Regular Reduplication Across Modalities

Amy Li[1] **Jonathan Rawski**[1,2]
[1]Massachusetts Institute of Technology [2]San Jose State University
{amyli785, rawski}@mit.edu

Overview

Reduplication is a common morphological process of copying, with a wide-ranging typology. Reduplication is among the most computationally complex phenomena in natural language, is difficult for modern machine learning methods to learn (Deletang et al., 2022), and yet is inhabits a highly restricted and structured computational class. However, this restrictiveness is mostly known based on spoken language typology, when reduplication is far more ubiquitous, expressive, and varied in signed languages. Here we consider signed reduplication, and analyze a unique type of 'embedded' reduplication which may require supra-regular computation. We show that in fact it is still regular, further evidence that reduplication is regular across modalities. We discuss some lingering issues related to reduplicative computation across speech and sign.

Generative Capacity of Copying

Reduplication is a morphological process of copying, i.e. a function which output some finite number of copies of a word: $w \to ww^n$, which can be the whole string or a bounded substring. While the output language of reduplication, $\{ww | w \in \Sigma^*\}$, is mildly-context-sensitive, the function itself is a ***regular*** transduction, the generalization of regular languages, which have various algebraic, automata, and logical characterizations (Filiot and Reynier, 2016). Regular transductions possess several defining properties: they are composition-closed, the output length is at most a linear growth on the input $|f(w)| = \mathcal{O}(|w|)$, and the number of copies is independent of the input length. Rawski et al. (2023) show that partial, total, triplication, and other reduplication types are regular functions, since they possess these properties. Sign languages ubiquitously exhibit partial and total copying, and make far more use of triplication for plurals, recirocals, and more (Wilbur, 2009).

Embedded Copying

As well as possessing all the types of reduplication that spoken languages do, signed languages have types of reduplication that spoken languages do not. Wilbur (2009) analyzes a case of 'embedded' reduplication from Klima and Bellugi (1979) as a productive process. An example from ASL using the verb 'GIVE' is shown below. Aspectual inflection is handled via reduplication of a certain number of times, and each of these reduplications can be composed to get different meanings, as seen below. This process is intriguing because it potentially violates the property of linear growth inherent to regular functions. if the output size of various embedded reduplications is supra-linear in terms of the input, so for some polynomial k, $|f(w)| = \mathcal{O}(|w|^k)$, then the function is polyregular, a superclass of the regular functions (Bojanczyk, 2022).

Embedded Reduplication is Regular

Our main contribution is to show that embedded reduplication is regular. To do so, we first note that each of the individual aspectual inflections constitute a regular function, since they each copy the underlying form some finite number of times (Wilbur et al., 1983). Signs exhibit significant simultaneous structure compared to strings, and are normally formalized as graphs as opposed to strings. We formalize the regularity using MSO-transductions (Courcelle and Engelfriet, 2012) which can make finite copies of an input graph and define orderings on the output using formulas in Monadic second-order logic, and compute exactly regular functions as noted earlier. We show an total reduplication example below using Sandler (1989)'s Hand Tier model of the sign.

Following the literature on model theory (Courcelle and Engelfriet, 2012) we represent a sign as a graph. There are many possible

Proceedings of the Society for Computation in Linguistics (SCiL) 2024, pages 370-373.
Irvine, California, June 27-29, 2024

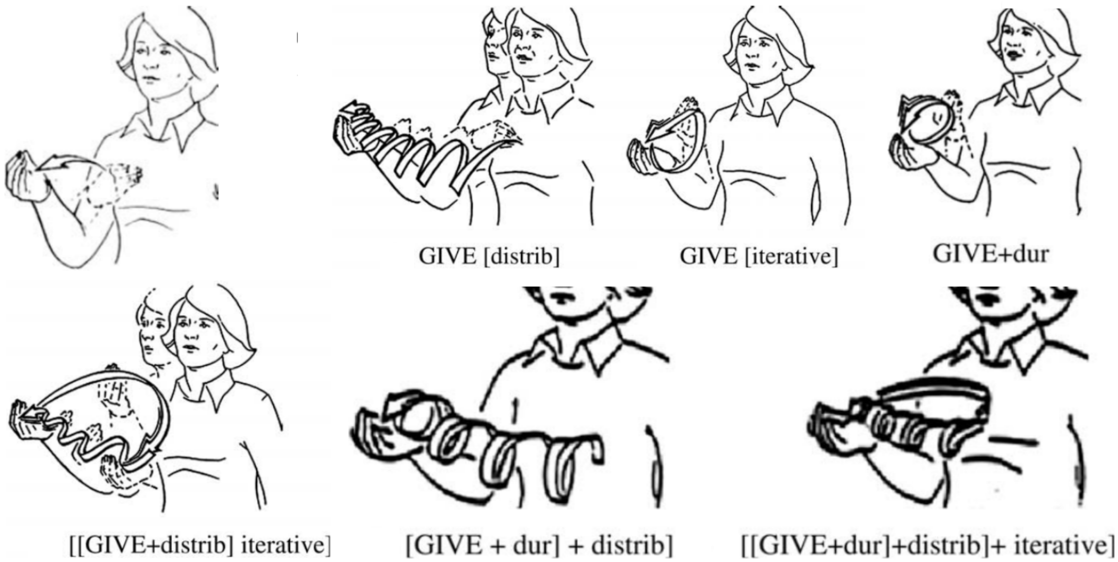

Figure 1: Top: ASL 'GIVE' in citation form (left), and inflected for distributive, iterative, and durative aspect (right). Bottom: Embedded reduplication of 'GIVE'. Left: 'give to each of them over and over again'. Center: 'keep on giving to each person, one after another'. Right: 'keep on giving to each person, one after another, this event sequence recurring regularly over expanses of time.' Adapted from Wilbur (2009); Klima and Bellugi (1979)

representations of the sign, but here we use Sandler (1989)'s autosegmental 'Hand-Tier' model of the sign, whose signature is $\mathcal{S}^{HT} \stackrel{\text{def}}{=} \langle \mathcal{D}; L, M, H_i, P_j; A(x,y), loc(x,y); \triangleright(x,y)\rangle$. The domain $\mathcal{D}$ is a finite set of word elements $\{1,\ldots,n\}$. A unary relation, $\sigma(x)$, holds true if the sign labels the x-th element with a symbol σ from some alphabet Σ. We consider Location, Movement, Handshape, and Place as unary relations $\{L, M, H_i, P_j\}$, where i, j refer to one out of a set of possible handshapes or places, respectively. The binary relation $x \triangleright y$ is the standard successor relation between temporally adjacent positions. The association relation $A(x,y)$ associates elements on the LM-tier to the handshape tier, and the location relation $loc(x,y)$ relates elements on the LM tier to specific elements on the Place tier. Note that these are both association relations, but location is labeled for clarity. A representation of a monosyllabic sign with two place features and one handshape is given in Figure 2.

We define first-order (FO) formulas ϕ using countably many first-order variables $x, y, \ldots$, incorporating the usual set of Boolean connectives $\wedge, \vee, \neg, \rightarrow$, and employing quantifiers $\forall x, \exists x$. Additionally, monadic second-order (MSO) formulas allow for countably many second-order set variables $X, Y, \ldots$, use a predicate $x \in X$ that holds true when the position assigned to x is part of the set assigned to X, and permit quantification over set variables ($\forall X, \exists X$).

Given a formula $\phi(x_1, \ldots, x_m)$ and structure $w \in \Sigma^*$, we write $w \models \phi$ to denote that $\mathcal{S}_w$ satisfies ϕ. The language defined by a formula ϕ is the set $\{w \mid w \models \phi\}$.

A logical *transducer* uses this setup to transform input structures to output structures, rather than to Boolean values, defining the output word structure in terms of the input word structure. It does so by creating a finite number of *copies* of the input word structure's domain, and using formulas to define the output's labeling and precedence relations over these copies.

A logical transducer is a tuple $T = \left(\Sigma, \Gamma, \phi_{\mathcal{D}}, C, \phi^c_\sigma(x), \phi^{c,c'}_\triangleright(x,y)\right)$ where

- Σ and Γ are the input and output alphabets,
- C is a finite set of (indices of) copies of the input domain,
- $\phi_{\mathcal{D}}$ defines the set of structures on which the transduction is defined,
- $\phi^c_\gamma(x)$ is true iff the symbol at position x of copy c is labeled γ, for each $\gamma \in \Gamma$ and $c \in C$,
- $\phi^{c,c'}_\triangleright(x,y)$ is true iff position x of copy c immediately precedes position y of copy c', for each $c, c' \in C$,

and all formulas $\phi_{\mathcal{D}}$, ϕ^{c}_{σ}, and $\phi^{c,c'}_{<}$ are FO- (or MSO-) definable.

For a total reduplication transduction, there are two copies. Nothing is deleted, so the domain function $\phi_{\mathcal{D}}$ is true for every element, i.e. each input element survives in the output. Each unary relation $\phi^{c}_{\gamma}(x)$ in the output is merely the identity from the input, i.e. labels are preserved from input to each copy. The interesting piece of the transduction is the ordering relation, which connects the last and first elements of the base and reduplicant, respectively. To refer to these, we can define some helper predicates

$$\texttt{first}(x) \overset{\text{def}}{=} \neg\exists y, y \triangleright x \tag{1}$$

$$\texttt{last}(x) \overset{\text{def}}{=} \neg\exists y, x \triangleright y \tag{2}$$

$$\texttt{first}_H(x) \overset{\text{def}}{=} \texttt{first}(x) \wedge H(x) \tag{3}$$

$$\texttt{first}_{LM}(x) \overset{\text{def}}{=} \texttt{first}(x) \wedge (L(x) \vee M(x)) \tag{4}$$

$$\texttt{first}_P(x) \overset{\text{def}}{=} \texttt{first}(x) \wedge H(x) \tag{5}$$

$$\texttt{last}_H(x) \overset{\text{def}}{=} \texttt{last}(x) \wedge H(x) \tag{6}$$

$$\texttt{last}_{LM}(x) \overset{\text{def}}{=} \texttt{last}(x) \wedge (L(x) \vee M(x)) \tag{7}$$

$$\texttt{last}_P(x) \overset{\text{def}}{=} \texttt{last}(x) \wedge H(x) \tag{8}$$

Using these predicates, we can define the successor relation in the output. To do so, we must define the relation between each copy i, j

$$\begin{aligned} &\phi^{i,j}_{\triangleright}(x,y) \\ &\quad \overset{\text{def}}{=} ((x < y) \wedge (i = j)) \vee \\ &(\texttt{last}_H(x) \wedge \texttt{first}_H(y) \wedge (i = j-1)) \vee \\ &(\texttt{last}_{LM}(x) \wedge \texttt{first}_{LM}(y) \wedge (i = j-1)) \vee \\ &(\texttt{last}_P(x) \wedge \texttt{first}_P(y) \wedge (i = j-1)) \end{aligned} \tag{9}$$

An example of this transduction over a monosyllabic sign like 'GIVE' is shown graphically in Figure 2. What happens when this output itself is reduplicated? Applying the total reduplication transduction again results in two copies of the structure, which are joined together in the same way, shown graphically in Figure 2. It is clear that if one were to define another transduction to copy an input 4 times, the output would be the same as copying, then copying again. All that would change is the number of copies in the copy set C, from 2 to 4. Additionally, composing two copying functions whose number of copies differ, say reduplication followed by triplication does not make a difference. Since regular transductions are closed under composition, the overall mapping is regular, meaning embedding aspectual reduplications does not increase the expressivity of the mapping.

Issues in Regular Signed reduplication

At first glance, the result that reduplication across modalities is regular falls out straightforwardly from the nature of the functions. However, there are some factors that impact the expressive power of a reduplicative function as noted by Rawski et al. (2023). The first issue is the bound on copies. Each individual reduplication copies a sign some finite number of times. Signers may variably copy more or less even when the reduplicant is fixed, as in the case of pluralization by triplication where users may copy between 2 and 5 times when the canonical form is 3 (van Boven, 2021). Another issue is whether the composition is finite, i.e. whether something like [[GIVE + durative] + distributive] but also [[[[GIVE + durative] + distributive] + durative] + distributive] is allowed. Wilbur et al. (1983) provide statistical evidence suggesting that certain combinations are strongly dispreferred, but more typological and production evidence may assist here. If such evidence suggests the number is unbounded, regularity depends on how the semantic input is structured.

This semantic input is also crucial: if the input is a word w plus a sequence of identical features F i.e. $wF^n \rightarrow ww^n$, then it is polyregular, but if it is just w, then it is regular. As Rawski et al. (2023) note, the computation depends on if a theory uses cyclicity, bounds on the number of copies, and how it represents the semantically-based input to reduplication. If factorized with cycles of the same function, then each cycle of a function would be regular. Using a single collapsed function, the computational expressivity depends on how many times a copy can be generated. If a bound on the number of copies exists, then the finite composition of multiple total reduplication functions is a single regular function. If there is no bound, regularity depends on how the input w looks and how

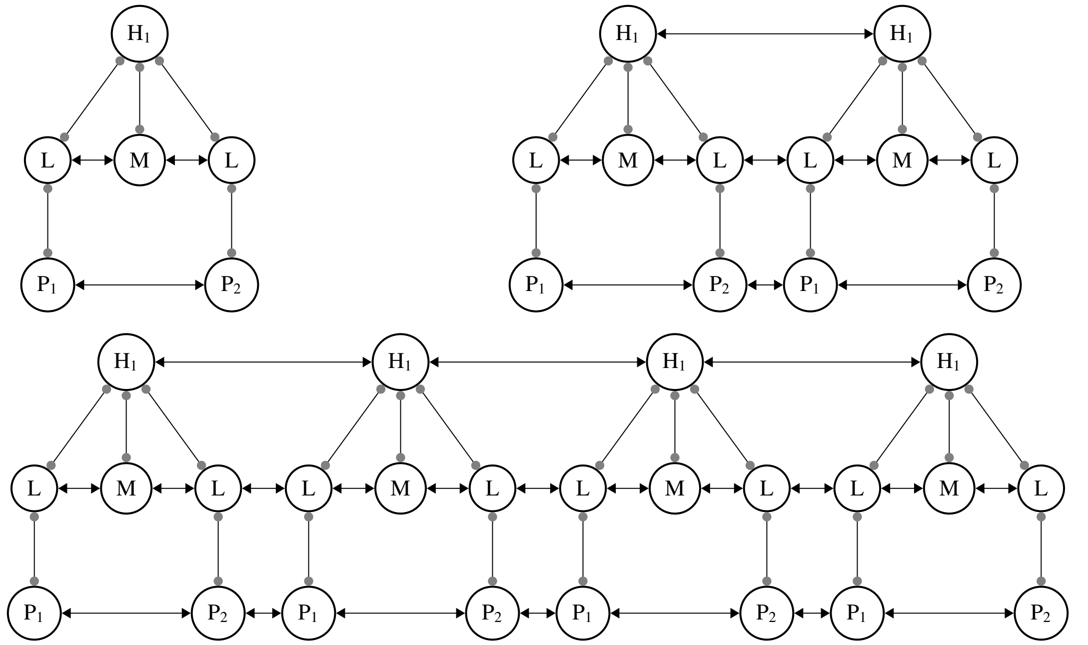

Figure 2: Total reduplication (top right) and multiple reduplication (bottom) of a monosyllabic sign with one handshape and two place features (top left)

the mapping between semantic features F to the input works. If the input it treated as a sign plus a sequence of identical features F, then this process is reduced to the polyregular 'input-specified' copying: $w\ F^n \rightarrow ww^n$.

This perspective provides an empirical question for sign researchers regarding the nature of the input to reduplication. Undoubtedly, there will be modality effects, but this computational perspective allows us to ground those findings. Reduplication appears to be regular across speech and sign. Our finding supports a computationally grounded amodal upper bound on natural language morphophonology.

References

Mikolaj Bojanczyk. 2022. Transducers of polynomial growth. In *Proceedings of the 37th Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 1–27.

Bruno Courcelle and Joost Engelfriet. 2012. *Graph structure and monadic second-order logic: a language-theoretic approach*, volume 138. Cambridge University Press.

Gregoire Deletang, Anian Ruoss, Jordi Grau-Moya, Tim Genewein, Li Kevin Wenliang, Elliot Catt, Chris Cundy, Marcus Hutter, Shane Legg, Joel Veness, et al. 2022. Neural networks and the chomsky hierarchy. In *The Eleventh International Conference on Learning Representations*.

Emmanuel Filiot and Pierre-Alain Reynier. 2016. Transducers, logic and algebra for functions of finite words. *ACM SIGLOG News*, 3(3):4–19.

Edward S Klima and Ursula Bellugi. 1979. *The signs of language*. Harvard University Press.

Jonathan Rawski, Hossep Dolatian, Jeffrey Heinz, and Eric Raimy. 2023. Regular and polyregular theories of reduplication. *Glossa: a journal of general linguistics*, 8(1).

Wendy Sandler. 1989. *Phonological representation of the sign: Linearity and nonlinearity in American Sign Language*, volume 32. Walter de Gruyter.

Cindy van Boven. 2021. Phonological restrictions on nominal pluralization in sign language of the netherlands: evidence from corpus and elicited data. *Folia Linguistica*, 55(2):313–359.

R.B. Wilbur, E.S. Klima, and U. Bellugi. 1983. Roots: on the search for the origins of signs in asl. In *Chicago Linguistic Society 19*.

Ronnie B Wilbur. 2009. Productive reduplication in a fundamentally monosyllabic language. *Language Sciences*, 31(2-3):325–342.