exclude two outlier Y&R/D&S systems with extremely high complexities from Figure 5 and 6.

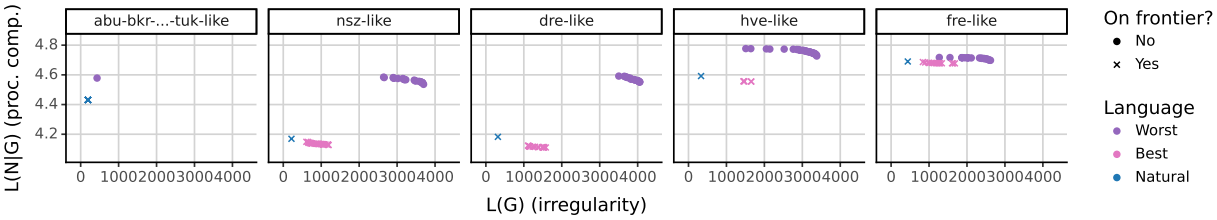

Figure 4: Irregularity (x-axis) and processing complexity (y-axis) of the most (purple) and least (pink) efficient natural-(D, M, C, L_{Num})-matched artificial systems, as well as their corresponding natural systems (blue). Natural systems are consistently on the approximately local frontier (cross). Leftmost panel shows a case where the space is very small with one best–occupied by natural systems like Karo Batak–and one worst system. Subsequent panels (Sierra Nahuatl-, Drehu-, Huave- and French-like systems) show cases where there is room for variation, though $L(N \mid G)$ varies to different degrees depending on exact (D, M, C, L_{Num}). French-like systems exhibit little variation in $L(N \mid G)$, whereas Nahuatl-like systems exhibit more.

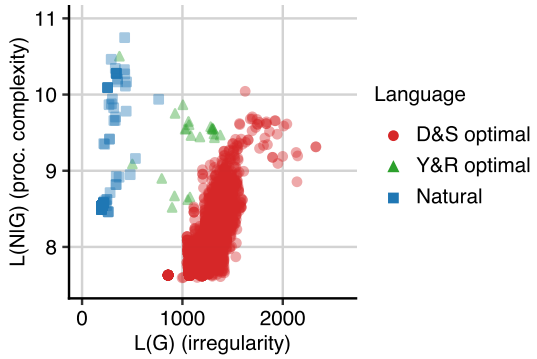

Figure 5: Irregularity (x-axis) and unweighted processing complexity (y-axis) of natural systems (blue, square) and optimal systems in D&S (red, circle) and Y&R (green, triangle).

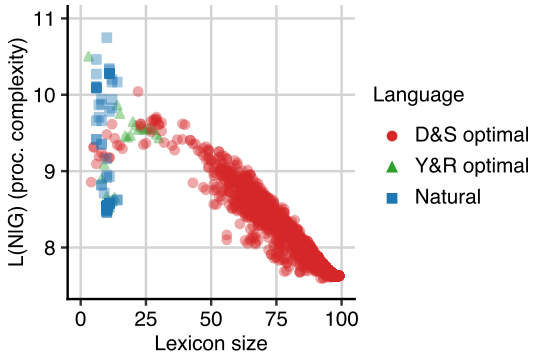

Figure 6: Lexicon size (x-axis) and unweighted processing complexity (y-axis) of natural systems (blue, square) and optimal systems in D&S (red, circle) and Y&R (green, triangle).

it, resulting in $L(N \mid G) = log_2(99) + 1 \simeq 7.63$ bits per number. On the other hand, to parse a numeral like 96 in a highly regular system like Karo Batak (Figure 1), we would need $\underbrace{log_2(9)}_{\lambda} + \underbrace{log_2(1) + 1}_{D} + \underbrace{log_2(1)}_{D*} + \underbrace{log_2(1) + 1}_{D*10} + \underbrace{log_2(1)}_{D*10+} + \underbrace{log_2(9) + 1}_{D*10+D} \simeq 9.34 > 7.63$ bits. While the possible transitions are fewer, Karo Batak is harder to process than a 99-word system due to the extra bits required by the larger number of accepting states.

Regardless of the shape of the correct prior over numbers, our results show how natural systems are much more efficient in both measures (Figure 3) or at least in regularity (Figure 5) than the optimal artificial systems obtained by both D&S and Y&R when optimizing for lexicon size and average morphosyntactic complexity.

6 Discussion

We have argued that prior literature on the efficiency of recursive numeral systems (Denić and Szymanik, 2024; Yang and Regier, 2025) has not satisfactorily accounted for the key characteristics of human-like systems, namely their regularity and low processing complexity. In this paper, we have shown that human recursive numeral systems are more regular and easier to process than other theoretically possible systems, even when controlling for the set of number, morphemes and numeral lengths. We are skeptical that the relevant prior is the power law communicative need distribution (Dehaene and Mehler, 1992; Piantadosi, 2016) as assumed in prior work (Xu et al., 2020; Denić and Szymanik, 2024; Yang and Regier, 2025). We show that even if the relevant prior was uniform, our measures would still clearly separate human-like and other possible recursive numeral systems.

One issue that we have not considered here is the

role of cognitive cost of encoding certain meanings or relations. For example, systems using highly unnatural numerals like $(((2*2+1)*2+1)*2+1)*2$ (Section 4.1) are currently considered efficient. We have shown that this might be because of the power law prior assumption (Section 5), but the absence of this kind of numeral could also be due to the high cost of computation involved in the multiply nested multiplications, which are harder to learn in numeral systems than operations like additions (Holt and Barner, 2025; Silvi et al., 2025), while our measure of processing complexity assumes that all transition symbols are equally costly. Similarly, expressing numerals with multiple nested parentheses would also introduce ambiguity in how they are interpreted (e.g., $2 * 10 + 5$ could be either $(2 * 10) + 5$ or $2 * (10 + 5)$).

Another issue concerns the granularity of the annotated data from D&S. We notice that the annotation was not sensitive to the order of morphemes in each numeral, and they would always annotate the numerals according to Hurford's left-headed grammar. For example, English *-teen*s were annotated as $10 + D$ instead of the real order $D + 10$ (e.g., four-teen should be $4 + 10$). The same is true for multiplications; the product of D and M is annotated as $D * M$. For example, *ìri īse* '50' in Igbo is $10*5$ (Emenanjo, 1987) but was annotated $5*10$. This presents no problem for their measures, which are not sensitive to morpheme order. However, our measures are, and this annotation results in overestimated regularity, since, for instance, the orders of M and D in natural languages often change depending on number ranges, reducing regularity. We have begun the process of reannotating their data, in order to re-run the analyses here taking this order variation into account. Early results suggest that the order-sensitive reannotation does not change our qualitative results, at least with regard to the efficiency of recursive numeral systems relative to other theoretically possible, including D&S- and Y&R-optimal systems. This is likely because natural systems maintain a high degree of regularity in order over large number ranges (e.g., only use $M + D$ over the teens range, as in English).

Unlike most work in the efficient communication literature, which only considers the *mapping* between form and meaning (cf. Kemp et al., 2018), work on recursive numeral systems like ours, D&S and Y&R focuses on the *forms* themselves. Our work specifically examines the *relationship* among forms, i.e., the regularity of the system, and shows that it is a vital property of efficient, human-like linguistic systems. This suggests, for example, that simplicity might be underestimated in work that treats compositional signals as equally complex as unanalysable ones (e.g., in grammatical categories like person systems; Zaslavsky et al., 2021). Moreover, our re-examination of D&S and Y&R also suggests that it is not sufficient simply to find a set of measures under which natural numeral systems could be shown to be optimal. To actually pinpoint why certain systems are human-like, one would want to identify the properties, like regularity and processing complexity, which explain the natural preferences of human-like systems over other possible ones, and which have plausible cognitive bases.

One clear extension of our work here is an investigation into whether natural recursive numeral systems are on the Pareto frontier of the irregularity/processing complexity trade-off. This is particularly challenging, because the space of all possible recursive numeral systems is extremely large, even when we only consider Hurford-grammar-conforming systems and limit the maximum search depth. Unlike D&S and Y&R, we would also like to avoid constraining the space by simply favouring shortest numerals or adding ad-hoc constraints. One potential solution is to adopt Y&R's enhanced grammars as representations while dropping their ad-hoc implementational constraints, allowing us to consider a much larger solution space while still keeping compact recursive numeral representations. Another complementary path would be to provide a mechanistic explanation of how these systems have been shaped over time via general learning mechanisms such as reinforcement learning, similar to Carlsson et al. (2021) and Silvi et al. (2025), and also via behavioural experiments.

Finally, a more general study that fully incorporates form-meaning mappings would yield a unifying model for numeral systems, spanning from recursive numeral systems, which have one-to-one form-meaning mappings, to exact-restricted and approximate ones, which have one-to-many mappings. In fact, a general approach to measuring structure-sensitive form-meaning systematicity would enable communicative efficiency studies on domains which are expressed compositionally in natural languages, which have hitherto been underexplored. We hope to continue the work in these directions in the future.

7 Limitations

First, we have limited computational resources, so we were not able to use a higher search depth to explore even larger spaces of possible numeral systems. However, as Figure 2 shows, the space we explore is already larger than that considered in previous work (D&S and Y&R) combined.

The second limitation concerns the format of the data. As we pointed out in the discussion, the D&S annotation always employs the same order (i.e., consistent with Hurford's left-headed grammar), which may inflate estimated regularity under our measure. While preliminary results suggest that this does not change the key findings, in future work we will analyse the newly annotated data in more detail.

The next limitation also concerns the data. D&S used the language sample from the World's Atlas of Language Structures (WALS) (Comrie et al., 2013). While, as they pointed out, WALS was constructed with a focus on language diversity, we found that the sample comprises many languages from large language families such as Indo-European and Austronesian. This could be taken as "representative" in so far as the sample is roughly proportional to the number of languages across families, but a more appropriate sample might be one with more balance across large and small families, and is controlled for geographical effects (Dryer, 1989; Ladd et al., 2015).

Finally, while our work is focused on comparing the efficiency of natural and other theoretically possible recursive numeral systems, including the previously "optimal" systems obtained in previous work, one might still wonder how efficient natural numeral systems actually are relative to the theoretical optima. Both of the previous work have shown that natural recursive numeral systems lie close to their Pareto frontier, and, as noted in the discussion, we hope to do the same in future work.

References

Henry Brighton. 2003. *Simplicity as a Driving Force in Linguistic Evolution*. Phd thesis, The University of Edinburgh.

Emil Carlsson, Devdatt Dubhashi, and Fredrik D Johansson. 2021. Learning approximate and exact numeral systems via reinforcement learning. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 43.

M. Charikar, E. Lehman, D. Liu, R. Panigrahy, M. Prabhakaran, A. Sahai, and A. Shelat. 2005. The Smallest Grammar Problem. *IEEE Transactions on Information Theory*, 51(7):2554–2576.

Sihan Chen, Richard Futrell, and Kyle Mahowald. 2023. An information-theoretic approach to the typology of spatial demonstratives. *Cognition*, 240:105505.

Bernard Comrie. 2022. The arithmetic of natural language: Toward a typology of numeral systems. *Macrolinguistics*, 10(1):1–35.

Bernard Comrie, Matthew S. Dryer, David Gil, and Martin Haspelmath. 2013. Introduction (v2020.4). In Matthew S. Dryer and Martin Haspelmath, editors, *The World Atlas of Language Structures Online*. Zenodo.

Stanislas Dehaene and Jacques Mehler. 1992. Cross-linguistic regularities in the frequency of number words. *Cognition*, 43(1):1–29.

Milica Denić and Jakub Szymanik. 2024. Recursive numeral systems optimize the trade-off between lexicon size and average morphosyntactic complexity. *Cognitive Science*, 48(3):e13424.

Milica Denić, Shane Steinert-Threlkeld, and Jakub Szymanik. 2022. Indefinite Pronouns Optimize the Simplicity/Informativeness Trade-Off. *Cognitive Science*, 46(5):e13142.

Matthew S Dryer. 1989. Large linguistic areas and language sampling. *Studies in Language. International Journal sponsored by the Foundation "Foundations of Language"*, 13(2):257–292.

Emmanuel Nolue Emenanjo. 1987. *Elements of modern Igbo grammar: A descriptive approach*. Oxford University Press.

Edward Gibson, Richard Futrell, Steven P Piantadosi, Isabelle Dautriche, Kyle Mahowald, Leon Bergen, and Roger Levy. 2019. How efficiency shapes human language. *Trends in cognitive sciences*, 23(5):389–407.

Sebastian Holt and David Barner. 2025. Learning a novel number system: The role of compositional rules and counting procedures. *Cognitive Science*, 49(6):e70071.

James Hurford. 1975. *The linguistic theory of numerals*, volume 16. Cambridge University Press.

James Hurford. 2007. A performed practice explains a linguistic universal: Counting gives the packing strategy. *Lingua*, 117(5):773–783. Linguistic perspectives on numerical expressions.

Charles Kemp and Terry Regier. 2012. Kinship categories across languages reflect general communicative principles. *Science*, 336(6084):1049–1054.

Charles Kemp, Yang Xu, and Terry Regier. 2018. Semantic typology and efficient communication. *Annual Review of Linguistics*, 4(1):109–128.

D. Robert Ladd, Seán G. Roberts, and Dan Dediu. 2015. Correlational Studies in Typological and Historical Linguistics. *Annual Review of Linguistics*, 1(1):221–241.

Ming Li, Paul Vitányi, et al. 2008. *An introduction to Kolmogorov complexity and its applications*, volume 3. Springer.

Stoyan Mihov and Klaus U. Schulz. 2019. *Finite-State Techniques: Automata, Transducers and Bimachines*, 1 edition. Cambridge University Press.

Francis Mollica, Geoff Bacon, Noga Zaslavsky, Yang Xu, Terry Regier, and Charles Kemp. 2021. The forms and meanings of grammatical markers support efficient communication. *Proceedings of the National Academy of Sciences*, 118(49):e2025993118.

Steven T. Piantadosi. 2016. A rational analysis of the approximate number system. *Psychonomic Bulletin & Review*, 23(3):877–886.

Terry Regier, Charles Kemp, and Paul Kay. 2015. Word meanings across languages support efficient communication. *The handbook of language emergence*, pages 237–263.

Andrea Silvi, Jonathan Thomas, Emil Carlsson, Devdatt Dubhashi, and Moa Johansson. 2025. Learning efficient recursive numeral systems via reinforcement learning. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 47.

Wataru Uegaki. 2022. The informativeness/complexity trade-off in the domain of boolean connectives. *Linguistic Inquiry*, pages 1–23.

Geoff Woollams. 1996. *A Grammar of Karo Batak, Sumatra*. Number 130 in Pacific Linguistics. Dept. of Linguistics, Research School of Pacific Studies, Australian National University.

Yang Xu, Emmy Liu, and Terry Regier. 2020. Numeral systems across languages support efficient communication: From approximate numerosity to recursion. *Open Mind*, 4:57–70.

David Yang and Terry Regier. 2025. Re-examining the tradeoff between lexicon size and average morphosyntactic complexity in recursive numeral systems. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 47.

Noga Zaslavsky, Charles Kemp, Naftali Tishby, and Terry Regier. 2019. Color naming reflects both perceptual structure and communicative need. *Topics in Cognitive Science*, 11(1):207–219.

Noga Zaslavsky, Mora Maldonado, and Jennifer Culbertson. 2021. Let's talk (efficiently) about us: Person systems achieve near-optimal compression. In *Proceedings of the 43rd Annual Meeting of the Cognitive Science Society*.

A Genetic Algorithm of Denić and Szymanik (2024)

In this Appendix we describe the Genetic Algorithm used by Denić and Szymanik (2024) to generate the Pareto frontier of optimal recursive numeral systems (Algorithm 1). Note that D&S to evaluate a (D, M) pair utilise Hurford's grammar to generate numerals for each number from $1-99$, keeping the shortest numeral for each of them. Yang and Regier (2025) utilise a similar algorithm but includes also suppletives S beside digits and multipliers, use the constrains over Hurford's grammar discussed in Section 2. While D&S perform mutations only on the sets D and M, Y&R includes the suppletives S and exception rules.

Algorithm 1: Genetic algorithm for Pareto frontier.

Result: Pareto-optimal population of (D, M) pairs.

Sample initial population of (D, M) pairs;
Evaluate population (via lexicon size and average morphosyntactic complexity);
for $i \in [1, \ldots, \textit{max generation}]$ **do**
  Select only Pareto-dominant (D, M) pairs;
  Perform 1 to 3 random mutations;
  Evaluate dominant systems and offspring;

B Estimating natural languages' local neighbourhood frontiers

Algorithm 2 (Section 4.2) approximates the local neighbourhood of natural recursive numeral systems. It works as follows. For a given local neighbourhood (D, M, C, L_{Num}), the algorithm works as follows. First, we fix all the numerals that have no alternative form other than the real one (e.g., Mandarin $10 + 2$ only has one alternative of length 3, namely $10 + 2$). From this initial partial system, we enumerate all the possible partial systems that have all the initial numerals, plus the combination of all numerals for the next γ largest numbers. We identify the Pareto-optimal partial systems in terms of our measures and discarded the rest. If the number of Pareto-optimal partial systems exceeded β, we sample only β systems. For each of these β systems, we enumerate again all the possible partial systems that included the combinations of numer-

Algorithm 2: Local neighbourhood frontier estimation.

Requires: digits D, multipliers M, combinators C, numeral lengths L_{Num}; max sample size β, expansion size γ, depth d.

Result: The set of local neighbourhood's optimal systems $\mathcal{O}$.

```
for i ∈ [1, ..., 99] do
    Generate numerals of number i up to depth d;
    Keep only the numeral(s) of i whose length matches L_Num[i];
N_0 = {numerals with no alternative form};
O = {N_0};
while |O[i]| < 99 ∀i ∈ {1, ..., |O|} do
    Enumerate in S the combinations of the next γ numbers' alternative numerals;
    for optimal sub-system O in O do
        for set of alternatives S in S do
            Calculate L(G) and L(N | G) of O ∪ S.
    O = {O ∪ S : O ∪ S is Pareto-dominant};
    if |O| > β then
        Sample β systems from O.
```

als for the next γ numbers. We repeat this process until we obtained the full systems that can express the numbers from 1 to 99. Note that when we are searching for the least efficient systems in the local neighbourhood, we keep in $\mathcal{O}$ only the sub-systems that are Pareto-optimal for the *negative* of our measures.

We run the algorithm with 37 out of 38 recursive numeral system types inferred from the D&S sample of natural systems. We exclude the Kayah Li-like type because they require a search depth deeper than our max depth for this task ($d = 6$) to generate a space that includes Kayah Li itself. Out of the 37 types, we set $\beta = 30$ and $\gamma = 3$ for 30 of them. We set $\beta = 10, \gamma = 2$ for the other 7 language types (Kilivila-, Abkhaz-, Gola-, Otomi-, Supyire-, Khana-, and Mangab-Mbula-like), for which sampling is too computationally expensive.