



Learning Efficient Recursive Numeral Systems via Reinforcement Learning

Downloaded from: <https://research.chalmers.se>, 2026-03-07 06:42 UTC

Citation for the original published paper (version of record):

Silvi, A., Thomas, J., Carlsson, E. et al (2025). Learning Efficient Recursive Numeral Systems via Reinforcement Learning. Proceedings of the Annual Meeting of the Cognitive Science Society, 47: 4700-4707

N.B. When citing this work, cite the original published paper.

UC Merced

Proceedings of the Annual Meeting of the Cognitive Science Society

Title

Learning Efficient Recursive Numeral Systems via Reinforcement Learning

Permalink

<https://escholarship.org/uc/item/3cc5053z>

Journal

Proceedings of the Annual Meeting of the Cognitive Science Society, 47(0)

Authors

Silvi, Andrea
Thomas, Jonathan
Carlsson, Emil
[et al.](#)

Publication Date

2025

Copyright Information

This work is made available under the terms of a Creative Commons Attribution License, available at <https://creativecommons.org/licenses/by/4.0/>

Peer reviewed

Learning Efficient Recursive Numeral Systems via Reinforcement Learning

Andrea Silvi Jonathan Thomas Emil Carlsson Devdatt Dubhashi Moa Johansson
Chalmers University of Technology and University of Gothenburg

Abstract

It has previously been shown that by using reinforcement learning (RL), agents can derive simple approximate and exact-restricted numeral systems that are similar to human ones Carlsson et al. (2021). However, it is a major challenge to show how more complex recursive numeral systems, similar to for example English, could arise via a simple learning mechanism such as RL. Here, we introduce an approach towards deriving a mechanistic explanation of the emergence of efficient recursive number systems. We consider pairs of agents learning how to communicate about numerical quantities through a meta-grammar that can be gradually modified throughout the interactions. Utilising a slightly modified version of the meta-grammar of Hurford (1975), we demonstrate that our RL agents, shaped by the pressures for efficient communication, can effectively modify their lexicon towards Pareto-optimal configurations which are comparable to those observed within human numeral systems in terms of their efficiency.

Keywords: efficient communication; reinforcement learning; numeral systems

Introduction

While there is evidence to suggest that animals, young infants and adult humans possess a biologically determined, domain-specific representation of numbers and elementary arithmetic operations, only humans have a capacity for generating an infinite set of natural numbers, while all other species seem to lack such a capacity (Hauser et al., 2002; Chomsky, 1982,9; Dehaene and Changeux, 1993; Dehaene, 1997). This unique capacity is central to many aspects of human cognition, including, of course, the development of sophisticated mathematics. The fundamental mechanism underlying this is the use of a finite symbolic system to represent arbitrarily large discrete numerical magnitudes i.e. the positive integers. In the context of AI however, there is little work on developing and changing representations of mathematical concepts, such as number systems, while most primarily concerns revising representations of logical theories Bundy and Li (2023).

In cognitive science, a recent influential body of work suggests language is shaped by a pressure for efficient communication which involves an information-theoretic tradeoff between cognitive load and informativeness (Kemp and Regier, 2012; Gibson et al., 2017; Zaslavsky et al., 2019). This means that language is under pressure to be simultaneously informative, in order to support effective communication, while

also being simple, to minimize cognitive load. Exact and approximate numeral systems were studied in Xu et al. (2020), while a mechanistic explanation of how such schemes could arise was proposed in Carlsson et al. (2021) via reinforcement learning in signaling games. However, these studies were limited to simpler exact and approximate numeral systems and do not cover more complex systems capable of representing arbitrary numerical quantities.

How does one explain the origins and development of numeral systems capable of generating expressions for arbitrary numerical quantities? Chomsky (2008) hypothesises that a fundamental operation called *Merge* can give rise to the *successor* function (i.e., every numerosity N has a unique successor, $N + 1$) in a set-theoretic fashion ($1 = \text{one}$, $2 = \{\text{one}\}$, $3 = \{\text{one}, \{\text{one}\}\}$, ...) and that the capacity for discretely infinite natural numbers may be derived from this. However, representations of numbers in natural languages do not reveal any straightforward trace of the successor function.

Another proposal is that *Merge* is able to *integrate* the two more primitive number systems mentioned above, an approximate number system for large numerical quantities and a system of precise representation of distinct small numbers. A natural way to achieve this is via a *grammar* that has two primitive components corresponding to these notions: a *base* for large approximate numerical values and *digits* for precise representations of small quantities. An example of a meta-grammar which covers this family of grammars is that proposed in the seminal work of Hurford (1975).

Such a grammar in turn would be subject to pressures to achieve a system of *efficient communication* in the information-theoretic sense of (Kemp and Regier, 2012; Gibson et al., 2017; Zaslavsky et al., 2019). Taking this line of thought, we investigate here how pressure for efficient representations would lead to the evolution of recursive numeral systems that are efficient in an information-theoretic sense in the context of artificial agents. We build on previous work by (Carlsson et al., 2021) for approximate and exact restricted systems and extend their framework to consider recursive numeral systems representations, obtained via the use of Hurford (1975)'s meta-grammar. Tradeoffs in recursive numeral systems have been recently explored in Denić and Szymanik (2024), and we adapt their framework. However, a mechanistic system to learn such efficient numeral systems was not provided in Denić and Szymanik (2024). We provide such

Correspondence to: silvi@chalmers.se.

a system in a setting where two agents interact to communicate numerosity concepts. Our agents’ emergent languages tend closer to the Pareto frontier in terms of the tradeoff between average morphosyntactic complexity and lexicon size, providing a mechanistic explanation to the work in Denić and Szymanik (2024) and complementary to the one of Carlsson et al. (2021) for the other classes of numeral systems.

Numerals	1	2	3	4	5	6	7	...
Chiquitano	Blue	Orange	Orange	Orange	Orange	Orange	Orange	Orange
Awa Pit	Blue	Orange	Green	Red	Red	Red	Red	Red
English	Blue	Orange	Green	Red	Purple	Brown	Pink	Yellow-green

Figure 1: Three distinct numeral systems from three different languages are shown. These are the approximate numerals used in *Chiquitano*, the exact-restricted numerals used in *Awa Pit* and the recursive numerals used in *English*. Colors indicate how words are assigned to numeral concepts.

Efficiency of Recursive Numeral Systems

There is a large literature on the need for efficient communication as a unifying principle that shapes human languages across several domains, see e.g. Kemp and Regier (2012); Regier et al. (2015); Gibson et al. (2017); Xu et al. (2016); Kemp et al. (2018b); Zaslavsky et al. (2018); Xu et al. (2020); Steinert-Threlkeld (2021); Mollica et al. (2021); Zaslavsky et al. (2021); Uegaki (2023); Yin et al. (2024) and the review Kemp et al. (2018a) which covers several domains. A domain where this has been particularly well studied is that of numeral systems. Under this information-theoretic framework, Xu et al. (2020) argued that numeral systems, including approximate, exact restricted and recursive systems, support efficient communication (examples of these systems are shown in Figure 1). While this argument is compelling for exact and approximate systems, it was pointed out by Denić and Szymanik (2024) that for recursive systems, the picture is more complicated. The complexity measure of Xu et al. (2020) depends both on the number of lexicalized terms and number of rules in the grammar. As noted by Denić and Szymanik (2024), while exact and restricted systems lie very close to the information-theoretic frontier in the tradeoff between simplicity and informativeness, recursive systems seem to lie off the frontier, as shown in Figure 2. The definition of complexity used by Xu et al. (2020) might not be well suited to recursive systems because these systems always maximize informativeness, since they can express any numeral. Hence, for recursive systems there is not a clear tradeoff between informativeness and this measure of complexity and hence recursive systems do not lie on the frontier. Instead, Denić and Szymanik (2024) argue that there is (at least) one more pressure shaping systems of recursive numerals in addition to simplicity and informativeness, namely *average morphosyntactic*

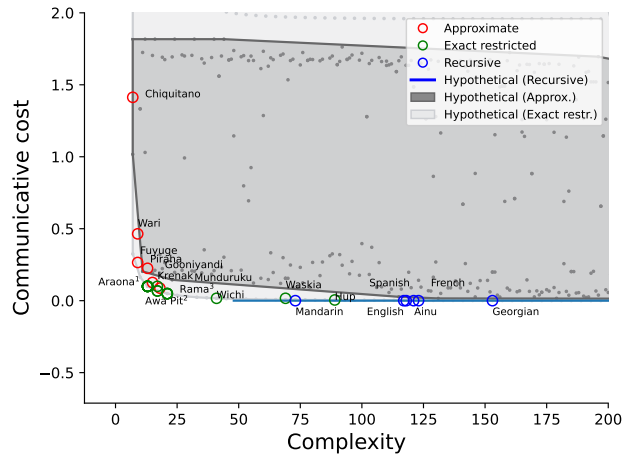


Figure 2: Reproduction of Figure 4b from Xu et al. (2020), showing that while restricted numeral systems seem to optimize the simplicity/informativeness tradeoff (here are plotted the complexity and communicative cost, their opposite), recursive numeral systems (plotted as blue dots) do not, as they lie far away from the Pareto-optimal recursive numeral system (the left-most point of the blue line).

complexity. Thus, since recursive systems achieve perfect informativeness, the appropriate tradeoff is along two axes: the number of lexicalized terms and the average morphosyntactic complexity. Denić and Szymanik (2024) show that recursive systems found in human languages optimize a tradeoff between these two measures. A natural question that emerges from this line of argument is: what mechanistic processes can optimize for an efficient tradeoff along these dimensions?

Like Denić and Szymanik (2024) we first show how the Pareto frontier of the number of lexicalised terms and average morphosyntactic complexity can be estimated using a genetic algorithm optimising a grammar for number systems like the one of Hurford (1975,9). We then introduce a slightly modified Hurford-grammar better suited for optimisation and provide a natural mechanistic procedure based on a neuro-symbolic multi-agent reinforcement learning approach, which leads to the emergence of numeral systems that are close to the Pareto frontier of efficiency between lexicon size and average morphosyntactic complexity.

Meta-grammars for recursive number systems

The grammar introduced by Hurford (1975,9) allows for the representation of numerals in natural languages. It relies on two sets: D – enumerating the set of lexicalised digits (e.g. one, two, three,...), and M – enumerating the set of multipliers (e.g. ten, hundred,...). Some examples are shown in Table 1. We use a slightly modified version of this grammar, which is given in equation (1). Our modification removes the single M from the construction of *Phrase* leaving only $Num * M$ as an option. When optimising D and M pairs this small modifica-

tion results in more natural, human-like systems as it captures that multipliers ought to have higher costs than digits.

$$\begin{aligned} \mathbf{Num} &= D \mid \mathit{Phrase} \mid \mathit{Phrase} + \mathit{Num} \mid \mathit{Phrase} - \mathit{Num} \\ \mathbf{Phrase} &= \mathit{Num} * M \end{aligned} \quad (1)$$

This meta-grammar is a mutually recursive (*Num* and *Phrase* are defined in terms of each other) non-free datatype (not every number has a canonical representation). A specific (human) number system consists of a combination of sets *D* and *M*, together with a commitment to a unique representation for each number where there are alternatives. That is, the same *D* and *M* pair can appear in several languages, differing in which representations are chosen for the numerals.

Table 1: Some *D* and *M* pairs associated with different human numeral systems.

Language	<i>D</i>	<i>M</i>
English	[1, 2, 3, 4, 5, 6, 7, 8, 9, 11]	[10]
French	[1, 2, 3, 4, 5, 6, 7, 8, 9]	[10, 20]
Kunama	[1, 2, 3, 4]	[5, 10]

Complexity Metrics

Optimization with the meta-grammar constitutes choosing appropriate sets *D* and *M*, and then the composition of the remaining numerals using the constructions operators (+, −, *).

The *morphosyntactic complexity* of a *Num* is simply defined as the size, i.e. number of symbols, in the expression built using the meta-grammar. For example, the single digit “2” has complexity 1 while “2 * 10” has complexity 3 etc. This measure is referred to as *ms_complexity*.

Recall that the meta-grammars allows non-canonical representation of numerals. For a given pair of *D* and *M*, a set of concrete languages, $\mathcal{L} = \{L_1, \dots, L_N\}$, can be induced. For each concrete language, the average morphosyntactic complexity under the need distribution can be calculated according to Equation 2. The need distribution is parameterised as $P(n) \propto n^{-2}$ which captures that smaller numbers are typically used more frequently and thus better expressed with low complexity constructs (Dehaene and Mehler, 1992; Xu et al., 2020). The evaluation is limited to natural numbers from 1 to 99, as numbers beyond this range have a very low probability under the given distribution, making their contribution to the average morphosyntactic complexity negligible.

$$\mathit{avg_ms_complexity}(\mathcal{L}) = \sum_{n \in [1, \dots, 99]} P(n) \cdot \mathit{ms_complexity}(n, \mathcal{L}) \quad (2)$$

We refer to the language which achieves the most compact representation as L_{min} :

$$L_{min} = \arg \min_{L \in \mathcal{L}} (\mathit{avg_ms_complexity}(L)) \quad (3)$$

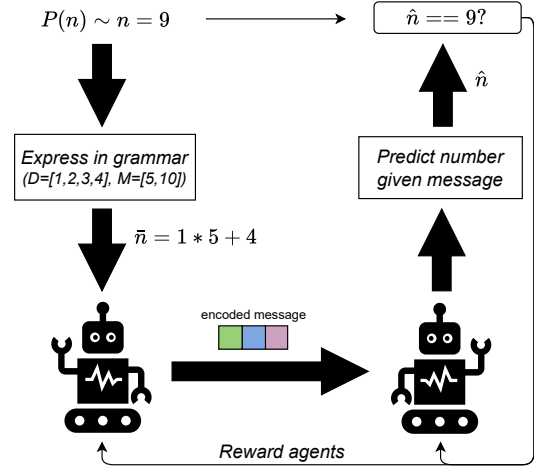


Figure 3: An example of a communication round: a number, e.g. $n = 9$, is sampled from the need distribution. The corresponding representation in terms of the current (*D*, *M*) is passed to the speaker, which encodes it neurally. The listener receives this message and outputs the number it thinks it refers to. Both agents are rewarded if the guess is correct.

L_{min} is the lower-bound on the average morphosyntactic complexity obtainable by a given *D* and *M* pair. We note that when optimising *D* and *M* pairs we are interested in minimising the measure shown in Equation 4.

$$\mathit{avg_ms_complexity}(D, M) = \mathit{avg_ms_complexity}(L_{min}) \quad (4)$$

The minimum languages associated with the final population of *D* and *M* pairs will provide an approximation to the set of optimal languages that lie on the Pareto frontier. Languages in this set are optimal in the sense that a reduction in average morphosyntactic complexity cannot be obtained without an increase in lexicon size and vice versa.

Reinforcement Learning for Grammars

Like in previous work by Carlsson et al. (2021) we study the emergence of number systems as a two-agent Lewis signaling game (Lewis, 1969) using reinforcement learning. One agent, the speaker, has to convey a numerosity encoded as a neural message, and is rewarded if the listener can reconstruct the number. Carlsson et al. (2021) left the extension to recursive numerals systems open for future work. Their agents were only capable of sending single messages, which is incompatible with the compositional nature of recursive numeral systems, and it was also unclear how to integrate a grammar structure like Hurford (1975)’s. To allow the agents to manipulate discrete grammars and send variable-length messages, we use Hurford’s grammar (Equation 1) to construct variable-length symbolic expressions that are used as input data for the speaker. These are then communicated neurally

with variable-length messages, using an LSTM architecture (Hochreiter, 1997) and reinforcement learning. The length of the speaker’s messages depends on the length of the expression of the numeral being communicated, given the current contents of D and M . The listener receives this message and outputs the number it believes it refers to. Figure 3 shows an example round of our framework.

Algorithm 1: Learning Efficient Recursive Numeral Systems

Result: Final (D, M) pair
Requires: initial (D_{curr}, M_{curr}) pair, randomly initialized speaker \mathcal{S} and listener \mathcal{L} ;
 Compute current dataset $L_{min,curr}$;
 Pre-train \mathcal{S} and \mathcal{L} on $L_{min,curr}$ via communication;
for $i \in [1, \dots, max\ steps]$ **do**
 Grammar Modification Phase
 Perform mutations to (D_{curr}, M_{curr}) from Table 2;
 Randomly sample (D_{alt}, M_{alt}) ;
 Initialize Q-values = $[0.0, 0.0]$;
 Communication Phase
 for $j \in [1, \dots, max\ iterations]$ **do**
 Sample a value n between 0.0 and 1.0;
 if $1 - \epsilon \leq n$ **then**
 Select highest Q-value grammar;
 else
 Select the least explored grammar;
 end
 Sample batch from picked grammar
 $G_k = (D_k, M_k)$ where $k \in \{curr, alt\}$;
 Train \mathcal{S} and \mathcal{L} via communication on batch;
 $Q(G_k) \leftarrow Q(G_k) + \alpha(r - Q(G_k))$;
 end
 Update (D_{curr}, M_{curr}) based on highest Q-value;
end

Algorithm 1 summarises our method for learning recursive number systems via RL communication. It takes a starting grammar (D_0, M_0) as input. From this, we compute $L_{min,0}$, i.e. the set of minimal representations for each numeral in the range 1 – 50, increasing it from Carlsson et al. (2021) who had a range of 1 to 20. The agents then use these variable-length expressions to play a signaling game: the speaker, given a numerical quantity sampled from the need distribution, retrieves the corresponding expression in $L_{min,0}$. The speaker then neurally encodes a message given this expression. The listener receives this message and then outputs the number they think the speaker is trying to represent. The agents start *tabula rasa*, i.e. are randomly initialized, and are jointly trained together via reinforcement learning: they are rewarded positively only if the listener predicts the right numerical quantity from the message communicated by the speaker. Note that in our setup only the speaker has access to the grammar, while the listener has it indirectly through the messages produced by the speaker, which are random at the

start but will obtain structure through successful communication.

After some rounds the agents stop communicating, allowing the speaker to consider modifying the current grammar (D_{curr}, M_{curr}) , with the aim of exploring possibly more efficient versions. We propose a small set of six conservative modifications that can be applied to (D_{curr}, M_{curr}) , given in Table 2. Through only applying small changes, the language shifts gradually allowing the listener to leverage some of what it has learned from previous interactions rather than starting from scratch at each iteration.

Table 2: Our set of modifications and their descriptions.

Action	Description
m_0	Add highest numeral not in D or M to D
m_1	Add highest numeral not in D or M to M
m_2	Move lowest numeral in M to D
m_3	Move highest numeral in D to M
m_4	Remove highest numeral in D
m_5	Remove highest numeral in M

We randomly sample one of these alternatives (D_{alt}, M_{alt}) , deriving a new language and the resulting alternative minimal expressions $L_{min,alt}$. This new alternative language allows us to set up recursive numeral systems-learning as a competitive game between the two grammars: at each step of communication, the speaker will sample expressions from one of the two grammars, and will then collect feedback on how successful they are at communicating numerical quantities with each of them. To choose between the two grammars, we employ a *bandit* algorithm (Lattimore and Szepesvári, 2020) (which are a class of algorithms for sequential decision making modelling the choice between arms of a slot machine as an analogy). The two number systems correspond to the bandit’s arms, which the speaker interacts with to learn which one generates the expressions that give the best reward in terms of communicative success. More specifically, the speaker stores and updates two Q-values that estimate the “goodness” of both grammars. The Q-value is updated at each communication step with the rewards collected when using the corresponding language, as $Q(G_i) \leftarrow Q(G_i) + \alpha(r - Q(G_i))$, where α is the learning rate. The Q-values are used to pick what grammar to choose: with probability ϵ the highest valued grammar is picked, otherwise the least used one is. This is to balance the *exploration-exploitation* problem, which means exploring new under-used actions while also exploiting the knowledge accumulated via the previous trials. At the end of communication, we set the new current grammar (D_{curr}, M_{curr}) to the grammar with the highest Q-value. The process then recurse until the maximum number of iterations is reached.

Implementation Details We parametrise the two agents policies as LSTMs (Hochreiter, 1997) with one embedding layer of size 5 and a single LSTM layer of hidden size 100. The speaker policy maps from expressions of variable length

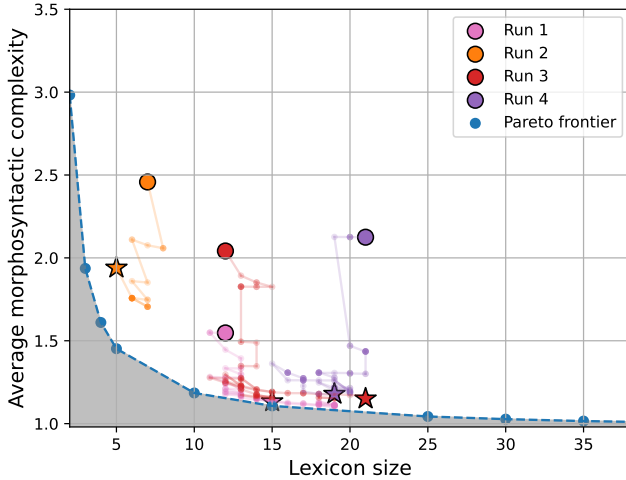


Figure 4: Trajectories showing the evolution of the languages during communication between our agents, starting from points 1-5 in Table 3 (for visual clarity not all trajectories are included). The agents tend towards languages that are closer to the Pareto frontier. The grey area represent unobtainable configurations.

to messages of the same length. The listener policy maps from variable length messages to the space of numerals considered, predicting directly the numerical quantity. We train both policies jointly using the REINFORCE loss (Williams, 1992) via two ADAM optimizers (Kingma and Ba, 2015) with learning rates of 0.002 using a batch size of 32. We additionally add an entropy term to the listener loss to encourage exploration. The default value of the bandit’s learning rate is $\alpha = 0.1$ and $\epsilon = 0.2$, while the Q-values are initialized everytime a new round of communication starts to 0.0. Each communication round lasts for 40000 epochs. For each modification to the grammar, we first check that the resulting new grammar can cover all the numbers in the range we are interested in, otherwise we discard it and thus it cannot be sampled as the alternative grammar.

Results and Discussion

Communication Leads to More Efficient Languages

In Figures 4-5, we show that our neuro-symbolic method results in (D, M) pairs that lie very close to the Pareto frontier, just like human numeral systems. The results are obtained by running Algorithm 1 for 100 iteration, starting from the (D, M) -configurations in Table 3. These are randomly selected among non-Pareto optimal (D, M) -combinations representing languages covering the entire numeral range between 1 and 50. We perform 2 runs for each such starting point (Figure 5), showing that we can obtain different end languages from the same starting point, but always tend towards the Pareto frontier.

The trajectories in 4 show that our agents tend to consistently prefer more efficient languages, which over time move

Table 3: Starting D and M pairs used in RL experiments.

#	D	M
1	[1, 2, 3, 20, 35, 37, 40, 47, 49]	[4, 25, 45]
2	[1, 4, 7, 8, 15]	[10, 33]
3	[1, 20, 25, 28, 31, 39, 41, 45]	[2,3,4,15]
4	[1, 4, 19, 21, 39, 40, 45, 47, 49]	[3,5,8,10,18,23,28,30,37,42,43,48]
5	[1, 2, 3, 4, 5, 10, 20, 30, 35, 40, 47, 49]	[6, 11, 13, 15, 45, 50]
6	[1, 2, 35, 37, 40, 47, 49]	[3, 5, 10, 20, 30, 40]
7	[1, 4, 12]	[9, 25]
8	[1, 4, 17, 22, 49]	[9, 10, 25, 28, 31, 41, 45]

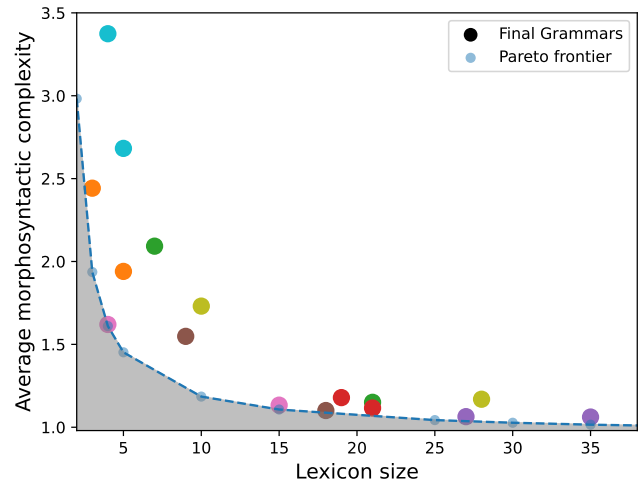


Figure 5: Final languages of our agents, plotted in terms of lexicon size and average morphosyntactic complexity. The final grammars tend to lie close to the Pareto frontier. Points are colour coded based on their starting (D, M) pair.

closer and closer to the Pareto frontier. This demonstrates that the pressures for efficient communication that come into play in our two-agents game produce languages that are more efficient in terms of tradeoff between average morphosyntactic complexity and lexicon size. These more efficient languages are easier to learn for our agents, connecting our results to works on the link between simplicity and ease of learning (Steinert-Threlkeld and Szymanik, 2020).

Our final languages in Figure 5 cover a wider span along the Pareto frontier than human languages, which tend to have a lexicon size of around 10-15. This might depend on our random starting points, or not-yet captured human constraints leading to a preference for smaller lexicon sizes. The need distribution might also influence our results: adopting the exponential need distribution used in Denić and Szymanik (2024) for the calculation of the average morphosyntactic complexity, we note that larger terms become very rare when it is used to sample numerical quantities for our agents to communicate. Further work with language evolution paradigms such as iterated learning (Smith et al., 2003) might solve this issue. We also note that some starting configurations that are further away from the Pareto frontier may require more iterations to fully reach the frontier. We kept the number of steps to 100 for consistency sake. Still, our exper-

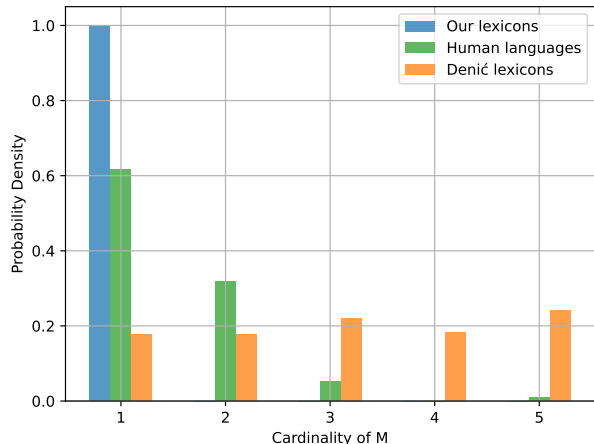


Figure 6: Comparison of lexicons induced by our meta-grammar and Hurford’s in terms of the cardinality of M .

iments show a consistent trend towards the Pareto frontier.

To summarise, our approach shows that more efficient languages are preferred, which provides a simple mechanistic explanation of why human recursive numeral systems developed to be efficient in the tradeoff between average morphosyntactic complexity and lexicon size.

Reward Structure for Numeral Systems

Compared to approximate and exact restricted numeral systems, where a word may refer to a range of numerosities, recursive numeral systems refer to exact quantities. Therefore, the proximity-based reward function used in Carlsson et al. (2021), which promotes guesses that are closer to the target value, is not suitable here. We instead reward our agents via an exact reward, i.e. only if the listener predicts the correct numerosity. This in turn makes the multi-agent reinforcement learning problem much harder, as reward is *sparser*. Adapting larger LSTMs and using a larger batch size handled this.

Characteristics of Pareto Optimal Languages

We motivate our change of meta-grammar by comparing the cardinality of M for languages on the Pareto frontier (Figure 6). This shows that our meta-grammar results in optimal languages which contain a single multiplier in M . Instead, with Hurford’s original meta-grammar, the optimal languages show an approximately uniform distribution across cardinalities of M within the permitted range. Human languages tend to have grammars with just a few multipliers in M : numeral systems with one or two multipliers are mostly common. Although our resulting lexicons are not exactly matching this, they have a consistent bias towards fewer multipliers, which was not the case if we used Hurford’s meta-grammar. Furthermore, it is notable that the genetic algorithm utilised by Denić and Szymanik (2024) imposes an artificial constraint upon the meta-grammar which restricts it to systems with no more than five elements in M with the intention of better representing human systems. Without the constraint,

optimal lexicons would mostly have an even higher cardinality of M . Instead, our modification makes it so this hard constraint is not needed when finding the Pareto frontier.

Table 4, shows examples of Pareto-optimal languages and how they differ in their composition of D s and M s. The example deduced from Hurford’s original grammar results in an L_{min} which does not have a predictable recursive structure, as the expressions for close numbers vary a lot in terms of form and M s used as basis. This is in contrast to ours and a human language from the Mixtec group (see Denić and Szymanik (2024), appendix) referred to as *Type 4-MixtecA*. We suggest that our meta-grammar, when used for optimization, results in lexicons that do not overrely on M s, like with natural human languages. A downside of our approach is that when representing natural numeral systems with our modified meta-grammar some nuances get lost, e.g. the numeral 10 in English would have to be encoded as “1*10” with our modified meta-grammar. While this is satisfactory for the optimization purposes of our current work aims, future work should try to find a grammar variant that is both closer to natural languages and work for optimization.

Table 4: A comparison of different D and M pairs associated with Pareto-optimal languages with a lexicon size of 12.

Grammar	D	M
Hurford	[1, 2, 3, 5, 6, 9, 10, 11, 14]	[4, 7, 25]
(1) (ours)	[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]	[12]
4-MixtecA	[1, 2, 3, 4, 5, 6, 7, 8, 9]	[10, 15, 20]

Conclusions and Future Work

Our neuro-symbolic method shows how efficient recursive numeral systems develop as a trade-off between morphosyntactic complexity and lexicon size, through a combination of two-agent signaling games, reinforcement learning and symbolic language modifications. A minor modification to Hurford’s well-established meta-grammar for expressing numeral systems made it more suitable for optimisation via RL. Our contributions provide a mechanistic explanations for the emergence of efficient recursive numeral systems, supporting the work of Denić and Szymanik (2024). In future work, we intend to explore how cultural transmission paradigms like iterated learning might influence the languages our agents develop, possibly leading to languages even closer to human ones in terms of average morphosyntactic complexity and lexicon size, as Carlsson et al. (2024) demonstrates for the colours domain and Guo et al. (2019) for numerical quantities. We also want to explore how different need distributions affect the efficiency of our agents’ languages. Finally, we are interested in creating a unifying framework that can demonstrate how approximate, exact restricted and recursive numeral systems might emerge from the same system and possibly transition from one to the other through communication and inter-generational transmission.

Acknowledgments

This work was supported by funding from the Swedish Research Council and the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. The computations in this work were enabled by resources provided by the Swedish National Infrastructure for Computing (SNIC).

References

- Alan Bundy and Xue Li. Representational change is integral to reasoning. *Phil. Trans. R. Soc. A.*, 2023. <http://doi.org/10.1098/rsta.2022.0052>.
- Emil Carlsson, Devdatt Dubhashi, and Fredrik D. Johansson. Learning approximate and exact numeral systems via reinforcement learning. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, 43, 2021. URL <https://doi.org/10.48550/arXiv.2105.13857>.
- Emil Carlsson, Devdatt Dubhashi, and Terry Regier. Cultural evolution via iterated learning and communication explains efficient color naming systems. *Journal of Language Evolution*, 9(1-2):49–66, 2024.
- Noam Chomsky. A note on the creative aspect of language use. *The Philosophical Review*, 91(3):423–434, 1982.
- Noam Chomsky. *Knowledge of language: Its nature, origin, and use*. Greenwood Publishing Group, 1986.
- Noam Chomsky. On phases. *Current Studies in Linguistics Series*, 45:133, 2008.
- Stanislas Dehaene. What are numbers, really? a cerebral basis for number sense. *reprinted in Edge March*, 19:2016, 1997.
- Stanislas Dehaene and Jean-Pierre Changeux. Development of elementary numerical abilities: A neuronal model. *Journal of cognitive neuroscience*, 5(4):390–407, 1993.
- Stanislas Dehaene and Jacques Mehler. Cross-linguistic regularities in the frequency of number words. *Cognition*, 43(1):1–29, 1992.
- Milica Denić and Jakub Szymanik. Recursive numeral systems optimize the trade-off between lexicon size and average morphosyntactic complexity. *Cognitive Science*, 48(3):e13424, 2024. <https://doi.org/10.1111/cogs.13424>. URL <https://onlinelibrary.wiley.com/doi/abs/10.1111/cogs.13424>.
- Edward Gibson, Richard Futrell, Julian Jara-Ettinger, Kyle Mahowald, Leon Bergen, Sivalogeswaran Ratnasingam, Mitchell Gibson, Steven T. Piantadosi, and Bevil R. Conway. Color naming across languages reflects color use. *Proceedings of the National Academy of Sciences*, 2017. ISSN 0027-8424.
- Shangmin Guo, Yi Ren, Serhii Havrylov, Stella Frank, Ivan Titov, and Kenny Smith. The emergence of compositional languages for numeric concepts through iterated learning in neural agents. *arXiv preprint arXiv:1910.05291*, 2019.
- Marc D Hauser, Noam Chomsky, and W Tecumseh Fitch. The faculty of language: what is it, who has it, and how did it evolve? *science*, 298(5598):1569–1579, 2002.
- S Hochreiter. Long short-term memory. *Neural Computation MIT-Press*, 1997.
- James R. Hurford. *The Linguistic Theory of Numerals*. Cambridge University Press, Cambridge, 1975.
- James R. Hurford. *Language and Number: The Emergence of a Cognitive System*. 1987. URL <https://api.semanticscholar.org/CorpusID:144840779>.
- Charles Kemp and Terry Regier. Kinship categories across languages reflect general communicative principles. *Science (New York, N.Y.)*, 336:1049–54, 05 2012.
- Charles Kemp, Yang Xu, and Terry Regier. Semantic typology and efficient communication. *Annual Review of Linguistics*, 4:109–128, 01 2018a.
- Charles Kemp, Yang Xu, and Terry Regier. Semantic typology and efficient communication. *Annual Review of Linguistics*, 4(1):109–128, 2018b.
- Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL <http://arxiv.org/abs/1412.6980>.
- Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.
- David K. Lewis. *Convention: A Philosophical Study*. Wiley-Blackwell, 1969.
- Francis Mollica, Geoff Bacon, Noga Zaslavsky, Yang Xu, Terry Regier, and Charles Kemp. The forms and meanings of grammatical markers support efficient communication. *Proceedings of the National Academy of Sciences*, 118(49): e2025993118, 2021.
- Terry Regier, Charles Kemp, and Paul Kay. Word meanings across languages support efficient communication. *The handbook of language emergence*, pages 237–263, 2015.
- Kenny Smith, Simon Kirby, and Henry Brighton. Iterated learning: A framework for the emergence of language. *Artificial life*, 9(4):371–386, 2003.
- Shane Steinert-Threlkeld. Quantifiers in natural language: Efficient communication and degrees of semantic universals. *Entropy*, 23(10):1335, 2021.
- Shane Steinert-Threlkeld and Jakub Szymanik. Ease of learning explains semantic universals. *Cognition*, 195:104076, 2020.
- Wataru Uegaki. The informativeness/complexity trade-off in the domain of boolean connectives. *Linguistic Inquiry*, 55(1):174–196, 2023.
- Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8:229–256, 1992.

- Yang Xu, Terry Regier, and Barbara C Malt. Historical semantic chaining and efficient communication: The case of container names. *Cognitive science*, 40(8):2081–2094, 2016.
- Yang Xu, Emmy Liu, and Terry Regier. Numeral Systems Across Languages Support Efficient Communication: From Approximate Numerosity to Recursion. *Open Mind*, 4:57–70, 2020.
- Kayo Yin, Terry Regier, and Dan Klein. Pressures for communicative efficiency in american sign language. In *Annual Conference of the Association for Computational Linguistics (ACL)*, August 2024.
- Noga Zaslavsky, Charles Kemp, Terry Regier, and Naftali Tishby. Efficient compression in color naming and its evolution. *Proceedings of the National Academy of Sciences*, 115(31):7937–7942, 2018.
- Noga Zaslavsky, Charles Kemp, Naftali Tishby, and Terry Regier. Color naming reflects both perceptual structure and communicative need. *Topics in Cognitive Science*, 11(1): 207–219, 2019.
- Noga Zaslavsky, Mora Maldonado, and Jennifer Culbertson. Let’s talk (efficiently) about us: Person systems achieve near-optimal compression. In *Proceedings of the annual meeting of the cognitive science society*, volume 43, 2021.