Using Locality and Natural Classes to Infer Underlying Representations and a Phonological Grammar

Jane Chandlee (jchandlee@haverford.edu) and Adam Jardine (adam.jardine@rutgers.edu)

DRAFT (under review); May 2025

1 Introduction

A standard generative phonological analysis comprises two main parts: a set of *underlying representations* (URs) of the phonological information stored for each morpheme, and a *phonological grammar* that maps URs to *surface representations* (SRs). However, if such an analysis is to make a claim to psychological reality, it must be the case that native speakers are able to simultaneously infer the URs and phonology of their language at the time of acquisition. How exactly this inference is possible is still largely an unsolved problem. While work on the problem of acquiring phonological grammars from data has seen great strides in the past three decades, the harder problem of acquiring URs has only recently seen progress (see, e.g., Apoussidou, 2007; Simpson, 2010; Pater et al., 2012; Tesar, 2014; Cotterell et al., 2015; Barke et al., 2019; Rasin et al., 2020; Ellis et al., 2022).

In this paper, we show how a theory of phonology that explicitly refers to computational properties of phonological processes, enhanced with knowledge of natural classes, allows for a learning procedure that can simultaneously acquire both a phonology and a set of URs from positive data.[1] Specifically, building on work by Hua et al. (2021) and Hua and Jardine (2021), we conceive of the phonological learning problem in terms of *function decomposition*. Children are not directly exposed to URs; rather, they have to infer them from alternations in surface forms. We therefore model the input children get during acquisition as strings of morphemes paired with phonological SRs, e.g., (CAP-PL, [kæps]).[2] These pairs of strings are generated by the *composition* of two functions, the lexicon that maps morphemes to URs—e.g., CAP $\rightarrow$ /kæp/, PL $\rightarrow$ /z/—and the phonological grammar that maps URs to SRs—e.g., /kæp-z/ $\rightarrow$ [kæps]. The goal of the learner is then to *decompose* this composition in order to identify the lexicon and phonology themselves.

We present a learner that solves this problem in the case that the phonology function enacts a combination of several strictly 2-local (ISL_2) mappings, meaning that each process is conditioned by a context of length 1. This is a generalization of Hua and Jardine (2021)'s learner, which can only learn a phonology function representing a single process. We accomplish this generalization by giving the learner explicit knowledge of morphological

[1] In this paper we limit our attention to local phonological processes, but toward the end we discuss the potential for expansion to long-distance phonology.

[2] By *morpheme* here we refer to the meaning component of the sound-meaning correspondence that a morpheme represents. The learner is tasked with identifying the sound component. Throughout the paper we will notate these abstract forms with small caps, e.g. PL for plural, CAP for English 'cap', etc.

structure and the ability to reason about natural classes. Using a case study from Johor Malay (based on the data and description in Onn, 1976), we demonstrate that the proposed algorithm learns the lexicon and phonology from a dataset of attested forms that reflects six distinct phonological patterns including an opaque interaction.

Equally importantly, the fact that this learner is fully interpretable means we can also explain where (and why) it falls short, a discussion that will identify the crucial next steps for developing this learner into a more comprehensive model of morpho-phonological acquisition. Thus, while our proposed learner has various limitations, the contribution of this paper is to provide a proof-of-concept for an overall architecture for studying the UR learning problem. As we discuss, because the algorithms that form the base of the learner are provably correct, and because its hypothesis space is well-understood, there is a clear path forward for expanding this algorithm to one that covers a larger range of cases and learning challenges.

The outline of the paper is as follows. In Section 2, we provide a definition of the phonological learning problem in terms of function decomposition, first informally and then formally. Section 3 describes how the learning algorithm works using a toy example. Section 4 then demonstrates the learner on a set of morpho-phonological and allophonic rules in Johor Malay. Section 5 situates the current approach in the context of the previous literature on learning URs, and Section 6 outlines the next steps for developing the learner into a more general and comprehensive model of morpho-phonological acquisition. Section 7 then briefly concludes.

2 The learning problem

At the heart of phonological analysis is the principle that, wherever possible, a morpheme is assigned a single UR.[3] Alternations in the SRs of morphemes are then explained by the application of a phonological grammar to these URs (Kenstowicz and Kisseberth, 1977, pp. 26–27). For example, in Johor Malay the nominalizing prefix with UR /pəŋ-/ alternates among several SRs by assimilating to a following voiced obstruent.

(1) Allomorphy in Johor Malay (Onn, 1976)

/pəŋboroŋ/	[pəmboroŋ]	'wholesaler'
/pəŋjahit/	[pəɲjahit]	'tailor'
/pəŋdaki/	[pəndaki]	'climber'
/pəŋgali/	[pəŋgali]	'digger'
/pəŋarah/	[pəŋarah]	'director'

At a high level, then, there are two main stages to the production of surface forms: strings[4] of morphemes are assigned URs, and then the phonology applies to those URs to produce SRs. These two stages are schematized in Figure 1 with two examples from (1).

A learner, however, cannot directly observe either the UR assignment or the phonology of

[3]Suppletive allomorphy would be a case where this isn't possible. We leave for future work the interesting and challenging problem of deciding when to resort to suppletion.

[4]This, of course, assumes concatenative morphology. Non-concatenative morphology introduces additional challenges that could be addressed in future work that builds on the results discussed here.

$$
\begin{tikzcd}
\textsc{NomClimb} \arrow[d] \arrow[r, phantom, "\text{UR assignment}" near start, shift right=4ex] & \textsc{NomDirect} \arrow[d] \\
/\text{pəŋdaki}/ \arrow[d] \arrow[r, phantom, "\text{phonology}", shift right=4ex] & /\text{pəŋarah}/ \arrow[d] \\
[\text{pəndaki}] & [\text{pəŋarah}]
\end{tikzcd}
$$

Figure 1: Flow of UR assignment and application of phonology.

their target language(s). For present purposes, we assume a starting point where they do have the SRs of words paired with the semantic and morphosyntactic information those words encode, as schematized in (2).

(2) (NomClimb, [pəndaki])
(NomDirect, [pəŋarah])
...

This conception of the input data is of course an idealization that assumes the learner can accurately parse SRs from a phonetic output and that they already have fairly complete information about the morphology, including how many morphemes are present in the SR and the order in which they appear.[5] However, there are several reasons why it is still worth studying this more idealized version of the learning problem.

First, solving this version of the problem is an important step forward towards a more realistic model of phonological learning, as compared to earlier work that includes URs in the input to the learner (e.g. Tesar and Smolensky, 2000; Chandlee et al., 2014; Belth, 2023b). Second, it allows us to zero in on sub-problems of the problem of simultaneously learning the phonology and lexicon. For instance, the learner must parse SRs into the surface allomorphs of their constituent morphemes—given (NomClimb, [pəndaki]), is the Climb morpheme associated with the string [i], [ki], [aki], [daki], etc.? Furthermore, the learner must tease apart multiple, interacting phonological processes in their grammar. Solutions to these problems can then, in future work, be applied to the larger problem of learning in the face of incomplete morphological information.

The learning problem, then, can be stated (informally, for now) as follows:

(3) The learning problem (informal version)
Given a finite set of pairs of strings of morphemes and their SRs, return
1. a list of the URs assigned to each morpheme (i.e., a lexicon); and
2. a phonological grammar;
that are consistent with this set.

[5]This assumption is not unique to our approach and has many precedents in the literature on UR learning (see, e.g., Jarosz, 2006a,b; Apoussidou, 2007; Merchant, 2008; Tesar, 2014; Nyman and Tesar, 2019).

Our approach to solving this problem starts by representing the two stages in Figure 1 as *functions* that map one set of objects to another set of objects. For example, UR assignment can be viewed as a function L (for 'lexicon') from strings of morphemes to URs. The phonology is then a function P from URs to SRs.[6] The input pairs of strings of morphemes and their SRs, as in (2), are thus drawn from the *composition* $P \circ L$. For example, $P \circ L(\text{NOMCLIMB})$ = [pəndaki] because $P(L(\text{NOMCLIMB})) = P(/\text{pəŋdaki}/)$ =[pəndaki].

The learning problem is thus a *decomposition* problem: given pairs from $P \circ L$, determine L and P. By posing the problem in this way we can tailor the learning procedure based on our assumptions about the kinds of functions L and P are.[7] For example, because we are assuming one UR per morpheme, L is a *homomorphism*. That is, it maps any individual morpheme to the same UR regardless of where it is in the string of morphemes.

As for P, we draw on the body of work characterizing the computational complexity of phonological processes (Heinz, 2018, et seq.). In particular, we will assume that P is an *input strictly* 2*-local* (ISL_2) function, which means that changes to any segment in the UR are dependent on only a single adjacent segment (Chandlee, 2014; Chandlee and Heinz, 2018). The example phonology function above is ISL_2 because the place feature of the prefix nasal consonant is dependent only on the immediately following obstruent. Of course, this is an overly strong assumption about the nature of phonological functions—while many phonological processes are ISL_2, not all of them are. However, as we discuss later in §6, the ISL_2 assumption provides a useful test case (both theoretically and empirically) for techniques that can be expanded to other classes of functions.

We can now formally characterize the learning problem as follows:

(4) The learning problem (formal version)
Given a homomorphism L and an ISL_2 function P, from a finite set of pairs drawn from $P \circ L$, find a representation of L and P.

In the next section we outline the steps of the algorithm using a toy example for simplicity. After that, §4 presents a detailed demonstration using a more realistic problem instance drawn from multiple patterns of allomorphy and allophony in Johor Malay.

3 The algorithm

The algorithm is an extension of the Simplex Input Strictly 2-Local Decomposition Learning Algorithm (SI_2DLA) first proposed in Hua et al. (2021) and later refined and proved in Hua and Jardine (2021).[8] The basic steps of the algorithm are as follows:

[6]The use of functions abstracts away from the problem of optionality and variation (Anttila, 2007; Vaux, 2008), in which a single UR may be mapped to multiple SRs. However, there are established methods for learning *semi-deterministic* functions that map strings to sets of strings (Beros and de la Higuera, 2016) and therefore can be used to model optionality (see Heinz, 2020).

[7]Put another way, we can explore proposals for what kinds of functions they are by showing the consequences for learning, i.e., *if* L and P are these types of functions, how might a learner successfully decompose $P \circ L$?

[8]The term *simplex* here means that the phonology function P only makes a single change. The extension of SI_2DLA being presented in this paper relaxes this restriction.

(5) a. Initialize the phonological grammar to an ISL_2 finite-state transducer (FST) of the identity map (§3.1)
b. Create another finite-state representation of the training data and segment the morphemes (§3.2)
c. Merge states to reveal the generalizations governing allomorphy (§3.3)
d. Apply heuristics to select the UR from the set of surface forms for each morpheme (§3.4)
e. For each alternating morpheme, identify the change necessary to derive the SR from the UR and modify the corresponding transition in the phonology FST accordingly (§3.5)

We will elaborate on each of these steps through a demonstration on the following toy example. First consider an inventory $\Sigma = \{\mathrm{t, d, a}\}$; that is, two consonants, one voiced and one voiceless, and one vowel. Then assume a vocabulary of five abstract morphemes: three roots R_1, R_2, and R_3, and two suffixes S_1 and S_2. The target grammar in (6) includes a lexicon of URs for each of these morphemes (6a) and a phonological mapping that enacts a single process assimilating /t/ to [d] after another [d], given in rule format in (6b).

(6) a. Lexicon:

R_1	/tad/	S_1	/ta/
R_2	/tat/	S_2	/da/
R_3	/tada/		

b. Phonology: $\mathrm{t} \rightarrow \mathrm{d}$ / d __

A dataset that includes the SRs of each root both in isolation and affixed with each suffix is given in (7). Note that the morpheme S_1 alternates on the surface as [ta] following [t] and [a] (e.g., in R_2S_1 [tatta] and R_3S_1 [tadata]) and as [da] following [d] (e.g., in R_1S_1 [tadda]).

(7) A dataset generated by the lexicon and phonology in (6)

R_1	[tad]	R_1S_1	[tadda]	R_1S_2	[tadda]
R_2	[tat]	R_2S_1	[tatta]	R_2S_2	[tatda]
R_3	[tada]	R_3S_1	[tadata]	R_3S_2	[tadada]

The remainder of this section will walk through the steps in (5), demonstrating how the learner takes a dataset like (7) and returns a grammar equivalent to (6).

3.1 Initialization of the phonology FST

First, the learner initializes the phonological grammar to an ISL_2 FST that represents the identity map for the given inventory. FSTs are abstract automata that transform input strings into output strings using a set of states and transitions (Mohri, 1997; Sakarovitch, 2009, for an explanation for phonologists, see Chandlee and Heinz (2018)). The phonology FST for our toy example is given in Figure 2.

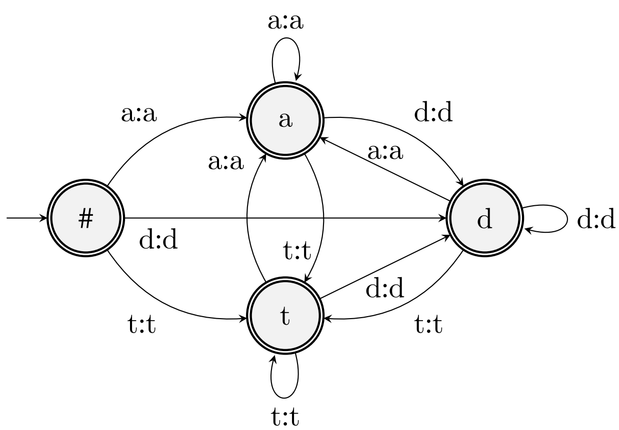

Figure 2: Initial phonology function: an ISL_2 identity map for $\Sigma = \{\mathrm{t, d, a}\}$.

Importantly, the states in this FST each represent a distinct 1-*suffix*, meaning the last segment seen in the input.[9] Every transition that reads an 'a' goes to state 'a', every transition that reads 'd' goes to state 'd', and every transition that reads 't' goes to state 't'. This means that a string used as input to this FST will end up in the state that corresponds to its 1-suffix. For example, the string /tad/ will end in state 'd', as will all other strings that share the 1-suffix 'd'.

As will become clear later, it is this structure of the FST (its states and transitions) that the learner will exploit when it reasons about the relevant contexts for the phonological processes being applied to URs. At this point, however, the FST represents the identity function ($f(x) = x$) and therefore makes no changes to any input. In other words, the learner initially assumes that the phonology maps all URs to fully faithful SRs.[10]

3.2 Morpheme segmentation

Step (5b) is to build a FST representation of the training data and use it to parse the strings into their component morphemes and surface forms. The particular FST representation used is called an *onward prefix-tree transducer*, which is a lossless representation of the dataset in which input strings and their associated outputs are stored based on shared prefixes.[11] Specifically, onwardness is achieved by determining the *longest common prefixes* (LCPs) of all surface forms that begin with a particular string of morphemes.[12]

For example, the set of SRs of all items that begin with R_2 is $\{\mathrm{tat}, \mathrm{tatda}, \mathrm{tatta}\}$, and the LCP of this set is [tat]. So we can associate [tat] with the morpheme R_2. From there, to

[9]The term *suffix* here does not refer to morphology but to a substring of a string that includes the end. An n-suffix is a suffix of length n.

[10]The initial state of the phonology in acquisition has been a subject of debate in the literature, particularly in work that assumes a constraint-based Optimality Theory (OT) grammar. Following Smolensky (1996), a significant amount of computational work on learning with OT assumed an initial state in which all markedness constraints outranked all faithfulness constraints (M $\gg$ F). But see Hale and Reiss (1998) for arguments for the opposite initial state, in which F $\gg$ M, which is more consistent with the identity map phonology that our model assumes.

[11]An algorithm for generating an onward PTT was first given by Oncina et al. (1993) in their *Onward Subsequential Transducer Inference Algorithm* (OSTIA), which we will also used for state merging in step (5c).

[12]Again the term *prefix* has no morphological meaning here; it is simply an initial portion of a string.

determine the surface form of S_1, we consider all inputs that begin with R_2S_1. There is only one such string, R_2S_1, and so trivially its LCP is [tatta]. As we already determined that [tat] is associated with R_2, this initial segment can be removed, leaving [ta] for S_1. And so on, for all of the morphemes in our dataset. The morpheme-SR pairings discovered through this method are stored in a prefix tree transducer (PTT), show in Fig. 3.

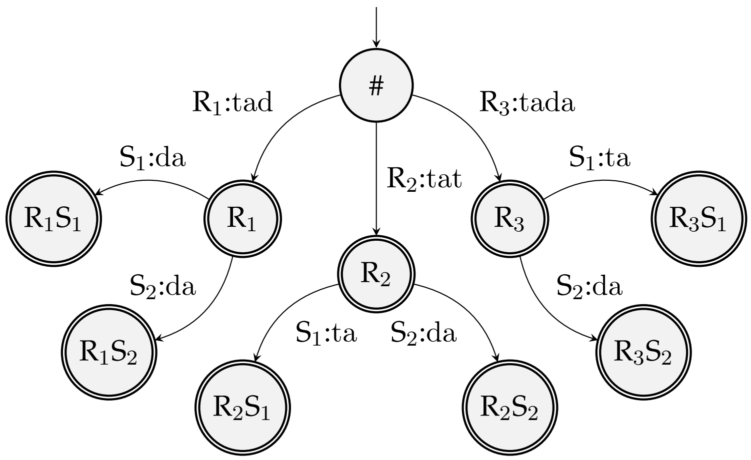

Figure 3: An onward prefix tree transducer for the data in (7).

Each pair of strings in the dataset is stored as a path through the PTT. For example, the pair (R_1S_1, [tadd]) is represented by the transition from the # state to the R_1 state on the transition R_1:tad and then by the transition S_1:da to state R_1S_1. State names represent the input prefix that reaches that state. This organization enables the learner to detect allomorphy and its associated contexts through state merging, which we turn to in the next step.

3.3 Detecting allomorphy through state merging

The next step of the algorithm is to *merge* states in the onward PTT that have the same behavior in terms of the allomorphy they induce on morphemes. The state merging strategy is the one used by the OSTIA (Onward Subsequential Transducer Inference Algorithm), which induces a subsequential (i.e., deterministic) function from a finite PTT by merging states with shared behavior (Oncina et al., 1993).

For example, looking at the PTT in Fig. 3, we see that the morphemes S_1 and S_2 are output as the same SRs—[ta] and [da], respectively—when they come out of both states R_2 and R_3. Thus, we can safely merge states R_2 and R_3 into a single state (preserving their incoming and outgoing transitions) without altering the function the FST represents. Put another way: regardless of the difference in the paths that lead *to* states R_2 and R_3, once we *leave* them the outputs for all morphemes are identical. So there is no need to maintain a distinction between these states. The same goes for all of the 'leaves' of the PTT: none of these states have any outgoing transitions at all, and so they trivially have the same behavior and can safely be merged.[13]

[13]Formally, the criterion for merging states in the OSTIA is that they share the same set of *tails*, or outgoing paths, and can be merged without creating irreparable nondeterminism. See Oncina et al. (1993) for details.

More specifically, we use a version of the OSTIA for learning partial functions, called the OSTIA-D (Oncina and Varò, 1996). The 'D' designation indicates that the learner makes use of explicit information about the domain of the target function. This is necessary when learning partial functions because if the learner otherwise assumes it is learning a total function (as the original OSTIA does), it will be expecting certain sequences in the training data that don't actually exist.

With respect to the current learning problem, the target function is a partial function because the language's morphotactics will prevent certain morpheme combinations from appearing in the training data. In our toy example, roots can appear unaffixed or with a single suffix. But suffixes don't appear without a root, or before a root, and roots do not combine with other roots. In other words, the data do not reflect a total function over the domain of morphemes. Instead, the language of the partial function's domain is the one accepted by the FSA in Fig. 4. This FSA is provided to the learner as explicit knowledge of the language's morphology.

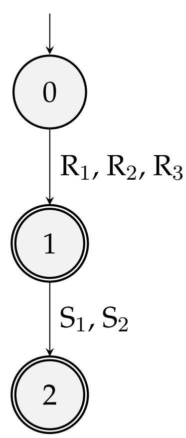

Figure 4: Domain FSA representing the morphotactics of toy example language.

Given this FSA, the OSTIA-D adds a second condition for which states can be merged in the PTT, which is that the prefixes that lead to the two states to be merged must also end in the same state of the domain FSA. For example, the strings that lead to states R_2 and R_3—which we determined above have the same behavior—also meet this second condition: they both lead to state 1 in the domain FSA. Likewise, the strings that lead to all of the leaf states—R_1S_1, R_1S_2, R_2S_1, R_2S_2, R_3S_1, and R_3S_2—all lead to state 2 in Fig. 4.

The resulting machine after all allowable state merges are performed is given in Fig. 5.[14] Note that the consequence of using the OSTIA-D is that the structure of the domain FSA has been imposed on this FST, with the important difference that two 'root' states have survived state merging. While R_1, R_2, and R_3 all reach the same state in Fig. 4, these states

[14] A note about the state labels in our figures. For clarity, in the prefix tree transducer the states are labeled with the prefix that leads to them, but after state merging a single state will represent a set of these prefixes. As we will see, our learner does not make use of state labels at all, but instead gathers the needed information from the transitions only. To emphasize this, we have arbitrarily labeled the states with integers, with the convention that state 0 is the start state. For reference, though, in Fig. 5 state 1 corresponds to state R_1 in the prefix tree, state 2 is the result of merging states R_2 and R_3, and state 3 is the result of merging all of the leaves of the tree.

cannot all be merged because they do not share the same behavior. Specifically, morpheme S_1 is output as $[\mathrm{da}]$ after R_1 in the PTT, but it is output as $[\mathrm{ta}]$ after R_2 and R_3. Therefore the merging of R_1 with either R_2 or R_3 is rejected (though R_2 and R_3 themselves can be merged as we've already discussed). In this way, state merging preserves all and only the contextual information that governs the allomorphy the learner needs to discover.

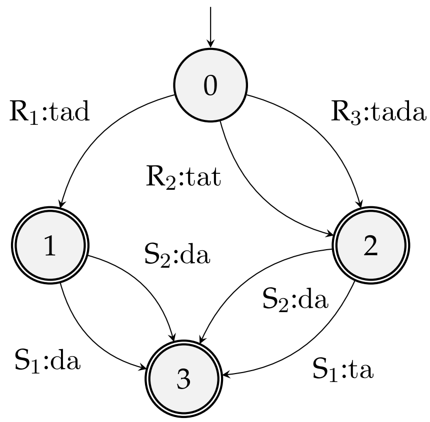

Figure 5: Allomorphy revealed by merging states in Fig. 3 using the OSTIA-D.

Using this FST, the learner will select a UR and then extract the relevant phonological information from the surface forms of each morpheme to determine the changes made by the phonology. How it does this will be explained in the next two subsections.

3.4 UR selection

With the information in Fig. 5, the learner can now infer the URs and build a lexicon. First, for those morphemes that do not alternate, it trivially assumes that the single SR is the UR. For those URs that *do* alternate, it zeroes in on the relevant phonological information that will govern the alternation. In our toy example, only one morpheme alternates: S_1.

What counts as 'relevant' phonological information is tied to the initial assumption (from §3.1) that the phonological function is an ISL_2 function: the context that governs the surface form of a morpheme is the 1-suffix of the morpheme that was output before it. We can therefore translate the states of Fig. 5 into the sets of 1-suffixes of the outputs leading into them. For example, the one surface form that is output going into state 1 ($[\mathrm{tad}]$) has the 1-suffix $\{\mathrm{d}\}$. In contrast, the surfaces forms going into state 2 have the 1-suffixes $\{\mathrm{t},\ \mathrm{a}\}$, (from $[\mathrm{tada}]$ and $[\mathrm{tat}]$). Fig. 6 presents the FST again with these states relabeled with their sets of 1-suffixes.

The original $\mathrm{SI}_2\mathrm{DLA}$ makes the assumption that allomorphs that are output from the state with the *larger* set of 1-suffixes are the URs: in this case that criterion identifies $/\mathrm{ta}/$ as the UR for S_1. Of course, in the general case (i.e., beyond the simplex functions assumed by $\mathrm{SI}_2\mathrm{DLA}$) this heuristic for selecting URs will be too simplistic, and so in our expanded learner we make use of two additional mechanisms. We will sketch those here and then give more details in the case study in §4. The first is a *natural class heuristic*,which takes advantage of the fact that often the UR appears with a motley collection of contexts that does

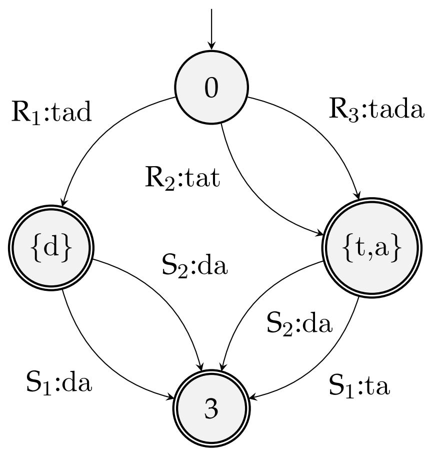

Figure 6: States from Fig. 5 relabeled with incoming 1-suffixes.

not constitute a natural class. Accordingly, if all but one of the surface variants appears with a set of 1-suffixes that constitute a natural class, that one variant will be selected as the UR.

To address cases in which this natural class heuristic is insufficient, the learner is also equipped with a *hypothesis testing* capability through which it considers each potential UR in turn and looks for a contradiction with the grammar it has identified so far. Specifically, given a hypothesized UR, the learner identifies the set of phonological changes that would be needed to generate the other variants. Using the lexicon it has constructed so far, the learner applies those changes and checks whether the SRs they generate match what is in the training data. If not, that hypothesis for the UR can be rejected.

3.5 Updating the phonology FST

With the URs selected, the learner then compares them to the other variants to identify the needed phonological changes. Continuing with our toy example, since /ta/ is the selected UR of S_1, the learner detects the change by comparing the *longest common suffix* (LCS)—that is, the longest shared *final* sequence—of /ta/ and the SR [da]. In this case, the LCS is 'a', as shown in (8).

(8) Comparing allomorphs of S_1

$$\begin{array}{lcc} \text{UR} & \text{t} & \text{a} \\ \text{SR} & \underbrace{\text{d}}_{\text{difference}} & \underbrace{\text{a}}_{\text{LCS}} \end{array}$$

Ignoring this LCS identifies the change that must be enacted by the phonology: an underlying /t/ changes to [d]. The contexts in which this change occurs are the set of 1-suffixes that correspond to the state that generates the SR, in this case {d}. This discovered generalization—/t/ becomes [d] after /d/—is added to the phonology by editing the corresponding transition in the phonology FST, as shown in Fig. 7.

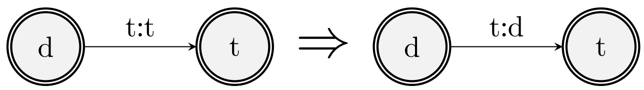

Figure 7: Editing the transition in Fig. 2 representing the phonological change.

With this transition edited in the phonology FST, the learner can now simplify the FST in Fig. 5 by changing the outputs of all transitions to the selected URs and then merging all states. The resulting 'lexicon' FST is depicted on the left-hand side of Fig. 8.

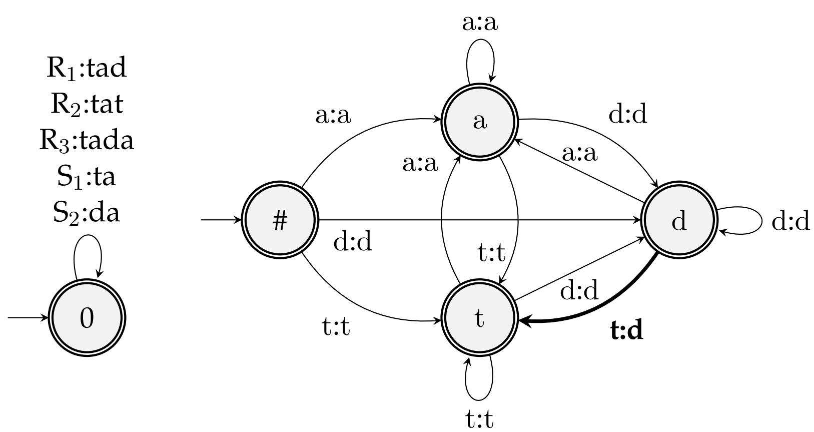

Figure 8: Final lexicon(on the left) and phonology (on the right), with the modified transition highlighted in bold.

The reader can confirm that the two FSTs in Fig. 8 describe the target correspondence between morphemes and URs in (6a) and the rule in (6b), and further that they recreate the dataset in (7) when used in composition, i.e., using a concatenation of the outputs of the lexicon machine as the input to the phonology machine.

3.6 Requirements for guaranteed correctness

Both the OSTIA-D and the SI_2LDA are provably correct algorithms, which means we know what kind of data is necessary to guarantee they will succeed. Importantly, that guarantee extends to *any* function in the designated class of functions (subsequential for OSTIA-D and simplex ISL_2 for SI_2DLA). Here we give an informal description of these guarantees (see Oncina et al., 1993; Oncina and Varò, 1996; Hua and Jardine, 2021, for the formal proofs) and discuss the extent to which they also apply to the learner we are presenting in this paper.

Essentially, the crucial step is generating the merged machine that accurately describes the morphophonological behavior of the morphemes—this is Fig. 5 in the above example. The data required to do this is as follows. First, the data must include some input string that exercises each transition in this target machine. In terms of the empirical problem at hand, this means the learner needs to see each combination of morphemes permitted by the morphotactics. (In the cases here the set of such combinations is finite; in cases with an unbounded number of morpheme combinations, it is sufficient to just exercise each transition in the domain FSA.) Second, to ensure that the morphemes are correctly parsed—i.e., that the outputs on each transition can be correctly calculated using the LCP procedure outlined in Sec. 3.2—for each transition, any *following* transition must also be exercised. This resolves any ambiguity as to which part of a surface form corresponds to which morpheme.

Finally, the SI_2DLA imposes the additional requirement that there are sufficient morphemes to make the inference in step (5d) (Sec. 3.4) about the URs of each morpheme. The current learner is a generalization of the SI_2DLA to phonologies with multiple processes and thus requires a more detailed UR selection procedure that makes use of at least natural classes and a hypothesis testing mechanism. As we will see in Sec. 4, this new UR selection procedure requires more information that cannot always be characterized in terms of the content of the dataset. As we will also see, a comprehensive understanding of what information is required to guarantee successful UR selection (in all cases) will ultimately fall beyond the scope of this paper. However, the interpretability of our overall approach to this problem will greatly facilitate our investigation into that question.

All of this will be demonstrated and discussed in great detail stating with the next section, in which we apply our learner to data from an actual language: Johor Malay. We will see that the learner can identify the lexicon and phonology even when faced with six distinct patterns of alternation, including an opaque interaction. These results thus represent a significant step forward from the simplex functions assumed by SI_2DLA.

4 Case study: Johor Malay

The data and description of the phonological patterns used in the following case study are based on Onn's (1976) dissertation.[15] As an instance of the phonological learning problem, Johor Malay provides a variety of pattern types and learning challenges, including allomorphy, allophonic variation, neutralization, deletion, epenthesis, optionality, derived environment blocking, and opaque interactions. While not all of these challenges are within reach of the current version of the learner, we will demonstrate the ones that are using a set of patterns that we will now describe.

Here we anticipate the following potential objection: that in leaving out patterns the current learner can't handle we have hand-crafted a test case that will exaggerate its abilities. In response we emphasize that this is a deterministic learner that succeeds by making use of the known formal properties of a target class of objects, in this case the class of ISL_2 functions. Because all of the phonological patterns we are about to describe belong to this class, the learner's ability to learn them (and *all* others that share the same formal properties) is already known. The purpose of the demonstration is not then to *assess* the learner's performance, but to *illustrate* its known behavior in a way that is intuitive to phonologists. Following this demonstration, we will discuss in detail the learner's limitations and the paths forward that they point to.

Turning now to the facts for Johor Malay, Tables 1 and 2 present the consonant and vowel inventories, respectively. Glottal stop is not phonemic but surfaces as an allophone of /k/ and is also used to break up vowel hiatus. The rhotic /r/ (represented in Onn (1976)

[15]Later revised and published as Onn (1980).

with the symbol /r̃/) is not trilled but is 'produced with the tongue somewhat retracted towards the front of the soft palate, and without radical constriction' (pg. 24). The learner is supplied with this inventory as well as the feature chart listed in Appendix A.2. (The role of features is discussed in more detail in Sec. 6.4.)

Bilabial	Alveolar	Palatal	Velar	Glottal
p b	t d		k g	(ʔ)
		c j		
	s			
m	n	ɲ	ŋ	
	l r			
w		y		h

Table 1: Consonant inventory of Johor Malay (Onn, 1976).

i		u
e	ə	o
	a	

Table 2: Vowel inventory of Johor Malay (Onn, 1976).

Our case study is focused on the allomorphy of two prefixes—the nominalizer /pəŋ-/ and the active voice morpheme /məŋ-/. We will illustrate their patterns of allomorphy using the nominalizer /pəŋ-/, but /məŋ-/ behaves exactly the same. As already described in §2, before voiced obstruents, the nasal in the prefix is subject to place assimilation. Examples (repeated from (1)) are given in (9). As seen in the last two examples, because they appear before vowels as well as velars, the variants /pəŋ/ and /məŋ/ are assumed to be the URs.

(9) Nasal place assimilation before voiced obstruents

/pəŋ-boroŋ/	[pəmboroŋ]	'wholesaler'
/pəŋ-jahit/	[pəɲjahit]	'tailor'
/pəŋ-daki/	[pəndaki]	'climber'
/pəŋ-gali/	[pəŋgali]	'digger'
/pəŋ-arah/	[pəŋarah]	'director'

Before voiceless obstruents, the obstruent also deletes.[16]

(10) Nasal assimilation and deletion before voiceless obstruents

[16] Except for /c/ (Onn, 1976, pg. 64-65). Note also that /s/ patterns with {c, j} rather than {t, d}. It is therefore described as alveopalatal, though in the feature chart it is specified as [+anterior, +coronal]. Regardless, because /s/ still forms a natural class with the palatals /c, j/ based on the feature strident, this discrepancy does not affect the performance of our learner with respect to UR selection. We thank the reviewers for pointing this out. For more on the deletion part of this pattern see §6.1.

/pəŋ-pandu/	[pəmandu]	'a guide'
/pəŋ-karaŋ/	[pəŋaraŋ]	'author'
/pəŋ-samun/	[pəɲamon]	'robber'
/pəŋ-tari/	[pənari]	'dancer'

Before sonorant consonants, the prefix nasal deletes.[17]

(11) Deletion before sonorant consonants

/pəŋ-layan/	[pəlayan]	'waitress'
/pəŋ-rayu/	[pərayu]	'appeal'
/pəŋ-malu/	[pəːmalu]	'shame'
/pəŋ-ɲaɲi/	[pəːɲaɲi]	'singer'
/məŋ-ŋaŋa/	[məːŋaŋə]	'to open one's mouth'

In addition to the prefix allomorphy, our data also reflect various patterns of neutralization and allophonic variation. First is obstruent devoicing in coda position, illustrated in (12). The stem /jawab/, 'to answer', surfaces as [jawap] without a suffix or when suffixed with the consonant-initial causative benefactive suffix /-kan/.

(12) Coda devoicing

/jawab/	[jawap]	'to answer'
/pəŋ-jawab-an/	[pəɲjawaban]	'the answering'
/məŋ-jawab-kan/	[məɲjawapkan]	'to cause to answer for'

Next, also in coda position, velar stops become glottal, as illustrated in (13), and /r/ deletes, as illustrated in (14).

(13) Velar codas → glottal

/masak/	[masaʔ]	'to cook'
/pəŋ-masak-an/	[pəmasakan]	'the cooking'
/məŋ-masak-kan/	[məmasaʔkan]	'to cause to cook for'

(14) Coda /r/ deletion

/kisar/	[kisa]	'revolve'
/kisar-an/	[kisaran]	'revolution'
/kisar-kan/	[kisakan]	'to cause to revolve for'

Lastly, word-final /a/ raises to schwa, as illustrated in (15):

[17] Onn (1976, pg. 60) notes that in the case of nasal-initial stems, the prefix vowel also lengthens. The current version of our learner will identify [pəː] and [məː] as additional allomorphs that appear with nasal-initial stems. However, these allomorphs are subject to two phonological changes: deletion and lengthening. As the latter process is ISL_3, it falls outside of the scope of the current iteration of the learner. We therefore leave this aspect of the pattern aside in the current analysis, though see §6 for discussion of extending the learner to non-ISL_2 processes.

(15) Word-final /a/ raising

/bawa/	[bawə]	'to carry'
/bawa-kan/	[bawakan]	'to cause to carry for'

These last two processes—/r/-deletion and /a/-raising—create an opaque interaction: deletion counterfeeds raising. This interaction is demonstrated using ordering in (16), though our learner is not targeting a grammar of ordered rules. The correct order is shown on the left; reversing the order generates the wrong surface form (shown on the right).

(16) Counterfeeding interaction

UR	/bakar/		/bakar/
/a/-raising	—	/r/-deletion	[baka]
/r/-deletion	[baka]	/a/-raising	[bakə]
SR	[baka]		*[bakə]
	'to burn'		

4.1 Data

The learner will be demonstrated using the set of stems listed in Table 3. These stems were selected to include all of the contexts for the set of patterns described above. Their glosses are given in small caps, which will correspond to their meaning representations in the learner's input strings.

TIE	/ikat/	INSTIGATE	/asut/
DIG	/gali/	PAY	/bayar/
CARRY	/bawa/	EMBRACE	/dakap/
FARM	/ladaŋ/	ASCEND	/naik/
ANSWER	/jawab/	REVILE	/cərca/
ROB	/rompak/	OPEN	/ŋaŋa/
PLAY	/main/	SING	/ɲaɲi/

Table 3: Stems used to demonstrate the learner.

The set of affixes that will be used are listed in Table 4. In addition to the prefixes already mentioned, there are three suffixes. The suffix /-i/ marks causative, /-kan/ marks causative benefactive, and /-an/ is a nominalizer. The nominalizing suffix can be used on its own or in combination with the nominalizing prefix /pəŋ-/.[18] Likewise, the causative and causative benefactive suffixes can be used on their own or in combination with the active voice prefix /məŋ-/. To avoid confusion, we will use NOMP for the prefix nominalizer and NOMS for the suffix nominalizer.

[18]According to Onn (1976, pg. 103), the nominalizing suffix contributes various meanings, including locative (/mandi/, 'to bathe', [pəmandiyan], 'place for bathing'), resultative (/baŋun/, 'to arise', [baŋunan], 'building'), collective (/darat/, 'land', [daratan], 'land mass'), and verbal noun (/jatuh/, 'to fall', [kəjatuhan], 'downfall').

ACT	/məŋ-/	active
NOMP	/pəŋ-/	nominalizer
CAUSE	/-i/	causative
BEN	/-kan/	causative benefactive
NOMS	/-an/	nominalizer

Table 4: Affixes used to demonstrate the learner.

The dataset given to the learner includes all of the stems unaffixed, with each suffix, with each prefix, and with each allowable prefix/suffix combination. For example, the set of (morpheme, SR) pairs for the stem /rompak/, 'rob', is listed in (17).

(17) (ROB, rompaʔ)
(ROBCAUSE, rompaki)
(ROBBEN, rompaʔkan)
(ROBNOMS, rompakan)
(ACTROB, mərompaʔ)
(NOMPROB, pərompaʔ)
(ACTROBCAUSE, mərompaki)
(ACTROBBEN, mərompaʔkan)
(NOMPROBNOMS, pərompakan)

As discussed in §3.3, the learner is also provided with the FSA in Figure 9, which represents the domain language from which the input strings of morphemes are drawn (i.e., the morphotactics).

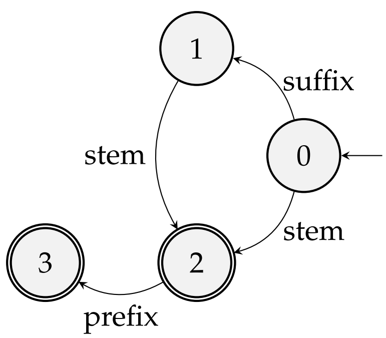

Figure 9: Domain FSA representing the morphotactics for Johor Malay case study.

A note about directionality. All of the patterns described above are regressive, i.e., the triggering context follows the target. Accordingly, the FST representations used by the learner will read the strings from right-to-left (i.e., suffixes will be read first, then stems, then prefixes). As a visual reminder of this, the finite-state diagrams that follow (as well as the one in Figure 9) are displayed with the start state on the right-hand side of the page. For now we consider direction to be a parameter of the learner that is set in advance, but we will say more about the role of directionality in §6.

4.2 Demonstration

As described in §3, the first steps of the learner are to 1) initialize an identity map ISL_2 phonological function for the segment inventory, 2) construct a prefix tree transducer for the dataset and then parse the morphemes by making the tree onward, and 3) merge states in the PTT using the OSTIA-D for partial functions. Given the dataset described above, the output of state merging is an FST with 9 states. For readability, only a fragment of this FST is shown in Figure 10, focusing only on two classes of stems: those beginning with labials and those beginning with sonorants.[19]

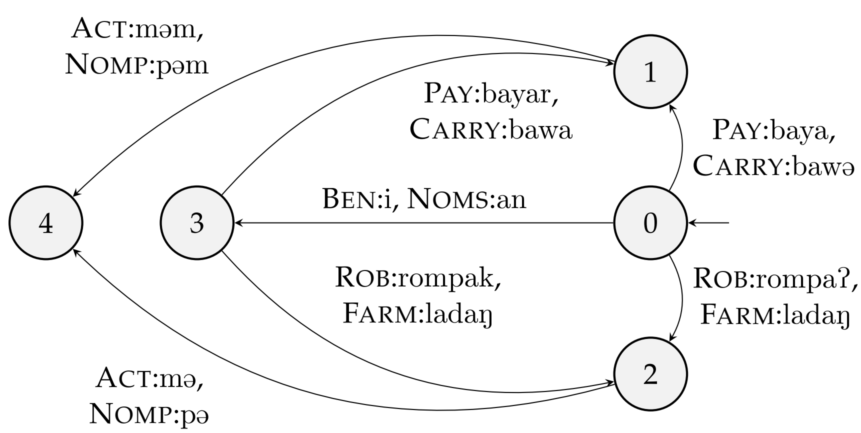

Figure 10: Fragment of the OSTIA-D output for Johor Malay test case.

Why nine states? As noted in §3.3, the structure of the domain FSA (Figure 9) is imposed on the FST that results from state merging. So this FST has distinct states for stems, suffixes, and prefixes, connected by transitions according to the language's morphotactics. But, state merging must also retain those states that reflect the crucial contextual information governing the patterns of alternation reflected in the data. For example, the FST retains two suffix states, one for vowel-initial suffixes and one for consonant-initial suffixes (only the former is shown in the figure as state 3), because different variants of some stems appear before these two subclasses. For example, because of /k/-glottaliztion and /r/-deletion, stems like PAY and ROB surface as [bayar] and [rompak], respectively, with vowel-initial suffixes (as shown in the figure), but they surface as [baya] and [rompaʔ] with consonant-initial ones and when unsuffixed (i.e., coming out of the start state 0).

Likewise, there are five distinct stem states because of the prefix allomorphy: one each for the subclasses of stems beginning with labials, sonorants, alveolars, palatals, and velars+vowels (only the first two of these are shown in the figure as states 1 and 2, respectively). So the nine states in total include: two suffix states, five stem states, a prefix state (state 4), and the start state. Note also that the remaining sonorant-initial stems (namely, PLAY, ASCEND, OPEN, and SING) all also follow the transition into state 2, but have been omitted from the transition label for readability.

To begin the construction of the lexicon and phonology, any morpheme that does not alternate—meaning any morpheme that has the same form on all transitions on which

[19]The complete FST is provided as a list of transitions in Appendix A.1.

it appears—is taken to be its own UR and placed in the lexicon. In this case, this initial lexicon of non-alternating morphemes includes the ones listed in Table 5:

Initial lexicon	
CAUSE	/i/
BEN	/kan/
NOMS	/an/
TIE	/ikat/
INSTIGATE	/asut/
DIG	/gali/
EMBRACE	/dakap/
FARM	/ladaŋ/
PLAY	/main/
SING	/ɲaɲi/

Table 5: Initial lexicon of URs: non-alternating morphemes.

For the morphemes that do alternate, we now follow the procedure outlined in 3.4: gather the relevant 1-suffixes for each variant and assess the natural class status of each set. We will demonstrate this procedure using the active voice morpheme, /məŋ/. For each of its surface variants, the learner gathers all of the transitions that lead *into* the state from which the variant is observed. For example, in Figure 10 the variant [məm] is observed on the transition that leaves state 1. We then gather all of the *incoming* transitions for state 1, which are listed in Table 6. (In addition to the four transitions depicted in Figure 10, this listing includes the two transitions from the consonant-initial suffix state, 5.)

State	Input	Output	State
0	PAY	baya	1
0	CARRY	bawə	1
3	PAY	bayar	1
3	CARRY	bawa	1
5	PAY	baya	1
5	CARRY	bawa	1

Table 6: Transitions leading into state 1 in Figure 10.

From here we gather all of the output strings from this set of transitions—{baya, bawə, bayar, bawa}—and then take the 1-suffixes of this set of strings: {b}. Importantly, because the FST in this case reads strings right-to-left, the 1-suffix of a string is its *first* segment, not its last. We repeat this procedure for all of the morpheme's variants, as summarized in Table 7.

Variant	Output strings	1-suffixes	Natural class?
[məŋ]	{ikat, asut, gali}	{i, a, g}	✗
[mən]	{dakap}	{d}	✓
[məm]	{baya, bawə, bayar, bawa}	{b}	✓
[məɲ]	{jawap, cərcə, jawab, cərca}	{c, j}	✓
[mə]	{ladaŋ, naiʔ, rompaʔ, ŋaŋə, main, ɲaɲi, naik, rompak, ŋaŋa}	{ɲ, ŋ, l, m, n, r}	✓

Table 7: 1-suffixes for all variants of active voice prefix.

The learner then determines whether each of these sets of 1-suffixes constitutes a natural class. Our definition of a natural class—inspired by Bale and Reiss (2018)'s use of set theory—is as follows: a set of segments is a natural class if there exists a feature specification (i.e., set of valued features) that is shared by all and only those segments. As shown in the last column of Table 7, this definition is met by all of the variants' sets of 1-suffixes *except* for that of məŋ-. This variant is then selected as the UR.

With the UR selected, the phonological changes to derive the other variants from that UR can be identified following the procedure described in §3.5 (again reversed since the learner is reading right-to-left). The complete list of transitions updated based on the analysis of the active voice prefix is shown in Table 8. Of course, the analysis of the nominalizing prefix will proceed in the exact same way, yielding the same changes and the selection of /pəŋ/ as the UR.

State	Input	Output	State
d	ŋ	n	ŋ
b	ŋ	m	ŋ
c	ŋ	ɲ	ŋ
j	ŋ	ɲ	ŋ
ɲ	ŋ	λ	ŋ
ŋ	ŋ	λ	ŋ
l	ŋ	λ	ŋ
m	ŋ	λ	ŋ
n	ŋ	λ	ŋ
r	ŋ	λ	ŋ

Table 8: Transitions updated in the phonology FST after the analysis of the active voice prefix /məŋ/.

This same procedure handles the other patterns reflected in the data. For example, consider the morpheme CARRY, which is subject to /a/-raising and therefore has two surface forms, [bawa] and [bawə], depending on whether or not a suffix follows it. Gathering the transitions and 1-suffixes as described above reveals that the contexts for these two SRs are {i, a, k} and {#}, respectively, where # indicates that a form is observed coming out of the start state. Comparing these, we note that {i, a, k} is not a natural class, and so /bawa/ is selected as the UR and the phonological mapping is updated to reflect the change of /a/ to [ə] out of the start state (i.e., word-finally).

The learner proceeds in this fashion through all of the stems with multiple SRs, culminating in the final lexicon from Tables 3 and 4 and the set of phonological changes listed in Figure 11. (These changes are listed in the familiar rule format, but again, the phonology is a single FST, not an ordered sequence.)

$$\begin{array}{l}
\text{ŋ} \rightarrow \text{m} \ / \ __\ \text{b} \\
\text{ŋ} \rightarrow \text{n} \ / \ __\ \text{d} \\
\text{ŋ} \rightarrow \text{ɲ} \ / \ __\ \{\text{c}, \text{j}\} \\
\text{ŋ} \rightarrow \emptyset \ / \ __\ \{\text{m}, \text{n}, \text{r}, \text{l}, \text{ŋ}, \text{ɲ}\} \\
\text{k} \rightarrow \text{ʔ} \ / \ __\ \{\text{k}, \#\} \\
\text{r} \rightarrow \emptyset \ / \ __\ \{\text{k}, \#\} \\
\text{a} \rightarrow \text{ə} \ / \ __\ \# \\
\text{b} \rightarrow \text{p} \ / \ __\ \{\text{k}, \#\}
\end{array}$$

Figure 11: Phonological changes detected by the learner.

In sum, the learner finds the URs even in the presence of multiple changes in multiple contexts, including the opaque interaction of /a/-raising and /r/-deletion mentioned previously. Because the phonology is enacted as a single map that operates directly on the underlying form, the counterfeeding exhibited by /bakar/ → [baka], 'to burn' is accommodated for free, in the spirit of the Direct Mapping Hypothesis (Kenstowicz and Kisseberth, 1977, 1979).[20]

The natural class heuristic demonstrated here succeeds when exactly one surface variant appears with a set of contexts that are not a natural class, a condition that will obviously not be met in all cases. Anticipating this, in the next section we describe an additional mechanism—hypothesis testing—that the learner can drawn on when the natural class heuristic fails to select a UR.

4.2.1 Hypothesis testing

A number of alternations in our test case depend on whether or not a suffix is attached to a stem, and if so, whether that suffix is vowel- or consonant-initial. Given the limited suffix inventory, this meant the competing sets of 1-suffixes are {i, a} (from *-i* and *-an*) and {k, #} (from *-kan* and no suffix). In such cases, the target UR is the variant that appears before {i, a}. Given the vowel inventory (Table 2), these two vowels are indeed not a natural class, and so the learner selects the correct UR.

However, it's fairly intuitive that such patterns are not about these two particular vowels, but instead whether the stem-final consonant is in onset or coda position (e.g., for ANSWER, compare [jawaban] and [jawapkan]). In a language with a wider array of suffixes, then, the learner could easily find itself in a position in which the natural class heuristic is indecisive (i.e., either both or neither sets of 1-suffixes are natural classes).

To address this, we now describe an additional mechanism that the learner can draw on in such cases. For demonstration purposes, we'll just force the learner to treat {i, a} as a

[20]Other types of opacity—namely those cataloged in Baković (2007)—have been shown by Chandlee et al. (2018) to also be ISL, though some cases require a larger k-value than the individual processes. As for transparent interactions like feeding and bleeding, these are predicted to be instead *output* strictly local (OSL) (Chandlee, 2021a). We will say more about non-ISL/ISL_2 processes in §6.

natural class. This affects the decision procedure for four of the verbal stems: PAY (baya, bayar), ASCEND (naik, naiʔ), ANSWER (jawab, jawap), and ROB (rompak, rompaʔ). For each stem, the learner takes each surface variant in turn as a hypothesized UR, and identifies the phonological changes that would be necessary under that hypothesis. For example, for morpheme PAY, if [baya] is the UR, then the rule in (18a) is necessary to generate the SRs [bayari] and [bayaran].

(18) UR: /baya/
  a. $\emptyset \rightarrow$ r / __ {a, i}

And likewise, if [bayar] is the UR, then the rule in (19a) is necessary to generate the SRs [baya] and [bayakan].

(19) UR: /bayar/
  a. r $\rightarrow \emptyset$ / __ {k, #}

The learner tests these two hypotheses using the lexicon it has constructed so far, looking for contradictions in its dataset of SRs.[21] In this case, it finds a contradiction for rule (18a), which would wrongly map the UR for TIE, /ikat/, to the SR *[rikrat]. This hypothesis can then be rejected in favor of the second one, which does not encounter any contradictions. A similar analysis will lead to the selection of /jawab/ as the UR for ANSWER, as the voicing rule that would be needed to map underlying /jawap-an/ to [jawaban] would incorrectly map /dakap-an/, EMBRACE, to *[dakaban].

However, the /k/-glottalization process affecting the other two stems—ASCEND (naik, naiʔ) and ROB (rompak, rompaʔ)—still presents a challenge. The hypothesis that needs to be rejected, shown in (20), will not encounter any contradictions in the data (and of course, neither will the correct hypothesis).

(20) URs: /rompaʔ/, /naiʔ/
  a. ʔ $\rightarrow$ k / __ {i, a}

But the reason this rule will not generate any ungrammatical forms is because it won't ever apply: no URs in the lexicon will include glottal stop, because it is not phonemic. For now we can address this hurdle by assuming the learner will reject outright any hypothesized UR that includes non-phonemes, but this example does raise interesting questions about how and when knowledge of the phoneme inventory (i.e., contrast) can be brought to bear on the learning of URs. More generally, this example illustrates the limits of our current UR selection procedure and a need to equip the learner with additional sources of information that can be brought to bear on that task. We will say more about this in the next section.

[21] Tesar and Prince (2007) employ a similar strategy for the learning of URs and an OT grammar. See also Bale and Reiss (2018)'s use of *modus tollendo ponens* reasoning for phonological analysis.

4.3 Epenthesis

To conclude the demonstration, we briefly show how the learner treats epenthesis processes, a notable omission from our case study so far. As shown in (21a) and (21b), epenthesis is employed in Johor Malay as a repair for vowel hiatus; if the first vowel is high, the corresponding glide is epenthesized; otherwise ((21c) and (21d)) it's a glottal stop.

(21) Epenthesis for vowel hiatus

a.	/bantu-an/	[bantuwan]	'aid, relief'
b.	/tari-an/	[tariyan]	'dance'
c.	/məŋ-gula-i/	[məŋgulaʔi]	'to cause to sweeten'
d.	/pəŋ-buka-an/	[pəmbukaʔan]	'opening'

When expressed in rule form, epenthesis to break up vowel hiatus differs from the other patterns we have dealt with in that it has a split context. This can be seen by comparing the word-final devoicing and vowel hiatus epenthesis rules in (22).

(22) a. $\mathrm{b} \rightarrow \mathrm{p}\ /\ \_\_\ \#$
b. $\emptyset \rightarrow \mathrm{w}\ /\ \mathrm{u}\ \_\_\ \mathrm{a}$

Despite this apparent difference, however, the structural descriptions of both rules—b# and ua, respectively—are of length 2 and so are both ISL_2. This is because the target of epenthesis is empty. Though it's not the convention, if we express epenthesis instead as in (23), its formal equivalence to (22a) and all of the rules found by the learner (Fig. 11) becomes more evident.

(23) $\mathrm{u} \rightarrow \mathrm{uw}\ /\ \_\_\ \mathrm{a}$

Epenthesis itself is therefore within reach of our learner, though it does treat it as a special case in which it must recover the 'target' segment from the end of the verb stem. However, epenthesis in Johor Malay does raise some challenges for our current UR selection procedure. Consider example (21a): the learner will be faced with a choice between [bantu] and [bantuw]. The necessary rules that correspond to these two UR hypotheses are shown in (24) and (25), respectively.

(24) UR: /bantu/
a. $\emptyset \rightarrow \mathrm{w}\ /\ \mathrm{u}\ \_\_\ \{\mathrm{i},\ \mathrm{a}\}$

(25) UR: /bantuw/
a. $\mathrm{w} \rightarrow \emptyset\ /\ \_\_\ \{\mathrm{k},\ \#\}$

The desired hypothesis, (24), will of course not find any contradictions in the dataset. But neither will (25), because no URs end in glides. The absence of such forms in the lexicon is not due to our particular data selection, but to a broader absence of glides in syllable-final position throughout the language (Onn, 1976, pg. 109, fn. 12). Perhaps after a sufficient lexicon has been constructed, the learner could begin to posit something akin to morpheme structure conditions to rule out unfalsifiable hypotheses like (25).

As for glottal epenthesis, because it affects stems that end in /a/ and those same stems are subject to word-final /a/-raising, the learner has to choose from three surface variants, [bawaʔ], [bawə], and [bawa], the last of which is the target UR. The first can be rejected because it contains the unparseable glottal stop, as discussed above. However, [bawa] will also be rejected, because its corresponding epenthesis rule ($\emptyset \rightarrow$ ʔ / a __ i) will wrongly apply to words like /main/. This is nonderived environment blocking, which is not a problem for ISL provided the representations include morpheme boundaries to distinguish derived from nonderived environments (see Chandlee, 2021b).[22]

Both types of epenthesis then point directly to the kinds of additional information the learner may need to draw on during UR selection. We leave an exploration of these potential solutions for future work, along with a number of other planned expansions that we will discuss in §6. But first, §5 situates the current results in the context of previous computational models of UR learning.

5 Previous work on UR learning

Foundational work on learning OT constraint rankings from (UR, SR) pairs (Tesar, 1995; Tesar and Smolensky, 1993, 1996, 1998, 2000) was later extended to include the learning of the lexicon of URs by Tesar et al. (2003) and Merchant (2008), as well as in Tesar (2014)'s work on output-driven maps. As in the present work, Tesar's Output-Driven Learner (ODL) capitalizes on the assumption that the target phonological map has a property that is independent of the grammatical formalism chosen to represent it intensionally. The output-driven property provides the following entailment relation: if A is mapped to X, and B is more similar to X than A is, B must also be mapped to X. Similarity here refers to the number of feature differences, but it was later extended to include insertion and deletion by Nyman and Tesar (2019). Following Tesar and Prince (2007), the ODL first establishes a preliminary ranking using phonotactics alone before drawing on information from alternations to both refine the ranking and identify the URs in an error-driven feedback loop. At each stage, the assumption of output-drivenness enables the learner to eliminate a great many hypotheses and efficiently converge on the combination of lexicon and constraint ranking that accounts for the observed surface forms.

Progress has also been made on learning constraint-based grammars using probabilistic approaches that identify the lexicon and grammar combination that maximizes the likelihood of the training data, using Expectation-Maximization and/or Maximum Entropy (Jarosz, 2006b,a, 2013; O'Hara, 2017; Wang and Hayes, 2022). In the course of learning, these approaches consider all possible URs in order to find the most likely one, in some cases by making use of lexical or UR constraints that either require or prohibit language-particular morpheme-UR pairings (Apoussidou, 2007; Pater et al., 2012; Nelson, 2019).

One commonality across constraint-based approaches to UR learning is that the learner is provided with a constraint set that corresponds to the patterns present in the data. In contrast, a goal of the present work is to determine how much learning can take place when

[22] Glide insertion is not subject to nonderived environment blocking, as it also applies within morphemes, e.g., [buwah] from underlying /buah/, 'fruit'.

the learner is given only the formal properties of these patterns. By taking advantage of this formal structure of the hypothesis space of possible maps—instead of the space of possible grammars—we aim to understand something about the phonological learning problem that is independent of the choice of grammatical formalism (i.e., rules, constraints, or something else).

Here we anticipate the objection that the finite-state formalism is itself an intensional description of the target map. We concur with that observation, but note that the use of finite-state representations is ultimately just an implementation choice. State merging as a generalization strategy has many precedents in the grammatical inference literature (de la Higuera, 2010; Heinz et al., 2016; Heinz and Sempere, 2016), and we are taking advantage of that foundation in order to make progress on the challenging problem of UR learning. Importantly, though, the formal *properties* (e.g., subsequentiality, strict locality) exploited by our learner are not in fact dependent on finite-state but have equivalent and converging characterizations in other formalisms including logic and algebra (Chandlee and Jardine, 2019; Bhaskar et al., 2020; Lambert, 2022), and there are some promising developments in learning with logical formalisms (Yolyan, 2025). Synthesizing the present results with such work will be an important avenue for future research, as logical characterizations of processes are more flexible with respect to the role of directionality and the choice of representation (i.e., linear or non-linear).

Other prominent examples of UR and grammar learning include the use of Minimum Description Length (MDL) (Rasin et al., 2020, 2021) and Bayesian approaches (Cotterell et al., 2015; Barke et al., 2019; Ellis et al., 2022). The MDL approach converges on the lexicon and grammar combination that is most economical in terms of the size of the grammar plus the encoding of the data according to that grammar. While it handles an impressive array of learning challenges, the rapid growth of the hypothesis space and the time needed to search it has limited its demonstration to toy examples. The Bayesian approaches face similar challenges with the tractability of their search procedures, employing various heuristics or restrictions on the type or number of rules the grammar will contain—for example, Barke et al. (2019) limit their rules to ISL_3. The current proposal instead takes typologically-motivated formal restrictions on possible phonological functions as the starting point in order to capitalize on the way such properties structure the hypothesis space. As discussed in §3.6, it is also grounded on an algorithm that is provably correct, as Hua and Jardine (2021) give a detailed analysis of the behavior of the $\mathrm{SI}_2\mathrm{DLA}$ and prove the conditions under which it will successfully converge to the target grammar.

Lastly, Belth (2023a) explores the learning of abstract URs. Working from morphologically-analyzed SRs, the learner initially lists each SR as its own lexicalized form, using the Tolerance Principle (Yang, 2016) as a cue for when to abstract over observed variants. Once the lexicon is compiled, a separate module learns the phonological mapping. Our learner instead interleaves lexicon construction and phonological learning in order to explore how these might interact with and inform each other over the course of learning. Nonetheless, the potential for learning abstract URs may be an important extension for our learner as well, which in its current form adheres to the *basic alternant* assumption that the UR is one of the observed SRs (Kenstowicz and Kisseberth, 1979).

6 Future directions

In addition to further developing the UR selection procedure as discussed above, there are a number of remaining limitations of the learner that will need to be addressed to make it a more general model of phonological learning. This section discusses those areas of future work.

6.1 Directionality

One pattern that was described in §4 that has not been accounted for yet is the combined nasal assimilation and deletion pattern that affects the prefixes /məŋ-/ and /pəŋ-/ when they attach to stems that begin with a voiceless obstruent (examples repeated from (10)):

(26) Nasal assimilation and deletion before voiceless obstruents

/pəŋ-pandu/	[pəmandu]	'a guide'
/pəŋ-karaŋ/	[pəŋaraŋ]	'author'
/pəŋ-samun/	[pəɲamon]	'robber'
/pəŋ-tari/	[pənari]	'dancer'

This pattern has been called *fusion* or *coalescence* (Lapoliwa, 1981; Pater, 2004), though for Onn (1976) it is the result of two ordered rules (nasal assimilation and deletion). In our model, the phonological grammar is a single function that maps inputs to outputs in a single step. For this pattern in particular, the mappings it has to enact are the ones listed in (27).

(27)
a. /ŋp/ → [m]
b. /ŋt/ → [n]
c. /ŋs/ → [ɲ]
d. /ŋk/ → [ŋ]

Formally, these mappings are ISL_2 (i.e., the input side is of length 2) and so are consistent with our assumption that the phonological grammar is an ISL_2 function. However, our learner does encounter an issue because of the *bidirectional* nature of this pattern: the nasal assimilates to a *following* obstruent, but the obstruent deletes in response to a *preceding* nasal. The current learner, however, is limited to looking in only one direction at a time (either leftward or rightward, set by parameter) when determining the contexts of patterns. Removing this limitation is necessary not just for languages with fusion but for those with any combination of patterns in both directions.

Importantly, though, ISL functions are in fact adirectional, which means for any ISL function described with a left-to-right FST there exists a right-to-left FST that will describe the same function. So the need for both directions is not inherently problematic for our assumption that the phonological grammar is an ISL function. This limitation of the current learner is an artifact of the SI_2DLA on which it is based and will be removed when we generalize further from the simplex functions assumed by that algorithm to the broader ISL class. We discuss this area for future work further in the next section.

6.2 Beyond ISL_2

As we noted at the outset (§2), the assumption that the phonology is an ISL_2 function is overly strong, as there is no shortage of examples of phonological patterns with structural descriptions of length greater than 2. Johor Malay itself includes several such examples, including vowel lowering and vowel lengthening:

(28) Vowel lowering (Onn, 1976, p.30)
  a. $\text{V}_{+hi} \rightarrow$ [−hi] / __C {#, C}
  b. /milik/ → [mileʔ], 'to own/possess'
  c. /pilih/ → [pileh], 'to choose'

(29) Vowel lengthening (Onn, 1976, p.59)
  a. V → [+long] / __ [+nasal][+cons]
  b. /tombaŋ/ → [toːmbaŋ], 'to fall'
  c. /gurindam/ → [guriːndam], 'poetry'

The structural descriptions of both of these patterns are of length 3, which means an ISL_2 grammar would overgenerate by lowering vowels before single consonants and lengthening them before all nasals (e.g., *[meleʔ] and *[toːmbaːŋ]). Initializing an ISL_3 phonological grammar instead is a trivial change, but the broader question is how to extract the correct generalizations from the output of state merging. In particular, how can the learner make use of sets of 2-suffixes for 3-local mappings like these, or more generally $(k-1)$-suffixes for ISL_k? In addition, rather than setting k as a parameter, how can the learner detect automatically that a particular value of k is insufficient and needs to be increased to correctly learn the lexicon and grammar? These are crucial open questions that are being addressed in current work.

Nonetheless, the existence of these open questions should not detract from the significance of the formal learnability results for SI_2DLA as well as the empirical coverage of the current learner. These represent great strides forward on the phonological learning problem as defined in this paper. It is therefore not unreasonable to assume that the next step of generalizing to all k is well within reach.[23]

Of course the other crucial aspect of the locality assumption for the phonological grammar is that it is specifically an *input* local function. What about languages with patterns in other formally-local classes, namely Output Strictly Local (OSL; Chandlee et al., 2015) functions (for iterative patterns) or the tier-based strictly local functions (TSL; Burness et al., 2021; Burness, 2022) (for long-distance patterns)? We turn to this question in the next section.

6.3 Beyond ISL

A primary goal of this paper has been to show how a lexicon of URs can be learned in tandem with a phonological grammar given a simple set of starting assumptions, one of which is that the grammar is an ISL_2 function. A natural extension of this work will be

[23] As a precedent we cite the previous work by Jardine and Heinz (2016) on learning tier-based strictly local languages (including the tier) under the assumption that $k = 2$, a result that was soon after generalized to k by Jardine and McMullin (2017).

to likewise show how learning can proceed when the grammar is instead an OSL or TSL function. Once we have investigated the learning problem under all of these conditions, we will be in a position to address the more realistic condition in which the language being learned includes processes in all of these categories. Looking ahead, then, we envision a factored learner with multiple, interacting functions comprising the phonological grammar. Crucial open areas of future work include investigating the nature of those function interactions as well as the cues in the data that will signal to the learner which function is appropriate for each pattern that is represented.[24]

The current results are then just the first step in a larger research project currently underway, one built around the central idea of function decomposition and the formalized notion of phonological locality that is captured by (I/O/T)SL.

6.4 The role of features

The current learner is provided with a feature chart, which amounts to an assumption of innate features. However, we do not wish to take here a strong position as to what extent features are specified by UG and to what extent they must also be learned as a language-specific part of the grammar (per the theories of, e.g., Mielke (2008) or Dresher (2009)). The goal, rather, has been to show how natural class information—however it is obtained—can be used to help solve the learning problem. However, we note that at least some aspects of contrast can be induced from distributional information (Goldsmith and Riggle, 2012). The current learner thus opens up avenues for studying how feature learning can interact with other learning problems.

In addition, recent work by Markowska and Heinz (2023) has explored ways of using feature-based generalization with a finite-state learner, which is another useful route for reducing the amount of data that the learner needs to identify the patterns at the right level of generality.

7 Conclusion

Understanding the formal properties of phonological grammars is an important step in understanding how they are learned. While Hua and Jardine (2021) define the SI_2DLA and prove its correctness in abstract terms, the result is almost directly applicable to the phonological learning problem as defined in this paper. Limitations remain, but as Chomsky (1957, pg. 5) writes, "By pushing a precise but inadequate formulation to an unacceptable conclusion, we can often expose the exact source of this inadequacy and, consequently, gain a deeper understanding of the linguistic data."[25] Put another way, this paper has shown how much progress on the morpho-phonological learning problem can be made by starting from very simple assumptions. This results in a transparent algorithm that establishes a clear path forward for further progress on the problem of how children acquire phonological grammars.

[24] An example of such a cue would be the presence of irreparable non-determinism in the output of OSTIA in the event the wrong function type is assumed. This same cue may assist with the identification of the right k-value for a function as mentioned in the previous section.

[25] We thank Charles Reiss for pointing this quote out to us.

A Appendix

A.1 Complete FST output by the OSTIA-D for Johor Malay test case

State labels can be interpreted as follows:

- 0 = start state
- 1 = stems whose phonological form begins with a bilabial
- 2 = stems whose phonological form begins with a sonorant consonant
- 3 = vowel-initial suffixes
- 4 = prefixes
- 5 = consonant-initial suffix (-*kan*)
- 6 = stems whose phonological form begins with a palatal
- 7 = stems whose phonological form begins with an alveolar
- 8 = stems whose phonological form begins with a velar or a vowel

State	Input	Output	State
0	TIE	ikat	8
0	CAUSE	i	3
0	BEN	kan	5
0	NOMS	an	3
0	INSTIGATE	asut	8
0	DIG	gali	8
0	PAY	baya	1
0	CARRY	bawə	1
0	EMBRACE	dakap	7
0	FARM	ladaŋ	2
0	ASCEND	naiʔ	2
0	ANSWER	jawap	6
0	REVILE	cərcə	6
0	ROB	rompaʔ	2
0	OPEN	ŋaŋə	2
0	PLAY	main	2
0	SING	ɲaɲi	2
8	ACT	məŋ	4
8	NOMP	pəŋ	4
3	TIE	ikat	8
3	INSTIGATE	asut	8
3	DIG	gali	8
3	PAY	bayar	1
3	CARRY	bawa	1
3	EMBRACE	dakap	7
3	FARM	ladaŋ	2
3	ASCEND	naik	2
3	ANSWER	jawab	6
3	REVILE	cərca	6
3	ROB	rompak	2
3	OPEN	ŋaŋa	2
3	PLAY	main	2
3	SING	ɲaɲi	2
5	TIE	ikat	8
5	INSTIGATE	asut	8
5	DIG	gali	8
5	PAY	baya	1
5	CARRY	bawa	1
5	EMBRACE	dakap	7
5	FARM	ladaŋ	2
5	ASCEND	naiʔ	2
5	ANSWER	jawap	6
5	REVILE	cərca	6
5	ROB	rompaʔ	2
5	OPEN	ŋaŋa	2
5	PLAY	main	2
5	SING	ɲaɲi	2
1	NOMP	pəm	4
1	ACT	məm	4
7	NOMP	pən	4
7	ACT	mən	4
2	NOMP	pə	4
2	ACT	mə	4
6	NOMP	pəɲ	4
6	ACT	məɲ	4

A.2 Feature chart used for natural class heuristic

From Onn (1976, pg. 40), with the addition of a 'segment' feature.

	Syl	Cons	Son	Nas	Bk	Fr	Hi	Rd	Lo	Cont	Ant	Cor	Str	Voi	DelRel	Seg
b	-	+	-	-	-	+	-	-	-	-	+	-	-	+	-	+
d	-	+	-	-	-	+	-	-	-	-	+	+	-	+	-	+
g	-	+	-	-	+	-	+	-	-	-	-	-	-	+	-	+
p	-	+	-	-	-	+	-	-	-	-	+	-	-	-	-	+
t	-	+	-	-	-	+	-	-	-	-	+	+	-	-	-	+
k	-	+	-	-	+	-	+	-	-	-	-	-	-	-	-	+
j	-	+	-	-	-	-	+	-	-	-	-	+	+	+	+	+
c	-	+	-	-	-	-	+	-	-	-	-	+	+	-	+	+
s	-	+	-	-	-	-	-	-	-	+	+	+	+	-	-	+
l	-	+	+	-	-	-	-	-	-	+	+	+	-	+	-	+
r	-	+	+	-	+	-	-	-	-	+	-	-	-	+	-	+
m	-	+	+	+	-	+	-	-	-	-	+	-	-	+	-	+
n	-	+	+	+	-	+	-	-	-	-	+	+	-	+	-	+
ɲ	-	+	+	+	-	-	+	-	-	-	-	+	-	+	-	+
ŋ	-	+	+	+	+	-	+	-	-	-	-	-	-	+	-	+
w	-	-	+	-	+	-	+	-	-	+	-	-	-	+	-	+
y	-	-	+	-	-	-	+	-	-	+	-	-	-	+	-	+
h	-	-	+	-	-	-	-	-	+	+	-	-	-	+	-	+
i	+	-	+	-	-	+	+	-	-	+	-	-	-	+	-	+
e	+	-	+	-	-	+	-	-	-	+	-	-	-	+	-	+
u	+	-	+	-	+	-	+	+	-	+	-	-	-	+	-	+
o	+	-	+	-	+	-	-	+	-	+	-	-	-	+	-	+
a	+	-	+	-	-	-	-	-	+	+	-	-	-	+	-	+
ə	+	-	+	-	-	-	-	-	-	+	-	-	-	+	-	+
#	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

References

Anttila, A. (2007). Variation and optionality. In Lacy, P. D., editor, *The Cambridge Handbook of Phonology*, chapter 22, pages 519–536. Cambridge University Press.

Apoussidou, D. (2007). *The Learnability of Metrical Phonology*. PhD thesis, University of Amsterdam.

Baković, E. (2007). A revised typology of opaque generalisations. *Phonology*, 24(2):217–259.

Bale, A. and Reiss, C. (2018). *Phonology: A Formal Introduction*. MIT Press.

Barke, S., Kunkel, R., Polikarpova, N., Meinhardt, E., Baković', E., and Bergen, L. (2019). Constraint-based learning of phonological processes. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (emnlp-ijcnlp)*, pages 6176–6186.

Belth, C. (2023a). Towards a learning-based account of underlying forms: A case study in Turkish. In *Proceedings of the Society for Computation in Linguistics*, volume 6. Article 31. https://scholarworks.umass.edu/scil/vol6/iss1/31.

Belth, C. A. (2023b). A learning-based account of local phonological processes. *Phonology*, 40(1–2):1–33.

Beros, A. and de la Higuera, C. (2016). A canonical semi-deterministic transducer. *Fundamenta Informaticae*, pages 431–459.

Bhaskar, S., Chandlee, J., Jardine, A., and Oakden, C. (2020). Boolean monadic recursive schemes as a logical characterization of the subsequential functions. In Leporati, editor, *Language and Automata Theory and Applications, 14th International Conference*, volume 12038, pages 157–169. Lecture Notes in Computer Science.

Burness, P. (2022). *Non-local phonological processes as Multi-Tiered Strictly Local maps*. PhD thesis, University of Ottawa.

Burness, P. A., McMullin, K. J., and Chandlee, J. (2021). Long-distance phonological processes as tier-based strictly local functions. *Glossa: a journal of general linguistics*, 6(1):99.

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

Chandlee, J. (2021a). Modulating between input and output locality: A case study on phonological opacity. In Özge Bakay, Breanna Pratley, E. N. and Deal, P., editors, *NELS 52: Proceedings of the Fifty-Second Annual Meeting of the North East Linguistic Society*, volume 1, pages 119–138, Amherst, MA. GLSA.

Chandlee, J. (2021b). Nonderived environment blocking and input-oriented computation. *Evolutionary Linguistic Theory*, 3(2):129–153.

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

Chandlee, J., Eyraud, R., and Heinz, J. (2015). Output strictly local functions. In Kornai, A. and Kuhlmann, M., editors, *Proceedings of the 14th Meeting on the Mathematics of Language (MoL 14)*, pages 52–63, Chicago, IL.

Chandlee, J. and Heinz, J. (2018). Strict locality and phonological maps. *LI*, 49:23–60.

Chandlee, J., Heinz, J., and Jardine, A. (2018). Input Strictly Local opaque maps. *Phonology*, 35:1–35.

Chandlee, J. and Jardine, A. (2019). Quantifier-free least-fixed point functions for phonology. In *Proceedings of MOL 16*, pages 50–62.

Chomsky, N. (1957). *Syntactic Structures*. Mouton de Gruyter, Berlin/New York.

Cotterell, R., Peng, N., and Eisner, J. (2015). Modeling word forms using latent underlying morphs and phonology. *Transactions of the Association for Computational Linguistics*, 3:433–447.

de la Higuera, C. (2010). *Grammatical inference: learning automata and grammars*. Cambridge UP.

Dresher, B. E. (2009). *The Contrastive Hierarchy in Phonology*. Cambridge: Cambridge University Press.

Ellis, K., Albright, A., Solar-Lezama, A., Tenenbaum, J. B., and O'Donnell, T. J. (2022). Synthesizing theories of human language with bayesian program induction. *Nature communications*, 13(1):5024.

Goldsmith, J. and Riggle, J. (2012). Information theoretic approaches to phonological structure: the case of Finnish vowel harmony. *NLLT*, 30(3):859–896.

Hale, M. and Reiss, C. (1998). Formal and empirical arguments concering phonological acquisition. *Linguistic Inquiry*, 29(4):656–683.

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

Heinz, J. (2020). Deterministic analyses of optional processes. Talk given at University of Leipzig.

Heinz, J., de la Higuera, C., and van Zaanen, M. (2016). *Grammatical Inference for Computational Linguistics*. Morgan and Claypool.

Heinz, J. and Sempere, J. M. (2016). *Topics in Grammatical Inference*. Springer.

Hua, W., Dai, H., and Jardine, A. (2021). Learning underlying representations and input-strictly-local functions. In Reisinger, D. K. E. and Huijsmans, M., editors, *Proceedings of the 37th West Coast Conference on Formal Linguistics*, pages 143–151. Cascadilla Proceedings Project.

Hua, W. and Jardine, A. (2021). Learning input strictly local functions from their composition. In Chandlee, J., Eyraud, R., Heinz, J., Jardine, A., and van Zaanen, M., editors, *Proceedings of the Fifteenth International Conference on Grammatical Inference*, volume 153 of *Proceedings of Machine Learning Research*, pages 47–65. PMLR.

Jardine, A. and Heinz, J. (2016). Learning tier-based strictly 2-local languages. *Transactions of the Association for Computational Linguistics*, 4:87–98.

Jardine, A. and McMullin, K. (2017). Efficient learning of tier-based strictly k-local languages. In Drewes, F., Martín-Vide, C., and Truthe, B., editors, *Language and Automata Theory and Applications, 11th International Conference*, Lecture Notes in Computer Science, pages 64–76. Springer.

Jarosz, G. (2006a). *Rich Lexicons and Restrictive Grammars: Maximum Likelihood Learning in Optimality Theory*. PhD thesis, Johns Hopkins University.

Jarosz, G. (2006b). Richness of the base and probabilistic unsupervised learning in Optimality Theory. In *Proceedings of the Eighth Meeting of the ACL Special Interest Group in Computational Phonology*. Association for Computational Linguistics.

Jarosz, G. (2013). Learning with hidden structure in Optimality Theory and harmonic grammar: beyond Robust Interpretive Parsing. *Phonology*, 30:27–71.

Kenstowicz, M. and Kisseberth, C. (1977). *Topics in Phonological Theory*. New York: Academic Press, Inc.

Kenstowicz, M. and Kisseberth, C. (1979). *Generative Phonology: Description and Theory*. New York: Academic Press, Inc.

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

Lapoliwa, H. (1981). *A generative approach to the phonology of Bahasa Indonesia*. Materials in Languages of Indonesia, No. 3. Pacific Linguistics, The Australian National University.

Markowska, M. and Heinz, J. (2023). Empirical and theoretical arguments for using properties of letters for the learning of sequential functions. In Coste, F., Ouardi, F., and Rabusseau, G., editors, *Proceedings of 16th edition of the International Conference on Grammatical Inference*, volume 217 of *Proceedings of Machine Learning Research*, pages 270–274. PMLR.

Merchant, N. N. (2008). *Discovering Underlying Forms: Contrast Pairs and Ranking*. PhD thesis, Rutgers.

Mielke, J. (2008). *The emergence of distinctive features*. Oxford: Oxford University Press.

Mohri, M. (1997). Finite-state transducers in language and speech processing. *Computational Linguistics*, 23(2):269–311.

Nelson, M. (2019). Segmentation and ur acquisition with ur constraints. In *Proceedings of the Society for Computation in Linguistics*, volume 2. Article 8.

Nyman, A. and Tesar, B. (2019). Determining underlying presence in the learning of grammars that allow insertion and deletion. *Glossa: a journal of general linguistics*, 4(1):37.

Oncina, J., García, P., and Vidal, E. (1993). Learning subsequential transducers for pattern recognition tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15:448–458.

Oncina, J. and Varò, M. A. (1996). Using domain information during the learning of a subsequential transducer. *Lecture Notes in Computer Science - Lecture Notes in Artificial Intelligence*, pages 313–325.

Onn, F. M. (1976). *Aspects of Malay Phonology and Morphology*. PhD thesis, University of Illinois at Urbana-Champaign.

Onn, F. M. (1980). *Aspects of Malay Phonology and Morphology: A Generative Approach*. Kuala Lampur: Universiti Kebangsaan Malaysia.

O'Hara, C. (2017). How abstract is more abstract? learning abstract underlying representations. *Phonology*, 34:325—345.

Pater, J. (2004). Austronesian nasal substitution and other NC effects. In McCarthy, J., editor, *Optimality Theory in Phonology: A Reader*, pages 271–289. Oxford and Malden, MA: Blackwell.

Pater, J., Jesney, K., Staubs, R., and Smith, B. (2012). Learning probabilities over underlying representations. In *Proceedings of the 12th Meeting of the Special Interest Group on Computational Morphology and Phonology (SIGMORPHON2012)*, pages 62—71, Stroudsburg, PA. ACL.

Rasin, E., Berger, I., Lan, N., Shefi, I., and Katzir, R. (2021). Approaching explanatory adequacy in phonology using minimum description length. *Journal of Language Modelling*, 9(1):17–66.

Rasin, E., Shefi, I., and Katzir, R. (2020). A unified approach to several learning challenges in phonology. In Asatryan, M., Song, Y., and Whitmal, A., editors, *Proceedings of NELS 50*, volume 3, page 73–87. GLSA.

Sakarovitch, J. (2009). *Elements of Automata Theory*. Cambridge University Press.

Simpson, M. (2010). From alternations to ordered rules: A system for learning derivational phonology. Master's thesis, Concordia University.

Smolensky, P. (1996). On the comprehension/production dilemma in child language. *Linguistic Inquiry*, 27(4):720–731.

Tesar, B. (1995). *Computational Optimality Theory*. PhD thesis, University of Colorado at Boulder.

Tesar, B. (2014). *Output-driven phonology: Theory and learning*. Cambridge University Press.

Tesar, B., Alderete, J., Horwood, G., Merchant, N., Nishitani, K., and Prince, A. S. (2003). Surgery in language learning. In *Proceedings of the Twenty-second West Coast Conference on Formal Linguistics*, pages 477–490, Somerville, MA. Cascadilla Press.

Tesar, B. and Prince, A. (2007). Using phonotactics to learn phonological alternations. In *Proceedings of the Thirty-Ninth Conference of the Chicago Linguistics Society. Vol. II: The Panels*, pages 209–237.

Tesar, B. and Smolensky, P. (1993). The learnability of Optimality Theory: an algorithm and some basic complexity results. Unpublished ms., University of Colorado, Boulder.

Tesar, B. and Smolensky, P. (1996). Learnability in Optimality Theory (long version). Technical Report 96:3, Department of Cognitive Science, Johns Hopkins University.

Tesar, B. and Smolensky, P. (1998). Learnability in Optimality Theory. *LI*, 29:229–268.

Tesar, B. and Smolensky, P. (2000). *Learnability in Optimality Theory*. Cambridge: MIT Press.

Vaux, B. (2008). Why the phonological component must be serial and rule-based. In Vaux, B. and Nevins, A., editors, *Rules, Constraints, and Phonological Phenomena*, pages 20–60. Oxford University Press, Oxford.

Wang, Y. and Hayes, B. (2022). Learning underlying representations: Expectation-maximization and the kk-hierarchy. Poster presented at the Annual Meeting on Phonology. UCLA, Los Angeles, CA. October 2022.

Yang, C. (2016). *The price of linguistic productivity: How children learn to break the rules of language*. MIT press.

Yolyan, T. (2025). *Phonological Complexity and Learning via Boolean Monadic Recursive Schemes*. PhD thesis, Rutgers University.