Phonology (2026), **Vol. 43**:e14 1–16
doi:10.1017/S0952675726100396

PHONOLOGY

ARTICLE

Interpreting state, with focus on metaphonic processes

Dakotah Lambert [1,2] and Jeffrey Heinz [3,4]

[1]Department of Computer Science, Haverford College, Haverford, USA
[2]Department of Mathematics and Computer Science, Lake Forest College, USA
[3]Department of Linguistics, Stony Brook University, USA
[4]Institute for Advanced Computational Science, Stony Brook University, USA

Corresponding author: Dakotah Lambert; Email: dakotahlambert@acm.org

Received: 16 May 2024 UTC; **Revised:** 12 July 2025 UTC; **Accepted:** 30 July 2025 UTC

Keywords: computational phonology; metaphony; harmony; algebra; representation

Abstract

Metaphony provides a unique window into the nature of the capacities of phonological grammars. The representational issues metaphonic processes highlight are exactly those core to computational analysis. Prominent examples of such processes, under various systems of phonological and morphological representation, are examined using algebraic methods, which describe the structure that arises when certain sequences exhibit the same behaviours. This analysis clarifies how representational choices interact with computational complexity. In particular, the simplest characterisations are obtained when the patterns are analysed at the syllabic level, as opposed to the segmental level, and when distinct morphemes are not demarcated by morpheme boundaries but by morphologically marked phonological material. Furthermore, these analyses speak directly to issues in learning, acquisition, production and perception.

Contents

1. Introduction 1
2. Algebraic representations of processes 4
3. Properties of semigroups 7
4. Analysis of metaphonic processes 9
   4.1. Ascrea harmony . . . 9
   4.2. Central Veneto harmony . . . 10
   4.3. Lena harmony . . . 11
5. Discussion 12
6. Conclusion 14
References 14

1. Introduction

Metaphony is an important class of phonological transformations because these transformations often create long-distance interactions involving segmental, prosodic and morphological representations (Rose & Walker 2011). In this way, they provide a unique window into the nature of the capacities of phonological grammars.

The computational study of phonological patterning aims to identify (computational) classes of patterns that are sufficiently expressive to approximate the diversity of attested phonological patterns,

© The Author(s), 2026. Published by Cambridge University Press. This is an Open Access article, distributed under the terms of the Creative Commons Attribution licence (https://creativecommons.org/licenses/by/4.0), which permits unrestricted re-use, distribution and reproduction, provided the original article is properly cited.

https://doi.org/10.1017/S0952675726100396 Published online by Cambridge University Press

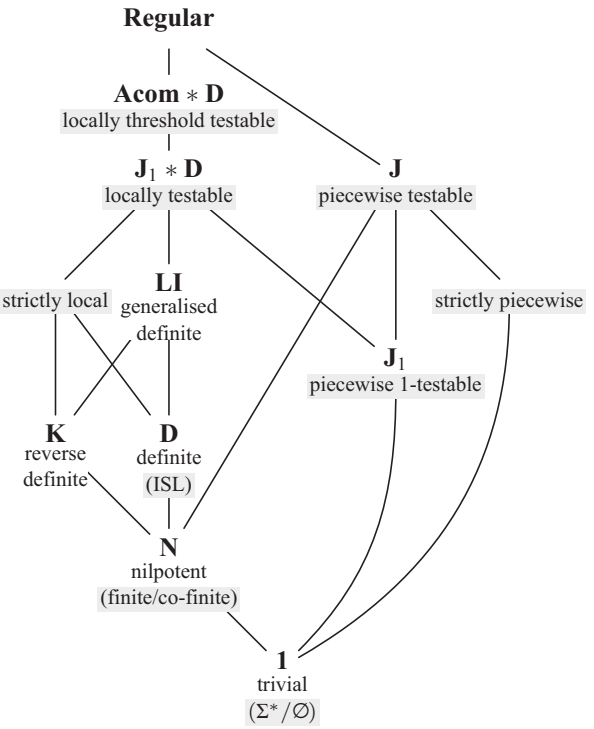


Figure 1. *A subregular hierarchy: higher classes contain connected lower classes.*

but also sufficiently restrictive to account for their learnability from examples. Throughout this article, we use the word 'pattern' to refer generically both to constraints on well-formedness and to processes.

Heinz (2011, 2018) further argues that these computational classes speak to abstract capacities of phonological grammars, irrespective of whether those capacities are instantiated with rules, constraints or something else. These capacities are often expressed in terms of particular properties of finite-state machines (Johnson 1972; Kaplan & Kay 1994; Heinz 2009; Chandlee & Heinz 2018), logical languages (Potts & Pullum 2002; Jardine 2017; Oakden 2020; Chandlee & Jardine 2021) or, more recently, algebraic structures (Lambert & Heinz 2023, 2024; Lambert 2026). Any cognitive mechanism which is capable of realising a particular phonological pattern must be structured, and make distinctions, according to these basic capacities and properties (Rogers & Pullum 2011; Rogers *et al.* 2013). Identifying these capacities thus informs questions of linguistic interest, including questions of learning, acquisition, production and perception.

To illustrate these ideas, consider Figure 1, which presents a subregular hierarchy of classes, including several studied previously in phonology (e.g., Heinz 2018), whose names are highlighted in grey. Classes corresponding to algebraic varieties include a codename from the algebraic literature in boldface.[1] The classes in Figure 1 are not exhaustive. Notably, each of the classes above admits a tier-based superclass (Lambert 2023).

These classes are defined in §3. For now, it suffices to realise that each class indicates the kind of information every pattern in the class is sensitive to. For example, a definite pattern is only sensitive to the most recent k-many symbols perceived. A constraint like *[−sonorant, +voice]# is definite because determining a constraint violation only depends on the last symbol in the word. ISL functions (Chandlee & Heinz 2018) are also definite, because the output string is determined by the most recently read

[1]The strictly local and strictly piecewise classes are not closed under complementation, so they cannot have algebraic characterisations based on equations like the others in Figure 1 (Almeida 1995).

https://doi.org/10.1017/S0952675726100396 Published online by Cambridge University Press

Table 1. *Preview of results.*

Language	Structure	Class	Distinctions
Ascrea	Subsequences	$\mathbf{J}_1$	Set of symbols
	Substrings	$\mathbf{J}_1$	Set of symbols
	Tiers	tier-**1**	None (on the tier)
Veneto (syllables)	Any	**N**	Finite set of words
Lena (boundary)	Subsequences	**Reg**	MSO logical formulae
	Substrings	**Reg**	MSO logical formulae
	Tiers	tier-$\mathbf{J}_1 * \mathbf{D}$	Set of length-2 tier-substrings
Lena (marked suffix)	Subsequences	$\mathbf{J}_1$	Set of symbols
	Substrings	$\mathbf{J}_1$	Set of symbols
	Tiers	tier-**1**	None (on the tier)

k-sized block of symbols in the input (Lambert & Heinz 2023). Importantly, both *k*-definite constraints and functions can be learned from positive examples. In fact, every class in Figure 1, except **Acom** $*$ **D** and Regular, can be parameterised by a positive integer *k,* and each of those *k*-classes can be learned.[2] However, it is the case as one moves higher up in this hierarchy, learning uses more time and resources in the worst case (Lambert *et al.* 2021).

This article examines metaphony patterns in light of subregularity. In particular, it uses *algebraic* methods for several reasons (Lambert 2022). First, as shown in Figure 1, the classes of patterns obtained from algebraic methods include many of the subregular classes previously identified as relevant to phonological patterning. Second, algebraic methods provide a much richer landscape of classes, allowing both coarser and finer granularity when examining phonological patterns. Third, algebraic methods apply equally well to both constraints and transformations, allowing a unified approach to these different aspects of phonological knowledge. Fourth, algebraic methods provide concrete procedures for deciding whether a given pattern belongs to a particular class, and, if it does not, which words witness this. This last point is especially vital, since these procedures can be automated by computer to accurately determine the kinds of patterns which exist in phonology. These automated techniques were used to verify the analyses in this article.

This article examines the metaphony found in Ascrea, Central Veneto Italian and Lena (Walker 2011 and the references therein). These examples are chosen primarily because they are prominent examples of metaphony in the literature. A more complete study would conduct an exhaustive typological analysis of all documented metaphony patterns, and this is something we hope to accomplish in future research. Nonetheless, these three examples are sufficient to make especially clear some trade-offs between representational and computational complexity.

As mentioned, the main outcome of the analysis is that metaphony is computationally simple, but the choice of representation is significant. These results are summarised in preview in Table 1, which shows the language whose metaphony is under analysis, the choice of representation and structure and the algebraic class the pattern belongs to with respect to those choices. The distinctions column indicates the kinds of information any grammar, or learner, must be sensitive to.

Algebraic analysis rarely reveals a single class for any particular pattern, because a pattern may satisfy the defining properties of multiple classes. Furthermore, as mentioned, the analyst's choice regarding how to represent information is also a factor in algebraic classification. For example, one should expect to see differences at different levels of representation (e.g., syllabic or segmental) or with different encodings of information (e.g., as we shall see for Lena, whether a morpheme boundary

[2] **Acom** $*$ **D** is parameterised by two positive integers, a window size *k* and a threshold *t*.

https://doi.org/10.1017/S0952675726100396 Published online by Cambridge University Press

is presented explicitly as a unique symbol or implicitly by using different sets of symbols for affix segments and for root segments). Altogether, this means that there can be more than one way to view the algebraic structure underlying a pattern. No one of these classifications is more or less valid than any other. Rather, taken together, they provide insight into the patterns at hand, and how they relate to other patterns that have been studied. The different classifications achieved under different representations demonstrate the importance of an analyst's choice of representation.

In particular, the metaphonic patterns of Ascrea and Lena, at the segmental level, belong to the algebraic class $\mathbf{J}_1$, which corresponds to the class that Burness & McMullin (2020) call INPUT PIECEWISE 1-TESTABLE. The defining characteristic of this class is that the behaviour of its patterns unfolds according to the *set* of symbols so far encountered. On the other hand, the metaphonic pattern of Central Veneto Italian, at the syllabic level, belongs to the algebraic class **N**, whose defining characteristic is that sequences longer than some fixed finite length must behave in the same way as one another; after this length, no distinctions can be made. The pattern in Central Veneto is simpler than input strictly local (**D**), while the others are simply different. All are simultaneously locally testable and piecewise testable. Consequently, for the learning, production and perception of these metaphonic processes, it is sufficient to attend to simple kinds of information. In all cases analysed, there is some representation where the behaviour is fully determined either by the previous symbol or by the set of symbols encountered so far. This provides a structure that can bias a learner towards the correct result, such as by the system presented by Jardine *et al.* (2014).

However, other representations lead to different results. If metaphony in Lena distinguishes morphemes using morpheme boundaries instead of by indexing the morphemes directly, and if metaphony in Veneto uses segmental representations, then these patterns become more computationally complex.

This article also considers analyses over phonological tiers. Lambert (2022) shows that phonological tiers are naturally expressed in algebraic structures, as discussed in §3. Tier-based analyses of the metaphonic patterns in Ascrea and Lena are shown to simplify the algebraic classification.

Overall, the algebraic analysis of metaphonic processes illustrates that processes that at first appear complex may, in fact, be simple when interpreted under a different level of representation or encoding of the relevant phonological and morphological material. More generally, the algebraic analysis reinforces the importance of metaphony as providing a valuable window into the nature of phonological competence, on account of the unique processes that metaphony entails.

2. Algebraic representations of processes

In order to use algebraic techniques to classify phonological processes, we must first produce algebraic representations of these processes. Specifically, we use semigroups, which can be expressed in a multiplication table. This section is necessarily brief; readers are referred to Lambert & Heinz (2023, 2024) for details. The next section explains how to analyse and exploit the structure present in those multiplication tables. This section explains how to obtain them for generalisations over strings.

A semigroup is a set of objects coupled with an associative multiplication operation, so $x(yz)$ and $(xy)z$ mean the same thing and it makes sense to write xyz. Sometimes, there will be a neutral element (sometimes called an 'identity') that we will denote λ, where $\lambda x = x = x\lambda$ for all objects x.

A mathematical example of such a structure is the multiplication of classes of integers. For instance, we can group integers into 'even' (e) and 'odd' (o) categories. Each class consists of infinitely many numbers, yet still we know how they behave with respect to multiplication. Recall from elementary arithmetic that two odd numbers multiply to produce an odd number ($o \times o = o$), and any product with an even number is even ($e \times o = o \times e = e \times e = e$). Integers can also be grouped by sign: positive (+) for integers greater than zero and negative (−) for integers less than zero, leaving zero itself (0) as a class of one element. Multiplication by a positive preserves the sign; multiplication by a negative inverts the sign and multiplication by zero yields zero. These two semigroups are summarised by their multiplication tables in Figure 2. The semigroups capture different aspects of the structure of multiplying integers.

https://doi.org/10.1017/S0952675726100396 Published online by Cambridge University Press

	o	*e*
o	*o*	*e*
e	*e*	*e*

(a) Even/odd

	+	−	0
+	+	−	0
−	−	+	0
0	0	0	0

(b) Signedness

Figure 2. *Two semigroups describing multiplication of classes of integers.*

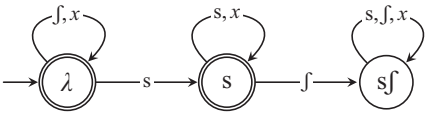


Figure 3. *Minimal DFA for Tsuut'ina sibilant harmony.*

One of the powerful facts about the semigroups above is that the classes are CONSISTENT with respect to multiplication. They would be INCONSISTENT if there were classes A and B such that one can find exemplars a_1 and a_2 from A, and exemplars b_1 and b_2 from B, where $a_1 \times b_1$ is not in the same class as $a_2 \times b_2$. If an operation is not inconsistent, it is consistent. The classes in the even/odd semigroup and the positive/negative/zero semigroup are consistent. For instance, the even/odd semigroup captures one aspect of the structure of multiplication because the product of any element of e multiplied with any integer also belongs to e.

Strings are objects which associatively multiply by concatenation. For example, $b \times a \times ab = baab$. (The symbol × is typically omitted.) Just as was the case for integer multiplication, it is useful to identify groups, or classes, of strings that behave the same way as one another with respect to some pattern.

As an example, consider the regressive sibilant harmony pattern in the Athabaskan language Tsuut'ina, with examples taken from Cook (1978) and Li (1930).

(1)	a.	/sì-tʃʊ́gʊ̀/	ʃìtʃʊ́gʊ̀	'my flank'
	b.	/sì-tsáɣà/	sìtsáɣá	'my hair'
	c.	/gi-si-tˀʃast/	giʃìtˀʃast	'they are piled up'
	d.	/na-dzá-na-dì-s-ɣʲìh/	nadʒánadìʃʃìh	'I'll be refreshed'

As a process, sibilants like [ʃ] cause sibilants like [s] earlier in words to assimilate in anteriority, as in (1a) and (1c). As a constraint, [+anterior] sibilants like [s] cannot precede [−anterior] sibilants like [ʃ] at any distance within words, although the reverse is not true, as evidenced by the last morpheme in (1c).

How can a semigroup be found for these generalisations? We explain the answer to this question as follows. First, we consider the constraint-based generalisation and explain how the states of a finite-state model of a constraint partition strings into equivalence classes. Next, we motivate finite-state models based on Myhill equivalence classes, which are guaranteed to be consistent with respect to concatenation. Finally, we explain how to extend this analysis to processes.

Common representations of phonological constraints use the minimal, deterministic finite-state acceptor (DFA; Heinz 2018). Every string that the DFA accepts obeys the constraint and every string it rejects violates the constraint. For the constraint in Tsuut'ina, a DFA representation is provided in Figure 3, with x representing arbitrary non-sibilant phones.

The states in a DFA partition strings into equivalence classes in terms of the states they lead to. For example, all strings that contain [s] after [ʃ] lead to the rejecting state labelled 'sʃ'. All strings that contain [s], but no [ʃ] after [s], lead to the state labelled 's'. And all other strings lead to the state labelled 'λ'.

https://doi.org/10.1017/S0952675726100396 Published online by Cambridge University Press

This partitioning is called NERODE EQUIVALENCE. Formally, x is considered equivalent to y if for all strings v, either xv and yv are both accepted or they are both rejected. The defining feature of the minimal DFA of a formal language is that there is exactly one state per Nerode-equivalence class (Nerode 1958).

However, Nerode classes of strings do not necessarily make concatenation consistent in the sense defined above. The strings 'ʃɒs' and 'sɒs' are both in the group corresponding to the state labelled 's'. The string 'si' is also in this group. But 'siʃɒs' is in the group corresponding to 'sʃ' while 'sisɒs' is in the group corresponding to 's'. These results contradict one another, so this grouping is inconsistent.

With a different perspective, we can refine the partition further to repair this issue. Each symbol in the alphabet labels edges in the finite-state model in a way that describes the behaviour of that symbol. We can treat them as actions, for example, [s] maps states 'λ' and 's' to state 's' and maps state 'sʃ' to state 'sʃ'. For a longer string, the edges get chained together as a function composition: [ʃs] first maps state 'λ' to state 'λ' via the [ʃ] and then to state 's' via the [s]. States 's' and 'sʃ' both map to state 'sʃ'. As there are only finitely many possible mappings over a finite set, the result is guaranteed to be finite. As function composition is associative, the resulting classes will be consistent with respect to concatenation. And finally, because a word is accepting whenever it maps state 'λ' to an accepting state, grouping strings by their corresponding functions guarantees that no group mixes accepted and rejected words.

This grouping, known as MYHILL EQUIVALENCE, gives the algebraic structure of the language. This semigroup is the coarsest partitioning of strings into classes compatible with the pattern such that the concatenation of those classes is consistent (Rabin & Scott 1959). It can be derived from the minimal DFA as per the preceding paragraph. Figure 4a depicts the DFA obtained using Myhill-classes as states.

This semigroup is also presented as a multiplication table in Figure 4b: the string at row x column y is the representative for the group containing xy. There are five classes of strings, named according to one short representative element. These are the sets of strings that behave the same way. The state labelled 'sʃ' captures all sets that contain the forbidden [+anterior]…[−anterior] subsequence. The state labelled 'ʃs' captures all other sets that contain its reverse, the [−anterior]…[+anterior] subsequence. Then 's' and 'ʃ' capture sets with at least one sibilant, where all sibilants are [+anterior] or [−anterior], respectively. Finally, 'λ' captures sets with no sibilants at all.

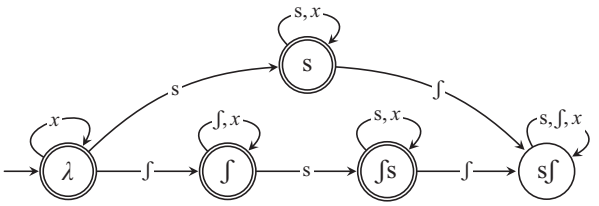
This multiplication table models the behaviour of sets of forms with respect to the constraint-based generalisation of Tsuut'ina. We emphasise that it is not the only model. For instance, an algebraic model may depend on whether the input is read from left to right as above (e.g., reading the underlying /sìtʃʊ́gʊ̀/ from its initial 's' towards its final 'ʊ̀') or from right to left (e.g., reading the underlying /sìtʃʊ́gʊ̀/ from its final 'ʊ̀' towards its initial 's'). In this case, the right-to-left version mirrors the one shown above (swap 's' and 'ʃ' everywhere).

The next section explains how to analyse the structures present in these tables, but first we explain how such tables can be constructed for processes.

Processes can also be modelled with finite-state devices, in this case deterministic finite-state transducers (DFTs). Unlike a DFA, the transitions in a DFT associate an output string to each transition. These outputs determine which input strings may be considered equivalent – that is, which input strings exhibit the same behaviour. Consequently, the algebraic structure of the process-based generalisation for Tsuut'ina is not necessarily identical to the one shown in Figure 4.

A DFT for Tsuut'ina is shown in Figure 5a. This transducer is specified to process strings from right to left. In this figure, and in all subsequent DFTs in this article, if the output differs from the input, this is indicated as 'input:output'; otherwise, only the input is shown. These outputs influence which strings can be considered equivalent. Let p_x represent the longest common prefix of the outputs across all inputs that begin with the string x. Every continuation xv of x will produce some output $p_x y$. The state corresponding to x, then, computes the subfunction mapping v to y. This corresponds to the notion of tails used by Chandlee (2014), Chandlee *et al.* (2014) and Chandlee & Heinz (2018). It results in equivalence classes analogous to Nerode equivalence.

https://doi.org/10.1017/S0952675726100396 Published online by Cambridge University Press

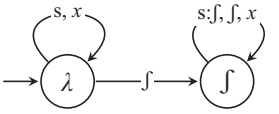


(a) DFA

	λ	s	ʃ	ʃs	sʃ
λ	λ	s	ʃ	ʃs	sʃ
s	s	s	sʃ	sʃ	sʃ
ʃ	ʃ	ʃs	ʃ	ʃs	sʃ
ʃs	ʃs	ʃs	sʃ	sʃ	sʃ
sʃ	sʃ	sʃ	sʃ	sʃ	sʃ

(b) Multiplication table

Figure 4. *Algebraic structure of sibilant harmony in Tsuut'ina.*



(a) DFT

	λ	ʃ
λ	λ	ʃ
ʃ	ʃ	ʃ

(b) Multiplication table

Figure 5. *Sibilant harmony in Tsuut'ina (right-to-left).*

To obtain Myhill equivalence classes, the strings can again be treated as actions on the states. In the case of the Tsuut'ina sibilant harmony process, the Myhill classes are exactly the Nerode classes. The multiplication table is shown in Figure 5b.

We sum up with some observations. Recall that directionality may affect algebraic representations. This is true for processes as well. However, some processes, such as Tsuut'ina sibilant harmony, may only have a representation in one direction (right-to-left shown in Figure 5), and not the other (no finite-state transducer can capture the process when input is read from left to right). Processes representable with left-to-right DFTs and processes representable with right-to-left DFTs (called LEFT-TO-RIGHT SEQUENTIAL and RIGHT-TO-LEFT SEQUENTIAL) are known to be distinct but overlapping classes (Elgot & Mezei 1965). Finally, it is interesting to note that this table is isomorphic to the even/odd example in Figure 5.

In general, given a pattern represented as a DFA or DFT, we can produce an algebraic model of that language or process. The remainder of this work provides finite-state descriptions in Myhill-minimal form, as opposed to the smaller and more common Nerode-minimal form, in order to better visualise the semigroup structure.

We have used the word 'state' frequently in this section and we will continue to use it throughout this work. It should be noted that, while this term comes from the perspective of finite-state models, its importance goes far beyond such systems. As discussed, the states in these models correspond to classes of strings that behave in the same way. That is, we use the term 'state' as a convenient notational proxy encompassing the finite-state model itself, the equivalence classes of strings, and the behaviours they induce.

3. Properties of semigroups

This section briefly describes the small collection of well-studied subregular classes that we looked to when analysing the patterns in §4. We include the piecewise testable, locally testable, locally threshold-testable and tier-based input strictly local classes because they have been used in prior literature to analyse long-distance patterns in phonology (Rogers *et al.* 2013; Lambert & Rogers 2019; Burness & McMullin 2020). Some other related classes are also included. Each class is accompanied by a test to determine if a given subregular pattern is a member of that class. These tests are given in the form of

https://doi.org/10.1017/S0952675726100396 Published online by Cambridge University Press

equations in terms of variables, such as $x = xx$ or $xy = yx$. A pattern is in the class if, and only if, the equations are true no matter which semigroup elements are substituted for which variables.

A pattern is piecewise k-testable if the set of subsequences of length k of each word determines the behaviour, that is, which state (read: Myhill-equivalence class) that word is in (Simon 1975). A pattern is piecewise testable if there is some finite k for which it is piecewise k-testable. Following algebraic tradition, we use **J** to represent the piecewise testable class, with k as a subscript if necessary.

Almeida (1995) provides the equations that characterise piecewise testability: a pattern is in the class if, and only if, for all x and y, it holds that $(xy)^\omega = (yx)^\omega$ and $x^\omega x = x^\omega$. This notation x^ω refers to the unique semigroup element of the form x^i that squares to itself: $x^i x^i = x^i$.[3] The elements that square to themselves, the x^ω, are called IDEMPOTENT.

The language of potential surface forms satisfying the sibilant harmony of Tsuut'ina, depicted in Figure 4, is piecewise 2-testable, as the only relevant information is whether a [+anterior]…[−anterior] subsequence occurs. The process mapping underlying to surface forms, depicted in Figure 5, has a different structure, but is still piecewise testable. The process is simpler than the surface structure, as the relevant factor size is $k = 1$ rather than 2: words which contain 's' are in [s] and all other words are in [λ]. The class $\mathbf{J}_1$ is characterised by the equations $xy = yx$ and $xx = x$; order and multiplicity are irrelevant.

Locally threshold-t k-testable (**Acom** $*$ **D**; $x^\omega a y^\omega b x^\omega c y^\omega = x^\omega c y^\omega b x^\omega a y^\omega$ and $x^\omega x = x^\omega$) patterns determine Myhill-equivalence classes by the multiset of substrings of length k counted up to a threshold t (Beauquier & Pin 1991).

Locally k-testable patterns ($\mathbf{J}_1 * \mathbf{D}$; $z^\omega xz^\omega yz^\omega = z^\omega yz^\omega xz^\omega$ and $z^\omega xz^\omega xz^\omega = z^\omega xz^\omega$) determine Myhill-equivalence classes by the set of substrings of length k. They are the subset of locally threshold testable patterns where the threshold is $t = 1$ (Brzozowski & Simon 1973). The semigroup of the process depicted in Figure 5 is locally testable, as the class [s] is the set of words which contain the 's' substring and [λ] is all other words. On the other hand, the language in Figure 4 is not locally testable, as setting $x = \mathrm{s}$, $y = \int$ and $z = \lambda$ yields $\lambda^\omega \mathrm{s} \lambda^\omega \int \lambda^\omega = \mathrm{s}\int \neq \int \mathrm{s} = \lambda^\omega \int \lambda^\omega \mathrm{s} \lambda^\omega$.

Definite (**D**; $yx^\omega = x^\omega$), reverse-definite (**K**; $x^\omega y = x^\omega$) and generalised definite (**LI**; $x^\omega y x^\omega = x^\omega$) are subclasses of locally testable patterns, where equivalence classes correspond to the k most recent symbols, the k first symbols or both taken as a pair, respectively (Almeida 1995). The total 'input strictly local' transducers of Chandlee *et al.* (2014) are exactly the definite processes (Lambert & Heinz 2023).

Nilpotent patterns (**N**) are those that are both definite and reverse-definite; it turns out that they are also piecewise testable (Almeida 1995). In these patterns, words up to length k may have their own equivalence classes, but all longer words share a single class.

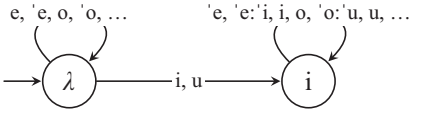
Finally, the class **1**, characterised by $x = y$, contains single-state patterns, called 'trivial' patterns. The classes discussed so far in this section are summarised in Figure 1.

Each of these classes can be extended by accounting for tier-projection. Heinz *et al.* (2011) define the tier-based strictly local languages to account for long-distance patterns using the notion of a phonological tier from Goldsmith (1976), first projecting away some symbols and then applying a strictly local constraint on what remains. Following Lambert (2023), we account for tiers by removing the equivalence class containing the empty string from consideration if this class cannot be reached by concatenation from any other class. This class corresponds to words that cannot result in a change of state. For instance, the language shown in Figure 4, while not locally testable, is tier-based locally testable. Only sibilants can result in state-change, and after projecting input strings to their sibilants, a word is accepted if, and only if, it does not contain the 'sʃ' substring.

These classes describe the structure of the pattern. Each gives a blueprint for constructing a learning algorithm. For example, for learning constraints, one can use the string-extension learning of Heinz (2010b), and for learning processes, one can use SOSFIA of Jardine *et al.* (2014). Some of this structure

[3] Note that the uniqueness refers not to the exponent i, but to the actual element x^i. If $x^m = x^m x^m$, then by repeatedly replacing one copy of x^m by two copies, we can show that $x^m = x^{nm}$ for any positive whole number n. Then if $x^n = x^n x^n$, we can similarly show $x^n = x^{mn}$. Since $mn = nm$, that means $x^m = x^{mn} = x^{nm} = x^n$.

https://doi.org/10.1017/S0952675726100396 Published online by Cambridge University Press



(a) DFT

	λ	i
λ	λ	i
i	i	i

(b) Multiplication table

Figure 6. *Stress-raising in Ascrea (right-to-left).*

will naturally arise from the constraints that shape the language or process, while some of it will be incidental. In the following analyses, we explore the interplay between the structure of representations and the structure of resulting patterns.

4. Analysis of metaphonic processes

In this section, we present analyses of several processes which target a prominent position. Each process under consideration is regressive, so our classifications are based on reading the input from right to left. We begin with two similar examples of raising processes that target stressed syllables: one in the Romance dialect of Ascrea in which intervening vowels are transparent (§4.1), and one in the Romance dialect of Central Veneto in which intervening mid vowels also raise (§4.2). In the next example, the Asturian dialect of Lena, the raising process is unambiguously triggered only by vowels in a morphological suffix (§4.3). We present two analyses that differ in how morphemes are separated. With explicit morpheme boundaries, the phenomenon appears complicated, but with suffix-vowels marked as distinct symbols from stem-vowels, it instead appears simple.

The semigroups for these patterns are available in the form of AT&T-format finite-state representations as Supplementary Material. Our analysis uses the Language Toolkit of Lambert (2024) to automatically classify the semigroups, which avoids the tedious and error-prone process of checking the multiplication tables by hand.

4.1. Ascrea harmony

Walker (2011: 43) provides an example of metaphony in the Romance dialect of Ascrea, where a high suffix vowel [i] or [u] causes a preceding stressed mid vowel to raise. This stressed vowel might occur in the immediately preceding syllable, as in (2a), or it might be in an earlier syllable, as in (2b). In the latter case, intervening unstressed mid vowels are transparent. They do not undergo change. The following examples are excerpted from Walker (2011: 171–172):

(2) a. meˈtesse meˈtiʃʃi ‘reap (1SG/2SG IPFV SBJV)’
b. ˈtoreu̯a ˈtureu̯u *ˈturiu̯u ‘cloudy (F.SG/M.SG)’

We shall assume for now that stress assignment and morphological concatenation have already occurred. Stress will be marked on the input to this process, and the suffix will already be attached. Notice that, as there is only ever one stressed syllable, it is ambiguous whether high stressed vowels are neutral undergoers or triggers that necessarily lack targets. The ‘trigger’ interpretation would have [ˈe] and [ˈo] pattern like [i] and [u] rather than like [e] and [o]; the two possibilities are in the same complexity classes, so, in order to follow the existing description, we treat them as neutral undergoers. The Myhill-minimal right-to-left sequential transducer for this process and the multiplication table of its semigroup are shown in Figure 6. Compare this to the Tsuut’ina sibilant harmony process of Figure 5.

The semigroup is commutative ($xy = yx$ for all x and y) and idempotent ($xx = x$ for all x). This corresponds to the class $\mathbf{J}_1$ and means that the state is uniquely determined by the set of length-one subsequences (or, equivalently, substrings) encountered so far in the input word. Namely, any set that

https://doi.org/10.1017/S0952675726100396 Published online by Cambridge University Press

includes [i] or [u] is in state i,[4] while all others are in state λ. This is the simplest nontrivial variety under precedence analysis, corresponding to piecewise 1-testable. In the terminology of Burness & McMullin (2020), this is an input piecewise 1-testable function. Under adjacency, $\mathbf{J}_1$ corresponds to locally 1-testable, so this would be an input locally 1-testable function. The next simplest class is **LI**, which asserts that $x^\omega y x^\omega = x^\omega$ for each idempotent x^ω and each element y. However, $\lambda i \lambda \neq \lambda$, so locally 1-testable is the simplest analysis under adjacency.

The tier-based analysis is the simplest. If the neutral λ is removed, the semigroup has only one element; this is the class **1**.

4.2. Central Veneto harmony

The Romance dialect of Central Veneto exhibits a similar phenomenon, where a high vowel in a suffix will cause a preceding stressed mid vowel to raise (Walker 2011). However, this pattern differs from that of Ascrea in that intervening unstressed mid vowels also raise, as shown in (3a). Example (3b) depicts a situation in which raising stops at the stressed syllable. The following examples are excerpted from Walker (2011: 116):

(3)	a.	ˈordeno	ˈurdini	‘order (1SG/2SG)’
	b.	boˈton	boˈtuni	‘button (M SG/PL)’

This appears to be a typical asymmetric harmony pattern with blocking. But the metaphony of Central Veneto is not actually so simple. As shown in the following examples taken from Walker (2011: 117), the true pattern more closely resembles the sour-grapes harmony studied by Padgett (1995), Baković (2000), Heinz & Lai (2013) and Smith & O’Hara (2019), in that the intervening vowels raise only if a stressed vowel is successfully targeted. Note also that a low penult blocks raising of the stressed syllable, as shown in (4c).

(4)	a.	ˈpɛrsego	ˈpɛrsegi	‘peach (fruit; M SG/PL)’
	b.	ˈangolo	ˈangoli	‘angle (M SG/PL)’
	c.	laˈ(v)orava	laˈ(v)oravi	‘work (1SG/2SG IPFV IND)’

There is one important factor distinguishing this from sour grapes: the stressed syllable is never before the antepenultimate position. The pattern would be circumambient unbounded in the terminology of Jardine (2016) if not for this mitigating circumstance. Indeed, to maintain sequentiality, we must either further impose that there is some maximum number of consonants, say three or four, that can intervene between vowels, or we must move to a different level of representation. Without this, the intervening consonants must be buffered indefinitely after seeing a potentially harmonising unstressed mid vowel in the penultimate syllable.

Figure 7 treats the input at the syllable level: H represents a syllable with a high vowel; $M_1 \ldots M_n$ represent the possible syllables with mid vowels and L represents a syllable with a low vowel. The acute accent indicates stress. There can only be one stressed syllable, and it cannot appear further to the left than the antepenult, but we include additional transitions for impossible placements to maintain totality. This essentially implements the principle of Prince & Smolensky ([1993] 2004) known as Richness of the Base. The prediction made by this particular state machine is that, if the main stress occurs further left than the antepenult, then harmony would not be triggered. If this does not hold in reality, then the system must be analysed differently, perhaps by using a two-way transducer (see Carton & Dartois 2015; Dolatian & Heinz 2020) to detect and act upon the harmonising context.

Notice that the initial state λ has no in-edges, and recall that this means that it is not part of the semigroup. The other elements are, but aside from HM_i every product xy is L. This is not in $\mathbf{J}_1$ for a

[4]The use of ‘i’ as a state representative is arbitrary; ‘state i’ could as well have been named ‘u’ or ‘io’, or merely given a number like ‘2’.

https://doi.org/10.1017/S0952675726100396 Published online by Cambridge University Press

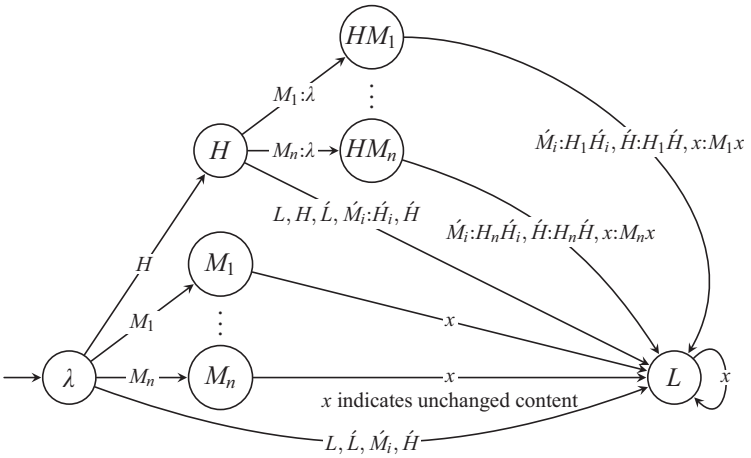


Figure 7. *Stress-raising in Veneto (right-to-left).*

few reasons. First, it is not commutative, as HM_i is distinct from $M_iH = L$. Further, there is only one idempotent element: L. Because this is the only idempotent and because $xL = L = Lx$ for all elements x, this semigroup is in **N**. States correspond to a finite set of strings alongside one sink state.

4.3. Lena harmony

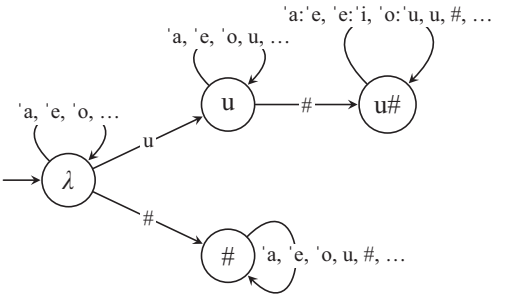
Like Ascrea and Central Veneto, the Asturian dialect of Lena undergoes regressive metaphonic harmony. A suffix containing [u] triggers a preceding stressed vowel to raise: [e] and [o] raise to [i] and [u], while [a] becomes [e], as shown in (5a)–(5c). As in Ascrea, intervening vowels are transparent, as demonstrated in (5d). Moreover, the trigger is necessarily in the morphological suffix; a high vowel in the stem will not trigger the raising, as demonstrated in (5e). For this reason, Finley (2009) argues that this process is not a phonological process at all, but instead a form of morphological harmony. The following examples are excerpted from Walker (2011: 167–168):

(5)	a.	ˈnena	ˈninu	‘child (F.SG / M.SG)’
	b.	ˈgatos	ˈgetu	‘cat (M.PL / M.SG)’
	c.	ˈtsobos	ˈtsubu	‘wolf (M.PL / M.SG)’
	d.	burˈwebanos	burˈwibanu	‘wild strawberry (M.PL / M.SG)’
	e.	aˈbaxu		‘down’

Because vowels in the morphological suffix and vowels in the stem act differently, this difference must be accounted for in the transducer. One may account for this difference by explicitly encoding the morpheme boundary, or by treating suffix vowels as distinct symbols from stem vowels. We present both analyses to show that explicit morpheme boundaries result in a higher complexity.

First, suppose that morpheme boundaries are explicitly encoded as ‘#’. The Myhill-minimal right-to-left sequential transducer is shown in Figure 8 alongside the multiplication table of its semigroup. This is not input piecewise testable at all. Recall that states correspond to sets of subsequences seen in the input when and only when the semigroup satisfies $(xy)^\omega = (yx)^\omega$ and $x^\omega x = x^\omega$. In this case, we find that for x = u and y = #, an equation is not satisfied. We have $(\text{u\#})^\omega \neq (\text{\#u})^\omega$. That is, the two words $(\text{u\#})^k$ and $(\text{\#u})^k$ have the same subsequences of length k, but they lead to different states and produce different behaviours. It is for this reason that this function is not input piecewise k-testable for any k, and therefore it is not input piecewise testable.

https://doi.org/10.1017/S0952675726100396 Published online by Cambridge University Press

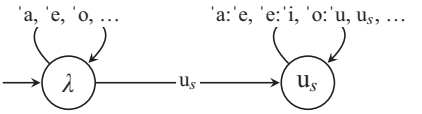


(a) DFT

	λ	u	#	u#
λ	λ	u	#	u#
u	u	u	u#	u#
#	#	#	#	#
u#	u#	u#	u#	u#

(b) Multiplication table

Figure 8. *Stress-raising in Lena, with explicit morpheme boundaries (right-to-left).*



(a) DFT

	λ	u$_s$
λ	λ	u$_s$
u$_s$	u$_s$	u$_s$

(b) Multiplication table

Figure 9. *Stress-raising in Lena, with marked suffix vowels (right-to-left).*

This phenomenon also appears complex under adjacency. Recall that to be locally threshold testable, the semigroup must satisfy $x^\omega ay^\omega bx^\omega cy^\omega = x^\omega cy^\omega bx^\omega ay^\omega$. Fixing $x = y = b = \lambda$, this simplifies to $ac = ca$. Yet we have elements a = u and c = # that do not satisfy this condition: u# ≠ #u. This tells us that ˈo^{k}uˈo^{k}ˈoˈo^k#ˈo^k and ˈo^k#ˈo^{k}ˈoˈo^{k}uˈo^k have the same set of k-factors counting up to any threshold t, but they map to different states and produce different behaviours. Thus, the process is not input locally threshold testable.

If we project away the neutral symbols, we are left with a smaller semigroup containing only u, # and u#. This smaller semigroup satisfies the equations that correspond to being locally testable, and indeed we find that the set of 2-factors on the tier containing both u and # suffices to determine the state. If the projection is empty, then the state is λ. If it begins with #, then the state is #. Otherwise, it begins with u, and whether it contains # at all determines whether the state is u# or u. Thus, this pattern is tier-based locally 2-testable. Interestingly, as asking whether a string contains # is equivalent to asking whether it begins with # after projecting to only #, this is also multiple-tier-based reverse definite, in that the state is fixed by the first symbol encountered on each of the u, # and # tiers.

This concludes the analysis with explicit morpheme boundaries. Next, we suppose instead that vowels in the morphological suffix are marked up as a distinct type of symbol. The transducer for this analysis is shown alongside its multiplication table in Figure 9, where the subscript 's' indicates a vowel in a suffix. Compare this to the metaphony of Ascrea as depicted in Figure 6, and note that the structures are identical. With this perspective, the metaphony in Lena is in $\mathbf{J}_1$, meaning it is input piecewise 1-testable and input locally 1-testable, and it is also in the tier-based extension of **1**. Like Ascrea, the classification of Lena does not change under the syllable-based analysis used for Central Veneto.

5. Discussion

The analyses above carry some interesting implications for morphophonological theories of representation, as well as raising some interesting questions for future research.

https://doi.org/10.1017/S0952675726100396 Published online by Cambridge University Press

What is the 'right' way to represent morphological, phonological and prosodic units? Earlier theories of phonology, such as that of Chomsky & Halle (1968), considered representations which included morpheme boundaries, syllable boundaries and segmental material within a single stream. More recent approaches consider some linguistic representations, such as syllable structure, metrical structures and morphological structures, to be hierarchical (Nespor & Vogel 1986; Selkirk 2011; Inkelas 2014). One principle used to help answer this question is to prefer the representations that lead to simpler analyses.

On the basis of this principle, and given the complexity classes in Figure 1, one would draw three conclusions. First, explicit marking of boundaries is less preferred than marking phonological material in suffixes differently from phonological material in stems (Lena). Second, some processes may be better represented over syllables than segments (Veneto). Third, phonological tiers simplify analyses (Lena, Veneto and Ascrea). We consider these conclusions more carefully in turn.

What precisely does it mean to mark phonological material in affixes differently from phonological material in stems? We do not mean to imply that the right representation is to literally use a different symbol – say, lowercase [t] for stems and uppercase [T] for suffixes – or to use a feature [±affix], though these representations would be compatible with the conclusion drawn. Instead, we emphasise that theories with hierarchical representations highlight one limitation of the algebraic techniques currently employed, which are fundamentally about sequences. In other words, if one adopts hierarchical representations and wants to conduct an algebraic analysis, as we have done here, one must first flatten the hierarchy into a string. The use of different symbols in different constituents is one way to do this. Therefore, we argue that its success supports the view that morphological representations are hierarchical. This conclusion is compatible with multiple theories of morphophonology, including ones based on correspondence in an OT setting (Walker & Feng 2004), and ones based in mathematical logic (Dolatian 2020).

Regarding syllable structure, similar issues persist. One can capture differences between onset and coda consonants in different ways: by referring to their position relative to an explicit syllable boundary symbol present in the same string, with features like [±onset] or with hierarchical structures, where a segmental unit inherits properties of the domain to which it belongs. Strother-Garcia (2019) compared these alternatives and concluded that they are inter-translatable (though see Danis 2025 for a more critical view). In the present analysis, however, this level of detail is irrelevant: all that matters is that syllabic units be the basic level of analysis. This approach, which considers the syllabic level instead of the segmental, converges with others, such as in the analysis of metrical stress (Hayes 1995; van der Hulst 2014; Lambert 2026).

Regarding tier-based representations, the results provide new mathematical arguments for the use of phonological tiers in phonological theory, even in contexts where they may not have been previously considered (such as in the analysis of metaphony). Locality has been an important principle in linguistic theory, and long-distance relationships are often mediated through a relativised kind of adjacency; see Poser (1982), Rose & Walker (2004), Heinz (2010a) and Lambert (2026), among others, for many examples, and Lambert (2023) for mathematical analysis of tier representations. This result is reinforced by recent theoretical and empirical work, which shows that tiers can be learned from data (Jardine & Heinz 2016; Jardine & McMullin 2017; Burness & McMullin 2019; Gouskova & Gallagher 2020; Burness & McMullin 2021; Lambert 2021).

An important area of future research regards the learning and acquisition of metaphony. It is generally the case that smaller, well-structured classes are more easily learnable than larger, less-structured classes (Kearns & Vazirani 1994; Mohri *et al.* 2018; Lambert *et al.* 2021; see Heinz & Riggle 2011 and Valiant 2013 for accessible introductions). This is because the distinctions learners need to attend to are fewer with smaller, well-structured classes. For instance, the k-definite languages can generally be learned from fewer examples than the generalised k-definite languages. In other words, the containment hierarchy of pattern types in Figure 1 also reflects a hierarchy for pattern learning: in general, learning algorithms which target the lower classes can be more efficient in time and data than the higher classes.

https://doi.org/10.1017/S0952675726100396 Published online by Cambridge University Press

As we have seen, since morphological and syllabic representations affect pattern complexity, which affects ease of learnability and acquisition, the question is naturally raised: Which representations do learners use, and is the choice of representation itself part of the learning and acquisition problem? We conclude that this constitutes an interesting direction for future work, one in which metaphony would play an especially important role.

Another question regards algebraic analysis of a particular process in the context of a larger phonological grammar. In the present analysis, it was assumed that stress was already marked in the input. Clearly, when stress is predictable, there is a process placing stress in the appropriate position. In addition to studying the complexity of individual processes, we can also study the complexity of the whole phonological system.

In general, the algebraic complexity of a whole system cannot straightforwardly be determined from the properties of the individual processes within it. In some cases, the complexity can increase. For example, in the context of metaphony, if stress assignment were definite (**D**) and the metaphony itself were $\mathbf{J}_1$, then the complexity of the whole phonology would be at most $\mathbf{J}_1 * \mathbf{D}$. In short, examining the issues raised by these considerations in greater detail is also warranted.

6. Conclusion

We examined three metaphonic processes studied by Walker (2011) and analysed them with respect to a small subregular hierarchy shown in Figure 1. In doing so, we showed that the basic metaphony in Ascrea is computationally equivalent to the asymmetric sibilant harmony in Tsuut'ina. The metaphony in Central Veneto appears more complex, with blockers and effects akin to sour-grapes harmony. However, by analysing a syllabic representation rather than a purely segmental one, and by exploiting the fact that the stress pattern is bounded, this pattern becomes significantly simpler. Finally, in the metaphony of Lena, it is necessary to distinguish the morphological suffix from the stem. There are two simple methods of doing so: either a morpheme boundary symbol may be inserted, or symbols may be marked differently depending on whether they belong to the suffix. Under the first interpretation, the pattern appears complex. But under the other interpretation, it becomes significantly simpler.

In short, the most important factor in computational complexity analysis is the representation over which the computational analysis operates. With appropriate choice of representation, each of the metaphony processes discussed is in the class $\mathbf{J}_1$ or **N**, which means that they are simultaneously piecewise testable and locally testable. Patterns with this property are also called OVER TESTABLE; we conjecture that all metaphony may be over testable. But using other representations, the systems may be significantly more complex.

Acknowledgements. We thank the reviewers and the guest editors for their feedback on an earlier version of this article.

Funding statement. J.H.'s research was supported by NSF Grant No. 2416183.

Competing interests. The authors declare no competing interests.

Supplementary material. The supplementary material for this article can be found at https://doi.org/10.1017/S0952675726100396.

References

Almeida, Jorge (1995). *Finite semigroups and universal algebra*. Singapore: World Scientific.

Baković, Eric (2000). *Harmony, dominance and control*. PhD dissertation, Rutgers University.

Beauquier, Danièle & Jean-Éric Pin (1991). Languages and scanners. *Theoretical Computer Science* **84**, 3–21.

Brzozowski, Janusz Antoni & Imre Simon (1973). Characterizations of locally testable events. *Discrete Mathematics* **4**, 243–271.

Burness, Phillip & Kevin McMullin (2019). Efficient learning of output tier-based strictly 2-local functions. In Philippe de Groote, Frank Drewes & Gerald Penn (eds.) *Proceedings of the 16th Meeting on the Mathematics of Language*. Stroudsburg, PA: Association for Computational Linguistics, 78–90.

https://doi.org/10.1017/S0952675726100396 Published online by Cambridge University Press

Burness, Phillip & Kevin McMullin (2020). Modelling non-local maps as strictly piecewise functions. *Society for Computation in Linguistics* **3**, 493–495.
Burness, Phillip & Kevin McMullin (2021). Learning multiple independent tier-based processes. In Chandlee *et al.* (2021), 66–80.
Carton, Olivier & Luc Dartois (2015). Aperiodic two-way transducers and FO-transductions. In Stephan Kreutzer (ed.) *24th EACSL Annual Conference on Computer Science Logic (CSL 2015)*. Wadern: Leibniz-Zentrum für Informatik, 160–174.
Chandlee, Jane (2014). *Strictly local phonological processes*. PhD dissertation, University of Delaware.
Chandlee, Jane, Rémi Eyraud & Jeffrey Heinz (2014). Learning strictly local subsequential functions. *Transactions of the Association for Computational Linguistics* **2**, 491–503.
Chandlee, Jane, Rémi Eyraud, Jeff Heinz, Adam Jardine & Menno van Zaanen (eds.) (2021). *Proceedings of the fifteenth International Conference on Grammatical Inference*. Cambridge: PMLR.
Chandlee, Jane & Jeffrey Heinz (2018). Strict locality and phonological maps. *LI* **49**, 23–60.
Chandlee, Jane & Adam Jardine (2021). Computational universals in linguistic theory: using recursive programs for phonological analysis. *Lg* **93**, 485–519.
Chomsky, Noam & Morris Halle (1968). *The sound pattern of English*. New York: Harper & Row.
Cook, Eung-Do (1978). The synchronic and diachronic status of Sarcee γ^{y}. *IJAL* **44**, 192–196.
Danis, Nick (2025). Logical transductions are not sufficient for notational equivalence. In Gérard Avelino, Merlin Balihaxi, Quartz Colvin, Vincent Czarnecki, Hyunjung Joo, Chenli Wang, Utku Zobarlar, Adam Jardine & Adam McCollum (eds.) *Supplemental proceedings of the 2023 and 2024 Annual Meetings on Phonology*. Amherst, MA: University of Massachusetts Amherst Libraries, 10 pp.
Dolatian, Hossep (2020). *Computational locality of cyclic phonology in Armenian*. PhD dissertation, Stony Brook University.
Dolatian, Hossep & Jeffrey Heinz (2020). Computing and classifying reduplication with 2-way finite-state transducers. *Journal of Language Modelling* **8**, 179–250.
Elgot, Calvin C. & Jorge E. Mezei (1965). On relations defined by generalized finite automata. *IBM Journal of Research and Development* **9**, 47–68.
Finley, Sara (2009). Morphemic harmony as featural correspondence. *Lingua* **119**, 478–501.
Goldsmith, John A. (1976). *Autosegmental phonology*. PhD dissertation, Massachusetts Institute of Technology.
Goldsmith, John, Jason Riggle & Alan C. L. Yu (eds.) (2011). *The handbook of phonological theory*, 2nd edition. Oxford: Wiley-Blackwell.
Gouskova, Maria & Gillian Gallagher (2020). Inducing nonlocal constraints from baseline phonotactics. *NLLT* **38**, 77–116.
Hayes, Bruce (1995). *Metrical stress theory*. Chicago: University of Chicago Press.
Heinz, Jeffrey (2009). On the role of locality in learning stress patterns. *Phonology* **26**, 303–351.
Heinz, Jeffrey (2010a). Learning long-distance phonotactics. *LI* **41**, 623–661.
Heinz, Jeffrey (2010b). String extension learning. In Jan Hajič, Sandra Carberry, Stephen Clark & Joakim Nivre (eds.) *Proceedings of the 48th annual meeting of the Association for Computational Linguistics*. Stroudsburg, PA: Association for Computational Linguistics, 897–906.
Heinz, Jeffrey (2011). Computational phonology, part I: foundations. *Language and Linguistics Compass* **5**, 140–152.
Heinz, Jeffrey (2018). The computational nature of phonological generalizations. In Larry M. Hyman & Frans Plank (eds.) *Phonological typology*. Berlin: Mouton de Gruyter, 126–195.
Heinz, Jeffrey & Regine Lai (2013). Vowel harmony and subsequentiality. In András Kornai & Marco Kuhlmann (eds.) *Proceedings of the 13th Meeting on the Mathematics of Language*. Stroudsburg, PA: Association for Computational Linguistics, 52–63.
Heinz, Jeffrey, Chetan Rawal & Herbert G. Tanner (2011). Tier-based strictly local constraints for phonology. In Dekang Lin, Yuji Matsumoto & Rada Mihalcea (eds.) *Proceedings of the 49th annual meeting of the Association for Computational Linguistics: human language technologies*, vol. 2. Stroudsburg, PA: Association for Computational Linguistics, 58–64.
Heinz, Jeffrey & Jason Riggle (2011). Learnability. In Marc van Oostendorp, Colin J. Ewen, Elizabeth V. Hume & Keren Rice (eds.) *The Blackwell companion to phonology*, vol. 1. Oxford: Wiley-Blackwell, 54–78.
van der Hulst, Harry (ed.) (2014). *Word stress: theoretical and typological issues*. Cambridge: Cambridge University Press.
Inkelas, Sharon (2014). *The interplay of morphology and phonology*. Oxford: Oxford University Press.
Jardine, Adam (2016). Computationally, tone is different. *Phonology* **33**, 247–283.
Jardine, Adam (2017). The local nature of tone-association patterns. *Phonology* **34**, 385–405.
Jardine, Adam, Jane Chandlee, Rémi Eyraud & Jeffrey Heinz (2014). Very efficient learning of structured classes of subsequential functions from positive data. In Alexander Clark, Makoto Kanazawa & Ryo Yoshinaka (eds.) *Proceedings of the twelfth International Conference on Grammatical Inference*. Cambridge: PMLR, 94–108.
Jardine, Adam & Jeffrey Heinz (2016). Learning tier-based strictly 2-local languages. *Transactions of the Association for Computational Linguistics* **4**, 87–98.
Jardine, Adam & Kevin McMullin (2017). Efficient learning of tier-based strictly *k*-local languages. In Frank Drewes, Carlos Martín-Vide & Bianca Truthe (eds.) *Language and Automata Theory and Applications, 11th International Conference, LATA 2017*. Cham: Springer, 64–76.
Johnson, C. Douglas (1972). *Formal aspects of phonological description*. The Hague: Mouton.

https://doi.org/10.1017/S0952675726100396 Published online by Cambridge University Press

Kaplan, Ronald M. & Martin Kay (1994). Regular models of phonological rule systems. *Computational Linguistics* **20**, 331–378.
Kearns, Michael & Umesh Vazirani (1994). *An introduction to computational learning theory*. Cambridge, MA: MIT Press.
Lambert, Dakotah (2021). Grammar interpretations and learning TSL online. In Chandlee *et al.* (2021), 81–91.
Lambert, Dakotah (2022). *Unifying classification schemes for languages and processes with attention to locality and relativizations thereof*. PhD dissertation, Stony Brook University.
Lambert, Dakotah (2023). Relativized adjacency. *Journal of Logic, Language and Information* **32**, 707–731.
Lambert, Dakotah (2024). System description: a theorem-prover for subregular systems: the Language Toolkit and its interpreter, Plebby. In Jeremy Gibbons & Dale Miller (eds.) *Functional and logic programming: 17th International Symposium, FLOPS 2024*. Singapore: Springer Nature, 311–328.
Lambert, Dakotah (2026). Multitier phonotactics with logic and algebra. *Phonology* **43**, article no. e9, 31 pp.
Lambert, Dakotah & Jeffrey Heinz (2023). An algebraic characterization of total input strictly local functions. *Society for Computation in Linguistics* **6**, 25–34.
Lambert, Dakotah & Jeffrey Heinz (2024). Algebraic reanalysis of phonological processes described as output-oriented. *Society for Computation in Linguistics* **7**, 129–138.
Lambert, Dakotah, Jonathan Rawski & Jeffrey Heinz (2021). Typology emerges from simplicity in representations and learning. *Journal of Language Modelling* **9**, 151–194.
Lambert, Dakotah & James Rogers (2019). A logical and computational methodology for exploring systems of phonotactic constraints. *Society for Computation in Linguistics* **2**, 247–256.
Li, Fang-Kuei (1930). A study of Sarcee verb-stems. *IJAL* **6**, 3–27.
Mohri, Meyhar, Afshin Rostamizadeh & Ameet Talwalkar (2018). *Foundations of machine learning*, 2nd edition. Cambridge, MA: MIT Press.
Nerode, Anil (1958). Linear automaton transformations. *Proceedings of the American Mathematical Society* **9**, 541–544.
Nespor, Marina & Irene Vogel (1986). *Prosodic phonology*. Dordrecht: Foris.
Oakden, Chris (2020). Notational equivalence in tonal geometry. *Phonology* **37**, 257–296.
Padgett, Jaye (1995). Partial class behavior and nasal place assimilation. In Keiichiro Suzuki & Dirk Elzinga (eds.) *Proceedings of South Western Optimality Theory Workshop 1995*. Tucson: Department of Linguistics, University of Arizona, 145–183.
Poser, William J. (1982). Phonological representation and action-at-a-distance. In Harry van der Hulst & Norval Smith (eds.) *The structure of phonological representations, part 2: linguistic models 3*. Dordrecht: Foris, 121–158.
Potts, Christopher & Geoffrey Pullum (2002). Model theory and the content of OT constraints. *Phonology* **19**, 361–393.
Prince, Alan & Paul Smolensky ([1993] 2004). *Optimality Theory: constraint interaction in generative grammar*. Oxford: Blackwell. Originally published in 1993 as technical report no. 2 of the Rutgers Center for Cognitive Science.
Rabin, Michael Oser & Dana Scott (1959). Finite automata and their decision problems. *IBM Journal of Research and Development* **3**, 114–125.
Rogers, James, Jeffrey Heinz, Margaret Fero, Jeremy Hurst, Dakotah Lambert & Sean Wibel (2013). Cognitive and sub-regular complexity. In Glyn Morrill & Mark-Jan Nederhof (eds.) *Formal Grammar: 17th and 18th International Conferences, FG 2012/2013*. Heidelberg: Springer, 90–108.
Rogers, James & Geoffrey K. Pullum (2011). Aural pattern recognition experiments and the subregular hierarchy. *Journal of Logic, Language and Information* **20**, 329–342.
Rose, Sharon & Rachel Walker (2004). A typology of consonant agreement as correspondence. *Lg* **80**, 475–531.
Rose, Sharon & Rachel Walker (2011). Harmony systems. In Goldsmith *et al.* (2011), 240–290.
Selkirk, Elisabeth (2011). The syntax–phonology interface. In Goldsmith *et al.* (2011), 435–484.
Simon, Imre (1975). Piecewise testable events. In Helmut Brakhage (ed.) *Automata Theory and Formal Languages: 2nd GI Conference*. Berlin: Springer, 214–222.
Smith, Caitlin & Charlie O'Hara (2019). Formal characterizations of true and false sour grapes. *Society for Computation in Linguistics* **2**, 338–341.
Strother-Garcia, Kristina (2019). *Using model theory in phonology: a novel characterization of syllable structure and syllabification*. PhD dissertation, University of Delaware.
Valiant, Leslie (2013). *Probably approximately correct: nature's algorithms for learning and prospering in a complex world*. New York: Basic Books.
Walker, Rachel (2011). *Vowel patterns in language*. Cambridge: Cambridge University Press.
Walker, Rachel & Bella Feng (2004). A ternary model of morphology–phonology correspondence. *WCCFL* **23**, 787–800.

https://doi.org/10.1017/S0952675726100396 Published online by Cambridge University Press