The Logical Nature of Phonology Across Speech and Sign

Jonathan Rawski, Stony Brook University

Abstract

This article examines whether the computational properties of phonology hold across spoken and signed languages. Model-theoretic representations of spoken and signed words, as well as logical mappings over these structures, are introduced as a powerful framework for structural and computational comparisons. Several phonological processes in sign are shown to require the same logical complexity as their spoken counterparts, suggesting an amodal sensitivity to notions of locality and memory, as well as a computational tradeoff between sequentiality and simultaneity in specific modalities. These analyses provide a necessary and sufficient condition for amodal aspects of phonology, and allow for promising new means to analyze issues of linguistic modality and the cognitive status of phonological knowledge.

1 Introduction

Sign languages arise spontaneously in deaf communities, are acquired during childhood through normal exposure without instruction, and exhibit all of the facets and complexity found in spoken languages (see Sandler and Lillo-Martin (2006) for a groundbreaking overview). However, if human language operates without respect to modality, we should find hearing communities that just happen to use sign language instead of spoken language, and we do not. Sign languages are thus "an adaptation of existing physical and cognitive systems for the purpose of communication among people for whom the auditory channel is not available" (Sandler, 1993).

The presence of sign language phonology poses a challenge for theories of the phonological module, since the structure and organization of phonological form is influenced in part by the physiology of the systems that produce and perceive them. However, striking similarities have been found in phonological systems across modalities, leading some to argue for an "algebraic" phonology of computational rules (Berent, 2013). Others argue for an abstract "substance-free" phonology in either modality, which is cognitively independent from the physiology but mediates between grammatical form and perceptual form (Reiss, 2018). Sandler and Lillo-Martin (2006) take up this challenge and define phonology across modalities as "the level of linguistic structure that organizes the medium through which language is transmitted".

Sign languages offer, as Sandler (1993) puts it, "a unique natural laboratory for testing theories of linguistic universals and of cognitive organization." They give insight into the concrete contents of grammatical form, and conditions on which aspects of grammar are amodal and which are tied to the modality. Schlenker (2018), speaking about semantics, notes that "investigating Universal Semantics from the standpoint of sign language might help reconsider foundational questions about the logical core of language, and its expressive power".

How then can we study in a principled and meaningful way the **expressivity** of phonology across speech and sign? Computational characterizations of phonological processes distill the necessary and sufficient conditions of any system that performs them. Heinz (2018) writes that understanding the computational nature of phonology requires determining the nature of 1) the abstract, underlying representations, 2) the concrete, surface representations, and 3) the transformation from underlying forms to surface forms. Each clarifies the properties to which a cognitive mechanism needs to be sensitive in order to correctly classify and process forms. As an alternative to developing a specific computational model, one can determine abstract measures of complexity that "are invariant across all possible cognitive mechanisms and depend only on properties that are necessarily common to all computational models that can distinguish a pattern" (Rogers et al., 2013).

This article analyzes the nature of such representations by using elements of Finite Model Theory, a branch of mathematics known in semantics but not often applied in the study of linguistic form. Model Theory refines the computational characterization by describing the *content* of linguistic structures. The computational expressivity of processes over these representations is analyzed with statements in mathematical logic. This approach has previously been used in linguistics to characterize and compare particular grammatical theories in syntax and phonology (Rogers, 1998; Potts and Pullum, 2002; Pullum, 2007; Graf, 2010). This article applies the model-theoretic and logical approach to the phonology itself and examines the relationship between representation and computation in phonological processes across modalities, on their own terms.

Why the focus on logic? The connections between language, mathematical logic and computational models have a long research history, dating back to the foundations of theoretical computer science. Some of the earliest results in linguistic theory established the place of natural language patterns within the theory of computable functions (Chomsky, 1956). Phonological transformations are sufficiently characterized by finite-state machines (Johnson, 1972; Kaplan and Kay, 1994; Karttunen, 1993), meaning the computation is **regular**, or characterized by a finite bounded memory. Foundational work by Büchi (1960) and Engelfriet and Hoogeboom (2001) showed that finite-state machines are equivalent to statements in monadic second-order logic. This logic-computation connection shows how to relate the **specification** of the phonological system (as given by a logical formula) to a possible **implementation** (as the finite-state behavior), a distinction used in the Declarative Phonology framework (Scobbie et al., 1996). One can understand the nature of a process in terms of the information and computation needed to perform it, especially for representations where the structure of the automaton is not readily apparent.

The main result of this study is that phonology is amodally sensitive to a logical notion of computational locality, relativized over different structural properties given by the modality. The boundedness of signed words, and the tendency of phonological typology towards this locality, predicts that most sign phonology is sufficiently characterized by this property. This suggests tradeoffs in the organization of phonological representations — more sequential structure in speech, and more simultaneity in sign. The model-theoretic approach in this article clarifies these issues and their implications for the nature of phonological cognition across modalities.

The article proceeds as follows. Section 2 overviews the model-theoretic approach to phonological representation and considers various representations of spoken and signed words. Section 3 describes phonological processes as logical transformations over word models. It distinguishes the logical power in terms of the type of quantification needed: Quantifier-Free, First-Order, or Monadic Second-Order. Section 4 demonstrates that many processes are computationally local, i.e. Quantifier-Free, across spoken and signed phonology. Section 5 discusses tradeoffs in sequentiality and simultaneity in modalities to meet this computational restriction. Section 6 discusses some implications for the cognitive status of phonological knowledge.

2 Model-Theoretic Representations of Spoken and Signed Words

This section defines the central ideas of model-theoretic representations and considers various representations of words and signs. This involves deciding what kind of objects we are reasoning about and what relationships between them we are reasoning with. Model theory provides a unified ontology and a vocabulary for representing many kinds of objects, by considering them as **relational structures** (see Libkin (2004) for a thorough introduction). This allows flexible but precise definitions of the structural information in an object, by explicitly defining its parts and the relations between them. This makes model-theoretic representations a powerful tool for analyzing the information characterizing a certain structure.

For example, one common way of representing phonological words is as a string. Strings are sequences of events, each labeled with particular symbols that describe **properties** of those events. These properties might be the speech sound symbols in the IPA, phonological features, or a set of orthographic symbols, but for the present purposes let us, without loss of generality, consider a set of properties $\Sigma = \{a, b, t\}$. Strings are combinations of these symbols at certain events, like the word *baba*.

Model-theoretic representations for finitely-sized objects like strings contain two parts. The first is a finite set of elements $\mathcal{D}$, called the *domain*, taken here to be elements of the natural numbers, as is common. The second is a finite set of k-ary *relations* $\mathcal{R}$, and *functions*[1] $\mathcal{F}$, which are subsets of the domain. The relations and functions provide information about the domain elements. The *model signature* $\mathcal{M} = \langle \mathcal{D}; \mathcal{R}; \mathcal{F} \rangle$ collects these parts and defines the nature of the model in terms of the information in the representation. One model signature for words, called the *successor model*, is given below in (1)

(1) $$\mathcal{M}^S \stackrel{\text{def}}{=} \big\langle \mathcal{D}; \mathcal{R} = \{R_\sigma \mid \sigma \in \Sigma\}; \mathcal{F} = \{p(x), s(x)\} \big\rangle$$

This model says that for every property σ in the set of properties Σ, there is a unary relation R_σ in $\mathcal{R}$ that can be thought of as a labeling relation for that symbol. For our set $\Sigma = \{a, b, c\}$, $\mathcal{R}$ includes the unary relations R_a, R_b, R_c. The two unary functions in $\mathcal{F}$, $p(x)$ and $s(x)$, describe a linear order over the domain elements by picking out the immediate *predecessor* and *successor* of some given position, respectively. In general, $s(x) = x + 1$ and $p(x) = x - 1$. Additionally, the predecessor function is often defined so that the initial position is its own predecessor, i.e. $p(0) = 0$, so that it is a total function. Similarly, the final position is its own successor, so the successor function is also total. Then in a string of n positions, $s(n) = n$.

As an example, the model for the word $baba$ under this theory is denoted $\mathcal{M}^S_{baba}$. For the properties $\Sigma = \{a, b, t\}$, $\mathcal{M}^S_{baba}$ is defined and represented visually in Figure 1. Here the word model's domain $\mathcal{D}$ consists of four nodes, each as a node with an index below it. Unary relations are illustrated as node labels. For example, node 2 is labeled a. The successor and predecessor functions are illustrated by directed edges (arrows), with a black or white arrowhead, respectively (i.e. $s(1) = 2$).

Model vocabulary items are also called *atomic formulas*, because they are the primitive terms from which larger logical expressions are built. Let x, y, etc., be variables and then let $p(x) = y$, $s(x) = y$ and $R_\sigma(x)$ for each σ in the set of properties be atomic predicates. These variables will be

[1] It is true that every k-ary function can be considered as a $(k+1)$-ary relation, and every constant as a 0-ary function or (singleton) unary relation, but for explicitness I keep them this way

$$\mathcal{M}^S_{baba}$$
$$\mathcal{D} = \{1,2,3,4\}$$
$$R_a = \{2,4\}$$
$$R_b = \{3\}$$
$$R_t = \{\varnothing\}$$
$$p(x) = \begin{cases} 1 & when \quad x \in \{1,2\} \\ 2 & when \quad x = 3 \\ 3 & when \quad x = 4 \end{cases}$$
$$s(x) = \begin{cases} 2 & when \quad x = 1 \\ 3 & when \quad x = 2 \\ 4 & when \quad x \in \{3,4\} \end{cases}$$

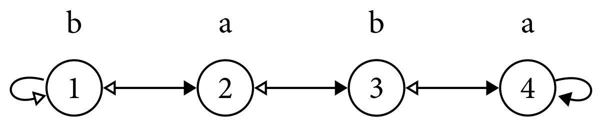

Figure 1: Successor word model for *baba*

assigned values in $\mathcal{D}$ in a model. $s(x) = y$ will evaluate to true if and only if y is the successor of x in $s(x)$ in that model. For example, $s(x) = y$ is true in Fig. 1 when x is assigned to 1 and y is assigned to 2, but not when x is assigned to 1 and y is assigned to 4. Similarly, $a(x)$ is true in a model only when x is assigned to an element in a in that model. For example, $a(x)$ is true in Fig. 1 when x is assigned to 2 (denoted $2 \in R_a$, $R_a(2) = \texttt{TRUE}$, or, equivalently, $a(2) = \texttt{TRUE}$), but not when it is assigned to 1. I also add an atomic predicate $x = y$ that evaluates to true when x and y are assigned to the same element in $\mathcal{D}$.

Atomic formulas are also predicates, so for a model theory we can implicitly define the full set of Boolean connectives such as conjunction, disjunction, implication, and negation, as well as quantifiers. This means that given a model theory $\mathcal{M}$, logical predicates can be defined to make it easier to refer to certain types of information in $\mathcal{M}$.

For example, if the atomic properties of the sequential elements are taken to be a set of phonological features like $+\texttt{voice}(x)$ or $-\texttt{continuant}(x)$, one may use multiple unary relations to describe the same domain element, and therefore refer to individual phonemes with user-defined predicates, such as in (2-3). One might also wish to pick out certain privileged elements in the word, such as word edges, as in (4-5). Constants may also be a part of the model signature. A modified representation of the word *baba* in the successor model using these predicates is shown in Figure 2.

(2) $$\mathrm{b}(x) \overset{\text{def}}{=} +voice(x) \wedge +\texttt{labial}(x) \wedge -\texttt{continuant}(x)$$

(3) $$\mathrm{a}(x) \overset{\text{def}}{=} +\texttt{vocalic}(x) \wedge +\texttt{back}(x) \wedge +\texttt{low}(x)$$

(4) $$\texttt{first}(x) \overset{\text{def}}{=} p(x) = x$$

(5) $$\texttt{last}(x) \overset{\text{def}}{=} s(x) = x$$

The model-theoretic perspective gives us a flexible position from which to compare the representational content of spoken and signed phonological words. One obvious strategy is to say that that the representation for signs is essentially equivalent to that of spoken words, i.e. they are strings.

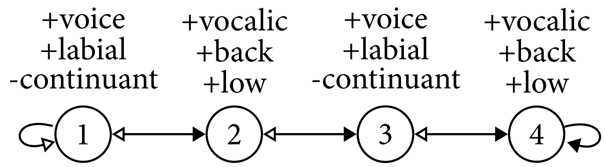

Figure 2: Visual of feature-based word model for 'baba' with word-edge constants

Virtually all models of sign phonology allow sequentiality, as a sequence of static and dynamic segments (Liddell, 1984; Sandler, 1986; Liddell and Johnson, 1989; Perlmutter, 1993; Newkirk, 1998), or a sequence of abstract timing units where only the non-dynamic endpoints ultimately associate (van der Hulst, 1993; Brentari, 1998).

This is essentially the position taken by Liddell (1984), who represents signs as strings of Hold and Movement segments (usually 3) along with various phonological features. Model-theoretically, this just means adapting our set of properties and thus the unary relations in the signature. This representation explicitly makes the claim that the only difference between spoken and signed representations is the size and content of the feature system. However, considering the string-based representation of the ASL sign IDEA given in Figure 3, it is easily apparent that such representation is highly redundant, as most featural information is stable across the whole sign.

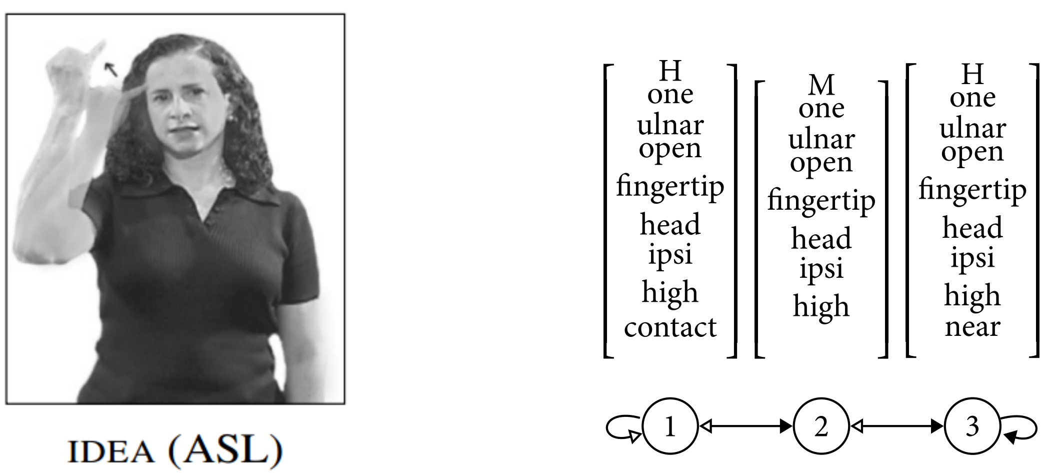

Figure 3: Left: ASL 'IDEA' (Image copyright Diane Lillo-Martin and Wendy Sandler). Right: Visual of string based word model for 'IDEA'. 'ipsi' refers to major body area location, ipsilateral side of forehead

Advances in phonological theory liberated certain feature content from a single temporal dimension by positing *Autosegmental Representations (AR)* (Williams, 1976; Goldsmith, 1976). In ARs, utterances are made up of several kinds of simultaneous levels, with each level related to but ordered independently of any other level. Phonological primitives are arranged in distinct strings or tiers, with an association relation relating units on different tiers. ARs have been argued to provide natural accounts of many tone and segmental processes because the non-local interactions between elements on a tier are always mediated through associations to a common element on another tier (see Jardine (2016b, 2017a) for a computational account).

To augment the model theory for words in order to make them ARs, Chandlee and Jardine (2019) note that all that is required is to add a binary association relation $A(x,y)$ to the model signature. This results in a new model signature, $\mathcal{M}^{AR}$, distinct from the previous string model signature,

(6) $$\mathcal{M}^{AR} \overset{\text{def}}{=} \langle \mathcal{D}; \{a, b, H, L\}; \{A(x,y)\}; \{p(x), s(x)\} \rangle$$

A model of our string *baba* and a visual representation are given in Figure 4. Here the number of domain elements has increased, and some are now only labeled with tonal features and some with segmental information. Successor and predecessor functions hold for all elements, and on different tiers. Note that now two elements precede themselves and two succeed themselves, though this does not have to be the case. The association relation $A(x,y)$ holds between elements on the two tiers, shown graphically with gray lines with dotted ends. This captures a structure where *baba* has a high tone H associated to the first vowel, and a low tone L associated to the second.

$$\mathcal{M}^{AR}_{baba}$$
$$\begin{aligned}
\mathcal{D} &= \{1,2,3,4,5,6\}\\
A(x,y) &= \{\langle 2,5\rangle, \langle 4,6\rangle\}\\
a &= \{2,4\}\\
b &= \{1,3\}\\
t &= \{\varnothing\}\\
H &= \{5\}\\
L &= \{6\}\\
s(x) &= \begin{cases} 2 & when \quad x = 1\\ 3 & when \quad x = 2\\ 4 & when \quad x \in \{3,4\}\\ 6 & when \quad x \in \{5,6\} \end{cases}\\
p(x) &= \begin{cases} 1 & when \quad x \in \{1,2\}\\ 2 & when \quad x = 3\\ 3 & when \quad x = 4\\ 5 & when \quad x \in \{5,6\} \end{cases}
\end{aligned}$$

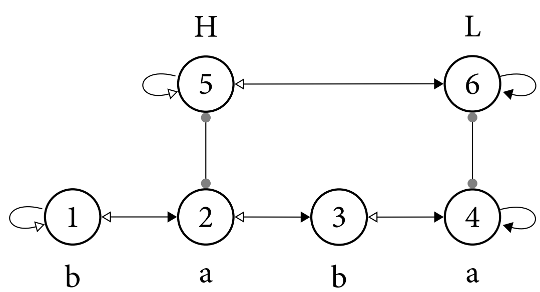

Figure 4: Autosegmental word model for "bábà"

While some sequential structure is acknowledged in almost all models of signs, sign representations have increasingly been argued to be inherently autosegmental in nature. Liddell and Johnson (1989) presented an updated version of their Move-Hold model where handshape features are autosegmentally associated to Movement and Hold elements on a segmental tier. This representational change can be captured using the autosegmental model signature, with appropriately different unary labeling relations. A model of the sign 'IDEA' is shown in Figure 5

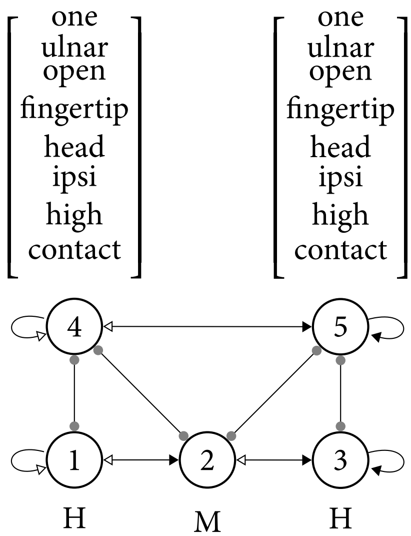

Figure 5: Autosegmental word model for ASL 'IDEA'

Sandler (1986, 1989), demonstrated that hand configurations can be independent morphemes (classifiers), and exhibit autosegmental stability phonologically. She proposed the Hand Tier model, which represents sequential Location (L) and Movement (M) segments on a skeletal tier, allowing explicit reference to sequential information, and represents Handshape Configuration (H) autosegmentally, where one handshape characterizes the whole sign, in contrast to the one-per-segment in the Move-Hold model. ARs for place of articulation features are a more recent innovation, seen in Brentari's (1998) Prosodic Model and in van der Hulst's Dependency Phonology Model (van der Hulst, 1993, 1994).

Incorporating these autosegmental attributes of sign into a model theory is again straightforward, since it simply amounts to adding more relational structure.

(7) $$\mathcal{M}^{HT} \stackrel{\text{def}}{=} \langle \mathcal{D}; \{L, M, H_1, H_2, H_3, P_1, P_2, P_3\}; \{A(x,y), \texttt{loc}(x,y)\}; \{p(x), s(x)\}\rangle$$

Location, Movement, Handshapes, and Place features form unary relations. For the purposes of this paper, we will consider 3 possible handshapes and places for simplicity and convenience. One could easily expand the Handshape or Place nodes to consider feature geometries by, again, adding more relational structure. We will also refer to arbitrary Handshape and Place features, using the following predicates, which would be extended if more handshape types are added:

(8) $$H(x) \stackrel{\text{def}}{=} H_1(x) \vee H_2(x) \vee H_3(x)$$

(9) $$P(x) \stackrel{\text{def}}{=} P_1(x) \vee P_2(x) \vee P_3(x)$$

The association relation $A(x, y)$ associates elements on the skeletal tier to the handshape tier, and the location relation $\texttt{loc}(x, y)$ relates elements on the skeletal 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 one place feature and one handshape feature is given in Figure 6.

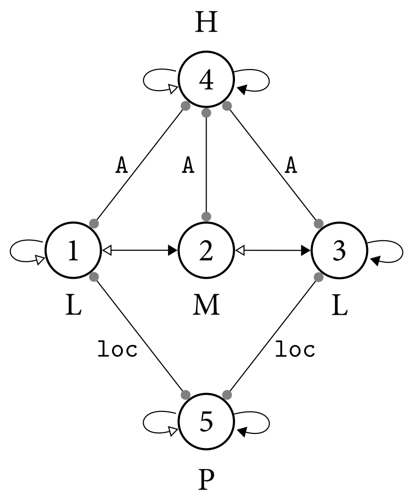

Figure 6: Autosegmental Hand Tier word model of a monosyllabic sign

While (Rawski, 2017) characterized sign phonology using strings, this paper will consider various autosegmental model signatures, to incorporate the rich simultaneous structure in sign. This is, however, not a claim that the particulars of what is presented make up exactly the "correct" model of spoken or signed phonological words, or that they are supposed to be a perfect formulation of one of the aforementioned theories — far from it. The proper characterization of the phonological word is very much an open question, even for spoken phonology.

The real advantage is that Model Theory allows for freedom and preciseness in choosing the representations (embedded in the atomic predicates) that one takes to be linguistically relevant, and in the strong case, cognitively real, and to see what effect these have on the structures that are allowable. If one has a commitment to a certain phonological representation, say graphs for feature geometries (van der Hulst, 1993; Sandler, 1989), or trees for representing syllabic or prosodic constituencies (Brentari, 1998), one can represent them model-theoretically and compare their effects on the computation. The earlier mentioned logic-automaton connection for finite string models holds for many different data structures, including infinite strings, trees, infinite trees, hypergraphs, graphs of bounded tree-width, among others (see Thomas (1997) for an overview of formal languages within the framework of mathematical logic).

One salient property that emerges across model signatures concerns the boundedness of the sign. Existence of a syllable-like unit in sign languages suggest that the movement corresponds to the syllable nucleus (Brentari, 1990; Wilbur, 1982, 2011; Sandler, 1989; Perlmutter, 1993) While internal movement resulting from a change in finger position or palm orientation may coincide with a path movement from one location to another, the simultaneous movements still constitute one syllable. Two movements in succession are counted as two syllables. This means that most monomorphemic words, and multimorphemic words are monosyllabic. This tendency, combined with the overwhelmingly nonconcatenative nature of sign morphology, has resulted in what some call a "monosyllable conspiracy" (Sandler and Lillo-Martin, 2006), with others using a CVC template as a heuristic comparison with the monosyllabic LML or HMH sequences (Perlmutter, 1993). While signs may be bounded in size, a surface feature distinct from spoken language, the phonology, the nature of a signed process may be factored out of this, as discussed in further detail in Section 4.

3 Logical Mappings and Computational Expressivity

Model Theory is about representations, but also provides a clear path to define the nature of *transformations* from underlying to surface forms. The theory of logical transformations states that we can define a mapping from the set of input structures in one model signature to the set of output structures in another model signature by defining each relation in the signature of the output in terms of the logic of the input (Courcelle, 1994; Engelfriet and Hoogeboom, 2001; Filiot, 2015). For an input signature $\mathcal{M}^I$ and an output signature $\mathcal{M}^O$, a logical mapping $\mathcal{T}$ specifies a finite number of output *copies* of the domain $\mathcal{D}$, and for each copy defines each function, relation, and constant in $\mathcal{M}^O$ in terms of the input $\mathcal{M}^I$. Mappings additionally define a *licensing function* that specifies which domain elements survive in the output. If there are multiple copies of the input domain, one must specify relations between them. Since each of these formulas are terms, they are semantically interpreted with respect to the input structure.

Consider a mapping that changes a b to an a only if it occurs between two a's, so a phonological rewrite rule of the form b → a / a_a, with simultaneous application. Here $\mathcal{M}^I = \mathcal{M}^O = \mathcal{M}^S$, the successor model. Then given the set of properties $\Sigma = \{\texttt{a}, \texttt{b}, \texttt{c}\}$, this mapping $\mathcal{T}_{ba}$ defines the set of predicates over the input structure where the superscript O indicates the relations over the output. (10) says that a domain element is labeled an a in the output iff it was labeled an a in the input, or was labeled a b in the input and its predecessor and successor were both a's. (11) specifies which output elements have the b relation: those that were b's in the input but whose predecessor and successor are not a. (12-13) say that the predecessor and successor functions are true for any input-output pair as they were in the input. Finally, (14) specifies which elements of the domain are present (licensed) in the output, in this case all of them. For example, using the model from above, when applied to the input $\mathcal{M}^S_{baba}$, the mapping changes the label of the third position from b to a and leaves the remaining positions unchanged, as shown below.

(10) $$a^O(x) \overset{\text{def}}{=} a(x) \lor [b(x) \land a(p(x)) \land a(s(x))]$$

(11) $$b^O(x) \overset{\text{def}}{=} b(x) \land \neg[a(p(x)) \land a(s(x))]$$

(12) $$p^O(x) \overset{\text{def}}{=} p(x)$$

(13) $$s^O(x) \overset{\text{def}}{=} s(x)$$

(14) $$license(x) \overset{\text{def}}{=} \texttt{TRUE}$$

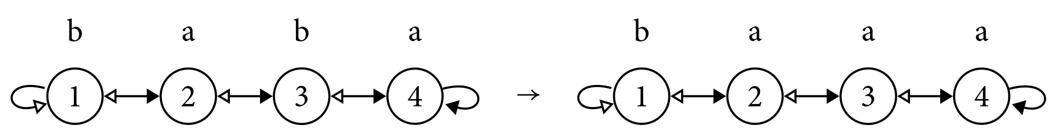

Figure 7: Visual of input and output word models for the mapping in (4-8)

Now consider a mapping that deletes a b only if it occurs between two a's, so a phonological rewrite rule of the form b → ∅ / a_a, with simultaneous application. In this function, all of the parts are the identity function, since the order and labels of segments remain the same. The licensing function, however, now evaluates to `TRUE` only if a segment was a b not in the conditioning environment. If a domain element is not licensed, it is deleted, and all relations it satisfies disappear, coalescing the structure. Unlicensed segments will be shown pictorially using dotted gray lines

(15) $$a^O(x) \overset{\text{def}}{=} a(x)$$

(16) $$b^O(x) \overset{\text{def}}{=} b(x)$$

(17) $$p^O(x) \overset{\text{def}}{=} p(x)$$

(18) $$s^O(x) \overset{\text{def}}{=} s(x)$$

(19) $$license(x) \overset{\text{def}}{=} \neg[b(x) \wedge a(p(x)) \wedge a(s(x))]$$

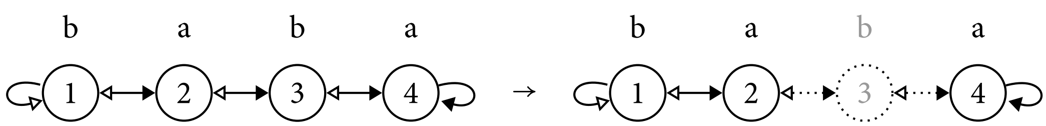

Figure 8: Visual of input and output word models for the mapping in (4-8)

Now we turn to classifying the logical complexity of transformations. We consider three levels of complexity, ordered by the types of distinctions they can make. None of the statements in the preceding examples involve the use of universal or existential quantifiers ($\forall, \exists$). Such statements are quantifier-free fragments of first-order logic, and a mapping that is completely quantifier-free is called a *Quantifier-Free (QF) mapping*. Statements in *First-Order (FO) logic* allow universal and existential quantifiers to range over elements of the domain. *Monadic Second-Order (MSO) logic* can allow quantifiers over *sets* of domain elements[2]. Importantly, these are logical statements about models, not about the logical axioms that define parts of the model signature.

To see why quantification is important, compare (20) to (21), adapted from Strother-Garcia (2018). The former states that an output position x will be labeled a if the corresponding input position is an a *or* if there is a position labeled b *somewhere* in the input. Checking whether $a^O(x)$ is true requires evaluation of the entire string to see if any position is labeled b. This is due to the existential quantifier $\exists$, which makes (20) strictly FO. In contrast, (21) lacks any quantification. $a^O(x)$ can be evaluated independently at every position in the string.

(20) $$a^O(x) \overset{\text{def}}{=} a(x) \vee (\exists y)[b(y)]$$

(21) $$a^O(x) \overset{\text{def}}{=} a(x) \vee b(x)$$

This example illustrates the connection between quantification and locality. Computing the truth value of a predicate that involves quantification requires *global* evaluation. If the predicate does not use quantifiers, however, its truth evaluation must be possible over a substring of bounded

[2]For formal definitions of MSO and FO, see Enderton (2001), Fagin et al. (1995), and Shoenfield (1967).

size. Thus a QF mapping amounts to a constraint-checking function that operates *locally*, within a bounded window of evaluation. Defining a QF formula that identifies targets of the process in terms of information present in the input provides a rigorous, independent notion of what it means for a process to be local.

There is a direct link between the type of logic used in a mapping and the computational power required of that mapping. It was stated earlier that MSO statements over model signatures are equivalent to Finite-State acceptors, which define the Regular languages (Büchi, 1960). The set of MSO-definable *mappings* uniquely defines 2-way deterministic finite-state *transducers* (machines with input and output symbols), which compute the set of *regular relations* (Engelfriet and Hoogeboom, 2001). There is thus a deep connection between MSO logic and the property "being regular", meaning that the amount of memory required does not grow with the size of the input. Weaker logics define subclasses of the regular languages and relations (see Rogers and Pullum (2011) for an overview of subregular languages, and Filiot (2015) for transducers).

Chandlee and Lindell (2016) showed that the set of mappings satisfiable by QF formulas over the successor model $\mathcal{M}^S$ share a deep connection to a proper subset of the regular relations, the *Input Strictly Local (ISL) functions* (not to be confused with Israeli Sign Language, often abbreviated ISL). ISL functions determine an output string for an input based only on contiguous substrings of bounded length (Chandlee and Heinz, 2018). For example, intervocalic voicing is ISL since it only has to be sensitive to 3-segment input substrings of the form VTV. Phonologically, this class is extremely relevant, because Chandlee (2014) showed that a full 95% of the process in P-base (Mielke, 2007), a database of phonological processes, are ISL functions. Additionally, ISL functions have efficient learning algorithms from positive data (Chandlee et al., 2014) and are learned more easily by humans in learning experiments (Finley, 2009). Intuitively, the QF-ISL connection stems from the fact that all the needed information can be found within a bounded window of material surrounding a given input position. Just as in the MSO-Regular case, there is a deep connection between Quantifier-Freeness and Strict Locality, because both rigorously define the notion of "being local".

As an example, Strother-Garcia (2018) provides an analysis of syllabification in Imdlawn-Tashlhiyt Berber, and shows that it can be succinctly captured using a QF mapping. This is important, since Berber syllabification traditionally was a strong motivation for global optimization in constraint-based frameworks, but the analysis shows that the process at heart is in fact local.

Chandlee and Jardine (2019) extend QF mappings to consider Autosegmental model signatures. They show that an input-output map is *Autosegmental-Input Strictly Local (A-ISL)* if it can be described with a QF mapping where the model signature is an AR. Model-theoretically, this generalizes the notion of locality from considering "substrings" to "*sub-structures*", where now the chunks that are being evaluated are members of an autosegmental graph. The A-ISL class is more powerful than the ISL class, but each preserves the notion of Strict Locality. They further prove that if an AR map is A-ISL, then the individual map on each tier is an ISL function.

The flexibility given by the model-theoretic perspective for defining linguistic representations, combined with the precise connections between logical statements and computation, enables a powerful ability to characterize the nature of phonological (and indeed any linguistic) processes across modalities. Model-theoretically, representations can be defined for each modality on its own terms, using the information characteristic of that modality. Parallels and divergences then emerge in a way that enables precise comparison. One can then understand the nature of a process in terms of the minimal information and computation needed to perform it.

4 Logical Mappings in Sign Language Phonology

This section provides logical characterizations of several phonological processes in sign language, in order to compare their complexity to their spoken equivalents. The first, compound reduction, demonstrates autosegmental spreading and segment deletion, processes known to be ISL or A-ISL in spoken language. The second, final syllable reduplication, was shown by Chandlee (2014, 2017) to be ISL in spoken language, despite the addition of structure. Epenthesis and deletion have been demonstrated to be ISL processes, with deletion shown in the preceding section. Additionally, these two processes feed each other and interact with the outputs of compound reduction and reduplication in sign language.

The main result of this section is that all these mappings are Quantifier Free over autosegmental representations, and thus A-ISL functions. This generalizes the result of Rawski (2017), who demonstrated the ISL nature of several sign processes when signs are represented as strings. Examples are mostly drawn from American Sign language (ASL) and Israeli Sign Language, but these processes are attested across the phonological typology of sign.

As mentioned in Section 2, signs have an overwhelming tendency to be mono- or bisyllabic, and most of the examples used in the following analyses fit this form. This should not suggest that the mappings are limited to only words of that type, implying a finite language or data overfitting. The phonological processes are characterized to apply to any word of unbounded but finite size, which is the scope of Finite Model Theory. This distinction is familiar from the prosodic approaches to Semitic morpho-phonology (McCarthy, 1981, 1982; McCarthy and Prince, 1990), to which sign language is often compared. Savitch (1993) discusses this distinction mathematically: one can factor the infinite and finite factors of an analysis, since the composition of an infinite language with a finite language is finite. The following analyses characterize the infinite side.

In the analyses that follow, I will often omit formulas for relations and functions whose definition is identical to their input formula, i.e. the identity relation, for ease of exposition.

4.1 Compound Reduction

Many lexicalized sign compounds undergo a type of phonological reduction to preserve the monosyllabic character of canonical signs (Frishberg, 1975). Compound reduction is an amalgam of several processes. Often, sequential segments of both members of the compound delete (Liddell, 1984; Liddell and Johnson, 1989), the hand configuration of the first member also deletes, and the hand configuration autosegment of the second member spreads to characterize the whole surface compound (Sandler 1986; 1989). Other compounds reduce in different ways. Some maintain all segments and both hand configurations. Others reduce segmental structure only, maintaining two hand configurations (though see Lepic (2015)).

As an example, consider the ASL compounds 'FAINT' ('MIND' + 'DROP') and 'BELIEVE' ('THINK' + 'MARRY') as well as the Israeli Sign Language compound 'SURPRISED' ('THINK' + 'STOP'). These compounds are characterized by regressive total handshape spreading from the second sign to the first, deletion of the initial location segments of both signs and the first movement, and coalescence of the signs such that place information is uniquely specified for both L segments. These are shown in Figure 9, along with Sandler (1989)'s Hand Tier model analysis of the reduction of ASL 'BELIEVE'.

An input model to compound reduction inhabits requires both sign elements of the compound

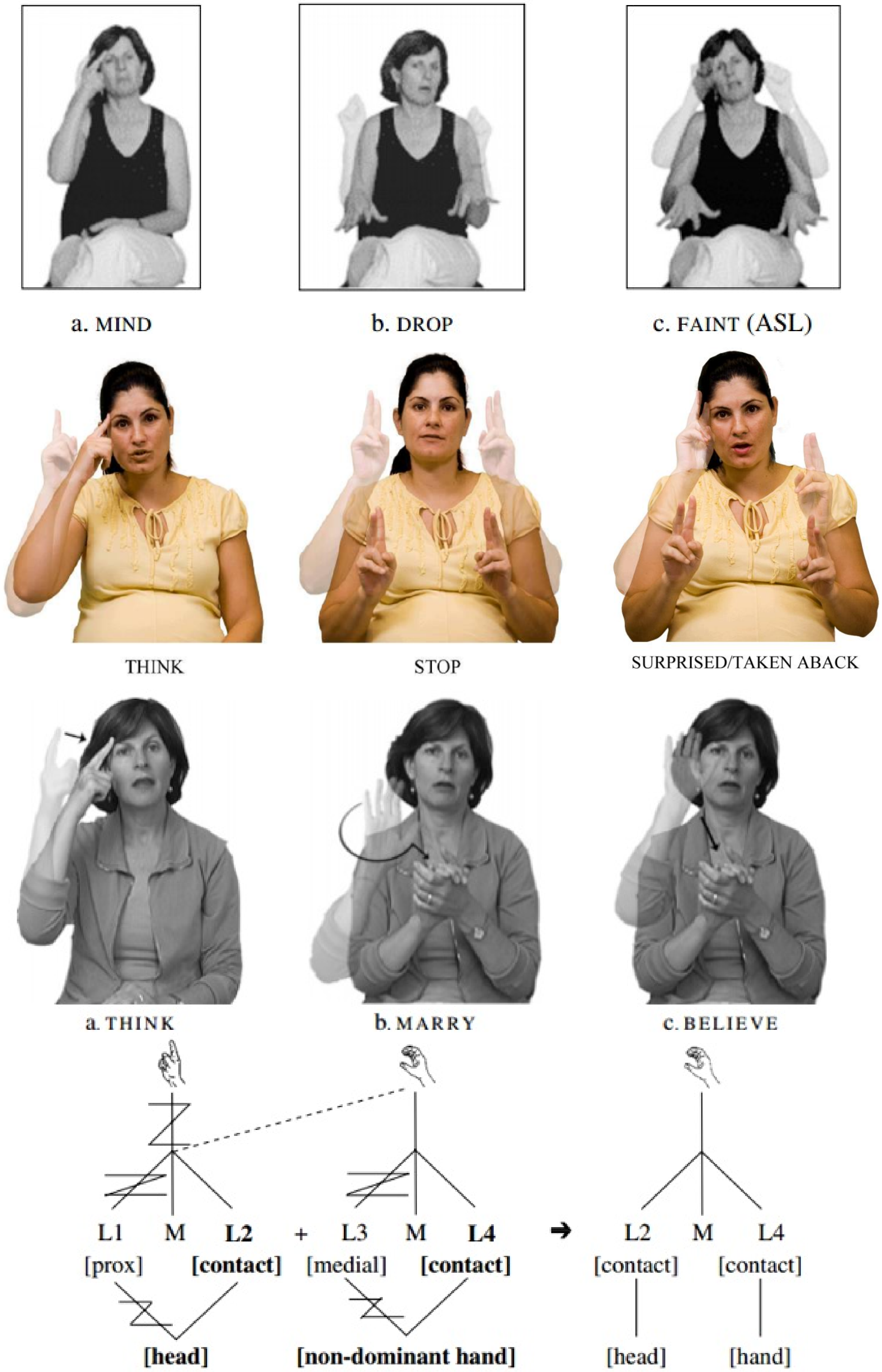

Figure 9: Compound Reduction. Top: ASL MIND+DROP=FAINT; Middle: ISL THINK+STOP=SURPRISED; Bottom: ASL THINK+MARRY=BELIEVE, with Hand Tier model of reduction (Images copyright Wendy Sandler & Diane Lillo-Martin)

to be fully present. Here we use the Hand Tier model signature $\mathcal{M}^{HT}$. The complete model signature is defined in (22), and a visual representation is on top in Fig. 10. For simplicity, this model signature includes only two handshape and two place labels, but this may easily be extended. Since this is an underlying representation of the fully specified morphological compound, and thus no epenthetic M segment, they are represented as immediately adjacent.

(22) $$\mathcal{M}^{HT} \stackrel{\text{def}}{=} \langle \mathcal{D}; \{\{H_1, H_2, H_3\}\{P_1, P_2, P_3\}L, M\}; \{A(x,y), \texttt{loc}(x,y)\}; \{\texttt{p}(x), \texttt{s}(x)\}\rangle$$

Now we can define the reduction mapping from this input model to an output. Since no unary labels are added, deleted, or changed, and the successor and predecessor functions stay the same, these amount to the identity relation. I omit them for clarity. The second part of the mapping concerns the handshape spreading. (23) states that elements on the skeletal and handshape tiers are associated in the output if they are associated in the input to the second handshape or its predecessor, i.e. the first handshape. In this way, the handshape spreading proceeds locally, as it only has to account for a bounded distance between the element it is considering and one that was associated in the input. This bears a striking parallel to Chandlee and Jardine (2018)'s characterization of local autosegmental spreading in tone.

(23) $$A^O(x,y) \stackrel{\text{def}}{=} L(x) \vee M(x) \wedge \texttt{last}(y) \wedge [A(x,y) \vee A(x, \texttt{p}(y))]]$$

The deletion of timing and handshape segments is handled by the licensing function $lic(x)$, which specifies the domain elements that survive in the output. For the metathesis case, the licensing function always evaluated to TRUE. Here it picks out specific elements based on their properties. I specify licensing functions for each tier, and then a more general function. (24) says that only the final handshape of the compound is licensed. (25) says that elements labeled L or M are licensed if they are final, penultimate, or the two elements which precede the final element. (26) says that a place tier element is licensed if it is the location of the last element in either of the compounding words. (27) says that in general, an element is licensed only if it satisfies one of these conditions.

(24) $$license_H(x) \stackrel{\text{def}}{=} \texttt{last}(x) \wedge H(x)$$

(25) $$license_{LM}(x) \stackrel{\text{def}}{=} (L(x) \vee M(x)) \wedge [\texttt{last}(x) \vee \texttt{last}(p(x)) \vee (\texttt{first}(p(p(x))) \wedge \neg\texttt{first}(x))]$$

(26) $$license_P(x) \stackrel{\text{def}}{=} P(x) \wedge [\texttt{first}(x) \vee \texttt{last}(x)]$$

(27) $$license(x) \stackrel{\text{def}}{=} license_H(x) \vee license_{LM}(x) \vee license_P(x)$$

As an example, we can apply this mapping to the model of the ASL compound BELIEVE (THINK + MARRY) under the Hand tier model signature $\mathcal{M}^{HT}$. As shown in Figure 9.

The separability of functions is what gives the logical transduction its power, by factoring a process into its necessary parts. It also captures the nature of the compound reduction, the parts of which can vary. For example, in this transduction, the spreading of the second handshape and the deletion of segments are independent. The $license_{LM}$ function can pick out different segments on the LM tier which survive deletion. For example, in the ASL compound GOOD-NIGHT (GOOD

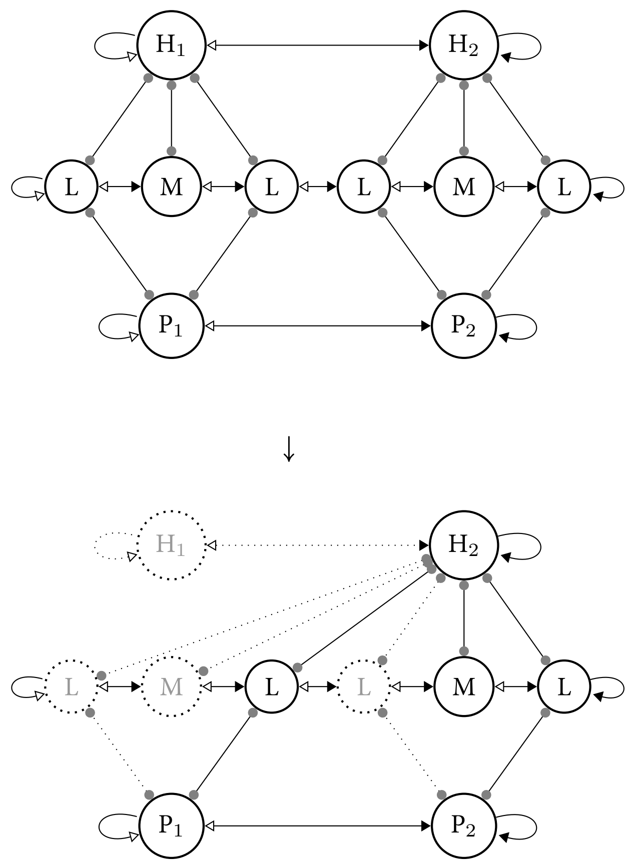

Figure 10: Visual of Input (top) and Output (bottom) word models for compound reduction of ASL BELIEVE (THINK+MARRY). Unlicensed elements are dashed. Domain indices are omitted for readability.

Figure 11: ASL GOOD (left), NIGHT (middle), and the compound GOOD-NIGHT (right). Illustration from Liddell and Johnson (1986)

+ NIGHT) (Figure 11), the second handshape still spreads, but it is the first L of the first compound member and the last member of the second compound which survive. Handling reductions of this flavor involves merely changing the licensing function, while the rest of the transduction remains unchanged. Similarly, hand configuration assimilation may take place regardless of whether or not the compound loses segments. For example, total hand configuration assimilation occurs on the reduced monosyllabic ASL compound HUSBAND (MAN + MARRY) and on the unreduced disyllabic compound OVERSLEEP (SLEEP+SUNRISE) (Figure 12) (Liddell and Johnson, 1986).

Most importantly, this mapping is quantifier-free, because every formula in it is quantifier-free, meaning the reduction process doesn't require global search. There are many more types of compound reduction, involving partial handshape assimilation, and/or deleting different elements, but the input and output structure is always bounded. This ensures that the output compound structure is only determined by a finite amount of information present in the input structure, and so does not need the power of quantification to describe the mappings. As stated in the preceding section, this means the mapping is A-ISL. Thus the notion of locality required for compound reduction is the same even when expanding the representation to include a complex interaction of processes, showcasing the local nature of the process even with additional relational structure, a point which has been made for ISL functions and opaque processes (Chandlee et al., 2018).

4.2 Final Syllable Reduplication

The nature of the output of compound reduction plays an additional role in other morphological processes. Many aspectual inflections in sign, in particular those expressing duration or iteration, involve total reduplication of the monosyllabic base (Klima and Bellugi, 1979). However, when compounds are reduplicated, the reduplicated element is the final syllable (Sandler, 1989), shown in various forms in Fig. 12. If the compound is reduced and monosyllabic, like ASL 'FAINT', then the whole LML form is reduplicated. However, if the compound is disyllabic, like ASL 'OVERSLEEP', only the final LML syllable is reduplicated. It doesn't matter whether the last syllable has path movement only or internal movement only; each type of movement is regarded as a syllable nucleus by this reduplicatory process.

Chandlee (2014) describes a class of reduplication patterns of this sort as local reduplication, since the reduplicant is affixed adjacent to the portion of the base it is copied from. Another category that meets this condition is where the reduplicant is a suffix copied from the end of the base. Chandlee cites reduplicative prefixation in Tagalog, and reduplicative suffixation in Marshallese,

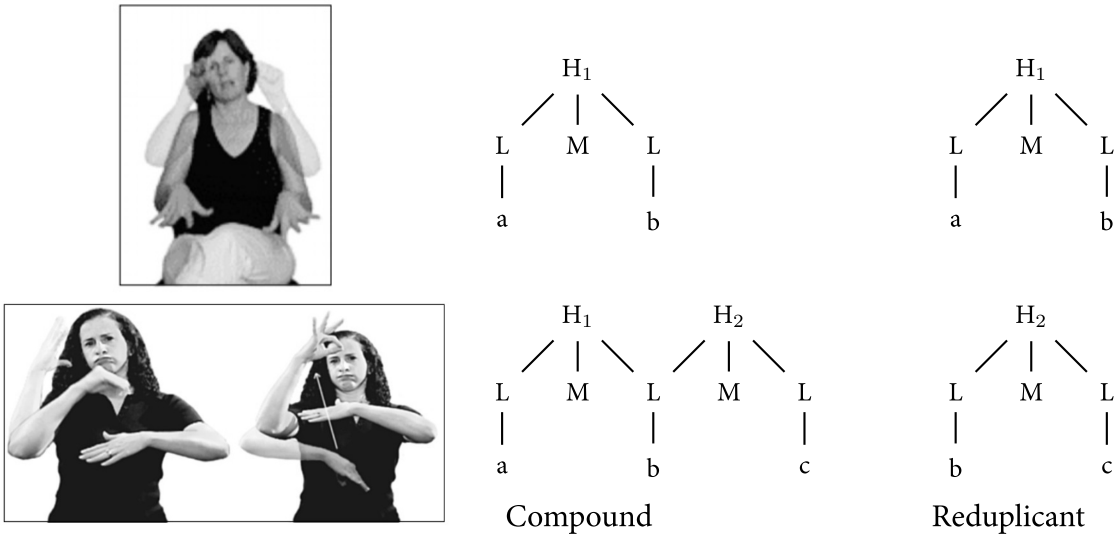

Figure 12: Final-syllable reduplication in monosyllabic ASL FAINT (top), and disyllabic OVER-SLEEP (bottom), Images Copyright Wendy Sandler and Diane-Lillo-Martin

shown in Ex. 1 and 2, respectively, along with rewrite rules for the particular processes:

(1) Tagalog (Blake, 1917)
súlat 'write' → su-súlat 'will write'
$\varnothing \rightarrow C_1V_1$ / $\#\_C_1V_1$

(2) Marshallese (Byrd, 1993)
ebbok 'to make full' → ebbok-bok 'puffy'
lɣŋɔŋ 'fear' → lɣŋɔŋ-ŋɔŋ 'very afraid'
$\varnothing \rightarrow C_1V_1C_2$ / $C_1V_1C_2\_\#$

Chandlee (2017) shows that all local reduplication patterns are Input Strictly Local, since the partial information copied from the input is attached to the same edge it is copied from. She contrasts this with non-local reduplication, where the copied portion attaches to the opposite edge from where it is copied, and shows that this is not an ISL function. Consider a simplified spoken language example where the reduplicant is the last two segments. The transduction in (28-37) describes this mapping using the successor model over strings. An example using the string *taba* is shown in Figure 13.

In this compound reduction mapping, only *one* output copy set was required, since structural information was only changed or deleted, not added. Since reduplication necessarily adds structural information by copying part of the word, the mapping requires *multiple* copies of the domain elements, in this case two. The mapping must be explicit about the output relations for *each* copy set. First, we require two licensing functions, one for each copy set. Each specifies which domain elements will surface: all of them in the base copy, and only the last two in the reduplicant copy.

In the transduction the unary labels are unchanged, but since we have multiple copy sets, they must be specified for each copy set. To join the two copy sets and make a reduplicated word, we must specify the successor and predecessor functions within each copy set, and also *between* copies. For clarity I simplify each of these using the following notation. In (36-37), (x, i) refers to the ith copy of element x in the output structure, and in Figure 13 this is shown as x_i. The successor function explicitly specifies that the relations are unchanged except for the last segment, whose successor is the penultimate segment in the second copy (i.e. its successor in copy set 2 is $(s(x), 2)$. The predecessor function is defined similarly.

(28) $$a_1^O(x) \overset{\text{def}}{=} a(x)$$

(29) $$a_2^O(x) \overset{\text{def}}{=} a(x)$$

(30) $$t_1^O(x) \overset{\text{def}}{=} t(x)$$

(31) $$t_2^O(x) \overset{\text{def}}{=} t(x)$$

(32) $$b_1^O(x) \overset{\text{def}}{=} b(x)$$

(33) $$b_2^O(x) \overset{\text{def}}{=} b(x)$$

(34) $$license_1(x) \overset{\text{def}}{=} \texttt{TRUE}$$

(35) $$license_2(x) \overset{\text{def}}{=} \texttt{last}(x) \vee \texttt{last}(p(x))$$

(36) $$s^O(x,i) \overset{\text{def}}{=} \begin{cases} (s(x),1) & \textit{when} \quad \neg(\texttt{last}(x)) \quad i=1 \\ (p(x),2) & \textit{when} \quad \texttt{last}(x)) \quad i=1 \\ (s(x),2) & \textit{when} \quad i=2 \end{cases}$$

(37) $$p^O(x,i) \overset{\text{def}}{=} \begin{cases} (p(x),1) & \textit{when} \quad i=1 \\ (s(x),1) & \textit{when} \quad \texttt{last}(s(x)) \wedge \neg\texttt{last}(x) \quad i=2 \\ (p(x),2) & \textit{when} \quad \neg[\texttt{last}(s(x)) \wedge \neg\texttt{last}(x)] \quad i=2 \end{cases}$$

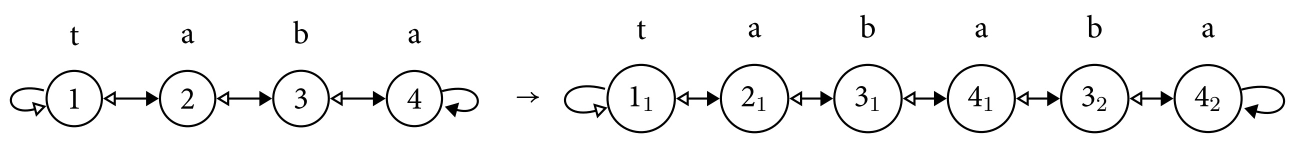

Figure 13: Visual of input and output word models for reduplication mapping *taba*. Unlicensed elements are omitted for readability.

Turning to final syllable reduplication in sign language as a partial copying mapping, we see that the nature of the mapping is almost identical, and is sufficiently characterized by a Quantifier-Free mapping. Just as in the spoken language example, this mapping uses the model-theoretic ability to specify multiple copies of the input structure. Specifying two copies of the input domain captures the base an reduplicant. The second copy set needs to refer to elements of the final syllable, which can be up to an LML sequence. In order to preserve readability, and in the spirit of simply good programming practice, I introduce several user-defined predicates to refer to such particular segments in (38-42), similarly to the examples in formulas (2-5). These will be used in the licensing, successor, and predecessor functions. The labels of the predicates are self-explanatory.

$$\texttt{last}_{LM}(x) \overset{\text{def}}{=} \texttt{last}(x) \wedge [L(x) \vee (M(x)] \tag{38}$$

$$\texttt{last}_{H}(x) \overset{\text{def}}{=} \texttt{last}(x) \wedge H(x) \tag{39}$$

$$\texttt{last}_{P}(x) \overset{\text{def}}{=} \texttt{last}(x) \wedge P(x) \tag{40}$$

$$\texttt{penult}_{P}(x)(x) \overset{\text{def}}{=} \texttt{last}(s(x)) \wedge \neg\texttt{last}(x) \wedge P(x) \tag{41}$$

$$\texttt{antipenult}_{LM}(x) \overset{\text{def}}{=} \texttt{last}(s(s(x))) \wedge \neg\texttt{last}(x) \wedge [L(x) \vee (M(x)] \tag{42}$$

Now we define the reduplication mapping. Each labeling, association, and location relations are the identity map (formulas omitted for space). In the first output copy set, all elements are licensed, as shown in (43), where the the subscript denotes the corresponding copy set. (44) says that the final, penultimate, and antipenultimate LM-tier segments are preserved, and that only the final handshape defined for the `last` segment is preserved. This captures the 'final syllable' reduplication copy.

$$\mathit{license}_1(x) \overset{\text{def}}{=} \texttt{TRUE} \tag{43}$$

$$\begin{aligned}\mathit{license}_2(x) \overset{\text{def}}{=}\ & \texttt{last}_{LM}(x) \vee \texttt{last}_{LM}(s(x)) \vee \texttt{last}_{LM}(s(x)) \vee \\ & \texttt{last}_{H}(x) \vee \texttt{last}_{P}(x) \vee \texttt{last}_{P}(s(x))\end{aligned} \tag{44}$$

To join the two copy sets and make a reduplicated word, we must again specify the successor and predecessor functions for each copy set, and between copies.

Now we use these predicates to specify the output predecessor and successor functions for the reduplication mapping. As in the spoken language example, in (45-46), (x, i) refers to the ith copy of element x in the output structure, and in Figure 14 this is shown as x_i. (45) defines the successor function in a particular copy set i. The function says that for all non-final elements in the first and second copies, their successor is the same as in the input. For the final element in copy 1, its successor is the antipenultimate element of copy 2. The `pred` function in (46) is defined symmetrically, but picking out the antipenultimate LM-tier segment and penultimate place tier segment as the explicit location where the reduplicant attaches the base. Just like compound reduction, the parameters of the mapping are independent of the overall structure of the map.

$$\texttt{s}((x,i)) \overset{\text{def}}{=} \begin{cases} (\texttt{s}(x), 1) & \textit{when} \quad \neg\texttt{last}(x), \quad i = 1 \\ (p(p(x)), 2) & \textit{when} \quad \texttt{last}_{LM}(x), \quad i = 1 \\ (x, 2) & \textit{when} \quad \texttt{last}_{H}(x), \quad i = 1 \\ (p(x), 2) & \textit{when} \quad \texttt{last}_{P}(x), \quad i = 1 \\ (\texttt{s}(x), 2) & \textit{when} \quad\quad i = 2 \end{cases} \tag{45}$$

(46)

$$\texttt{p}((x,i)) \overset{\text{def}}{=} \begin{cases} (p(x), 1) & \textit{when} & & i = 1 \\ (s(s(x)), 1) & \textit{when} & \texttt{antepenult}_{LM}(x), & i = 2 \\ (x, 2) & \textit{when} & \texttt{last}_{H}(x), & i = 2 \\ (s(x), 1) & \textit{when} & \texttt{penult}_{P}(x), & i = 2 \\ (\texttt{p}(x), 2) & \textit{when} & \neg[\texttt{antepenult}_{LM}(x) \vee \texttt{last}_{H}(x) \vee \texttt{penult}_{P}(x)] & i = 2 \end{cases}$$

As an example, consider the reduplicated disyllabic compound 'OVERSLEEP' (Figure 12), which copies the final syllable (an LML sequence, along with its associated handshape and place autosegments). The input word model for 'OVERSLEEP' is shown visually on top in Figure 14, and the output model of the mapping is shown on the bottom, with the base copyset shown on the left and reduplicant copyset on the right. Deleted segments again shown using dashed lines.

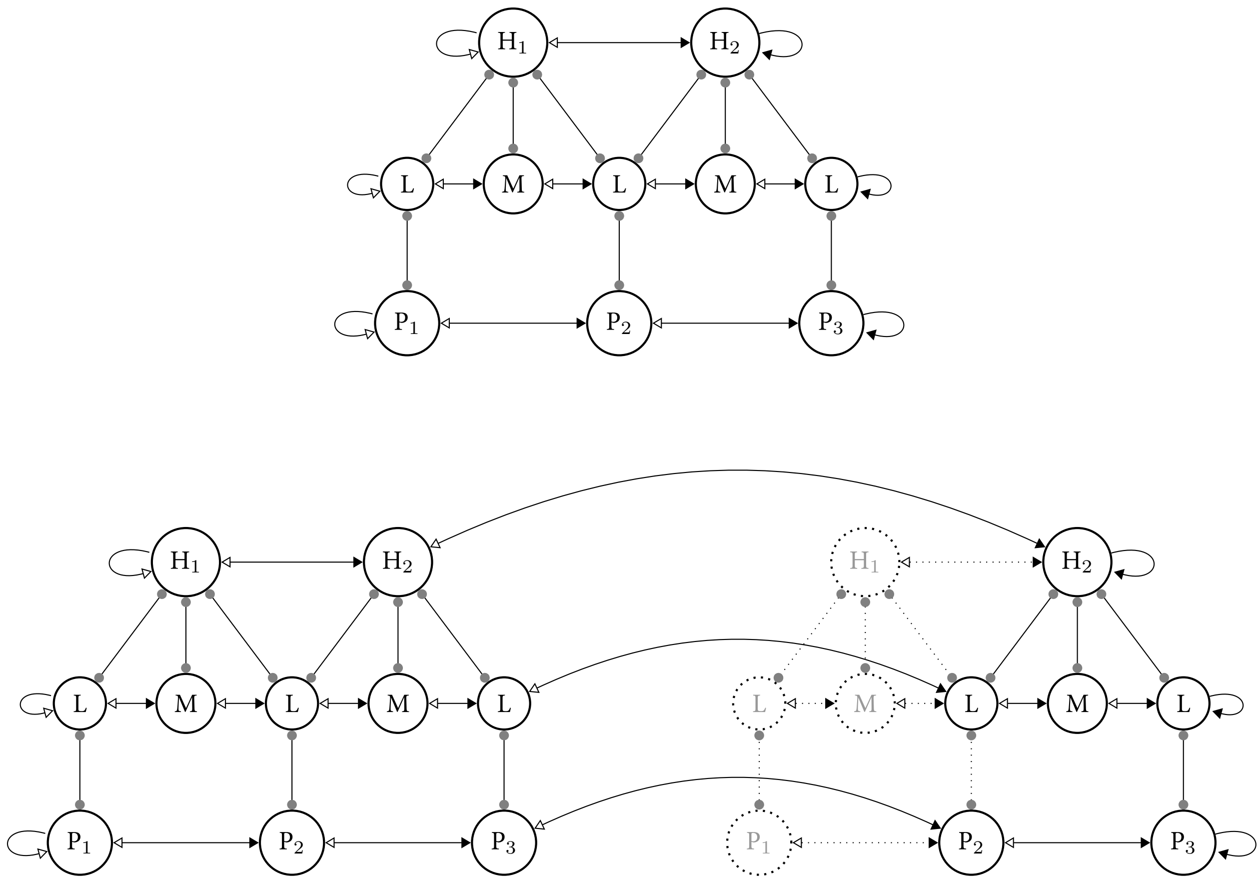

Figure 14: Visual of Input Model (top) and Output Copy Sets (bottom left and right) for Reduplicated ASL 'OVERSLEEP'. Unlicensed elements are dashed. Domain indices are omitted for readability

4.3 Movement Epenthesis and Hold Deletion

The outputs of compound reduction and reduplication processes also interacts with additional phonological processes. Newport (1981) shows that a linking movement is epenthesized in some reduplicated forms. Liddell and Johnson (1986) analyze a phenomena of Movement Epenthesis that occurs after compounding. Movement Epenthesis inserts a movement between articulatorily distinct contiguous segments. Its most common function is to move the hand from the ending configuration of one set of segments to the beginning configuration of the next set of segments. Because it is seldom the case that a sign begins with exactly the same articulatory specifications as the final segment of the previous sign, Movement Epenthesis often applies at word boundaries. Because the compounding process necessarily involves the juxtaposition of two words, the Movement Epenthesis Rule often applies between the two parts of a compound.

(47) $$\varnothing \rightarrow M \;/\; H\_H$$

(48) $$H \rightarrow \varnothing \;/\; M\_M$$

Sandler (1989) characterises this process with the following rule, using the Hand Tier model.

(49) $$\varnothing \rightarrow M \;/\; L_i\_L_j \quad where \quad L_i \neq L_j$$

For example, consider a monosyllabic form with distinct start and end locations like FAINT. After undergoing reduplication, an M segment is epenthesized in between adjacent L segments since they are distinct.

(50) $$L_1ML_2 + L_1ML_2 \rightarrow L_1ML_2ML_1ML_2$$

Under the Hand Tier model signature $\mathcal{M}^{HT}$ used up to this point, and specifying multiple copy sets, describing the epenthesis mapping is straightforward. However, to showcase the generality of the model-theoretic approach, and to show that the Quantifier-Free nature of the processes does not inherently rely on one particular representation of the sign, I describe the process of movement epenthesis using a variant of the autosegmental model signature $\mathcal{M}^{AR}$, which, recall from Section 2, characterized an autosegmental view of the sign more in line with Liddell and Johnson (1989)'s Move-Hold model, with appropriately different feature labels. This model signature, called $\mathcal{M}^{MH}$ for clarity, is fully specified below, with $H(x), M(x)$ specifying Movements and Holds, appropriately, and a, b, c, d as predicates specifying a particular feature bundle for ease of exposition. Note that there is only one autosegmental relation in this model signature.

(51) $$\mathcal{M}^{MH} \overset{\text{def}}{=} \langle \mathcal{D}; \{a, b, c, d, H, M\}; \{A(x, y)\}; \{p(x), s(x)\} \rangle$$

Since epenthesis requires adding structure to the word, we specify two copy sets just like in the reduplication case. Here the two licensing functions specify that domain elements in the first are all licensed, while in the second only a segment that occurs in the HH environment are.

(52) $$license_1(x) \overset{\text{def}}{=} \texttt{TRUE}$$

(53) $$license_2(x) \overset{\text{def}}{=} H(x) \wedge H(s(x)) \wedge \neg(\texttt{last}(x))$$

Notice that the licensing functions must necessarily specify an epenthetic segment in terms of the input. In general, which segment is epenthesized is irrelevant from a mathematical perspective. The epenthetic segment can be any element of the domain, and the choice is completely determined by the nature of the logical formula. This is a distinct departure from rule-based accounts or correspondence theory accounts in constraint violation frameworks like Optimality Theory or Harmonic Grammar (McCarthy and Prince, 1995). These frameworks do not specify the origin of an epenthetic segment in terms of input, and may prohibit such a specification. In logical transductions, specification is a necessary feature by design. representational choices make claims about the universe of discourse. One can make motivated choices about the nature of an epenthetic statement, or be agnostic, as I have done here. This explicitness is an advantage, as one might discover that there are independently motivated reasons for positing a certain element to be the epenthetic one.

Additionally, the flexibility of specifying the epenthetic segment gives further advantages, by allowing one to relabel the epenthetic segment directly in the mapping, just as was done earlier. This is handled by the unary labels, and may be done in any number of ways. Here, since we have two output copies, we must specify the output labels for each copy. In this transduction, unary labels in the first copy remain as they were in the input (54-56), and the copy that defines the epenthetic segment has all its segments relabeled as M's, while all H information is eliminated, as shown in (55-57), where $M(x)$ is true of every segment, and $H(x)$ is true of none.

(54) $$M_1^O(x) \overset{\text{def}}{=} M(x)$$

(55) $$M_2^O(x) \overset{\text{def}}{=} \texttt{TRUE}$$

(56) $$H_1^O(x) \overset{\text{def}}{=} H(x)$$

(57) $$H_2^O(x) \overset{\text{def}}{=} \texttt{FALSE}$$

Next we specify the successor and predecessor functions. The environment for the rule is an HH sequence.

(58) $$s((x,i)) \overset{\text{def}}{=} \begin{cases} (s(x),1) & when \quad \neg[H(x) \wedge H(s(x)) \wedge \neg(\texttt{last}(x))], \quad i = 1 \\ (x,2) & when \quad [H(x) \wedge H(s(x)) \wedge \neg(\texttt{last}(x))], \quad i = 1 \\ (s(x),1) & when \quad [H(x) \wedge H(s(x)) \wedge \neg(\texttt{last}(x))] \quad i = 2 \end{cases}$$

(59) $$p((x,i)) \overset{\text{def}}{=} \begin{cases} (p(x),1) & when \quad \neg[H(x) \wedge H(p(x) \wedge \neg(\texttt{first}(x))], \quad i = 1 \\ (p(x),2) & when \quad [H(x) \wedge H(p(x) \wedge \neg(\texttt{first}(x))], \quad i = 1 \\ (x,1) & when \quad [H(x)) \wedge H(s(x)) \wedge \neg(\texttt{last}(x))] \quad i = 2 \end{cases}$$

As mentioned above, epenthetic segments absorb autosegmental feature characteristics of their surrounding segments. This is handled at the same time in the mapping in exactly the same way as handshape spreading in compound reduction. Again, since there are multiple copies, we must specify multiple association relation mappings, one for each copy, and one between copy sets 1 and 2. (60-61) says that the first copy has the same relations as in the input. However, since only one element survives in the second copy, the association disappears. Association between the word and the epenthetic segment are handled by (62), which spreads the association lines of the successor or its predecessor if they were both non-final H's, exactly the phonological environment of the mapping. Note again that these are independent of the unary label of the epenthetic segment, though the flexibility of the logic allows one to place further substantive restrictions, if desired.

(60) $$A(x,y)_{1,1} \stackrel{\text{def}}{=} A(x,y)$$

(61) $$A(x,y)_{2,2} \stackrel{\text{def}}{=} A(x,y)$$

(62) $$A(x,y)_{1,2} \stackrel{\text{def}}{=} A(x,y) \wedge [[H(y) \wedge H(s(y)) \wedge \neg(\texttt{last}(x))]$$

(63) $$\vee [H(x) \wedge H(p(y) \wedge \neg(\texttt{first}(x))]]$$

Insertion of the M by the movement epenthesis rule often creates the environment for the deletion of a Hold segment (or L segment in the Hand Tier model) (Liddell and Johnson, 1986). Thus movement epenthesis feeds the Hold Deletion Rule which deletes holds that occur directly between the two movements in a string. It applies equally within words and across word boundaries. The rule deletes only the segmental bundle (the H) and does not affect the articulatory bundle. A more complete demonstration may be found in Liddell and Johnson (1989).

(64) $$H \rightarrow \varnothing \;/\; M\_M$$

The deletion mapping is handled exactly the same as before, requiring the licensing function to prevent licensing of domain elements whose predecessor and successor are both M, as shown below. I omit other parts of the mapping for space as they are simply identities.

(65) $$license(x) \stackrel{\text{def}}{=} \neg[H(x) \wedge M(p(x)) \wedge M(s(x))]$$

The ASL phrase GOOD IDEA (Figure 15) nicely illustrates the combination of both Movement Epenthesis and Hold Deletion. Both the sign GOOD and the sign IDEA have the form HMH. The final H ends away from the body at about the level of the sternum. The sign IDEA begins with the little finger in contact with the side of the forehead. Movement epenthesis brings the hand from the end of GOOD to the beginning of IDEA. The complete input model for the compound is shown in Figure 16, and bears close resemblence to the description of it by Liddell and Johnson (1986). The output model of movement epenthesis is described in Figure 17. In the output model I have indexed the copy set that each segment comes from using a superscript. The epenthesized segment, 3 in this case, has as predecessor its corresponding segment in Copy set 1 (3, an H) and its successor in copyset 1 (4, also an H). It has been relabeled an M. This output model creates the MHM sequence that triggers the Hold Deletion mapping, the output of which is shown in Figure 18.

Figure 15: Citation form of ASL 'GOOD' (left) 'IDEA' (middle), and the form 'GOOD IDEA' (right). Illustrations from Liddell and Johnson (1986)

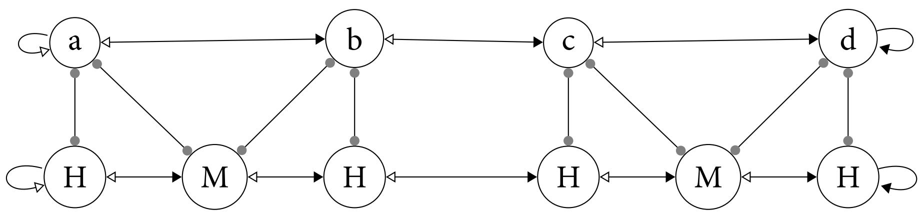

Figure 16: Input Word Model for ASL 'GOOD'. Domain element indices omitted for readability.

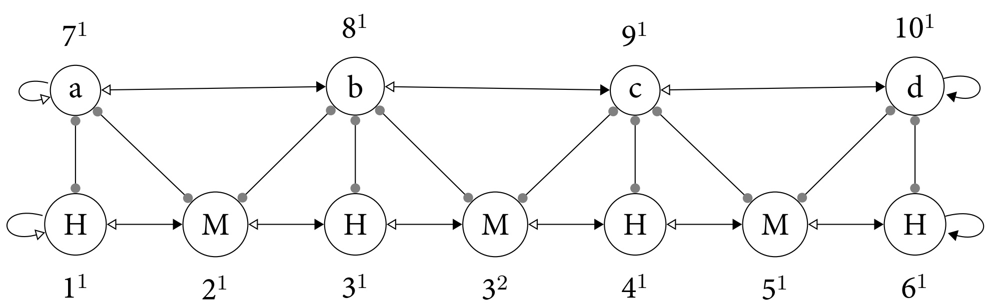

Figure 17: Output of Move epenthesis. Domain element indices indicated outside each circle, with superscript indicating the copy set. Unlicensed elements omitted for readability

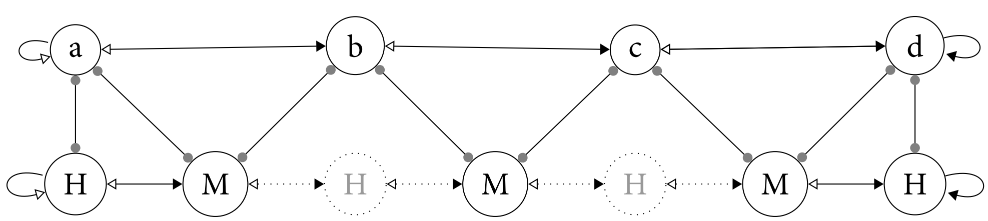

Figure 18: Output of Hold deletion. Domain element indices omitted for readability

4.4 Predictions and Further Analyses

To conclude, this section presented a model-theoretic characterization of several phonological processes in sign language. Each of the processes was sufficiently described with Quantifier-Free logical mappings over autosegmental representations, meaning each of the mappings is A-ISL. These results generalize those of Rawski (2017), and show that the computational nature of the processes is fundamentally strictly local across modalities, even relativized over different representations. Finite Model Theory and logical mappings provide a firm mathematical framework for analyzing the nature of phonological representation and computation in speech and sign on their own terms, without utilizing a framework designed around either modality.

While independent of a particular framework, the preceding analyses, combined with the monosyllabic tendencies in sign, generate a prediction: that the nature of other processes will also be computationally local. The most direct way to test this is to analyze other phonological processes observed in speech, sign, or both, show that a process is *necessarily non-QF*. There are many avenues to consider. For example, beginning and end locations in signs may undergo environmentally-conditioned metathesis triggered by the immediately preceding sign's location, a "bounded non-local" process that Chandlee (2014) shows is ISL in spoken language. Handshape configuration and place autosegments contain a dizzying feature geometry which participates in spreading and assimilation processes in subtle ways that may enrich the view of autosegmental spreading in ways that examining A-ISL tone mappings (Chandlee and Jardine, 2019) could not.

The complexity of other processes, such as gemination of edge locations to form the resultative (Sandler, 1989), are a natural domain for model-theoretic characterization given the feature geometry. The sign reduplicative typology is also rich, and nonconcatenatively interacts with morphology and phonology in subtle ways, such as triplication preceding epenthesis and spreading in habitual aspect, or templatic subject-object inflection followed by reduplication. Analysis of the limited cases of sequential affixation, such as negative incorporation (Woodward Jr, 1974) can provide a direct comparison with the strictly local morphological work of Chandlee (2017).

Additionally, the logical perspective can shed light on the advantages of particular theories of the signed word, of which there are many, often conflicting. Comparisons between the representations of one theory vs another, the complexity of operations over them, or the complexity of translating between theories, is a common use of the model-theoretic approach. One can determine whether theories are substantively different from one another, or notational variants that can be intertranslated between. For an example of this approach in phonology, see (Jardine et al., 2020)

Strict Locality appears to be representationally salient for phonological computation across modalities. This provides some intriguing consequences for a longstanding debate in phonology on the relationship between sequentiality and simultaneity across modalities.

5 Locality, Sequentiality, and Simultaneity

This section examines the nature of sequentiality and simultaneity in light of the Strict Locality discussed above. Languages in both modalities have sequential structure, but there are striking differences in the nature of that structure, as the model-theoretic perspective shows. Spoken languages vary in syllable structure, word length, and stress patterns among syllables. Sign languages appear limited in all these aspects. They are overwhelmingly monosyllabic, have no clusters, and show extremely simple stress patterns, due to few polysyllabic words apart from fully reduplicated forms

(Wilbur, 2011).

As the previous sections showed, the structural organization in signed or spoken language has a direct effect on the phonology. Strikingly few phonological rules that are not morphosyntactically triggered have been discovered in sign languages, mainly due to sign's lack of sequential structure. Significant sequential structure in sign mainly appears under morphosyntactic operations that concatenate morphemes and words (affixation, compounding, and cliticization). Predictably, when this occurs, a smorgasbord of phonology arises.

In general, sequential affixation is rare across sign languages (Sandler, 1996; Aronoff et al., 2005), and sign exhibits a strong tendency to express concatenative morphology through compounding (Meir, 2012). Aronoff et al. (2005) show that affixation usually results from grammaticalization of free words, via a series of diachronic changes concerning phonological and semantic factors. They cite the relative youth of sign languages as causing their lack of affixes compared with verbal languages. No known sign languages are over 300 years old, with some like Nicaraguan Sign Language, as young as 40 (Woll et al., 2001).

The lack of sequential structure in sign languages does not imply structural simplicity, however. Sign languages routinely employ nonconcatenative morphology (Emmorey, 2001; Meier, 2002), incorporating morphological material simultaneously in the phonology with restricted sequential form. Of course, simultaneous phonological structure exists in all languages, but differ across modalities in the amount. Very few sign features actually become sequenced, while in spoken language features are overwhelmingly sequenced, rarely simultaneous. Comparing hand configuration and place autosegments to autosegments in spoken language shows further differences. Unlike spoken autosegments for tone or harmony patterns, which typically consist of one or two features, the hand configuration autosegment in sign languages is extremely complex, containing almost half of sign features organized in an intricate feature geometry (van der Hulst, 1995; Sandler, 1996).

The model-theoretic perspective brings two important perspectives into stark relief. The first is the salience of autosegmental representation: out of all the possible data structures sign language could be designed with, given its two-dimensional modality, it is still fundamentally autosegmental. Model theory makes clear the massive universe of types of structures that can be computed over — strings, trees, hypergraphs, etc. That sign uses patterns that can be represented with autosegmental structures, an extremely restricted class of structures also salient in spoken phonology, suggests a striking representational universal.

The second point concerns tradeoffs. The logical and computational nature of tonal patterns in spoken languages is among the most computationally powerful in spoken phonology. Jardine (2016a) shows that tonal patterns with string representations require the full power of the regular relations, or equivalently, full Monadic Second-Order Logic. In contrast, all known segmental processes occupy subclasses of these relations, which an overwhelming majority of them sitting in the ISL region (Chandlee, 2014), sufficiently characterized by quantifier-free maps. Jardine conjectures that tonal patterns are by nature computationally more expressive. Jardine (2017b), and Chandlee and Jardine (2019) show that representationally switching to graphs that capture the autosegmental nature of tone pushes the complexity of some processes down, but some are still beyond A-ISL.

If sign languages were to take full advantage of the simultaneous abilities that the modality provides, and at the same time have access to the sequential range that spoken processes have, the complexity of a linguistic processes might dramatically change, perhaps out of the ISL class, when it is filtered through the representational capacity given by the modality. A similar claim could be made for spoken phonology if they had access to the same simultaneous range that sign has, given the range of suprasegmental process complexity.

Alternatively, if the phonological system is computationally limited to, or at least strongly biased toward Strict Locality in its mappings, as I conjectured earlier, then a representational tradeoff arises naturally. Spoken languages, limited in their simultaneity but not their sequentiality, can satisfy the computational ISL requirements (bounded locality and a limited memory) by expressing the majority of their phonology sequentially, and limiting the simultaneous expressivity to certain circumstances. Conversely, sign languages may limit the amount of sequentiality in phonological operations, satisfying the computational requirements via the representation by taking advantage of the rich simultaneity afforded them by the nature of the modality and the articulation system. The resulting small number of sequential distinctions in signs may also be compensated for by a larger number of features, to maintain a similar number of lexical contrasts as spoken language.

While the capacity for linearity and non-linearity are common to both spoken and signed languages, the relative computational centrality of each differs in the phonological organization of each modality. The model-theoretic and logical perspective thus allows a more nuanced version of the "adapted system" view of sign language in humans specialized by evolution for use of spoken language. If aspects of the articulatory/perceptual system are somehow compromised, the computational pressures characteristic of phonological organization combined with the modular features of the sign modality enable representations where simultaneity takes power over sequentiality.

6 Phonological Cognition Across Modalities

A central challenge of the cognitive neuroscience of language is to "identify representations and operations that can be linked to the types of operation that simple electrical circuits can execute" (Poeppel, 2012). Poeppel suggests "theoretically well-motivated units of representation or processing deriving from cognitive science research (here, say linguistics); then one attempts to decompose these into elementary constituent operations that are formally generic". This is exactly what Model Theory provides. Model-theoretic representations of linguistic structure, and the logical information needed to compute the constraints and processes characterizing human language, make principled claims about the nature of cognitive representations and operations that underlie them. This has direct implications for the interplay between phonology and modality.

Recall that phonology across speech and sign is sufficiently characterized by Regular languages and relations, which are describable with MSO logic and finite-state machinery. Any cognitive mechanism that can perform computations over a finite-state set must be capable of classifying input-output members into a finite set of abstract categories and be sensitive to the sequence of those categories (Rogers et al., 2013). This subsumes any processing mechanism in which the amount of information inferred or retained is limited by a fixed finite bound. Any cognitive mechanism that has such a fixed finite bound in processing sequences of events will be able to recognize only finite-state sets of structures or processes.

Across spoken and signed processes, "regularity" is a sufficient, but not necessary, condition on the cognitive capacity required for processing. The restriction to or bias toward processes describable by QF mappings or ISL functions is a stronger sufficient claim, as these are a proper subset of the Regular relations. Any cognitive mechanism that can distinguish input-output relations of this kind must be sensitive, at least, to a bounded number of blocks of events that occur in the presentation of the structure. The sufficiency of local evaluation inherent to QF logic in computing the functions characteristic of natural language phonology is important. It shows that the phonological module is amodally sensitive to bounded chunks of structural information at each moment of the process. The model-theoretic perspective makes this clear, by precisely defining the nature of the structures and the computation working over them. In more cognitive terms, this means looking at finite chunks of structure and a very restricted notion of memory.

This is not to say that the discovery of a process in either modality that is not describable by a QF mapping, or an ISL function, over graphs, invalidates the claims. In fact, several have been discovered, though with increasing dispreference as one approaches full regular power (see Heinz (2018)). What it does say is that one has choices. If one is committed to a cognitive view where computation is limited and works in this local way, then one can choose to impose more structure into the model signature, or adjust the structures which are already there. This is the approach taken by Heinz (2010), who makes the case for a general precedence relation rather than a successor relation, and by Jardine (2017a), who argues for incorporating more structure to handle autosegments.

Alternatively, if one is committed to the view of phonology with a particular structure, meaning a particular signature or part of a signature, then one can understand the relationship of that structure to the range of computational power needed to handle phonological processes using that structure. Most of the work in understanding the complexity of phonological processes has assumed a string structure, and there is extensive work carving out the precise range of computations required for phonology (Graf, 2017). This too has consequences, since this carving may require a further modular view of the phonological system depending on the classes of computations one finds to be necessary and sufficient. This also modularizes the view of the learning system, as successful learning algorithms are often tied to the data structures which characterize these function classes (Heinz, 2010; Heinz and Idsardi, 2013).

The salience of certain representations and computational aspects across modalities suggests that certain parts of the phonological module are amodal, that is, independent of modality, and in some cases constrain representation in the specific modality the phonology is expressed in. So where does this leave modality effects? Model theory allows explicit comparison of the nature of phonological properties across modalities to see where these differences lie: representation, or computation? If a particular process differs across modalities, one can precisely characterize which aspect of representation or computation is responsible. This gives a promising avenue for future research on modality and amodality in the phonological system, and concrete testable hypotheses for experimental approaches.

As mentioned in the preceding section, the phonological module may accommodate the representational abilities of the particular articulatory/perceptual system to satisfy the requirements of computation. This view has some independent support. van der Hulst and van der Kooij (2018), cite Brentari (2002) and Emmorey (2001) that visual perception of signs (even with sequential properties) is more "instantaneous" than auditory speech perception, and adapt Goldsmith (1976)'s division of phonology in terms of the notions of "vertical and horizontal slicing of the signal". They state

> an incoming speech signal is first spliced into vertical slices, which gives rise to a linear sequence of segments. Horizontal slicing then partitions segments into co-temporal feature classes and features. In the perception of sign language, however, the horizontal slicing takes precedence, which gives rise to the simultaneous class nodes that we call handshape, movement, and place. Then, a subsequent vertical slicing of each of these

can give rise to a linear organization

Of course, the phonetics and phonology of sign language differ from spoken language in many ways, and this is expected. Lillo-Martin (1997) cites Blakemore (1974)'s result that exposure to vertical and horizontal lines in the environment affects development of feline visual perception, and asks "why shouldn't exposure to the special acoustic properties of the modality affect perception, especially auditory perception?" For example, Sandler and Lillo-Martin (2006) note that unlike spoken syllables in many languages, sign language syllables prohibit location clusters comparable to consonant clusters, or diphthong-like movement clusters, and there must be a movement between locations due to the physiology of the system. Additionally, sign syllables do not have onset-rhyme asymmetries, which affects syllable structure and stress assignment, and they typically align intonation, conveyed by facial expression, with phonological/intonational phrases, not syllables inside those phrases, where spoken languages usually do (Nespor and Sandler, 1999).

However, the fact remains that locality and a bounded memory are representationally salient for and computationally exploited by the phonological module across modalities. These properties constrain the expression of phonological content by taking advantage of the particular properties of the modality. This represents a sufficient condition for amodality, and offers a promising route to exploring other necessary and sufficient conditions for it. The freedom and preciseness given by model-theoretic phonology and the computational tradeoffs that come with it give a promising answer to Poeppel (2012)'s call to "focus on the operations and algorithms that underpin language processing", since "the commitment to an algorithm or computation in this domain commits one to representations of one form or another with increasing specificity and also provides clear constraints for what the neural circuitry must accomplish."

7 Conclusion

This article presented a logical characterization of the nature of phonological computation in spoken and signed language. Model-theoretic analyses of phonological processes in sign were shown to require the same logical power as their spoken counterparts, namely, quantifier-free mappings, or ISL functions. It was predicted that almost all phonological processes in sign share this complexity due to the bounded nature of the sign, just as most phonological processes in spoken language fall into the ISL class. It was further conjectured that this computational constraint causes a tradeoff in the organization of phonological representations in each modality — more sequential structure in speech, and more simultaneous structure in sign. This has strong implications for the nature of the phonological module as an aspect of the cognitive capacity for language, highlighting the relevance of model-theoretic methods in addressing representational and cognitive questions, and providing a principled way to investigate the nature of phonology across speech and sign.

References

Aronoff, M., Meir, I., and Sandler, W. (2005). The paradox of sign language morphology. *Language*, 81(2):301.

Berent, I. (2013). *The Phonological Mind*. Cambridge University Press.

Blake, F. R. (1917). Reduplication in tagalog. *The American Journal of Philology*, 38(4):425–431.

Blakemore, C. (1974). Developmental factors in the formation of feature extracting neurons. *The neurosciences: Third study program*, pages 105–113.

Brentari, D. (1990). Licensing in asl handshape change. *Sign language research: Theoretical issues*, pages 57–68.

Brentari, D. (1998). *A prosodic model of sign language phonology*. Mit Press.

Brentari, D. (2002). Modality differences in sign language phonology and morphophonemics. *Modality and structure in signed and spoken languages*, pages 35–64.

Büchi, J. R. (1960). Weak second-order arithmetic and finite automata. *Mathematical Logic Quarterly*, 6(1-6):66–92.

Byrd, D. (1993). Marshallese suffixal reduplication. In *Proceedings of the Eleventh West Coast Conference on Formal Linguistics*, pages 61–77.

Chandlee, J. (2014). *Strictly Local Phonological Processes*. PhD thesis, U. of Delaware.

Chandlee, J. (2017). Computational locality in morphological maps. *Morphology*, pages 1–43.

Chandlee, J., Eyraud, R., and Heinz, J. (2014). Learning strictly local subsequential functions. *Transactions of the Association for Computational Linguistics*, 2:491–503.

Chandlee, J. and Heinz, J. (2018). Strict locality and phonological maps. *Linguistic Inquiry*.

Chandlee, J., Heinz, J., and Jardine, A. (2018). Input strictly local opaque maps. *Phonology*, 35(2):171–205.

Chandlee, J. and Jardine, A. (2018). Computational locality and autosegmental processes. In *Proceedings of the 54th Annual meeting of the Chicago Linguistics Society*, volume 54.

Chandlee, J. and Jardine, A. (2019). Autosegmental input strictly local functions. *Transactions of the Association for Computational Linguistics*.

Chandlee, J. and Lindell, S. (2016). Local languages. Paper presented at the 4th Workshop on Natural Language and Computer Science, in affiliation with LICS at Columbia University, NY.

Chomsky, N. (1956). Three models for the description of language. *IRE Transactions on information theory*, 2(3):113–124.

Courcelle, B. (1994). Monadic second-order definable graph transductions: a survey. *Theoretical Computer Science*, 126(1):53–75.

Emmorey, K. (2001). *Language, cognition, and the brain: Insights from sign language research*. Psychology Press.

Enderton, H. B. (2001). *A Mathematical Introduction to Logic*. Academic Press, 2nd edition.

Engelfriet, J. and Hoogeboom, H. J. (2001). MSO definable string transductions and two-way finite-state transducers. *ACM Transactions on Computational Logic (TOCL)*, 2(2):216–254.

Fagin, R., Stockmeyer, L., and Vardi, M. (1995). On monadic NP vs monadic co-NP. *Information and Computation*, 120(1):78 – 92.

Filiot, E. (2015). Logic-automata connections for transformations. In *Indian Conference on Logic and Its Applications*, pages 30–57. Springer.

Finley, S. (2009). *Formal and cognitive restrictions on vowel harmony*. PhD thesis, The Johns Hopkins University.

Frishberg, N. (1975). Arbitrariness and iconicity: historical change in american sign language. *Language*, pages 696–719.

Goldsmith, J. A. (1976). *Autosegmental phonology*, volume 159. Indiana University Linguistics Club Bloomington.

Graf, T. (2010). Logics of phonological reasoning. Master's thesis, University of California, Los Angeles.

Graf, T. (2017). The power of locality domains in phonology. *Phonology*, 34(2):385–405.

Heinz, J. (2010). Learning long-distance phonotactics. *Linguistic Inquiry*, 41(4):623–661.

Heinz, J. (2018). The computational nature of phonological generalizations. In Hyman, L. and Plank, F., editors, *Phonological Typology*, Phonetics and Phonology, chapter 5, pages 126–195. De Gruyter Mouton.

Heinz, J. and Idsardi, W. (2013). What complexity differences reveal about domains in language. *Topics in cognitive science*, 5(1):111–131.

Jardine, A. (2016a). Computationally, tone is different. *Phonology*, 33(2):247–283.

Jardine, A. (2016b). *Locality and non-linear representations in tonal phonology*. PhD thesis, University of Delaware.

Jardine, A. (2017a). The expressivity of autosegmental grammars. *Journal of Logic, Language and Information*, pages 1–46.

Jardine, A. (2017b). The local nature of tone-association patterns. *Phonology*, 34:385–405.

Jardine, A., Danis, N., and Iacoponi, L. (2020). A formal investigation of q-theory in comparison to autosegmental representations. *Linguistic Inquiry*, 1:1–51.

Johnson, C. D. (1972). *Formal aspects of phonological description*. The Hague: Mouton.

Kaplan, R. M. and Kay, M. (1994). Regular models of phonological rule systems. *Computational linguistics*, 20(3):331–378.

Karttunen, L. (1993). Finite-state constraints. *The last phonological rule*, pages 173–194.

Klima, E. S. and Bellugi, U. (1979). *The Signs of Language*. Harvard University Press.

Lepic, R. (2015). *Motivation in morphology: Lexical patterns in ASL and English*. PhD thesis, UC San Diego.

Libkin, L. (2004). *Elements of Finite Model Theory*. Springer.

Liddell, S. K. (1984). Think and believe: sequentiality in american sign language. *Language*, pages 372–399.

Liddell, S. K. and Johnson, R. E. (1986). American sign language compound formation processes, lexicalization, and phonological remnants. *Natural Language & Linguistic Theory*, 4(4):445–513.

Liddell, S. K. and Johnson, R. E. (1989). American sign language: The phonological base. *Sign language studies*, 64(1):195–277.

Lillo-Martin, D. (1997). The modular effects of sign language acquisition. *Relations of language and thought: The view from sign language and deaf children*, pages 62–109.

McCarthy, J. J. (1981). A prosodic theory of nonconcatenative morphology. *Linguistic inquiry*, 12(3):373–418.

McCarthy, J. J. (1982). Prosodic templates, morphemic templates, and morphemic tiers. In der Hulst, H. V. and Smith, N., editors, *The structure of phonological representations*, pages 191––223. Dordrecht: Foris.

McCarthy, J. J. and Prince, A. S. (1990). Foot and word in prosodic morphology: The arabic broken plural. *Natural Language & Linguistic Theory*, 8(2):209–283.

McCarthy, J. J. and Prince, A. S. (1995). Faithfulness and reduplicative identity.

Meier, R. P. (2002). Why different, why the same? explaining effects and non-effects of modality upon linguistic structure in sign and speech. *Modality and structure in signed and spoken languages*, pages 1–25.

Meir, I. (2012). Word classes and word formation. *Sign language. An international handbook. Berlin: De Gruyter Mounton*, pages 77–112.

Mielke, J. (2007). P-base 1.92. *Available (July 2010) at http://aix1. uottawa. ca/~ jmielke/pbase/index. html*.

Nespor, M. and Sandler, W. (1999). Prosody in israeli sign language. *Language and Speech*, 42(2-3):143–176.

Newkirk, D. E. (1998). On the temporal segmentation of movement in american sign language. *Sign language & linguistics*, 1(2):173–211.

Newport, E. L. (1981). Constraints on structure: Evidence from american sign language and language learning. In *Minnesota symposium on child psychology*, volume 14, pages 93–124. Erlbaum Hillsdale, NJ.

Perlmutter, D. M. (1993). Sonority and syllable structure in american sign language. In *Current issues in ASL phonology*, pages 227–261. Elsevier.

Poeppel, D. (2012). The maps problem and the mapping problem: two challenges for a cognitive neuroscience of speech and language. *Cognitive neuropsychology*, 29(1-2):34–55.

Potts, C. and Pullum, G. K. (2002). Model theory and the content of OT constraints. *Phonology*, 19:361–393.

Pullum, G. K. (2007). The evolution of model-theoretic frameworks in linguistics. In Rogers, J. and Kepser, S., editors, *Model-Theoretic Syntax at 10*, pages 1–10, Dublin, Ireland.

Rawski, J. (2017). Phonological complexity across speech and sign. In *Proceedings of the 53rd Chicago Linguistics Society Annual Meeting*.

Reiss, C. (2018). Substance free phonology. *The Routledge Handbook of Phonological Theory*, pages 425–452.

Rogers, J. (1998). *A descriptive approach to language-theoretic complexity*. CSLI Publications Stanford, CA.

Rogers, J., Heinz, J., Fero, M., Hurst, J., Lambert, D., and Wibel, S. (2013). Cognitive and sub-regular complexity. In Morrill, G. and Nederhof, M.-J., editors, *Formal Grammar*, volume 8036 of *Lecture Notes in Computer Science*, pages 90–108. Springer.

Rogers, J. and Pullum, G. K. (2011). Aural pattern recognition experiments and the subregular hierarchy. *Journal of Logic, Language and Information*, 20:329–342.

Sandler, W. (1986). The spreading hand autosegment of american sign language. *Sign Language Studies*, 50(1):1–28.

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

Sandler, W. (1993). Sign language and modularity. *Lingua*, 89(4):315–351.

Sandler, W. (1996). *Representing handshapes*, volume 1. Lawrence Erlbaum Associates, Inc., Mahwah, NJ.

Sandler, W. and Lillo-Martin, D. (2006). *Sign language and linguistic universals*. Cambridge University Press.

Savitch, W. J. (1993). Why it might pay to assume that languages are infinite. *Annals of Mathematics and Artificial Intelligence*, 8(1-2):17–25.

Schlenker, P. (2018). Visible meaning: Sign language and the foundations of semantics. *Theoretical Linguistics*, 44(3-4):123–208.

Scobbie, J. M., Coleman, J. S., and Bird, S. (1996). *Key aspects of Declarative Phonology*. European Studies Research Institute, Salford.

Shoenfield, J. R. (1967). *Mathematical logic*, volume 21. Reading: Addison-Wesley.

Strother-Garcia, K. (2018). Imdlawn tashlhiyt berber syllabification is quantifier-free. In *Proceedings of the Society for Computation in Linguistics*, volume 1.

Thomas, W. (1997). Languages, automata, and logic. In *Handbook of Formal Languages*, volume 3, chapter 7. Springer.

van der Hulst, H. (1993). Units in the analysis of signs. *Phonology*, 10(2):209–241.

van der Hulst, H. (1994). Dependency relations in the phonological representation of signs. *Sign language research*, pages 11–38.

van der Hulst, H. (1995). The composition of handshapes. *Trondheim Work. Papers*, 23:1–17.

van der Hulst, H. and van der Kooij, E. (2018). Phonological structure of signs – theoretical perspectives. In Josep Quer Villanueva, R. P. and Herrmann, A., editors, *The Routledge Handbook of Theoretical and Experimental Sign Language Research*. Routledge.

Wilbur, R. (1982). A multi-tiered theory of syllable structure for american sign language. In *Annual Meeting of the Linguistic Society of America, San Diego*.

Wilbur, R. (2011). Sign syllables. *The Blackwell companion to phonology*, 1:1309–1334.

Williams, E. S. (1976). Underlying tone in margi and igbo. *Linguistic Inquiry*, pages 463–484.

Woll, B., Sutton-Spence, R., and Elton, F. (2001). Multilingualism: The global approach to sign languages. *The sociolinguistics of sign languages*, pages 8–32.

Woodward Jr, J. C. (1974). Implicational variation in american sign language: negative incorporation. *Sign Language Studies*, 5(1):20–30.