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 (comments welcome)
August 26, 2024

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 solving 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 (recent example include Tesar, 2014; Cotterell et al., 2015; Rasin et al., 2018; Barke et al., 2019; Rasin et al., 2020).

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. We also demonstrate that this procedure 'scales up' to successfully learn on data from a real language, Johor Malay (Onn, 1976, 1980), that exhibit a number of phonological patterns. This is possible because the learner is attuned to the local nature of phonological processes (Chandlee, 2014; Chandlee and Heinz, 2018), which prevents the hypothesis space from becoming intractably large.

More specifically, building on work by Hua et al. (2021) and Hua and Jardine (2021), this paper conceives of the phonological learning problem in terms of *function decomposition*. The data available to children during acquisition consist of strings of morphemes paired with a phonological SR, e.g., (Cap-Pl, [kæps]). These string pairs 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. Hua and Jardine (2021) present the algorithm called $\mathrm{SI_2DLA}$ (Simplex Input Strictly 2-Local Decomposition Learning Algorithm), which provably solves this problem provided the phonological function belongs to the class of simplex input strictly 2-local ($\mathrm{ISL_2}$) functions, which means it makes a single change in a context of length 1. The current paper expands on their formal result with a modified version of this algorithm that can handle more realistic phonological functions that make multiple changes in multiple contexts simultaneously.

The original algorithm succeeds because of the strength of its assumption that the phonology is simplex ISL_2. This formal property greatly reduces the learner's hypothesis space and structures it in such a way that function decomposition becomes possible. The modified learner's expansion to more realistic problem instances is made possible by incorporating additional sources of information, namely 1) morphological structure to better identify the contexts governing the allomorphy and allophony reflected in the data, and 2) natural classes as a heuristic to identify the UR among the surface variants (i.e., the UR is the one whose set of contexts does *not* constitute a natural class).

Using the case study from Johor Malay, we demonstrate that this modified algorithm can learn 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 how (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.

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 first presents the original SI_2DLA and then explains the current proposed expansions needed for a realistic problem instance. Section 4 demonstrates the learner on a set of morpho-phonological and allophonic rules in Johore Malay (Onn, 1976, 1980). Section 5 situates the current findings in the context of the previous literature on learning URs, and then Section 6 outlines our next steps for scaling the learner up into a more comprehensive model of morpho-phonological acquisition. Section 7 then concludes.

2 The Learning Problem

2.1 In informal terms

At the heart of phonological analysis is the principle that, wherever possible, a morpheme is assigned a single UR. 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). To borrow Kenstowicz and Kisseberth (1977)'s example, the English plural is variously pronounced as [ɨz], [z], and [s], depending on its phonological context.

(1) English plural alternations (adapted from Kenstowicz and Kisseberth (1977, pp. 26–27) and Odden (2005, p. 77))

a.	[kæbz]	'cabs'
	[lædz]	'lads'
	[fla͡ɪz]	'flies'
	[baɹz]	'bars'
b.	[kæps]	'caps'
	[kæts]	'cats'
	[bæks]	'backs'
c.	[bʌsɨz]	'buses'
	[bʊʃɨz]	'bushes'

The standard phonological analysis explains these alternations by assigning the plural morpheme the UR /z/ and positing two phonological processes: one that devoices obstruents following voiceless consonants; and one that inserts [ɨ] in between sibilants.

There are thus, at a high level, two main stages to the production of surface forms: strings[1] of morphemes are assigned URs, and then the phonology applies to those URs to produce the surface forms. (These are the two levels in the two-level morphology of Koskenniemi (1983).) Note that 'morpheme' here refers to the bundle of semantic and/or syntactic information, as distinct from its phonological form. We will notate these abstract forms with all caps, e.g. Pl for plural, Cat for the noun root meaning 'cat,' etc.

The two stages of morphophonology are schematized in Figure 1, with the English words [kæts] 'cats' and [fla͡ɪz] 'flies' as examples.

$$\begin{tikzcd}
\text{Cat-Pl} \arrow[d] \arrow[r, phantom, "\text{UR assignment}"] & \text{Fly-Pl} \arrow[d] \\
\text{/kæt-z/} \arrow[d] \arrow[r, phantom, "\text{phonology}"] & \text{/fla͡ɪ-z/} \arrow[d] \\
\text{[kæts]} & \text{[fla͡ɪz]}
\end{tikzcd}$$

Figure 1: Flow of UR assignment and application of phonology in English [kæts] 'cats' and [fla͡ɪz] 'flies.'

A child learner, however, has no way of directly observing either UR assignment or the phonology of their native language. They do have access to the SRs of their language. They also, at best, have some knowledge of the morphological composition of these surface forms—that is, their semantic and morphosyntactic information. We can schematize the data that a child has access to, then, as strings of morphemes paired with their SRs. For the English data above, for example, the child's input might look something like in (2).

(2) (Cab-Pl, [kæbz])
(Lad-Pl, [lædz])
(Cap-Pl, [kæps])
(Bus-Pl, [bʌsɨz])
(Bush-Pl, [bʊʃɨz])
...

The learning problem, then, is 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; and
2. a phonological grammar;
that are consistent with this set.

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

This is of course an idealization that assumes that the child already has fairly complete information about their morphology, including how many morphemes are present in the SR and the order in which they appear.[2] However, there are several reasons why this problem is worth studying. First, even with this idealization, solving this version of the learning problem is a step towards a more realistic model of phonological learning, as compared to earlier work on learning phonology that assumes the URs are part of the input to the learner (Tesar and Smolensky (2000); Chandlee and Heinz (2018);Belth(in press)). Second, it allows us to study the sub-problems of the problem of simultaneously learning the phonology and the URs. For instance, the child must parse SRs into the surface allomorphs of their constituent morphemes—given (CAB-PL, [kæbz]), is the CAB associated with the string [k], [kæ], [kæb], or [kæbz]? Furthermore, the child 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.

2.2 A formal statement of the learning problem

The two stages in Figure 1 are *functions*. That is, each is a map from 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.[3] The pairs of strings of morphemes and their SRs, as in (2), that we are assuming as the input to the problem are thus drawn from the *composition* $P \circ L$ of L with P. That is, for some string m of morphemes we pair it with $P(L(m))$. For example, if $m =$CAT-PL then we pair it with $P(L(\text{CAT-PL})) = P(/\text{kætz}/) =$[kæts].

The learning problem is thus a *decomposition* problem: given pairs of $P \circ L$, determine L and P. The advantage of posing this problem in this way is that we can ask the question: what kinds of functions are L and P? This will allow us to tailor the learning procedure to the nature of L and P. As for L, because we are assuming one UR per morpheme, L is a *homomorphism*.[4] That is, it maps any individual morpheme to the same UR regardless of where it is in the string of morphemes. For example, in the English L above, L will map any instance of CAT to /kæt/, regardless of what preceded or followed it. Thus, identifying the UR for each individual morpheme is both necessary and sufficient for finding L.

As for P, we draw on the body of work characterizing the computational complexity of phonological processes (Chandlee and Heinz, 2018; Heinz, 2018, et seq.). In particular, we will assume that P is an *input strictly* 2*-local* (ISL_2) function. This means that any change in the string is dependent on the preceding segment in the input string—in more technical terms, its 1*-suffix* (Chandlee, 2014; Chandlee and Heinz, 2018). The above phonology

[2] The assumption that the morphemes have already been identified and ordered is not unique to our approach and has several precedents in the literature on UR learning (Jarosz, 2006a,b; Apoussidou, 2007; Merchant, 2008; Tesar, 2014; Nyman and Tesar, 2019). The rationale of these authors is similar to ours: focusing on the learning of the phonology in isolation is a valuable simplification in the interest of making progress on the very difficult problem of learning both phonology and morphology.

[3] This abstracts away from the problem of optionality and variation (Antilla, 2007; Vaux, 2008), in which a single UR may be mapped to multiple surface forms. However, there are established methods for learning functions mapping strings to sets of strings (Beros and de la Higuera, 2016) that can be applied to the UR learning problem in the future. We will say more about this possibility in §4.3.

[4] More technically, it is a letter-to-string homomorphism. Formally, this means that for any strings of morphemes m and n, $L(m \cdot n) = L(m) \cdot L(n)$, where $\cdot$ indicates concatenation.

function P in the English example is ISL_2: whether /z/ is mapped to [s], [z], or [ɨz] depends entirely on the preceding segment in the input (namely, whether it is voiced or a sibilant). Empirically, this is an overly strong assumption—while many phonological processes are ISL_2, not all are, and even some are not local in a computational sense (see Heinz 2018 for an overview). However, as we discuss later in the paper, the ISL_2 assumption provides a useful test case (both theoretically and empirically) for techniques that can be expanded to other classes of functions.

Importantly, all ISL_2 functions can be represented with a *finite-state transducer* (FST) with a very specific structure. FSTs are abstract automata that transform input strings into output strings using a system of states and transitions (Mohri (1997); Sakarovitch (2009); for an explanation for phonologists, see Chandlee and Heinz (2018)). Importantly, the states in an ISL_2 FST transparently represent the possible 1-suffixes that a function may depend on. It is this structure that will allow the learning algorithm to reason about the relevant contexts for the phonological process. This also means that we can extend this reasoning process to classes of functions for which the states represent information other than 1-suffixes, as will be discussed in §6.

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.

The following section outlines a partial solution to this problem—specifically, the case in which P makes a single change. We will expand on this solution in subsequent sections.

3 The algorithm

3.1 The base SI_2DLA algorithm

The proposal here is based on 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). The term *simplex* here means that the phonology function P only makes a single change. The following discussion essentially follows the procedure in Hua and Jardine (2021).

The basic steps of the algorithm are as follows:

(5) Steps of the SI_2DLA (informal)
1. Initialize the grammar to an ISL_2 FST of the identity map;
2. Parse strings of morphemes into their constituent allomorphs;
3. Create a finite-state representation of the generalizations governing allomorphy;
4. For any morpheme M for whom there are alternating allomorphs, choose the alternant for M that appears in the most contexts (i.e., after the most 1-suffixes) as the UR for M;

5.To determine the phonological change:

(a)For some alternating morpheme M, identify:

- The change necessary to derive the SR from the UR
- The 1-suffix s in the output after which the change occurs

(b)Modify the corresponding transition in the grammar FST at the state representing the 1-suffix

We will use the following toy example to demonstrate this algorithm. First consider an inventory $\Sigma = \{\mathrm{t}, \mathrm{d}, \mathrm{a}\}$; that is, two consonants, one voiced and one voiceless, and one vowel. Then assume a vocabulary of five abstract morphemes: three roots A, B, and C, and two suffixes S and T. The target grammar in (6) assigns a UR to each of these morphemes (6-a) and includes a single process that assimilates $/\mathrm{t}/$ to $[\mathrm{d}]$ after another $[\mathrm{d}]$, given as a rule in (6-b).[5]

(6) A target set of URs and grammar

a.

A	$/\mathrm{tad}/$	S	$/\mathrm{ta}/$
B	$/\mathrm{tat}/$	T	$/\mathrm{da}/$
C	$/\mathrm{tada}/$		

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

A dataset giving the surface forms of each root both in isolation and affixed with each suffix would look as in (7).

(7) A dataset generated by the URs and grammar in (6)

A	$[\mathrm{tad}]$	AS	$[\mathrm{tadda}]$	AT	$[\mathrm{tadda}]$
B	$[\mathrm{tat}]$	BS	$[\mathrm{tatta}]$	BT	$[\mathrm{tatda}]$
C	$[\mathrm{tada}]$	CS	$[\mathrm{tadata}]$	CT	$[\mathrm{tada}]$

Note that the morpheme S alternates on the surface as $[\mathrm{t}]$ following $[\mathrm{t}]$ and $[\mathrm{a}]$ (e.g., in BS $[\mathrm{tatta}]$ and CS $[\mathrm{tadat}]$) and $[\mathrm{d}]$ following another $[\mathrm{d}]$ (in AS $[\mathrm{tadda}]$).

The next three subsections will demonstrate how the procedure in (5) takes a dataset like (7) and returns a grammar equivalent to (6).

3.1.1 Initialization and parsing

First, as per Step 1, the learning algorithm initializes the phonological grammar to an ISL_2 FST for the identity map for the given inventory. This is given in Figure 2.[6] This FST represents the kind of information the learner will pay attention to. Each of its states represents a distinct 1-suffix, meaning the last symbol seen in the input. Every transition that reads

[5] The use of a rule format here is just for convenience; the grammar the algorithm learns is in the form of a single FST, not a set of distinct and ordered generalizations.

[6] Technically, Hua and Jardine (2021)'s algorithm uses a more compact 2-state FST that only distinguishes 'environment' and 'elsewhere' states. However, each of these states represents a set of the states in Figure 2, and so their method is equivalent to the one presented here, which itself is taken from Hua et al. (2021). Using the full ISL_2 machine is more conceptually transparent and, as we will see later, is better suited to representing multiple processes.

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’.

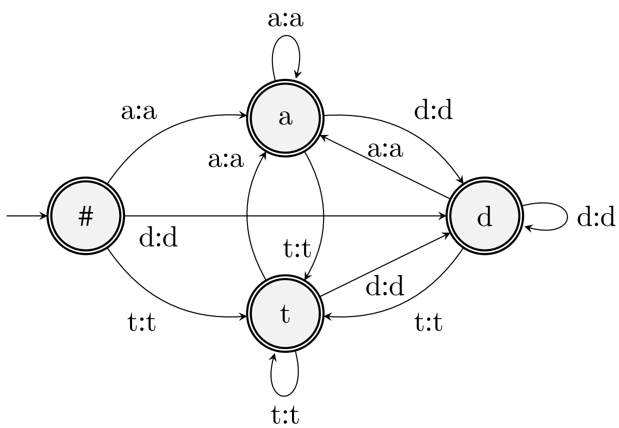

Figure 2: An ISL_2 FST for the identity function for $\Sigma = \{\mathrm{t, d, a}\}$

Importantly, however, this FST represents the identity function 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.

Step 2 then parses the input into individual morphemes and their surface forms. It does this by comparing *longest common prefixes* (LCPs) of all surface forms that begin with a particular (string of) morphemes. The LCP of a set of strings is the longest initial substring that all strings in the set share. For example, in the set $\{\mathrm{tat, tatda, tatta}\}$ the LCP is tat, as this is the longest initial sequence that all of these strings share. Similarly, the LCP of $\{\mathrm{tad, tat}\}$ is ta.

The LCP provides a useful heuristic for parsing a surface form into its constituent allomorphs. As an example, consider the morpheme string BS, whose surface form is $[\mathrm{tatta}]$. To discover the portion of the string associated with B, the learner first considers the LCP of all surfaces forms associated with strings of morphemes that begin with B. That is, it examines the outputs of the strings B, BS, and BT, which are $[\mathrm{tat}]$, $[\mathrm{tatta}]$, and $[\mathrm{tatda}]$, respectively. The LCP of this set of outputs is $[\mathrm{tat}]$, and so the learner associates $[\mathrm{tat}]$ with the morpheme B. For the S of BS, the learner then takes the LCP of all outputs of strings of morphemes that begin with BS, *minus* what the learner has already discovered for B. There is only one such string, BS, so trivially its LCP is $[\mathrm{tatta}]$. As the learner knows that $[\mathrm{tat}]$ is associated with B, this initial segment is removed, leaving $[\mathrm{ta}]$ for S. This principle for aligning input symbols with portions of the output using the LCP is known as *onwardness* (Oncina et al., 1993).

Of course there may be various surface allomorphs for a particular morpheme. For example, if we go through the above process for AS, we get $[\mathrm{tad}]$ associated with A and $[\mathrm{da}]$ associated with S. This contradicts the value associated with S, $[\mathrm{ta}]$, discovered using this procedure with the string BS above. In order to keep track of this contextual information, the learner arranges morphemes and their associated allomorphs into an onward *prefix-tree transducer* (PTT). An onward PTT is a lossless representation of the dataset in which input strings and their associated outputs are stored based on shared prefixes. A provably correct algorithm for generating an onward PTT was first given by Oncina et al. (1993) in their *Onward Subsequential Transducer Inference Algorithm* (OSTIA), which forms the first part of the implementation of the ISL_2DA.

An example onward PTT based on the data in (7) is given in Fig. 3.

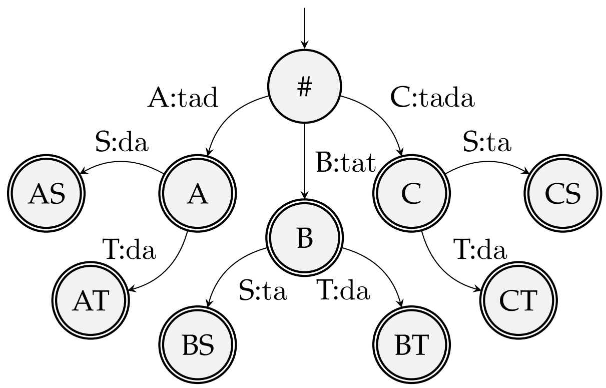

Figure 3: An onward prefix tree transducer for the data in (7)

Each pair of strings in the dataset in (7) is stored as a path through the machine in Fig. 3. For example, the pair AS [tadd] is represented by the transition from the # state to the A state on the transition A:tad and then by the transition S:da to state AS. State names represent the input prefix that reaches that state. This organization will allow the learner to detect allomorphy and its associated contexts in the next step.

3.1.2 Detecting allomorphy

The next step in the SI_2DLA is Step 3: create a FST representation of the generalizations governing allomorphy. That is, we want a representation of the rules that map morphemes to surface forms depending on what morphemes they follow; for example, S is mapped to [d] following A but it is mapped to [t] following B and C. We want these generalizations to be as general as possible, such that the learner can discover phonological commonalities among morphemes. In order to do this, we *merge* states in the PTT that have the same behavior in terms of the allomorphy they induce on morphemes. Specifically, the SI_2DLA uses the OSTIA, which induces a general function from a finite PTT by merging states with shared behavior.

For example, looking at the PTT in Fig. 3, we see that the morphemes S and T output the same allomorphs when they come out of both states B and C. Thus, we can safely merge these states—that is, combine them into a single state with the same transitions—without altering the function the FST represents. Put another way: regardless of the difference in the paths that lead to states B and C, once we leave them the outputs for S and T 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.[7] The resulting

[7] Formally, the criterion for merging states in the OSTIA is that they share the same set of *tails*, or outgoing paths. See Oncina et al. (1993) for details.

machine is given in Fig. 4 (where S/T abbreviates all states that end in either S or T); the reader can verify that it has exactly the same mapping between inputs and outputs as Fig. 3.

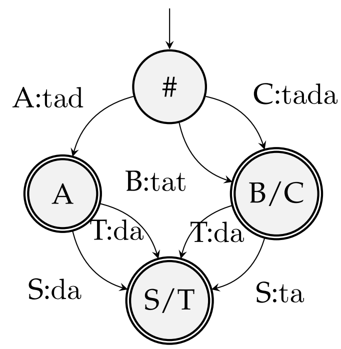

Figure 4: Result of merging states with shared behavior in Fig. 3

The advantage of this state merging technique is that it allows the learner to collect phonological information about morphemes that behave in a certain way, as the states that survive merging must represent crucial contextual information. In this example, the state labeled B/C represents morphemes which induce a following S to output as [ta], whereas the A state represents morphemes which induce S to output as [da]. (Currently there is only one such morpheme, but hopefully it is clear that other root morphemes ending in [d] would be merged with this state.) The learner can then compare their phonological forms; more specifically, it extracts the relevant phonological information from the surface forms of each morpheme.[8]

What is 'relevant' is based on the states in the phonology FST in Fig. 2—that is, what state is reached by the surface form of each morpheme? Because the FST in Fig. 2 is an ISL_2 machine, again, this boils down to the 1-suffix of the surface form. For example, the surface form that leads into state A in Fig. 4 has the 1-suffix [d], and thus would lead to the 'd' state in Fig. 2. In contrast, the surfaces forms leading into B/C arrive at two different states: 'a' (from morpheme C, [tada]) and 't' (from morpheme B, [tat]). We can begin to see the familiar pattern: state A represents a specific environment ([d]) whereas state B/C represents the 'elsewhere' environments ([a,t]). This information will be used to infer both the UR and the phonological changes that are occurring.

To see how these states actually represent *phonological*, and not *morphological* information (as it may look from the labels in Fig. 4), a fully merged allomorphy transducer is given in Fig. 5. This transducer describes the allomorphy in an ideal setting in which there are no morphotactic constraints; i.e., the morphemes can appear in any order. This is of course not realistic in the context of language acquisition, but it allows us to see the full set of environments conditioning the allomorphy. For clarity, these environments replace the state names in the figure, with '#' representing the beginning of the word.

[8]While the SI_2DLA initially looks at *surface* phonological information, the merged machine allows it to also keep track of the *underlying* phonological behavior. In this way, phonological behavior that crucially refers to underlying information—such as opaque interactions—can still be uncovered. This will be illustrated concretely in the Malay case study.

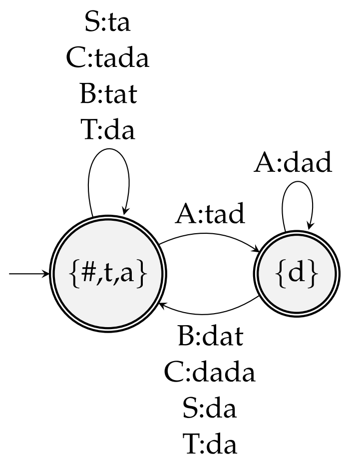

Figure 5: Ideal allomorphy FST

Obtaining this FST would require the additional data in Table 1. Again, such data is not necessarily plausible, as certain morpheme combinations will violate a given language's morphotactic constraints. A solution to this problem will be presented in §3.2 and illustrated with the Malay case study in §4.

S	[ta]	T	[da]	AS	[tadda]	BS	[tatta]
CS	[tadata]	AT	[tadda]	BT	[tatda]	CT	[tadada]
AA	[taddad]	AB	[taddat]	AC	[taddada]	BA	[tattad]
BB	[tattat]	BC	[tattada]	CA	[tadatad]	CB	[tadatat]
CC	[tadatada]	SA	[tatad]	SB	[tatat]	SC	[tatada]
ST	[tada]	SS	[tata]	TA	[datad]	TB	[datat]
TC	[datada]	TS	[data]	TT	[dada]	AAA	[taddaddad]

Table 1: Additional data needed to generate the transducer in Fig. 5 through state merging.

3.1.3 Inferring the phonology and URs

Identifying the allomorphs as in Fig. 5 is only a step towards the end goal of the learning problem, not the goal itself. The goal of the learning problem is instead to find a mapping from morphemes to their URs—as given in (6-a)—and then *explain* the allomorphy with a phonological map. With the information in Fig. 5, the SI_2DLA can infer both these URs and the phonological map.

First, as an initial assumption, the SI_2DLA takes the allomorphs that are output from the 'elsewhere' state {#, t, a} to be the URs: /ta/ for S, /da/ for T, /tad/ for A, /tat/ for B, and /tada/ for C.[9] The learner then compares these provisional URs to the outputs from the 'environment' state {d}. For example, the output of S from this state is [da], compared to its provisional UR of /ta/. The learner detects the change by comparing the *longest common*

[9]In cases in which the application of the process obscures the partition between sets of 1-suffixes, e.g., in a form like /tadt/ → [tadd], the SI_2DLA has the ability to revise this initial hypothesis based on further reasoning about the information presented in the allomorphy FST. Our case study does not make use of this aspect of the learner, but see Hua and Jardine (2021) for details.

suffix (LCS)—that is, the longest shared *final* sequence—of the two strings: in this case, 'a', as shown in (8).

(8) Comparing allomorphs of S

Output of S from state {#,t,a}	t	a
Output of S from state {d}	d	a
	difference	LCS

Ignoring this LCS identifies the difference in these forms in the initial segment: the UR begins with /t/ and the derived form begins with [d]. (The same can be seen with the outputs for A: /tad/ and [dad] share the LCS 'ad', leaving /t/ changing to [d].)[10] Thus, the learner has found evidence that an underlying /t/ changes to [d] in the context of a /d/ 1-suffix (i.e., immediately following /d/).

The $\mathrm{SI_2DLA}$'s can then use this information to edit the two FSTs in Figs. 2 and 5 such that it is the *phonological* grammar (Fig. 2) that enacts this change, leaving the machine in Fig. 5 to simply represent the lexicon mapping from morphemes to URs. The 1-suffix /d/ corresponds exactly to state 'd' in Fig. 2. The change of a /t/ to [d] following another /d/ can thus be implemented by changing the transition on /t/ from this state, as shown in Fig. 6.

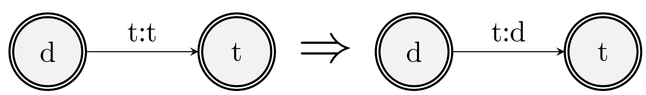

Figure 6: Editing the transition in Fig. 2 representing the phonological change

With this transition edited in the phonology FST, the learner can now simplify the FST in Fig. 5 by changing the output of all transitions to the provisional URs and then merging the two states. The resulting FST is depicted on the left-hand side of Fig. 7.

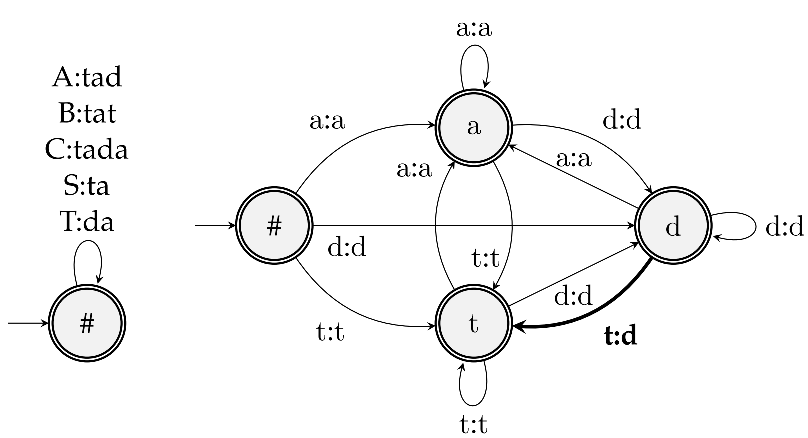

Figure 7: Final grammar, with lexicon FST on the left and phonology FST on the right, with the modified transition highlighted in bold

[10]While this was a simple, one-to-one change, the LCS-based comparison procedure outlined in Hua and Jardine (2021) can also handle insertion and deletion.

The reader can confirm that the two FSTs in Fig. 7 describe the target correspondence between morphemes and URs in (6-a) and the the rule in (6-b). This can be further checked by using these FSTs in composition, i.e., using an output of the morpheme-to-UR machine as the input of the phonology machine. For example, the machine on the left would output the morpheme sequence AS as $/\mathrm{tadta}/$, which when passed into the machine on the left would be correctly output as $[\mathrm{tadda}]$.

3.1.4 Summary

This simple example used for illustration has a limited set of morphemes, but the $\mathrm{SI_2DLA}$ is robust in that it is *provably* correct for any phonology in the class of simplex $\mathrm{ISL_2}$ functions (Hua and Jardine, 2021) (regardless of the size of the lexicon). Still, there are two main limitations to the $\mathrm{SI_2DLA}$ that are worth mentioning. First is the assumption that the phonology only enacts a single change (i.e., is simplex). The goal of the remainder of this paper is to show that this limitation can be relaxed with two simple modifications.

A second limitation is directionality. As the $\mathrm{ISL_2}$ machine used by the $\mathrm{SI_2DLA}$ to represent the phonology can only represent 1-*suffixes*, it can only deal with processes with a left-context. (Or, if we reverse the direction in which the FST reads strings, it can only deal with processes with a right-context.) We will return to this problem and discuss some possible solutions in Sec. 4.3.

3.2 Proposed modifications

The modifications to the $\mathrm{SI_2DLA}$ that enable it to work with a more realistic problem instance include 1) making use of information about morphological structure during state merging, and 2) applying a natural class heuristic during the UR selection step. These modifications will be summarized here and then demonstrated on the Malay test case in the next section.

Morphological information. The use of morphological information comes in the form of recognizing that the target composed function is actually a partial function. The domain from which the input morpheme strings are drawn does not include all possible combinations of morphemes, because (as already noted) the morphology will enforce restrictions on the order in which morphemes are concatenated. However, instead of including impossible combinations in the learner's input data as described above, we take the more plausible route of providing the learner with explicit knowledge of these morphotactics.[11]

The technical modification is to use a version of the OSTIA for learning partial functions, called OSTIA-D (Oncina and Varò, 1996). The 'D' designation indicates that the learner makes use of domain information, which it is provided in the form of a finite state acceptor (FSA) that accepts the language of the partial function's domain. The only change to the algorithm itself is an additional condition for which states can be merged in the PTT: states can only be merged if 1) they have the same behavior w.r.t. the allomorphy as described above, and 2) the prefixes that lead to those two states in the PTT end in the same state

[11] Again we acknowledge that this explicit knowledge must also be learned, either prior to or (more likely) in tandem with the phonology. In the present work we provide it to the learner in the interest of focusing on the already difficult problem of identifying URs, leaving the incorporation of morphological learning as an important area of future work.

of the domain FSA. The consequence of this added condition is that the structure of the domain FSA will be imposed on the FST that results from state merging.

Natural class heuristic. The original UR selection heuristic used by the SI_2DLA (i.e., choose the variant that appears with the larger set of contexts) will be insufficient in the general case, particularly when the allomorphy FST contains more than two states. Instead, the modified learner takes advantage of the fact that often (though not always) the UR is the form that appears with a motley collection of contexts that does not constitute a natural class. In other words, the size of the set of contexts is not as significant as whether or not it is a natural class. In the modified learner, when faced with the choice of which variant is the UR, if all but one of the variants appears in a set of contexts that can be described as a natural class, that one variant will be selected as the UR.

This modified heuristic will of course not always be decisive, as there won't always be exactly one non-natural set of contexts. To address such cases, 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.

The demonstration on the Malay test case to follow will show that these two modifications are sufficient to expand the learner's abilities to the point where it 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. While this modified learner is still not a complete solution to the phonological learning problem, its remaining limitations are fully interpretable and will be addressed at length in §4.3.

4 Case Study: Malay

We will demonstrate the modified algorithm using a set of phonological patterns in Johor Malay (Austronesian; Peninsular Malaysia), following the description in Onn's (1976) dissertation.[12] As an instance of the phonological learning problem, Malay provides a variety of pattern types and learning challenges, including allomorphy, allophonic variation, neutralization, deletion, epenthesis, optionality, derived environment blocking, and opaque interactions. We will first present the subset of these patterns that are within reach of the modified learner. After the demonstration on these patterns, we will discuss additional patterns that illustrate the modifications planned for future work that will further expand the learner's capabilities.

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 is already known. The purpose of the demonstra-

[12] Later published as Onn (1980).

tion is not then to assess the learner's performance, but to illustrate its known behavior in a way that is intuitive to phonologists.

Turning now to the facts for Johor Malay, Tables 2 and 3 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) with the symbol /r̃/) is not trilled as in 'standard literary Malay', but is rather described as being 'produced with the tongue somewhat retracted towards the front of the soft palate, and without radical constriction' (pg. 24). The fricative /s/ is classified as palatal based on its behavior in nasal place assimilation.

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

i		u
e	ə	o
	a	

Table 3: 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-/—that each have five surface variants. We will illustrate the patterns using the nominalizer /pəŋ-/, but /məŋ-/ behaves exactly the same. Before obstruents, the nasal in the prefix is subject to place assimilation:

(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:

(10) Nasal assimilation and deletion before voiceless obstruents

/pəŋ-karaŋ/	[pəŋaraŋ]	'author'
/pəŋ-samun/	[pəɲamon]	'robber'
/pəŋ-tari/	[pənari]	'dancer'

Before sonorant consonants, the prefix nasal deletes:

(11) Deletion before sonorant consonants

/pəŋ-layan/	[pəlayan]	'waitress'
/pəŋ-ɲaɲi/	[pəɲaɲi]	'singer'
/pəŋ-malu/	[pəmalu]	'shame'
/pəŋ-rayu/	[pərayu]	'appeal'

The five variants are thus {pəŋ, pəm, pən, pəɲ, pə} for the nominalizer and similarly {məŋ, məm, mən, məɲ, mə} for the active voice morpheme. Because they appear before both velar consonants *and* vowels, the variants with the velar nasals (pəŋ, məŋ) are assumed to be the defaults, or URs.

In addition to the prefix allomorphy, the surface forms 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/.[13]

(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 the glottal stop, 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):

(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 reduction. This interaction is demonstrated using rule ordering in (16), though again the proposed learner is not targeting a grammar of ordered rules.

[13]Whether the devoicing of /b/ in [jawapkan] is due to being in coda position or to voicing assimilation with /k/ cannot be discerned, because /-kan/ is the only consonant-initial suffix included in Onn's description. For present purposes, this distinction doesn't matter, since our learner is seeking a function that correctly performs the UR-SR mapping rather than the best rule-based intensional description. But we will say more later about the how the language's limited set of suffixes (only 3) interacts with the learner's heuristics.

(16) Counterfeeding interaction

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

4.1 Data

The learner will be demonstrated using the set of stems listed in Table 4. These stems were selected to include all of the contexts for the set of patterns described above. The capital letters listed with each stem will stand in for their meaning representations in the input strings.

A	/ikat/	'tie'	B	/asut/	'instigate'
D	/gali/	'dig'	E	/bayar/	'pay'
F	/bawa/	'carry'	I	/dakap/	'embrace'
L	/ladaŋ/	'farm'	M	/naik/	'ascend'
N	/jawab/	'answer'	P	/cərca/	'revile'
S	/rompak/	'rob'	T	/ŋaŋa/	'open'
U	/main/	'play'	V	/ɲaɲi/	'sing'

Table 4: Stems used to demonstrate the learner.

The set of affixes that will be used are listed in Table 5. 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-/.[14] Likewise, the causative and causative benefactive suffixes can be used on their own or in combination with the active voice prefix /məŋ-/. The affix meanings are coded with integers, but this is entirely for presentational convenience. No part of the learner makes use of the distinction between stems and affixes (i.e., capital letters versus integers).

1	/məŋ-/	active
2	/pəŋ-/	nominalizer
3	/-i/	causative
4	/-kan/	causative benefactive
5	/-an/	nominalizer

Table 5: 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).

[14] 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').

(17) (S, rompaʔ), (S3, rompaki), (S4, rompaʔkan), (S5, rompakan), (1S, mərompaʔ), (2S, pərompaʔ), (1S3, mərompaki), (1S4, mərompaʔkan), (2S5, pərompakan)

Importantly, all of these forms are attested in the language; no impossible combinations are provided. But we will say more in §6 about the potential for the learner to succeed with even less data (i.e., without complete paradigm coverage). In addition, as discussed in §3.2, the learner is provided with the FSA in Figure 8, which represents the language of the input strings of morphemes.

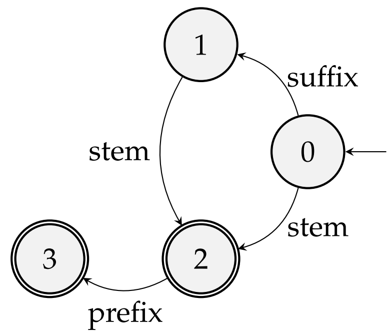

Figure 8: Domain FSA for Malay verbal stems.

A note about directionality. All of the patterns described above are regressive in nature, i.e., the triggering context follows the target. As noted above (§3.1.4) this means 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 8) are displayed with the start state on the right-hand side of the page.

4.2 Demonstration on Malay test case.

As described in §3.1.1, the first steps of the learner are to construct a prefix tree transducer for the dataset and then parse the morphemes by making the tree onward. State merging then proceeds, using the added condition of the OSTIA-D to incorporate the domain information. The output of state merging is an FST with 9 states.

For readability, only a fragment of this FST is shown in Figure 9. As noted above, the structure of the domain FSA is imposed on this FST, such that it includes a 'prefix' state (A1), a 'suffix' state (3, which is more specifically the state for the two vowel-initial suffixes), and then separate states for the classes of stems that behave differently according to the set of alternations described above. The figure includes the states for two of these classes: stems beginning with bilabials (state E) and stems beginning with sonorant consonants (state L).[15] The remaining states not shown in the figure include the ones for 1) consonant-initial suffixes, 2) alveolar-initial stems, 3) palatal-initial stems, and 4) the default class that includes both velar- and vowel-initial stems. Also for readability, the figure only includes one representative stem on the transitions to each state.

[15]The state names are retained from the prefix tree transducer; for a state representing a class of morphemes, the state's name is the morpheme in that class the learner encountered first and then merged all others with.

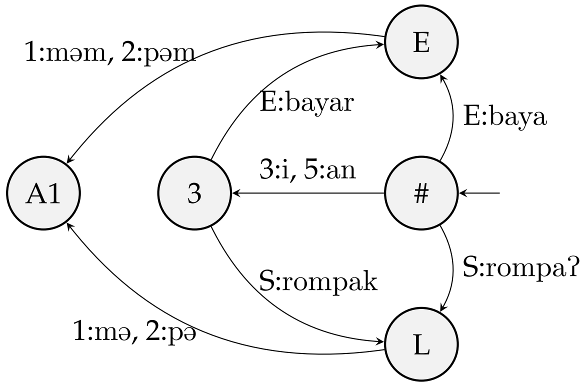

Figure 9: Fragment of the OSTIA-D output for Malay test case.

It can be seen in this figure how state merging reveals the contexts governing the target alternations, as the states that survive merging provide crucial information. Out of the start state (#), word-final processes like /r/-deletion and /k/-glottalization are reflected in the stem SRs, but out of the suffix state they are not. This is also why there are two suffix states, as out of the consonant-initial suffix state (not shown) these processes are again observed. And the prefix allomorphy is also reflected in the different variants of the prefixes leading out of each stem state and into the prefix state.

To begin the construction of the lexicon and phonological mapping, any morpheme that does not alternate—meaning any morpheme that has the same form on all transitions on which 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 6:

Initial lexicon	
3	/i/
4	/kan/
5	/an/
A	/ikat/
B	/asut/
D	/gali/
I	/dakap/
L	/ladaŋ/
U	/main/
V	/ɲaɲi/

Table 6: Initial lexicon of URs: non-alternating morphemes.

For the morphemes that do alternate, we need a general procedure for identifying which variant is the UR, one that does not rely on there being only two states to decide between. We will demonstrate this procedure using the active voice morpheme, /məŋ/, which as described above has five surface variants. For each of these variants, the learner gathers all of the transitions that lead *into* the state from which the variant is observed. For example, the variant [məŋ] is observed on the transition that leaves state A. The set of *incoming* transitions for state A is listed in Table 7. (A complete listing of all transitions in the FST is provided in Appendix A.1.)

State	Input	Output	State
#	A	ikat	A
#	B	asut	A
#	D	gali	A
3	A	ikat	A
3	B	asut	A
3	D	gali	A
4	A	ikat	A
4	B	asut	A
4	D	gali	A

Table 7: Transitions leading into state A.

From here we gather all of the output strings from this set of transitions—{ikat, asut, gali}—and then take the 1-suffixes of this set of strings: {i, a, g}. (Recall again that because the FST reads right-to-left, the suffix of the string is its first segment, not its last.) As discussed in §3.1.2, these 1-suffixes are the contexts that generate this variant of the morpheme, again under the overarching assumption that the phonology function is ISL_2. We repeat this procedure for all of the morpheme's variants, as summarized in Table 8.

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 8: 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. The feature set is the one provided by Onn (1976, pg. 40) for the Johor Malay inventory, with the addition of a [±segment] feature to distinguish segments from the word boundary (Chomsky and Halle, 1968). (The feature chart is provided in Appendix A.2.) A natural class is defined 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 8, this condition 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 §3.1.3. For example, based on the [mən] variant we know that an underlying /ŋ/ surfaces as [n] when it precedes a /d/ (second row of Table 8). The complete list of transitions updated based on the analysis of the active voice prefix is shown in Table 9. 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 9: Transitions updated in the phonological identity mapping after the analysis of the active voice prefix /məŋ/.

The exact same procedure handles the other patterns reflected in the data. For example, consider morpheme F ('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.

The learner proceeds in this fashion through all of the stems with multiple SRs, culminating in the final lexicon from Tables 4 and 5 and the set of phonological changes listed in Figure 10. (These changes are listed in the familiar rule format, but again, the phonology is a single FST, not an ordered sequence.)

ŋ → m / __ b
ŋ → n / __ d
ŋ → ɲ / __ {c, j}
ŋ → ∅ / __ {m, n, r, l, ŋ, ɲ}
k → ʔ / __ {k, #}
r → ∅ / __ {k, #}
a → ə / __ #
b → p / __ {k, #}

Figure 10: Phonological changes detected by the learner on the Malay test case.

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).[16] These results thus represent a significant step forward from the simplex functions assumed by the original algorithm.

[16]We leave it as an open question for future work whether other types of opacity (see Baković, 2007) will

Still, it is certainly not difficult to conceive of a case where the natural class heuristic will be unable identify the correct (or any) UR. Indeed, the point was not to argue that this heuristic alone is all that is needed for the broader problem of learning URs, but rather to show how much more can be handled—using data that includes only attested forms—-with minimal modifications to the $\mathrm{SI_2DLA}$. Also, in the modified learner, $\mathrm{SI_2DLA}$'s foundational idea—that a locality assumption allows us to extract the phonological map obscuring the URs on the surface—remains intact.

Anticipating more challenging cases, we now describe another augmentation to the UR selection procedure. While this augmentation was not strictly needed for the Malay test case, a closer look reveals that in fact the structure of the language is providing an added advantage, without which the natural class heuristic would indeed fail. We explain that advantage and demonstrate the augmentation—a hypothesis testing mechanism—in the next section.

4.2.1 Hypothesis testing

Let's take a closer look at the analysis of the morpheme /jawab/, 'answer', which leads to the discovery of what might be called 'final devoicing', or more accurately, 'coda devoicing', as this morpheme surfaces as [jawap] when it doesn't have a suffix and when it is suffixed with [-kan]:

(18) a. /jawab/ → [jawap], 'to answer'
b. /pəŋ-jawab-kan/ → [pəɲjawapkan], 'the answering'

The 1-suffix contexts for the SR [jawap] are then {#, k}, while [jawab] appears with {i, a}. Given the vowel inventory shown above in Table 3, these two vowels are indeed not a natural class, and so /jawab/ is correctly selected as the UR. However, it's fairly intuitive that this patterning is not actually about these two particular vowels, but rather the fact that before a vowel-initial suffix the obstruent at the end of this stem will be in onset position. Likewise, the devoicing of that obstruent before {#, k} is due to it being in coda position. Both of these contexts are predictable, making it unclear which should be considered the default. Indeed, the non-natural-ness of the set {i, a} is due entirely to the fact that *-i* and *-an* are the only two vowel-initial suffixes.[17]

We can obviously expect this scenario—all sets of contexts are equally natural (or possibly equally unnatural)—to arise in other test cases, and so here we describe an additional mechanism that the learner can draw on when the natural class heuristic is indecisive. For demonstration purposes, we'll just force the learner to treat {i, a} as a natural class. This affects the decision procedure for four of the verbal stems: E (baya, bayar), M (naik, naiʔ), N (jawab, jawap), and S (rompak, rompaʔ). For each stem, the learner in turn takes each surface variant as a hypothesized UR, and identifies the phonological changes that would be necessary under that hypothesis.[18] For example, for morpheme E ('pay'), if [baya] is the

---

prove to be problematic and what additional modifications might be needed for the learner to address those challenges.

[17]This is not in turn due to our limited data selection; according to Onn (1976, p. 102), the three suffixes we used are the only ones in the language.

[18]Tesar and Prince (2007) employ a similar strategy for the learning of URs and an Optimality Theory (OT) grammar.

UR, then the rule in (19-a) is necessary to generate the SRs [bayari] and [bayaran].

(19) UR: /baya/

a. $\emptyset \rightarrow$ r / __ {a, i}

And likewise, if [bayar] is the UR, then the rule in (20-a) is necessary to generate the SRs [baya] and [bayakan].

(20) 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. In this case, it finds a contradiction for rule (19-a), which would wrongly map the UR /ikat/, 'tie', to the SR *[rikrat]. This hypothesis can then be rejected in favor of the second one, which does not encounter any contradictions.[19] A similar analysis will lead to the selection of /jawab/ as the UR for N ('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—M (naik, naiʔ) and S (rompak, rompaʔ)—still presents a challenge. The hypothesis that needs to be rejected, shown in (21), will not encounter any contradictions in the data (and of course, neither will the correct hypothesis).

(21) 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 in the language. 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.

4.3 Additional process in Malay

In this section we discuss a few additional patterns from Johor Malay in order to illustrate some of the future developments that will be needed to expand the learner's capabilities. Perhaps most obvious is the fact that not all structural changes will be limited to a length of 2. Malay has at least two processes that require a locality window of length 3: vowel lowering and vowel lengthening, as illustrated in (22) and (23), respectively.

(22) Vowel lowering (Onn, 1976, p.30)

[19] The success of this method will of course depend on the content of the lexicon at the point it is attempted, as the crucial UR that identifies a contradiction may not be present. We leave it for future iterations of the learner to explore solutions for this dilemma, such as randomly selecting a UR and correcting it later if necessary.

a. $\mathrm{V}_{+hi} \rightarrow [-\mathrm{hi}]$ / __C {#, C}
b. /milik/ → [mileʔ], 'to own/possess'
c. /pilih/ → [pileh], 'to choose'

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

Clearly the use of 1-suffixes will be insufficient for the learner to recognize the contexts of these alternations, leading it to overgenerate and lower vowels before single consonants or lengthen them before all nasals. How to extend the learner to look at 2-suffixes for 3-local mappings like these, or more importantly, how to generalize it for any k, are interesting and challenging open questions. Answering them will of course also include broadening the UR selection heuristics (i.e., what does it mean for a set of $(k-1)$-suffixes to be a natural class or not?), a topic we will return to in §6.

Next, one process 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)):

(24) Nasal assimilation and deletion before voiceless obstruents

/pəŋ-karaŋ/	[pəŋaraŋ]	'author'
/pəŋ-samun/	[pəɲamon]	'robber'
/pəŋ-tari/	[pənari]	'dancer'

The combined effect of these patterns has been called *fusion* or *coalescence*, as the place feature of the deleted obstruent is taken on by the nasal (Lapoliwa, 1981; Pater, 2004), though for Onn (1976) it is the result of two ordered rules. In our model, the phonology must perform the mapping in one step, which poses a challenge given its bidirectional nature (i.e., the nasal assimilates to a following obstruent, but the obstruent deletes in response to a preceding nasal). Specifically, what goes wrong is the following. We'll use the stem /tari/, 'dance' (coded as morpheme W) to illustrate. As shown in Figure 11, when the prefix tree transducer is made onward to segment the morphemes, the stem-initial obstruent gets stranded because the longest common prefix (i.e., suffix, since the FST is right-to-left) of [mənari] and [tari] is just [ari].

This is not technically an error, however, as transitions with the input # are part of the definition of subsequential FSTs as a representation of their *final output function*. The outputs of these transitions are appended whenever an input string ends in that state. In this case, for the input string W, the FST will generate the output string [ari], and then (since there are no more inputs to transition on) the final output of state W is appended to generate the correct SR of [tari]. If more input did follow—for example if the string is 1W—the final output string [t] is not appended, as the FST moves on to state 1W and generates [mənari] (i.e., 'fusion'). The final transition that outputs [t] is retained through state merging, the result of which is shown in Figure 12.

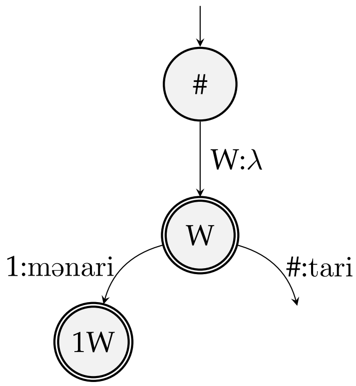

Figure 11: Fragment of prefix tree transducer (left) and onward prefix tree transducer (right) for the stem /tari/, 'dance'.

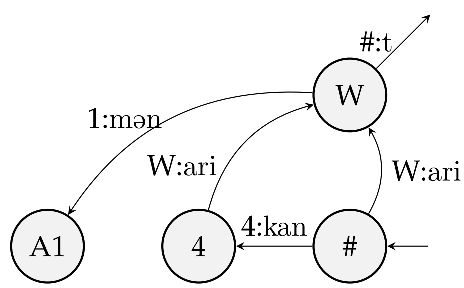

Figure 12: Fragment of the OSTIA-D output, focusing on fusion allomorph.

We can see from the figure that the existence of this final output on state W needs to be factored into the step that identifies the state's 1-suffixes. In particular, we know the [mən] variant of the active voice prefix is conditioned by alveolars, and so it's the [t] that needs to be included in the set of 1-suffixes, not the last segment that is output leading into that state (i.e., [a]). This is an easy fix—the learner can simply check whether the state has a non-empty final output string when it compiles the 1-suffixes—but a complete solution to fusion will require further modifications to the learner so that it can identify patterns in both directions. In particular, the selection of the UR for the stem fails because the determination between the two surface variants, [ari] and [tari], is not due to what follows (i.e., either the word boundary or a suffix) but to what precedes it (i.e., either the word boundary or a prefix).

Importantly, though, this does not mean the fusion mapping itself is formally too complex for a learner that employs a locality bias. The function that corresponds to the rules in (25) is straightforwardly ISL_2.

(25) a. ŋt → [n]
b. ŋp → [m]

c. ŋc → [ɲ]
d. ŋj → [ɲ]

The challenge for the learner is its current limitation to looking at one direction at a time (either leftward or rightward) when determining the contexts of patterns. We leave it for future work to explore ways to remove this limitation, which will be desirable not just for languages with fusion but for those with any combination of patterns in both directions.

Next, Johor Malay has a set of epenthesis processes that introduce some additional challenges. Epenthesis is employed as a repair for vowel hiatus, but which segment is epenthesized depends on the quality and location of the two vowels. As shown in (26-a) and (26-b), if the first vowel is high, the corresponding glide is epenthesized; otherwise ((26-c) and (26-d)) it's a glottal stop.

(26) 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'
e.	/di-ambil/	[diʔambel]	'to take (passive)'

However, that difference in the epenthesized segment is only observed when the hiatus straddles a suffix boundary. As shown in (26-e), at a prefix boundary a glottal stop is inserted after a high vowel. Lastly, as witnessed by stems like [main], 'to play' and [naik], 'to ascend', epenthesis is subject to nonderived environment blocking (Kiparsky, 1993) and only applies across, not within, morphemes. Once again, the challenge presented by these patterns is not due to formal complexity: as either individual maps or a single map, the epenthesis rules are ISL with a window size of 3 (in order to accommodate the morpheme boundary and only epenthesize in the context of, for example, a__−i and not a__i). The different behavior at prefix versus suffix boundaries can be handled straightforwardly with distinct boundary symbols.

As with fusion, the challenge for the current learner is again the way it looks for the local phonological generalizations. In particular, because of how morphemes are parsed (via onwardness), the epenthesized segments in the surface form must be attributed to one of the morphemes (specifically the verbal stems). This in turn means the UR selection procedure will be faced with a decision between, e.g., [bantu] and [bantuw]. The necessary rules that correspond to these two UR hypotheses are shown in (27) and (28), respectively.

(27) UR: /bantu/

a. $\emptyset \rightarrow$ w / __ − {i, a}

(28) UR: /bantuw/

a. w $\rightarrow \emptyset$ / __ − {k, #}

Neither of these rules is correct, however, as the epenthesis rule in (27) will overgenerate and apply to all stems when suffixed with *-i* and *-an*, not just the ones ending with [u].

Furthermore, all three epenthesis rules before suffixes will share the same context, such that only one will be reflected in the phonology map (whichever one is discovered last). These issues point to a need for the learner to recognize epenthesis as a distinct rule type in which its context is split (e.g., u__−{i, a}), but we leave this for future work with the hope that additional testing on other case studies will suggest a more general and elegant solution.

One additional important area for future work is adapting the learner to handle optional processes. For example, Malay has an optional rule that changes coda /s/ to [h]:

(29) s → h / __ {#, C} (Onn, 1976, p.26)
   a. /kipas/ → [kipah] ∼ [kipas], 'fan'
   b. /kipas-kan/ → [kipahkan] ∼ [kipaskan], 'to cause to fan for'

This rule has much in common with the rules learned in the demonstration: it is regressive, ISL_2, and conditioned by the word boundary or a following consonant. But, the current learner is not equipped for optional rules and so will enforce this change obligatorily. The path forward for this rule and optional rules in general is to move from *determinism* to *semi-determinism* (Beros and de la Higuera, 2016), which Heinz (2020) shows can model optional phonological changes. A semi-deterministic transducer is deterministic on the input, but the output of transitions can be a set of strings instead of a single string (e.g., $s : \{s, h\}$). This would preserve the structure of the FST (i.e., the states and where the transitions go) while accommodating the non-deterministic aspects of the data. Of course, the presence of these added output strings would have an impact on the UR selection procedure (as they would affect the (non)natural-ness of the sets of 1-suffixes), and so the use of semi-determinism is not necessarily a trivial development. Nonetheless, it offers a promising route toward a version of the learner that works with even more realistic data that reflects variation.

Before concluding this section, we offer two final notes on this test case and our results for it. First, our discussion of the limitations of the current version of the learner illustrates a key advantage of this overall approach to learning, which is *interpretability*. When the learner fails, we know exactly *why*, and this knowledge directly informs the identification of next steps. This bottom-up approach to development also offers a valuable means of studying the phonological learning problem itself, as we know definitively what class(es) of patterns can be learned under what conditions (i.e., data content, parameter settings, set of heuristics). Second, Malay is a very valuable test case for any learner, as it combines a variety of pattern types and recognized challenges for learning: substitutions, deletions, insertions, iterativity, opacity, optionality, and lexical exceptions.[20] We encourage anyone working on the computational modeling of morpho-phonological learning —particularly with a goal of learning URs—to draw on Malay for the development and testing of their learners.

In the next section we survey some of these existing learners, though a direct comparison of results is not particularly useful. Most of them are situated in particular theories of the phonological grammar, which has the effect of starting from a different problem definition

[20] Onn (1976) reports a class of /c/-initial stems that do not participate in fusion under prefixation but only show the nasal assimilation. There are also some words that do not undergo the optional s-to-h change.

than the one provided in §2 (i.e., learning a grammar as a function is a distinct problem from learning a ranking or weighting of a set of provided constraints, or learning a set of ordered rules). We instead offer this survey in order to situate our learner in the broader context of the work on UR learning and to highlight the ways it complements this line of work, all of which contributes in varying ways to our larger understanding of learning in phonology.

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 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 (under the common generative assumption that CON is innate). In contrast, one goal of the present work is to determine how much learning can take place when 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 dependant 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). Much less is known about learning in these other formalisms, but the present results suggest great potential for future work along those lines.

Beyond the work in constraint-based frameworks, other prominent examples of UR and grammar learning include the Minimum Description Length (MDL) learner proposed by Rasin et al. (2020, 2021) and Cotterell et al. (2015)'s use of loopy belief propagation over a Bayesian network. The MDL approach searches a space of possible rule-based grammars, converging on the one that is most economical in terms of both the size of the grammar itself as well as the encoding of the data according to that grammar. Cotterell et al. (2015) model the problem of learning morpho-phonological grammars as a problem of inference in a directed graphical model, representing the lexicon, morphology, and phonology with probabilistic maps as opposed to rule-based or constraint-based grammars. The current proposal differs from the above approaches in key respects. Primarily, it is grounded in grammatical inference techniques that capitalize on typologically-motivated formal restrictions, which dramatically reduce the hypothesis space that the MDL and Bayesian algorithms search without diminishing its capacity to represent actual morpho-phonological patterns. It is also *provably correct*, in that Hua and Jardine (2021) give a detailed analysis of the behavior of the algorithm and prove the conditions under which it will successfully converge to the target grammar.

Lastly, recent work by Belth (2023) has explored the learning of abstract URs, such as the underspecified suffixes in vowel harmony languages like Turkish (e.g., the plural suffix /-lAr/, which alternates between [-lɑr] and [-ler] depending on the backness of the stem vowel).[21] 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. The success of this approach on the Turkish case raises two questions regarding its viability as a model of acquisition. One is how the modules of lexicon construction and phonological learning interact with and inform each other over the course of acquisition, as realistically the entire lexicon will not be complete before the onset of phonological learning. Two is the question of whether URs are *always* abstract, or whether the use of abstract URs is reserved for those cases in which no other evidence (for example, contextual information) points to a default variant. The latter is a question for our learner as well, which in its current form does not allow for abstract URs at all, but rather adheres to the *basic alternant* assumption that the UR is one of the observed SRs (Kenstowicz and Kisseberth, 1979). Relaxing that assumption is one of the avenues for future work, which we turn to in the next section.

[21] Wang and Hayes (2022) also explore the learning of abstract URs, using an EM-MaxEnt learner to test the different levels of Kenstowicz and Kisseberth (1977)'s abstractness hierarchy.

6 Next steps and future directions

Our learner's results for the Malay test case demonstrate the viability of function decomposition as an approach to the phonological learning problem. With two fairly minor modifications, we have risen from a toy example to a test case on the level of an introductory phonology problem set. So where do we go from here?

In addition to the advantage of interpretability mentioned above, our approach also offers a number of advantages tied to its *modularity*. The state-merging procedure that reveals the contextual information governing the patterns is independent of the construction of the phonological mapping based on an assumption of a particular form of locality (ISL). And both of these components are also independent of the natural class heuristic used to select the UR. Future work will explore different versions of all of these components in order to broaden the scope of the patterns our learner can learn.

First, versions of the learner can be developed in which the phonological function is not limited to ISL but belongs to other classes that are based on distinct but not unrelated conceptions of locality, in particular the Output Strictly Local (OSL; Chandlee et al., 2015) functions (needed for iterative processes) or the tier-based strictly local (TSL; Burness et al., 2021; Burness, 2022) functions (needed for long-distance processes like vowel and consonant harmony). A language that includes patterns of these different types (ISL, OSL, TSL, for possibly different values of k) would then have some number of these functions that collectively enact its phonology. How those functions work together in a unified grammar is an important open question, one we defer for now in the interest of focusing on how such functions can be learned in tandem with a lexicon of URs.

Second, more sophisticated UR selection heuristics will inevitably be required in some cases. Identifying and implementing different versions of the UR selection procedure will amount to testing different hypotheses for what kinds of knowledge a learner might draw on when constructing the lexicon. For example, the natural class heuristic in its current form assumes the feature set has already been learned or else was provided by Universal Grammar (UG).[22] But it would be interesting to see how proposals for learning features (e.g., Odden, 2022) might be incorporated. In addition, relaxing the basic alternant assumption to accommodate cases in which the UR is not one of the SRs will require additional heuristics for when to conjecture an abstract UR (in the spirit of Belth (2023) discussed in the previous section) and how to construct it.

Lastly, the success of the learner's use of onwardness for morpheme segmentation in this test case is the result of it seeing all allowable morpheme combinations. As noted, this is already an improvement over a characteristic sample that also includes impossible sequences, but learning from less idealistic data coverage is an obvious desideratum. Testing of the learner with smaller datasets did reveal that under certain conditions initial segmentation errors due to not having all the needed forms can be corrected during state merging.[23] Identifying exactly what those conditions are is another important question for future work. 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

[22]See Hale and Reiss (2008) for arguments that features must be a part of UG.

[23]This is because the OSTIA's method for resolving the non-determinism that results from state merging involves pushing back prefixes that were pushed forward to make the transducer onward.

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 the current paper has shown, the use of unrealistic data samples in proofs of formal learnability does not mean these learners cannot succeed without them. The modifications made in the current work to obtain these results did not change the central role of the properties that define the class of functions. Limitations remain, but the transparent nature of the algorithm establishes a clear path forward for further progress on the problem of how children acquire phonological grammars.

A Appendix

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

State names can be interpreted as follows:

- # = start state
- E = bilabial stems
- L = sonorant consonant stems
- N = palatal stems
- I = alveolar stems
- A = velar and vowel-initial stems (default)
- 3 = vowel-initial suffixes
- 4 = consonant-initial suffixes
- A1 = prefixes

State	Input	Output	State
#	A	ikat	A
#	3	i	3
#	4	kan	4
#	5	an	3
#	B	asut	A
#	D	gali	A
#	E	baya	E
#	F	bawə	E
#	I	dakap	I
#	L	ladaŋ	L
#	M	naiʔ	L
#	N	jawap	N
#	P	cərcə	N
#	S	rompaʔ	L
#	T	ŋaŋə	L
#	U	main	L
#	V	ɲaɲi	L
A	1	məŋ	A1
A	2	pəŋ	A1
3	A	ikat	A
3	B	asut	A
3	D	gali	A
3	E	bayar	E
3	F	bawa	E
3	I	dakap	I
3	L	ladaŋ	L
3	M	naik	L
3	N	jawab	N
3	P	cərca	N
3	S	rompak	L
3	T	ŋaŋa	L
3	U	main	L
3	V	ɲaɲi	L
4	A	ikat	A
4	B	asut	A
4	D	gali	A
4	E	baya	E
4	F	bawa	E
4	I	dakap	I
4	L	ladaŋ	L
4	M	naiʔ	L
4	N	jawap	N
4	P	cərca	N
4	S	rompaʔ	L
4	T	ŋaŋa	L
4	U	main	L
4	V	ɲaɲi	L
E	2	pəm	A1
E	1	məm	A1
I	2	pən	A1
I	1	mən	A1
L	2	pə	A1
L	1	mə	A1
N	2	pəɲ	A1
N	1	məɲ	A1

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

Antilla, 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.

Barke, S., Kunkel, R., Polikarpova, N., Meinhardt, E., Bakovic, E., and Bergen, L. (2019). Constraint-based learning of phonological processes. In Inui, K., Jiang, J., Ng, V., and Wan, X., editors, *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, Hong Kong, China. Association for Computational Linguistics.

Belth, C. (2023). 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. (in press). A learning-based account of local phonological processes. *Phonology*.

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

Chomsky, N. and Halle, M. (1968). *The Sound Pattern of English*. Harper & Row.

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.

Hale, M. and Reiss, C. (2008). *The Phonological Enterprise*. Oxford University Press, Oxford.

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.

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.

Kiparsky, P. (1993). Blocking in nonderived environments. In Anderson, S. R., Hargus, S., Kaisse, E. M., and Keating, P. A., editors, *Phonetics and phonology 4: Studies in lexical phonology*, pages 277–313. Academic Press.

Koskenniemi, K. (1983). *Two-level morphology: a general computational model for word-form recognition and production*, volume 11. Helsinki: University of Helsinki Department of General Linguistics.

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.

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.

Odden, D. (2005). *Introducing Phonology*. Cambridge University Press.

Odden, D. (2022). Radical substance-free phonology and feature learning. *The Canadian Journal of Linguistics / La revue canadienne de linguistique*, 67(4):500–51.

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., and Katzir, R. (2018). Learning phonological optionality and opacity from distributional evidence. In Hucklebridge, S. and Nelson, M., editors, *Proceedings of NELS 48*, pages 269–282.

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.

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.