Algebraic Reanalysis of Phonological Processes Described as Output-Oriented

Dakotah Lambert
Université Jean Monnet Saint-Étienne, CNRS
Institut d Optique Graduate School
Laboratoire Hubert Curien
F-42023, Saint-Étienne, France
dakotahlambert@acm.org

Jeffrey Heinz
Stony Brook University
Department of Linguistics
Institute for Advanced Computational Science
jeffrey.heinz@stonybrook.edu

Abstract

Many phonological processes – including exemplars of local harmony, iterative spreading, and long-distance harmony patterns – have been shown to belong to the Output (Tier-based) Strictly Local (O(T)SL) functions. This article provides an algebraic analysis of these processes. The algebraic approach to subregular pattern complexity is important because it unifies the computational characterizations of constraints and processes, while leveraging a wealth of results in theoretical computer science on the structural properties of these classes. These structural properties are useful because they underlie algorithms for classifying and learning.

The first result shows that the O(T)SL class has no corresponding algebraic characterization. The second result establishes that canonical examples of these processes belong to the definite and reverse definite classes, or their tier-based extensions, some of the simplest algebraic classes. The third result provides a single learning algorithm for these classes which identifies them in the limit from positive data.

1 Introduction

Local harmony, iterative spreading, and long-distance harmony processes are ubiquitous in natural languages and have been the subject of much linguistic (Walker, 1998, 2011, 2014; Rose and Walker, 2004; Hansson, 2010; Nevins, 2010; van der Hulst, 2018) and computational (Heinz et al., 2011; Heinz and Lai, 2013; Chandlee et al., 2015; Aksënova and Deshmukh, 2018; Burness and McMullin, 2019; Burness et al., 2021; Lambert, 2023) research. Much of this latter work studies the computational properties of these processes when viewed as string-to-string functions.

Following Filiot et al. (2019), we consider algebraic analyses of such functions. Each function can be associated with a semigroup and classified according to properties of that semigroup. For example, Lambert and Heinz (2023) proved that, considering only total functions, the input strictly local functions (ISL) (Chandlee et al., 2014) are precisely the algebraic variety of **definite** functions.

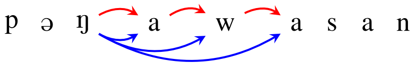

Figure 1: Two analyses of iterative spreading

Output Strictly Local (OSL) functions (Chandlee et al., 2015) are one way to characterize and represent phonological processes such as local harmony and iterative spreading (Chandlee and Heinz, 2018), and when combined with tiers, long-distance harmony (Burness et al., 2021). These processes are commonly understood as **output-oriented** because the output at any given point appears to depend on some prior output.

As an example, consider iterative spreading in Johore Malay where /pəŋawasan/ 'supervision' is pronounced as [pəŋãw̃ãsan] with nasalization on a successive sequence of vowels and semivowels (Heinz, 2010). Consider the rules shown below.

$$[-\text{cons}] \rightarrow [+\text{nas}]/[+\text{nas}]\_\_ \tag{1}$$

$$[-\text{cons}] \rightarrow [+\text{nas}]/[+\text{nas}][-\text{cons}]^{*}\_\_ \tag{2}$$

In order for Rule 1 to account for the nasal iterative spreading in Malay, it must apply iteratively from left to right. Consequently, the second [ã] is nasalized because the preceding glide has been nasalized in an earlier iteration of the rule. On the other hand, Rule 2 can apply simultaneously. The analysis with Rule 1 is considered output-oriented, but not the analysis with Rule 2. The applications of Rules 1 and 2 are schematized in Figure 1 in red and blue respectively. This issue is relevant today: Walker (2014) argues on empirical grounds that some harmony processes in some languages should be analyzed in the way suggested by Rule 2,

Proceedings of the Society for Computation in Linguistics (SCiL) 2024, pages 129-138.
Irvine, California, June 27-29, 2024

though her analysis uses Optimality Theory.

This article contains three main results. The first is that every finite semigroup is the syntactic semigroup of some OSL function. Since there are sequential functions that lie outside of this class, no algebraic variety can contain all and only the O(T)SL functions.

The second result is an algebraic analysis of a sample of canonical local harmony, iterative spreading, and long-distance harmony processes. By "canonical", we mean specific patterns in the literature that served to motivate the Output Strictly Local class and other related classes (Chandlee et al., 2015; Burness et al., 2021). The focus is less on the processes themselves, and more on the algebraic techniques by which they are classified. We make no claim that these processes are representative of the most complex of attested phenomena, which is often the subject of debate (see for example Kula and Syed, 2020).

For each process, we provide an algebraic analysis and discuss the classes in which it lies. We find that the patterns we consider have their behaviors fixed by either the k most recent symbols encountered or the first k symbols encountered, potentially projected onto a tier. This corresponds to the (tier-based) definite or (tier-based) reverse-definite classes of formal languages. In other words, the actual processes the O(T)SL functions were introduced to describe actually belong to some of the simplest and most restrictive algebraic classes. Furthermore, in the case of iterative spreading and long-distance harmony, the algebraic analysis indicates an interpretation akin to Rule 2. In this way, this paper provides a deeper insight into processes that have been described as output-oriented in the phonological literature.

Third, we present a learning algorithm, based on the smallest algebraically natural class which includes these functions, and prove it is learnable in the limit from positive data. As such, this algorithm does not take into account the output-oriented nature of the processes considered.

Section 2 recalls some relevant definitions. Then §3 shows how to conduct an algebraic analysis using post-nasal voicing as a running example. Then §4 demonstrates that no algebraic property can distinguish the output (tier-based) strictly local functions from arbitrary other sequential functions. §5 follows by providing algebraic analyses for several other processes that have been analyzed as output-oriented. §6 presents a learning algorithm based on SOSFIA (Jardine et al., 2014) that is powerful enough to handle the processes considered. Discussion and concluding remarks follow in §7.

2 Preliminaries

This section recalls basic definitions and notation. Given a finite alphabet Σ, let Σ^* denote the set of finite strings over Σ. Let λ denote the string of length 0 and $|w|$ the length of string w. For all strings $w \in \Sigma^*$, define $\mathrm{Suff}_k(w)$ to be the string v if there exists $u \in \Sigma^*$ such that $w = uv$ and $|v| = k$ and to be w otherwise.

A **tier** T is a subset of Σ and the **tier projection** of a string w is defined recursively as follows. For the base case, $\pi_T(\lambda) = \lambda$, and for the inductive case, $\pi_T(wa) = \pi_T(w)a$ iff $a \in T$ and $\pi_T(w)$ otherwise. Symbols in $\Sigma - T$ are called **neutral letters** and symbols in T are called **salient**.

A **semigroup** is a set S closed under an associative multiplication operation. An element a of S is **idempotent** whenever $aa = a$. If all elements of S are idempotent, then S is a **band**.

A **finite-state transducer** is an abstract machine that reads an input sequence, one symbol at a time, and produces one or more sequences as output (Raney, 1958). In this work, we are concerned only with **total**, **sequential** transducers, the subset of these machines in which computation is deterministic and each input sequence produces one and only one output sequence (Schützenberger, 1977). Formally, such a machine is a 8-tuple: $\mathcal{A} = \langle Q, \Sigma, \Gamma, \delta, q_0, \rho, \sigma \rangle$, where Q is a finite set of states, Σ a finite set of input symbols, Γ a finite set of output symbols, $\delta\colon Q \times \Sigma \to Q \times \Gamma^*$ a transition function, $q_0 \in Q$ an initial state, $\rho \in \Gamma^*$ a prefix prepended to all output sequences, and finally $\sigma\colon Q \to \Gamma^*$ a suffixing function.

The machine processing function μ is defined recursively. For the base case, let $\mu(q, \lambda, v) = v\sigma(q)$. The recurrence is given in Equation 3 below where $a \in \Sigma, \delta(q, a) = (q', w)$.

$$\mu(q, au, v) = \mu(q', u, vw) \tag{3}$$

Then the function $f\colon \Sigma^* \to \Gamma^*$ that $\mathcal{A}$ computes is $f(w) = \mu(q_0, w, \rho)$.

For every sequential function f, there is a unique (up to isomorphism) sequential transducer representing it, which is its **minimal onward form**. Informally, onwardness means the output is produced as early as possible. Readers are referred to Choffrut (2003) for technical details. The transducers introduced in this article, upon which the algebraic analyses are based, are all in minimal onward form.

Sequential functions come in two types: **left-to-right–sequential** and **right-to-left–sequential**. Left-to-right–sequential are defined as above. Right-to-left–sequential functions can be represented by transducers which process the input string from right to left.[1] In general, reversing the direction of the transducer computes the reversal of the process. For example, reversing the direction of a transducer for post-nasal voicing yields pre-nasal voicing, and reversing the direction of a transducer for regressive symmetric harmony yields progressive symmetric harmony.

It will be convenient to refer to the first component of the transition function. Whenever $\delta(q, a) = (q', w)$, we write $q * a = q'$.

For any tier $T \subseteq \Sigma$, a is a neutral letter if and only if for all $q \in Q$ it is the case that $q * a = q$ (Lambert, 2023). In other words, neutral letters are exactly those which never cause state to change.

A function f is **Output Strictly k-Local (k-OSL)** if there is sequential transducer representing f with the property that the current state is entirely determined by the $k - 1$ most recent symbols of output (Chandlee et al., 2015). In terms of the recurrence relation (Equation 3), this means that $q' = \mathrm{Suff}_{k-1}(vw)$.

The **Output Tier-based Strictly k-Local (k-OTSL)** functions are defined in the same way, except that the suffix is taken after projection to a fixed set of salient symbols (Burness et al., 2021). As with sequential functions, O(T)SL functions come in left-to-right and right-to-left variants.

Input Strictly k-Local (k-ISL) functions are also defined similarly where the suffix is taken over the input symbols (Chandlee et al., 2014).[2]

3 Algebraic Analysis

The algebraic theory of formal languages and functions provides a window into the kind of information to which a perceiver must attend when learning a pattern or when classifying it (Rogers et al., 2012; Filiot et al., 2016, 2019; Lambert, 2022).[3] This section explains the fundamentals of algebraic analysis of string-to-string functions using the phonological process of post-nasal voicing as a running example.

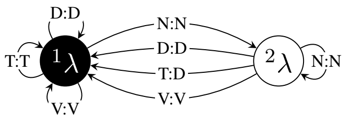

Figure 2: A minimal transducer for post-nasal voicing.

As an example, consider the phonological process of post-nasal voicing (PNV) in the Puyu Pungo dialect of Quechua, investigated by Burness et al. (2021). Here, a voiceless obstruent directly following a nasal becomes voiced. A transducer in minimal onward form for PNV is shown in Figure 2, where 'V' represents a vowel, 'N' a nasal consonant, 'T' a voiceless obstruent, and 'D' a voiced obstruent. States are labeled by an integer index and the output of the suffixing function σ. Edges are labeled with their input and output, in order, separated by a colon. The initial state is black.

Because this transducer is small, visual inspection is sufficient to establish that PNV is both ISL and OSL. The set of length-one input suffixes that lead to state 1 are $\{\$, \mathrm{D}, \mathrm{T}, \mathrm{V}\}$, where $\$$ represents the beginning of the string, while the set of those that lead to state 2 is $\{\mathrm{N}\}$. The sets are disjoint; thus, the function is ISL. Exactly the same analysis applies to output suffixes, so the function is OSL.

3.1 Transition Semigroups

Given a finite-state machine, its **transition semigroup**) is built from the **actions** of each letter, which are the changes they make to the state space. Formally, given a listing of the states in Q, $\langle q_0, \ldots q_n\rangle$, the action given by $a \in \Sigma$ is the tuple $\langle q_0 * a, \ldots q_n * a\rangle$. Note that distinct symbols may have the same action, which means they exhibit the same behaviors. Importantly, since neutral letters do not change state, their action is always the identity action $\langle q_0, \ldots q_n\rangle$, denoted $\mathbb{1}$.

The actions given by the letters form the **basis** of the transition semigroup. The rest of the semigroup is generated as follows. Given two actions a and b, one constructs the product ab by first applying a, then applying b to its result: $b \circ a$. The product is potentially a new action. However, as there are finitely many states, there are ultimately at most finitely many actions over these states. The transition semigroup is the set of actions generated under this composition, including the basis.

In the transducer for PNV in Figure 2, the list-

[1] One way for $\mathcal{A}$ to process w right-to-left is to give $\mathcal{A}$ the reverse of w and then reverse its output.

[2] The left-to-right and right-to-left ISL functions are the same class.

[3] A link to open source software for classifying and learning patterns will be provided upon acceptance.

Figure 3: Multiplication table (left) and eggbox diagram (right) for post-nasal voicing.

	x	y
x	x	y
y	x	y

x	y

ing of states is given by their index. Observe that $\langle 1{*}N, 2{*}N\rangle = \langle 2,2\rangle$, and $\langle 1{*}D, 2{*}D\rangle = \langle 1,1\rangle$. In fact, there are two actions which arise from individual letters in this transducer: $x = \langle 1,1\rangle$ from 'D', 'T' and 'V', and $y = \langle 2,2\rangle$ from 'N'. Thus, x, y are the basis of the transition semigroup.

Recall that successive actions in the transducer translate to multiplication in the semigroup. Observe that $xy = y$ because (i) $x = \langle 1,1\rangle$, which means that it maps state 1 to state 1 and state 2 to state 1, (ii) $y = \langle 2,2\rangle$ means that it maps state 1 to state 2 and state 2 to state 2, and (iii) when action x is followed by action y, the product action maps state 1 to state 2 and state 2 to state 2, which is the same action as y. Similar reasoning reveals that $xx = x$, $xy = y$, $yx = x$, and $yy = y$. Additional multiplication yields no new actions, so this pair of elements makes up the entire syntactic semigroup, shown in Figure 3 (left), where the cell at row x and column y is the product xy. The eggbox diagram shown at right in Figure 3 is another representation of the structure, which will be discussed in more detail in §3.4.

The transition semigroup of a transducer $\mathcal{A}$ *in minimal onward form* for a sequential function f is the **syntactic semigroup** of $\mathcal{A}$ (Filiot et al., 2016). Since the states of the minimal automaton correspond to minimally distinct behaviors, the syntactic semigroup indicates how input sequencing influences the behavior of f.

Since the transducer in Figure 2 for PNV is in minimal onward form, its transition semigroup is that function's syntactic semigroup.

3.2 Varieties

A **variety** is a class of semigroups closed under finitary direct products (tuples which multiply pointwise), quotients (structured merges of elements), and inverse nonerasing homomorphisms. Interested readers are referred to Almeida (1995) for more information on these operations in addition to the varieties discussed in this article and others. Pin (1984) discusses the relationship between varieties of semigroups and varieties of formal languages, which can be extended to string-to-string functions (Lambert, 2022). As a consequence of Eilenberg's variety theorem (Eilenberg, 1976), many important classes of formal languages and string-to-string functions are characterized by properties of their syntactic semigroup (Pin, 1984; Lambert, 2022). As an example, the class of ISL functions corresponds exactly to the variety of definite semigroups, defined below (Lambert and Heinz, 2023).

3.3 Green's relations

Many important varieties, including the definite variety, can be expressed in terms of binary relations defined by Green (1951). Given a semigroup S, Colcombet (2011) gives the following preorders.[4]

- $a \leq_{\mathcal{L}} b$ iff $a \in Sb \cup \{b\}$.
- $a \leq_{\mathcal{R}} b$ iff $a \in bS \cup \{b\}$.
- $a \leq_{\mathcal{J}} b$ iff $a \in SbS \cup Sb \cup bS \cup \{b\}$.

Then a is "$\mathcal{L}$-related" to b (denoted $a \mathrel{\mathcal{L}} b$) if and only if $a \leq_{\mathcal{L}} b$ and $b \leq_{\mathcal{L}} a$. If S contains no pair of distinct elements that are "$\mathcal{L}$-related" it is said to be **$\mathcal{L}$-trivial**. The relations $\mathcal{R}$ and $\mathcal{J}$, and the properties $\mathcal{R}$-trivial and $\mathcal{J}$-trivial, are defined similarly.

A semigroup belongs to the variety $\mathbf{D}$ of **definite** semigroups if and only if it is $\mathcal{L}$-trivial and the only idempotent elements lie in the minimal $\mathcal{J}$-class. Similarly, a semigroup belongs to the variety $\mathbf{K}$ of **reverse definite** semigroups if and only if it is $\mathcal{R}$-trivial and the only idempotent elements lie in the minimal $\mathcal{J}$-class (Almeida, 1995).

Recalling that neutral letters give rise to the identity action $\mathbb{1}$, Lambert (2023) defines a semigroup S to be **tier-based** definite (reverse definite) if and only if the elements of S *other than* $\mathbb{1}$ satisfy the conditions for definiteness and reverse definiteness. The tier-based definite and reverse definite classes are denoted $[\![\mathbf{D}]\!]_T$ and $[\![\mathbf{K}]\!]_T$, indicating the interpretation of the variety on some tier T.

A semigroup's multiplication table reveals which elements of the semigroup stand in which of Green's relations. Two elements are $\mathcal{R}$-related if, in the multiplication table of their semigroup, their rows contain the same set of elements, including the labels (the elements themselves). Figure 3 (left) for PNV shows x and y are $\mathcal{R}$-related, as each labels a row consisting of the set $\{x, y\}$.

Two elements in a semigroup are $\mathcal{L}$-related if the columns of the multiplication table contain the

[4] Note $Sb = \{xb : x \in S\}$ and similarly for bS and SbS.

same set of elements, including the labels. In Figure 3 (left), the column of x is $\{x\}$ while that of y is $\{y\}$, so no two distinct elements are $\mathcal{L}$-related.

Finally, the $\mathcal{J}$-order is defined such that $x \leq_{\mathcal{J}} y$ if and only if the union of the columns specified in the row of x is a subset of the union of the columns specified in the row of y. In Figure 3 (left), the row for x is $\{x, y\}$, and the union of those columns is $\{x\} \cup \{y\} = \{x, y\}$. The same holds for y, so $x \leq_{\mathcal{J}} y$ and $y \leq_{\mathcal{J}} x$. Thus $x \mathrel{\mathcal{J}} y$.

Synthesizing, no two elements in the syntactic semigroup for PNV are $\mathcal{L}$-related (it is $\mathcal{L}$-trivial). Also, x and y are idempotents and they are $\mathcal{J}$ related and thus belong to the same $\mathcal{J}$-class. This $\mathcal{J}$-class is minimal since it is the only one. Therefore, this semigroup satisfies the definition of a definite semigroup and belongs to $\mathbf{D}$. It does not belong to $\mathbf{K}$ because two of its elements are $\mathcal{R}$-related.

This algebraic analysis confirms the earlier ISL analysis. Moreover, it is a band. One consequence is that the degree of definiteness (the suffix length under consideration) is 1 (Lambert, 2022) and therefore the k-value for which it is ISL is 2.

3.4 Eggbox Diagrams

Another useful representation of a semigroup is given by what Clifford and Preston (1961) call the **eggbox diagram**, whose design is based on Green's relations. The eggbox diagram is constructed as a collection of grids. Within a grid, two elements share a row if and only if they are $\mathcal{R}$-related. They share a column if and only if they are $\mathcal{L}$-related. All elements within a grid are equal with respect to the $\mathcal{J}$-order. Grids are organized into a graph such that an edge exists from one to another if and only if the target is lower with respect to the $\mathcal{J}$-order than the source. There can be no cycles, so in depictions the source shall always be the higher grid. Finally, idempotent elements have their cells shaded. The eggbox diagram of the syntactic semigroup for PNV is shown in Figure 3 (right). The eggbox diagram makes clear that this semigroup belongs to the definite variety $\mathbf{D}$ because it shows there is one $\mathcal{J}$-class, so it is minimal, and its elements are idempotent. Furthermore, every column in this grid is of depth one and so no pair of distinct elements are $\mathcal{L}$-related. Eggbox diagrams are used for later analyses.

3.5 Directionality

The transducer in Figure 2 operates from left to right. A transducer operating right-to-left which computes the same function does not necessarily have the same structure. Figure 4 depicts a right-to-left machine for the same post-nasal voicing process. It is interesting to observe that this transducer has the same structure as would arise from a left-to-right transducer, in minimal onward form, computing *prenasal* voicing. When a voiceless obstruent is encountered, output is delayed until the following symbol, when it is known whether the output should be voiced or voiceless. The right-to-left computation remains ISL, as the set $\{\$, \mathrm{D}, \mathrm{N}, \mathrm{V}\}$ of suffixes lead to state 1, while the set $\{\mathrm{T}\}$ of suffixes lead to state 2. But it is does not have an OSL structure, as 'T' moves from state 1 to state 2 without outputting anything; the k-suffix of the output is unchanged while the state changes. This demonstrates the well-known fact that output strict locality is directional (Chandlee et al., 2015).

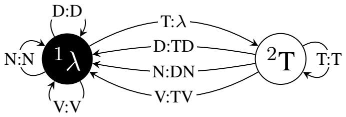

Figure 4: Right-to-left version of post-nasal voicing.

The basis of its syntactic semigroup, however, is the same as before: $x = \langle 1, 1 \rangle$ from 'D', 'N' and 'V', and $y = \langle 2, 2 \rangle$ from 'T'. Thus, the syntactic semigroup is also the same. This is coincidental and is not generally guaranteed, as witnessed by analyzing iterative spreading in §5.

4 OSL is not Algebraic

Given that algebraic results provide new tools for classifying and learning and that ISL functions correspond exactly to functions with definite semigroups, it is natural to ask what variety, if any, corresponds to OSL functions.

Theorem 1. *For any finite semigroup S, there is an OSL function whose syntactic semigroup is precisely S.*

Proof. Let S be a finite semigroup generated by a basis $B \subseteq S$. Let Γ be an alphabet containing at least two letters. Further, let n be $\lceil \log_{|\Gamma|} |S| \rceil$. Finally let $f \colon S \to \Gamma^n$ be an injective function assigning to each element of S a unique arbitrary string in Γ^n. At this point we can construct a sequential transducer $\mathcal{A} = \langle S, B, \Gamma, \delta, 1, \lambda, f \rangle$, where $\delta \colon S \times B \to S \times \Gamma^*$ where $\delta(x, y) = \langle xy, f(xy) \rangle$. The output is produced n symbols

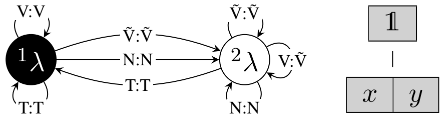

Figure 5: Iterative nasal spreading, with eggbox.

at a time and the state is fixed by the last n output symbols. So this transducer is clearly OSL.

It is also minimal, as every state yields a different output upon termination. Thus the syntactic semigroup of $\mathcal{A}$ is equivalent to its transition semigroup, which by construction is identical to S itself. □

It follows that for any non-OSL sequential function, there exists an OSL function with the same syntactic semigroup. Thus this class of functions does not correspond to a variety of semigroups and is not well-behaved under Eilenberg's theory.

5 Analysis of Output-Oriented Processes

We are thus led to ask which algebraic varieties the phonological processes that motivated the OSL class belong to, if any. This section examines canonical attested processes that have been analyzed in an output-oriented way: iterative spreading processes like nasal spreading, and harmony processes, both symmetric and asymmetric, such as sibilant harmony. The algebraic analyses show these processes to be (tier-based) definite or (tier-based) reverse definite. The results of the analysis are summarized in Table 1 (page 7).

5.1 Iterative Spreading

Post-nasal voicing is an example of noniterative assimilation. Chandlee et al. (2015) examine the process of local iterative nasal spreading in Johore Malay, where contiguous sequences of vowels and glides are nasalized following a nasal. This function is depicted in Figure 5, where 'N' represents a nasal, 'T' any other consonant, 'Ṽ' a nasalized vowel or glide, and 'V' any other vowel or glide. Here, there are three distinct actions that arise from the letters: $\mathbb{1} = \langle 1, 2\rangle$ from 'V', $x = \langle 1, 1\rangle$ from 'T', and $y = \langle 2, 2\rangle$ from 'N' and 'Ṽ'. The letter 'V' is neutral because it does not change state, and so it corresponds to the identity action $\mathbb{1}$ (Lambert, 2023). The eggbox diagram revealing Green's relations is shown in Figure 5.

This process is not definite, as there is an idempotent ($\mathbb{1}$) outside of the minimal $\mathcal{J}$-class. However, it is still $\mathcal{L}$-trivial: no two distinct elements are $\mathcal{L}$-related. It is still a band as well. However, if not for the neutral element it would be the *same* definite semigroup as the one witnessed for PNV. Algebraically, the semigroup satisfies the definition of tier-based definite. It belongs to $[\![\mathbf{D}]\!]_T$.

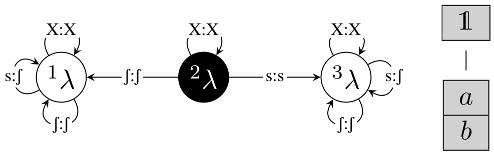

Figure 6: Symmetric harmony, with eggbox.

The tier-based behavior can also be understood from the transducer. The state is determined by the most recent input symbol after projection to the tier $T = \{\mathrm{T}, \mathrm{N}, \tilde{\mathrm{V}}\}$, as the suffixes $\{\$, \mathrm{T}\}$ lead to state 1 while $\{\mathrm{N}, \tilde{\mathrm{V}}\}$ lead to state 2. As mentioned, the only letter off the tier is 'V.'

The right-to-left version of this process is not sequential, as a stream of 'V' must be buffered indefinitely to determine whether they must become 'Ṽ' or stay 'V', and so it shall not be analyzed.

This section has shown how processes of iterative spreading can be understood as a local process operating on a tier.

5.2 Symmetric Harmony

Heinz (2010) describes the symmetric harmony pattern of Navajo, where the existence of a $[-\text{ anterior}]$ sibilant such as 'ʃ' triggers all prior $[+\text{ anterior}]$ sibilants such as 's' to assimilate and become $[-\text{ anterior}]$, and vice versa. The left-to-right version of this process is not sequential, as all sibilants must be buffered until the string ends to know which type surfaces. We therefore analyze only the right-to-left version, depicted in Figure 6, where 's' represents a $[+\text{ anterior}]$ sibilant, 'ʃ' a $[-\text{ anterior}]$ sibilant, and 'X' any other segment.

There are three actions induced by the letters: $\mathbb{1} = \langle 1, 2, 3\rangle$ from 'X', $a = \langle 1, 3, 3\rangle$ from 's', and $b = \langle 1, 1, 3\rangle$ from 'ʃ'. Composition yields no new elements, and $\mathbb{1}$ is neutral.

The eggbox diagram is also shown in Figure 6. The semigroup is does not belong to $\mathbf{D}$ because two elements are $\mathcal{L}$-related. On the other hand, no elements are $\mathcal{R}$-related, suggesting it may belong to $\mathbf{K}$. However, there is an idempotent outside the minimal $\mathcal{J}$-class and so it does not belong to $\mathbf{K}$. But it does satisfy Lambert's (2023) definition of tier-based reverse definite. It belongs to $[\![\mathbf{K}]\!]_T$.

Interestingly, symmetric harmony is the **dual**

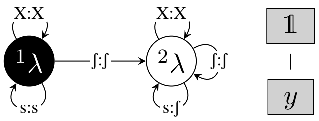

Figure 7: Asymmetric harmony, with eggbox.

of the structure for iterative spreading. For iterative spreading, the behavior is fixed by the k most recent symbols seen on the tier. For symmetric harmony, it is fixed instead by the *first* k symbols seen on the tier.

5.3 Asymmetric Harmony

Heinz (2010) also describes harmony in Sarcee, where only the [− anterior] sibilants are active. Again, this is not sequential when processing left-to-right, so we analyze only the right-to-left version. The minimal transducer is shown in Figure 7.

Two actions are induced by the letters: the neutral element $\mathbb{1} = \langle 1, 2\rangle$ from 'X' and 's', and $y = \langle 2, 2\rangle$ from 'ʃ'. Composition yields no new elements. The eggbox is shown in Figure 7. As with iterative spreading, this is tier-based definite, and the degree of definiteness is one because it forms a band. The behavior is fixed by the most recently seen symbol on the tier. However, not only is it in $[\![\mathbf{D}]\!]_T$ like iterative spreading, it is also in $[\![\mathbf{K}]\!]_T$ like symmetric harmony because it is $\mathcal{R}$-trivial.

5.4 Discussion

The aforementioned analyses establish that canonical examples of output-oriented phonological processes belong to one or more of $\mathbf{D}$ (definite), $\mathbf{K}$ (reverse definite), $[\![\mathbf{D}]\!]_T$ (the tier-based extension of definite), and $[\![\mathbf{K}]\!]_T$ (the tier-based extension of reverse definite), when examining their left-to-right or right-to-left sequential transducers in minimal onward form. These results are summarized in Table 1. The class $\mathbf{N}$ of semigroups are those which belong both $\mathbf{D}$ and $\mathbf{K}$, and it is also a variety (Almeida, 1995).

These results are striking because the algebraic analysis groups local, iterative spreading together with non-local iterative spreading since they each invoke neutral elements (i.e. involve projections onto tiers). Our analyses here show that these canonical *output*-oriented processes are in some sense local, after projection to some tier, on the *input* side as well.

It is also of interest to consider the smallest algebraic variety which includes these classes. The smallest variety containing both $\mathbf{D}$ and $\mathbf{K}$ is $\mathbf{LI}$, called "locally trivial" (Almeida, 1995) and sometimes "generalized definite" (Ginzburg, 1966; Brzozowski and Fich, 1984). Similarly, the tier-based extension of $\mathbf{LI}$, denoted $[\![\mathbf{LI}]\!]_T$ contains $[\![\mathbf{D}]\!]_T$ and $[\![\mathbf{K}]\!]_T$. Interestingly, none of these tier-based extensions are varieties because it can be shown they are not closed under products (Lambert, 2022). We are thus motivated to identify the smallest variety which contains $[\![\mathbf{LI}]\!]_T$.

Pattern	→	←
Post-Nasal Voicing	$\mathbf{D}$	$\mathbf{D}$
Prog. Iterative Spreading	$[\![\mathbf{D}]\!]_T$	–
Reg. Symmetric Harmony	–	$[\![\mathbf{K}]\!]_T$
Reg. Asymmetric Harmony	–	$[\![\mathbf{N}]\!]_T$
Pre-Nasal Voicing	$\mathbf{D}$	$\mathbf{D}$
Reg. Iterative Spreading	–	$[\![\mathbf{D}]\!]_T$
Prog. Symmetric Harmony	$[\![\mathbf{K}]\!]_T$	–
Prog. Asymmetric Harmony	$[\![\mathbf{N}]\!]_T$	–

Table 1: Algebraic classification for left-to-right (→) and right-to-left (←) processing.

Closing $[\![\mathbf{LI}]\!]_T$ under products and quotients has one advantage: by definition, it necessarily includes processes that occur over *multiple* tiers. Recall that the tier T in the above classes is singular; consequently, the total phonology of those languages lies outside any one such class. For this reason, we call this closure $\mathbf{MLI}$, which can be read as "locally trivial over multiple tiers." Almeida (1995) independently studied this class and others like it, considering $\mathbf{M}$ as a natural operator linking varieties of semigroups with varieties of monoids.[5]

To sum up, the smallest algebraic variety which includes all the canonical phonological processes we have considered in this paper, as well as combinations thereof, is $\mathbf{MLI}$. Figure 8 shows the classes discussed in this paper, along with their containment relationships.

6 Inference

We examine the processes and their input-oriented analyses discussed above within the tradition of grammatical inference (de la Higuera, 2010; Heinz et al., 2015; Heinz and Sempere, 2016; Wieczorek, 2017). Specifically, we are interested in whether there is an algorithm which identifies those processes in the limit from positive data (Gold, 1967)

[5] A monoid is a semigroup with an identity.

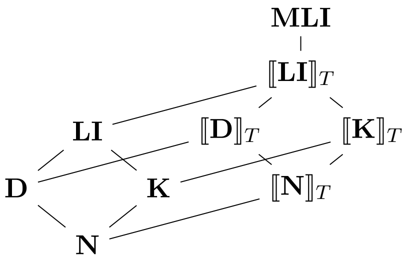
$\mathbf{MLI}$
$\llbracket \mathbf{LI} \rrbracket_T$
$\mathbf{LI}$
$\llbracket \mathbf{D} \rrbracket_T$
$\llbracket \mathbf{K} \rrbracket_T$
$\mathbf{D}$
$\mathbf{K}$
$\llbracket \mathbf{N} \rrbracket_T$
$\mathbf{N}$

Figure 8: Containment among varieties and extensions.

in linear time and data (de la Higuera, 1997). Of course, they are learnable given their output-oriented structure (Chandlee et al., 2015), but relying only on the structure of the input can simplify the system.

Jardine et al. (2014) present SOSFIA, an algorithm which identifies in linear time and data any class of sequential functions representable with a single deterministic transducer. For any function in $\mathbf{MLI}$, there is some k such that its behavior is fixed by the combinations of the first k symbols encountered and the most recent k symbols encountered, across all possible tiers. Consequently, we construct a family of learners, one for each value k. Each learning algorithm constructs a deterministic transducer for k-$\mathbf{MLI}$, and then uses SOSFIA to determine the output of its edges.

Given k and a fixed input alphabet Σ, the construction begins by fixing the state space. The states are in one-to-one correspondence with contexts, where a context is the k first symbols ("prefixes") and k most-recent symbols ("suffixes") for each tier (i.e. all subsets of Σ). In cases where fewer than k salient symbols have been encountered, these shorter strings constitute the context. A prefix of length k is **saturated**. Starting with the initial state corresponding to the empty string on all tiers, expand the state space iteratively as follows until no new edges are created. For each newly created state q, consider the effect of appending a single letter $a \in \Sigma$: Saturated prefixes remain unchanged, as are unsaturated prefixes on tiers that exclude a, but unsaturated prefixes on tiers that include a are extended by appending a. Similarly, suffixes on tiers that exclude a remain unchanged, but suffixes on tiers that include a are extended by appending a and, if now longer than k, contracted by removing their initial symbol. The result is a state r. If r is a new state, then it is added to the state space. In any case, an edge is created from q to r whose input is a and whose output is $\square$, representing a blank. Eventually, no new states will be created, and after the next iteration, no new edges will be created.

The state space is not small. There are $2^{|\Sigma|}$ possible tiers and more than $|\Sigma|^{2k}$ possible prefix–suffix pairs. Nonetheless, once the state space has been filled out, what remains is to assign outputs to the edges in a way that agrees with the observed data. This is precisely the problem SOSFIA solves (Jardine et al., 2014). Given a finite set of input–output pairs and an output-empty deterministic transducer as constructed above, this algorithm fills the outputs in such a way as to maintain onwardness.

If the sample contains sufficient information, which eventually it will in the identification in the limit paradigm, then all outputs will be filled. SOSFIA's time and data complexities are linear, but the constant is large due to the enormous state space.

7 Conclusion

We examined Output (Tier-based) Strictly Local maps in concept and in practice. It was shown that no algebraic property can determine whether a process belongs to these classes (§4). We also provided algebraic analyses for a sample of linguistically relevant O(T)SL processes (§5). Of the processes considered, all lay in $\llbracket \mathbf{D} \rrbracket_T$ or $\llbracket \mathbf{K} \rrbracket_T$, with behaviors fixed either by the k most recent symbols or the first k symbols encountered, for some fixed k, after projection to some fixed tier T. Interestingly, all of the output-oriented maps we discussed were also bands, with all elements idempotent.

These algebraic analyses reveal the unfolding behaviors of these output-oriented functions in terms of their inputs. In particular, iterative spreading was shown to be a local process on tier, and only different from symmetric and asymmetric harmony with regards to whether the first or most recent symbols on the tier trigger harmony. These analyses recall the application of Rule 2 (Figure 1) and Walker's (2014, p. 503) argument that "even in unbounded systems where harmony proceeds among adjacent vowels, the trigger-target relations may be nonlocal, with a single trigger related to many targets, both adjacent and nonadjacent." One area for future linguistic research is a more extensive algebraic cataloging of local and long-distance phonological processes, with particular attention to any that lie outside of $\mathbf{MLI}$ (Jardine, 2016).

The third contribution was an instantiation of the SOSFIA inference algorithm (Jardine et al., 2014) in order to learn processes of the variety $\mathbf{MLI}$ in the limit from positive samples (§6). While this is more powerful than necessary to capture the processes described in this work, it serves to demonstrate the learnability of the processes in question, even without relying on their Output (Tier-based) Strict Locality. Future research can examine imposing further restrictions to improve the space efficiency of the learning algorithm. Another important area of future research is to conduct a detailed comparison between this approach and others, such as the one in (Burness and McMullin, 2019) for 2-OTSL, and one for regular functions more generally (de la Higuera, 2010).

References

Alëna Aksënova and Sanket Deshmukh. 2018. Formal restrictions on multiple tiers. In *Proceedings of the Society for Computation in Linguistics*, volume 1, pages 64–73, Salt Lake City, Utah.

Jorge Almeida. 1995. *Finite Semigroups and Universal Algebra*, volume 3 of *Series in Algebra*. World Scientific, Singapore.

Janusz Antoni Brzozowski and Faith Ellen Fich. 1984. On generalized locally testable languages. *Discrete Mathematics*, 50:153–169.

Phillip Burness and Kevin McMullin. 2019. Efficient learning of output tier-based strictly 2-local functions. In *Proceedings of the 16th Meeting on the Mathematics of Language*, pages 78–90, Toronto, Canada. Association for Computational Linguistics.

Phillip Burness, Kevin McMullin, and Jane Chandlee. 2021. Long-distance phonological processes as tier-based strictly local functions. *Glossa: a journal of general linguistics*, 6(1):1–37.

Jane Chandlee, Rémi Eyraud, and Jeffrey Heinz. 2014. Learning strictly local subsequential functions. *Transactions of the Association for Computational Linguistics*, 2:491–503.

Jane Chandlee, Rémi Eyraud, and Jeffrey Heinz. 2015. Output strictly local functions. In *Proceedings of the 14th Meeting on the Mathematics of Language*, pages 112–125, Chicago, USA. Association for Computational Linguistics.

Jane Chandlee and Jeffrey Heinz. 2018. Strict locality and phonological maps. *Linguistic Inquiry*, 49(1):23–60.

Christian Choffrut. 2003. Minimizing subsequential transducers: A survey. *Theoretical Computer Science*, 292(1):131–143.

Alfred Hoblitzelle Clifford and Gordon Bamford Preston. 1961. *The Algebraic Theory of Semigroups*, volume 7 of *Mathematical Surveys and Monographs*. American Mathematical Society, Providence, Rhode Island.

Thomas Colcombet. 2011. Green's relations and their use in automata theory. In *Language and Automata Theory and Applications: Proceedings of the 5th International Conference, LATA 2011*, volume 6638 of *Theoretical Computer Science and General Issues*, pages 1–21, Heidelberg. Springer-Verlag.

Colin de la Higuera. 1997. Characteristic sets for polynomial grammatical inference. *Machine Learning*, 27(2):125–138.

Colin de la Higuera. 2010. *Grammatical Inference: Learning Automata and Grammars*. Cambridge University Press.

Samuel Eilenberg. 1976. *Automata, Languages, and Machines*, volume B. Academic Press, New York, New York.

Emmanuel Filiot, Olivier Gauwin, and Nathan Lhote. 2016. First-order definability of rational transductions: An algebraic approach. In *LICS '16: Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 387–396. Association for Computing Machinery.

Emmanuel Filiot, Olivier Gauwin, and Nathan Lhote. 2019. Logical and algebraic characterizations of rational transductions. *Logical Methods in Computer Science*, 15(4):16:1–16:42.

Abraham Ginzburg. 1966. About some properties of definite, reverse-definite and related automata. *IEEE Transactions on Electronic Computers*, EC-15(5):806–810.

Edward Mark Gold. 1967. Language identification in the limit. *Information and Control*, 10(5):447–474.

James Alexander Green. 1951. On the structure of semigroups. *Annals of Mathematics*, 54(1):163–172.

Gunnar Hansson. 2010. *Consonant Harmony: Long-Distance Interaction in Phonology*. Number 145 in University of California Publications in Linguistics. University of California Press, Berkeley, CA. Available on-line (free) at eScholarship.org.

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

Jeffrey Heinz, Colin de la Higuera, and Menno van Zaanen. 2015. *Grammatical Inference for Computational Linguistics*. Synthesis Lectures on Human Language Technologies. Morgan and Claypool.

Jeffrey Heinz and Regine Lai. 2013. Vowel harmony and subsequentiality. In *Proceedings of the 13th Meeting on the Mathematics of Language*, pages 52–63, Sofia, Bulgaria. Association for Computational Linguistics.

Jeffrey Heinz, Chetan Rawal, and Herbert G. Tanner. 2011. Tier-based strictly local constraints for phonology. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Short Papers*, volume 2, pages 58–64, Portland, Oregon. Association for Computational Linguistics.

Jeffrey Heinz and José Sempere, editors. 2016. *Topics in Grammatical Inference*. Springer-Verlag, Berlin Heidelberg.

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

Adam Jardine, Jane Chandlee, Rémi Eyraud, and Jeffrey Heinz. 2014. Very efficient learning of structured classes of subsequential functions from positive data. In *Proceedings of the Twelfth International Conference on Grammatical Inference*, volume 34 of *JMLR: Workshop and Conference Proceedings*, pages 94–108.

Nancy C. Kula and Nasir A. Syed. 2020. Non-myopic nasal spreading in Saraiki. *Radical: A Journal of Phonology*, 1:126–182.

Dakotah Lambert. 2022. *Unifying Classification Schemes for Languages and Processes with Attention to Locality and Relativizations Thereof*. Ph.D. thesis, Stony Brook University.

Dakotah Lambert. 2023. Relativized adjacency. *Journal of Logic, Language and Information*, 32(4):707–731.

Dakotah Lambert and Jeffrey Heinz. 2023. An algebraic characterization of total input strictly local functions. In *Proceedings of the Society for Computation in Linguistics*, volume 6, pages 25–34, Amherst, Massachusetts.

Andrew Nevins. 2010. *Locality in Vowel Harmony*. Linguistic Inquiry Monographs. MIT Press, Cambridge, Massachusetts.

Jean-Éric Pin. 1984. *Variétés de Langages Formels*. Masson, Paris.

George Neal Raney. 1958. Sequential functions. *Journal of the ACM*, 5(2):177–180.

James Rogers, Jeffrey Heinz, Margaret Fero, Jeremy Hurst, Dakotah Lambert, and Sean Wibel. 2012. Cognitive and sub-regular complexity. In Glyn Morrill and Mark-Jan Nederhof, editors, *Formal Grammar 2012*, volume 8036 of *Lecture Notes in Computer Science*, pages 90–108. Springer-Verlag.

Sharon Rose and Rachel Walker. 2004. A typology of consonant agreement as correspondence. *Language*, 80(3):475–531.

Marcel-Paul Schützenberger. 1977. Sur une variante des fonctions sequentielles. *Theoretical Computer Science*, 4(1):47–57.

Harry van der Hulst. 2018. *Asymmetries in Vowel Harmony*. Oxford University Press.

Rachel Walker. 1998. *Nasalization, Neutral Segments, and Opacity Effects*. Ph.D. thesis, University of California, Santa Cruz.

Rachel Walker. 2011. *Vowel Patterns in Language*. Cambridge University Press, Cambridge.

Rachel Walker. 2014. Nonlocal trigger-target relations. *Linguistic Inquiry*, 45(3):501–523.

Wojciech Wieczorek. 2017. *Grammatical Inference: Algorithms, Routines and Applications*. Springer.