# Linguistic Faithfulness Enables Decomposition Learning of Sequential Functions from Given Structure

**William Schilling** WSCHILLI@ANDREW.CMU.EDU
*Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, USA*

**David R. Mortensen** DMORTENS@CS.CMU.EDU
*Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA, USA*

## Abstract

Motivated by a learning problem in natural language phonology, this paper studies the learnability of series of sequential transductions from their composition when given a priori knowledge of their finite-state structures. Appealing to the notion of faithfulness, we propose linguistically-amenable conditions under which sequential transductions decompose uniquely into given finite-state structures. Under such conditions, we present a decomposition algorithm that enables strong in-the-limit identification of series of sequential transductions given data from their composition.

**Keywords:** decomposition learning, identification in the limit, faithfulness, sequential transductions, structured classes of transductions

## 1. Introduction

A premier objective of natural language phonology is the identification of phonological rules from natural language data. That is, given string pairs consisting of an *underlying form* and a *surface form*, to infer a series of rewrite rules—often modeled by subsequential transductions (Schützenberger, 1977; Mohri, 1997)—that maps every underlying form to its corresponding surface form. Though current in-the-limit identification algorithms such as OSTIA (Oncina et al., 1993) give strong theoretical learning results for subsequential models of individual rules or whole phonological processes, there are fewer methods for identifying the series of rules that constitutes such a process.

To address this, we turn to decomposition-style learning in the sense of Hua and Jardine (2021), identifying a series of transductions by decomposing, or factorizing, a learned representation of their composition. Their decomposition learning algorithm $SI_2DLA$ enables in-the-limit identification (Gold, 1967) of string homomorphisms composed with simplex input-strictly-2-local transductions (Vaysse, 1986-1987; Chandlee et al., 2014), which, for linguists, may model a linear morphology composed with a phonological rule.

This paper focuses instead on decomposition learning for series of many phonological rules, modeled by sequential transductions (a proper subclass of the subsequential transductions), and assuming a priori structural information in the sense of Jardine et al. (2014). Our main learning result is given by the Structured Faithful Sequential Decomposition Learning Algorithm (SFSDLA), which, under our proposed *faithfulness conditions*, enables strong in-the-limit identification (Gold, 1967) of series of total initial-empty sequential transductions given their individual finite-state structures and data from their composition. In the special case of two transductions, the identification algorithm is time efficient in the sense

© 2026 W. Schilling & D.R. Mortensen.

of de la Higuera (2010). We have implemented SFSDLA and deciders for the faithfulness conditions in Python.

To our knowledge, these are the first decomposition learning results for series of arbitrarily many transductions, instead of two transductions, and for transductions with arbitrary finite-state structures, instead of only strictly-local ones. Targeting arbitrary finite-state structures benefits the empirical coverage of our results. Though strictly-local classes model many common phonological processes (Chandlee and Heinz, 2018), they miss out on attested phenomena modeled by more complex finite-state structures. For instance, input-tier-based-local maps (Heinz et al., 2011; Hao and Andersson, 2019) are not strictly-local, but are still modeled by predictable finite-state structures, and are thus in reach for our given-structure approach.

To ensure that our target class of transductions decomposes uniquely—as is essential for decomposition learning—we propose three conditions motivated by linguistic faithfulness, the principle that surface forms tend to be similar to their underlying forms. Our focus on faithfulness comes from the observation that the simplex condition proposed by Hua and Jardine (2021) enforces a notion similar to faithfulness in that it limits the amount of change that a transduction may realize. The main condition, *feeding-faithfulness*, enforces that every change realized by a series of transductions be due to at most one transduction in the series. This benefits learnability in that any change observed in the composed transduction is guaranteed to accurately reflect its origin: some change from a transduction involved in the composition. As for how this restriction affects the empirical coverage of our results, we will argue that phonological rules exhibiting feeding interactions (Kiparsky, 1968) pose the greatest challenge.

The paper is organized as follows. Section 2 presents preliminaries on strings, transductions, and congruences. Section 3 describes sequential transducers and an algebraic characterization of sequential transductions. Section 4 defines the faithfulness conditions and proves a lemma that relates them. Section 5 presents the decomposition learning algorithm SFSDLA, and Section 6 presents a version optimized for the case of two transductions. Section 7 provides a linguistic analysis of the main faithfulness condition, which provides qualitative insight into the empirical coverage of our results. Section 8 concludes and proposes directions for future work.

## 2. Preliminaries

An alphabet $\Sigma$ is a finite nonempty set of characters. $\Sigma^*$ is the set of strings, or finite sequences of characters from $\Sigma$. $\lambda$ denotes the empty string. Let $u, v, w$ be strings. $uv$ or $u \cdot v$ is the concatenation of $u$ and $v$. If $w = uv$, then $u$ is a prefix of $w$, denoted $u \preceq w$. In this case, the left-quotient of $w$ by $u$, denoted $u^{-1} \cdot w$, is $v$. $\preceq$ is a partial order relation. If $u \preceq w$ but $u \neq w$, then we say that $u \prec w$.

A language over $\Sigma$ is a subset of $\Sigma^*$. A transduction between alphabets $\Sigma$ and $\Gamma$ is a maybe-partial function $f : \Sigma^* \to \Gamma^*$. The domain and image of $f$, which happen to be languages, are denoted $\mathrm{dom}(f)$ and $\mathrm{img}(f)$ respectively. $f$ is called total if and only if $\mathrm{dom}(f) = \Sigma^*$. The identity transduction, denoted $\mathtt{id}$, is defined via $\mathtt{id}(u) = u$, and is total. Let $L$ be a language and let $f, g, h$ be transductions. The composition of $f$ and $g$, denoted $f \circ g$, is defined via $\mathrm{dom}(f \circ g) = \{u \mid u \in \mathrm{dom}(g) \wedge g(u) \in \mathrm{dom}(f)\}$ and $(f \circ g)(u) = f(g(u))$.

We can restrict the domain of $f$ to $L$, similarly denoted $f \circ L$, via $\mathrm{dom}(f \circ L) = \mathrm{dom}(f) \cap L$ and $(f \circ L)(u) = f(u)$. If $h = f \circ g \circ L$, then $h \circ g^{-1}$ is defined as $f \circ \mathrm{img}(g \circ L)$. As the notation suggests, this is just the result of composing a minimal right-inverse of $g$ with $h$, and may be interpreted as the maximal specification of $f$ that is determined by knowing only $h$ and $g$. In Appendix A, we argue that these definitions are coherent and prove Equation 1.

$$(h \circ f \circ \mathrm{img}(g)) \circ f^{-1} = h \circ \mathrm{img}(f \circ g) \tag{1}$$

A right congruence $\sim$ is an equivalence relation over strings that satisfies the compatibility property $u \sim v \Longrightarrow uw \sim vw$ for all strings $u, v, w$. $\Sigma^*/\sim$ is the set of congruence classes resulting from $\sim$, where the congruence class $[u]_\sim \in \Sigma^*/\sim$ (or simply $[u]$) represents the set of strings that are equivalent to $u$ under $\sim$, so that $[u]_\sim = [v]_\sim$ if and only if $u \sim v$. If $\Sigma^*/\sim$ is finite, then we say that $\sim$ has finite index. The compatibility property allows us to lift right-concatenation to the level of congruence classes, so the congruence class $[uw]_\sim$ is well-defined whenever $[u]_\sim$ is a congruence class and $w$ is a string.

## 3. Sequential Transductions

We target the class of *initial-empty sequential transductions*, which are recognized by finite automata that read input strings from left to right and either emit an output string or choose to reject the input at every input character. Compared to the class of *subsequential transductions*, initial-empty sequential transductions emit no output before reading the input, and never emit output or choose to reject the input in response to termination of the input string.

In precise terms, an *initial-empty sequential transducer* between alphabets $\Sigma$ and $\Gamma$ is $\langle Q, \delta, o, q_0 \rangle$ where $Q$ is a finite set of states, $\delta : Q \times \Sigma \to Q$ is the maybe-partial transition function, $o : Q \times \Sigma \to \Gamma^*$ is the maybe-partial output function, and $q_0 \in Q$ is the initial state. It is convenient to enforce that $\mathrm{dom}(\delta) = \mathrm{dom}(o)$. We then define the extended transition function $\delta^* : Q \times \Sigma^* \to Q$ via Equation 2 and the extended output function $o^* : Q \times \Sigma^* \to \Gamma^*$ via Equation 3 for all $q \in Q$, $u, v \in \Sigma^*$, and $\sigma \in \Sigma$. Their domains are restricted according to the domains of $\delta$ and $o$, and it turns out that $\mathrm{dom}(\delta^*) = \mathrm{dom}(o^*)$. Every transducer is said to *recognize* a transduction $f : \Sigma^* \to \Gamma^*$ defined via $\mathrm{dom}(f) = \{u \mid (q_0, u) \in \mathrm{dom}(o^*)\}$ and $f(u) = o^*(q_0, u)$. The initial-empty sequential transductions are exactly the transductions that are recognized by some initial-empty sequential transducer.

$$\delta^*(q, \lambda) = q \qquad \delta^*(q, \sigma) = \delta(q, \sigma) \qquad \delta^*(q, uv) = \delta^*(\delta^*(q, u), v) \tag{2}$$

$$o^*(q, \lambda) = \lambda \qquad o^*(q, \sigma) = o(q, \sigma) \qquad o^*(q, uv) = o^*(q, u) \cdot o^*(\delta^*(q, u), v) \tag{3}$$

We can present transducers pictorially as in Figure 1. Each state in $Q$ becomes a node, with $q_0$ marked by an incoming transition with no label. For each $(q, \sigma) \in \mathrm{dom}(\delta)$, a transition is drawn from $q$ to $\delta(q, \sigma)$ with the label $\sigma : o(q, \sigma)$.

Let $f$ be an initial-empty sequential transduction. The *tail function* of $f$ after reading an input prefix $u$, denoted $f_u$, is defined via $\mathrm{dom}(f_u) = \{v \mid uv \in \mathrm{dom}(f)\}$ and $f_u(v) = f(u)^{-1} \cdot f(uv)$. $f_u$ is an initial-empty sequential transduction itself. Strings $u, v$ are said to be *tail congruent*, denoted $u \sim_f v$, if and only if $f_u = f_v$. Informally, $u \sim_f v$ means that after reading either $u$ or $v$, further output by $f$ will be equivalent. The tail congruence is a right congruence of finite index.

The following hold for all initial-empty sequential transductions $f, g$ and strings $u, v$.

$$
\begin{aligned}
f(\lambda) &= \lambda && && \text{(initial-empty)}\\
u \preceq v &\Longrightarrow f(u) \preceq f(v) && \text{if } v \in \mathrm{dom}(f) && \text{(prefix-closed)}\\
f(uv) &= f(u) \cdot f_u(v) && \text{if } uv \in \mathrm{dom}(f) && \text{(chain rule)}\\
(f \circ g)_u &= f_{g(u)} \circ g_u && \text{if } u \in \mathrm{dom}(g) && \text{(composition rule)}\\
(f_u)_v &= f_{uv} && \text{(concatenating input prefixes of tail functions)}
\end{aligned}
$$

As an aside, the composition rule, which implies that composition cannot cause output strings to be "pushed forward" in the resulting tail functions, does not hold for subsequential transductions. Because the alignment of output strings is crucial for our inference results, we have settled on the sequential class.

Conversely to the algebraic properties, $f$ is an initial-empty sequential transduction if and only if it is initial-empty, it is prefix-closed, and its tail congruence has finite index. In this case, Equation 4 constructs the canonical transducer that recognizes $f$. As a corollary, an initial-empty sequential transduction $f$ is uniquely determined by $\sim_f$ and a character-wise specification of the tails of $f$, that is, whether $u\sigma \in \mathrm{dom}(f)$ and what $f_u(\sigma)$ equals over all contexts $[u]_{\sim_f}$ and characters $\sigma$. This is exactly how we will specify the transductions in our inference results.

$$
\begin{aligned}
Q &= \Sigma^* / \sim_f\\
\mathrm{dom}(\delta) = \mathrm{dom}(o) &= \{([u], \sigma) \mid u\sigma \in \mathrm{dom}(f)\}\\
\delta([u], \sigma) &= [u\sigma]\\
o([u], \sigma) &= f_u(\sigma)\\
q_0 &= [\lambda]
\end{aligned}
\tag{4}
$$

Equation 4 also demonstrates by the specification of $\delta$ and $q_0$ that the tail congruence of $f$ encodes the finite-state structure of the canonical transducer that recognizes $f$. This makes it an adequate representation of our a priori structural information. On the implementation side, a tail congruence can be represented as the transition congruence of a total finite acceptor automaton, which is analogous to the output-empty transducers of Jardine et al. (2014).

Though transducers serve as a computational representation and are helpful for one's intuition, the theory we develop will be grounded in the algebraic characterization of initial-empty sequential transductions. The correspondence in Equation 4 gives all of our inference results immediate transducer-style interpretations, although the transducer-level implementation details are not the focus of this paper. It should also be noted that our inference results (Theorems 8 and 9) do not depend at all on the typical finitary restriction of sequential transductions, and even hold in the presence of infinitely many tail functions. We have outlined the finitary scenario here because it is necessary for algorithmic consideration.

## 4. Faithfulness Conditions on the Target Class of Transductions

As mentioned in Section 1, the simplex condition proposed by Hua and Jardine (2021) is related to linguistic faithfulness in that it limits the amount of change that a transduction

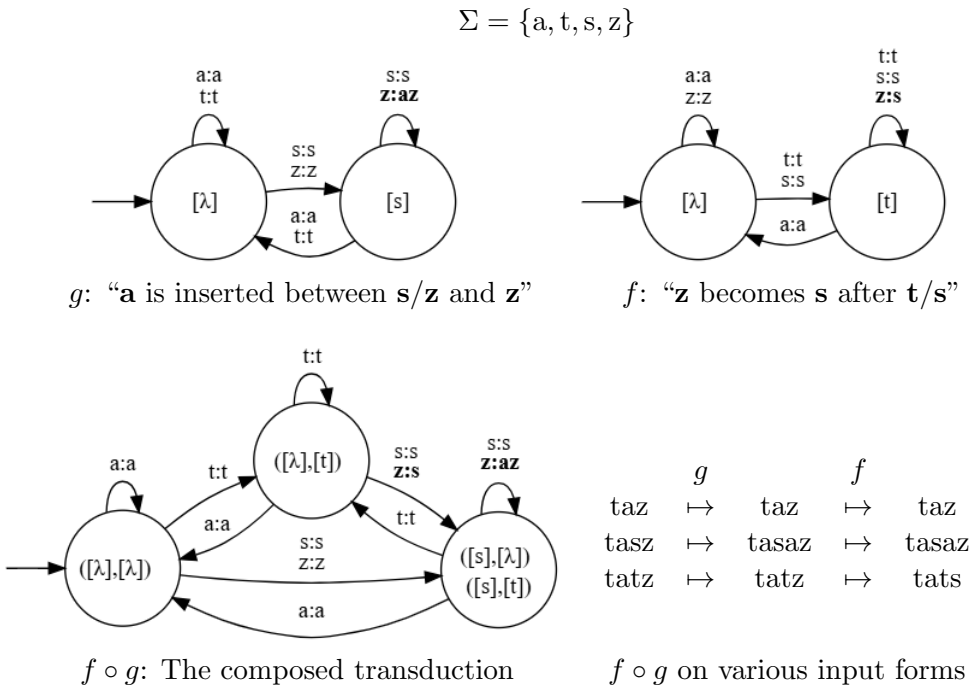



Figure 1: A simple phonological process modeled by the composition of two initial-empty sequential transductions, presented using their canonical transducers, as described by Equation 4. Transitions that realize a change have their labels bolded, and states in $f \circ g$ are labeled with one or more of $([u]_{\sim_g}, [v]_{\sim_f})$ according to their correspondence with states in $g$ and $f$.

may realize. In this section, we adapt their approach by proposing our own *faithfulness conditions*. Definition 1 presents a predicate meant to capture the notion of faithfulness. It allows us to state that no change is realized on a given input in a given context.

**Definition 1 (faithfulness)** *Let $f : \Sigma^* \to \Sigma^*$ be an initial-empty sequential transduction and let $u, v$ be strings. $f$ is faithful to $v$ in the context of $u$, denoted* $\text{faithful}_u^f(v)$*, if and only if for all nonempty prefixes $w\sigma$ of $v$, $f$ maps $\sigma$ to itself after reading $uw$. More precisely,* $\text{faithful}_u^f(v)$ *if and only if:*

$$uv \in \text{dom}(f) \wedge \forall w \in \Sigma^*, \forall \sigma \in \Sigma, w\sigma \preceq v \Longrightarrow f_{uw}(\sigma) = \sigma$$

**Example 1** *In Figure 1,* $\text{faithful}_{\text{t}}^f(\text{az})$ *holds because the input* az *starting from* $[\text{t}]_{\sim_f}$ *only traverses transitions that realize no change, and so we get $f_{\text{t}w}(\sigma) = \sigma$ for all $w\sigma \preceq$ az as desired. In contrast,* $\text{faithful}_{\text{t}}^f(\text{atz})$ *does not hold because the input* atz *starting from* $[\text{t}]_{\sim_f}$ *traverses the* **z:s** *transition, so it turns out that $f_{\text{tat}}(\text{z}) = \text{s}$, which contradicts the definition of* $\text{faithful}_{\text{t}}^f(\text{atz})$.

Lemmas 2 and 3—quantified over initial-empty sequential transducers $f, g$ and strings $u, v$—relate faithfulness to its intended semantics. Note that the converses do not hold in

general; faithfulness makes a character-wise statement that is stronger than the derived string-level equalities.

**Lemma 2** *If* $\text{faithful}_u^f(v)$*, then* $f_u(v) = v$*.*

**Proof** Assuming $\text{faithful}_u^f(v)$, the fact $f_u(w) = w$ for all $w \preceq v$ is by induction on $w$, with the initial-empty rule for the base case and the chain rule for the inductive step. ■

**Lemma 3** *If* $\text{faithful}_{g(u)}^f(g_u(v))$*, then* $(f \circ g)_u(v) = g_u(v)$*.*

**Proof** Lemma 2 gives $f_{g(u)}(g_u(v)) = g_u(v)$, which is as desired by the composition rule. ■

Definition 4 presents the main faithfulness condition that enables our inference results. $f$ being *feeding-faithful* to $g$ means that $f$ realizes no change on the output of any change that was due to $g$.

**Definition 4 (feeding-faithfulness)** *Let* $f, g : \Sigma^* \to \Sigma^*$ *be initial-empty sequential transductions.* $f$ *is* feeding-faithful *to* $g$ *if and only if whenever* $g$ *is not faithful to a character* $\sigma$ *in the context of a string* $u$*, it holds that* $f$ *is faithful to* $g_u(\sigma)$ *in the context of* $g(u)$*. More precisely,* $f$ *is* feeding-faithful *to* $g$ *if and only if:*

$$\forall u \in \Sigma^*, \forall \sigma \in \Sigma, u\sigma \in \text{dom}(f \circ g) \wedge \neg\, \text{faithful}_u^g(\sigma) \Longrightarrow \text{faithful}_{g(u)}^f(g_u(\sigma))$$

Definition 5 and Definition 6 present the *witnessing* conditions, which are meant to enforce that for every property we wish to observe of a transduction, there is some component of the other transduction that allows it observably influence their composition. In the case of Definition 5, $f$ *transition-witnessing* $g$ enforces that for every transition of $g$, there is some context in which $f$ realizes no change on its output.

**Definition 5 (transition-witnessing)** *Let* $f, g : \Sigma^* \to \Sigma^*$ *be initial-empty sequential transductions.* $f$ transition-witnesses $g$ *if and only if for all strings* $u$ *and characters* $\sigma$*, there is a string* $v$ *tail-congruent to* $u$ *where* $f$ *is faithful to* $g_v(\sigma)$ *in the context of* $g(v)$*. More precisely,* $f$ transition-witnesses $g$ *if and only if:*

$$\forall u \in \Sigma^*, \forall \sigma \in \Sigma, u\sigma \in \text{dom}(g) \Longrightarrow \exists v \in [u]_{\sim_g}, \text{faithful}_{g(v)}^f(g_v(\sigma))$$

In the case of Definition 6, $f$ being *change-witnessed* by $g$ enforces that for every change realized by $f$ on some input, there is some context in which $g$ has that input in its image.

**Definition 6 (change-witnessing)** *Let* $f, g : \Sigma^* \to \Sigma^*$ *be initial-empty sequential transductions.* $f$ *is* change-witnessed *by* $g$ *if and only if whenever* $f$ *is not faithful to a character* $\sigma$ *in the context of a string* $u$*, there is a string* $v$ *tail-congruent to* $u$ *where* $v\sigma$ *is in the image of* $g$*. More precisely,* $f$ *is* change-witnessed *by* $g$ *if and only if:*

$$\forall u \in \Sigma^*, \forall \sigma \in \Sigma, u\sigma \in \text{dom}(f) \wedge \neg\, \text{faithful}_u^f(\sigma) \Longrightarrow \exists v \in [u]_{\sim_f}, v\sigma \in \text{img}(g)$$

**Example 2** *In Figure 1:*

- $f$ *is feeding-faithful to* $g$ *because* $\text{faithful}^f_{g(u)}(\text{az})$ *whenever* $u \in [\text{s}]_{\sim_g}$*, exactly where* $g_u(\text{z}) = \text{az}$ *and* $\neg\,\text{faithful}^g_u(\text{z})$ *by the* **z:az** *transition (in fact,* $\text{faithful}^f_v(\text{az})$ *for all* $v$*).*
- $f$ *transition-witnesses* $g$*. We need to show that for every state* $[u]_{\sim_g}$ *and character* $\sigma$*, there is a string* $v \in [u]_{\sim_g}$ *for which* $\text{faithful}^f_{g(v)}(g_v(\sigma))$*. For all* $\sigma$*, the desired* $v$ *at* $[\lambda]_{\sim_g}$ *and* $[\text{s}]_{\sim_g}$ *are* $\lambda$ *and* s *respectively.*
- $f$ *is change-witnessed by* $g$*, as certified by the string* t *since* $\text{tz} \in \text{img}(g)$ *and the inputs* $u \in [\text{t}]_{\sim_f}$ *are exactly where* $\neg\,\text{faithful}^f_u(\text{z})$ *by the* **z:s** *transition.*

Change-witnessing is a relatively permissive condition. When combined with feeding-faithfulness, though, it provides stronger guarantees that will be important for our inference results. Lemma 7 states a stronger characterization of change-witnessing when given feeding-faithfulness, which is that for every change realized by $f$, not only must it be witnessed by some output of $g$, but also by some output of $g$ on which no change was realized. The main idea of the argument is that if the output of $g$ witnessing a change due to $f$ were itself changed by $g$, then it would contradict $f$ being feeding-faithful to $g$.

**Lemma 7 (strong change-witnessing)** *Let* $f, g : \Sigma^* \to \Sigma^*$ *be initial-empty sequential transductions where* $f$ *is feeding-faithful to* $g$*. Then,* $f$ *is* change-witnessed *by* $g$ *if and only if whenever* $f$ *is not faithful to a character* $\sigma$ *in the context of a string* $u$*, there is a string* $g(w)$ *tail-congruent to* $u$ *where* $g$ *is faithful to* $\sigma$ *in the context of* $w$*. More precisely,* $f$ *is* change-witnessed *by* $g$ *in this case if and only if:*

$$\forall u \in \Sigma^*, \forall \sigma \in \Sigma, u\sigma \in \text{dom}(f) \wedge \neg\,\text{faithful}^f_u(\sigma) \Longrightarrow \exists g(w) \in [u]_{\sim_f}, \text{faithful}^g_w(\sigma)$$

**Proof** As for the backward direction of the desired implication, assume the condition stated in the lemma. To show that $f$ is change-witnessed by $g$, let $u \in \Sigma^*$ and $\sigma \in \Sigma$ with $u\sigma \in \text{dom}(f)$, and assume that $\neg\,\text{faithful}^f_u(\sigma)$. It remains to show that $\exists v \in [u]_{\sim_f}, v\sigma \in \text{img}(g)$. We may conclude by our initial hypothesis that there is a $g(w) \in [u]_{\sim_f}$ such that $\text{faithful}^g_w(\sigma)$. Then, $g_w(\sigma) = \sigma$ follows by definition, and we can thereby show that $g(w) \cdot \sigma = g(w\sigma)$. Thus, $g(w) \in [u]_{\sim_f}$ is the desired string such that $g(w) \cdot \sigma \in \text{img}(g)$.

As for the forward direction of the desired implication, assume that $f$ is change-witnessed by $g$. Let $u \in \Sigma^*$ and $\sigma \in \Sigma$ with $u\sigma \in \text{dom}(f)$, and assume that $\neg\,\text{faithful}^f_u(\sigma)$. It remains to show that $\exists g(w) \in [u]_{\sim_f}, \text{faithful}^g_w(\sigma)$. We may conclude by our initial hypothesis that there is a $v \in [u]_{\sim_f}$ such that $v\sigma \in \text{img}(g)$, and so $v\sigma = g(w)$ for some $w \in \Sigma^*$.

Define $z \in \Sigma^*$ as the largest prefix of $w$ such that $g(z)$ is a prefix of $v$, which must exist because, even in the worst case where $z$ is $\lambda$, it holds that $g(z)$—which then equals $\lambda$ because $g$ is initial-empty—is a satisfying prefix of $v$. By the end of the proof, we will demonstrate that, in fact, $g(z)$ must equal $v$. First, observe that $z$ cannot equal $w$, because then we would have $v\sigma \preceq v$ due to $v\sigma = g(w)$ and $g(z) \preceq v$, which is a contradiction. That is, we have the stricter $z \prec w$, so then there is an additional character $\gamma \in \Sigma$ so that we still have $z\gamma \preceq w$. By the end of the proof, we will demonstrate that $\gamma$ is in fact equal to $\sigma$.

Observe now that $g(z\gamma) \preceq v\sigma$ because $g$ is prefix-closed and $g(w) = v\sigma$. So, either $g(z\gamma) = v\sigma$ or $g(z\gamma) \prec v\sigma$. We may rule out the right-hand case because it implies that $g(z\gamma) \preceq v$, which contradicts the maximality of $z$ from its definition. Thus, we conclude the left-hand case, that $g(z\gamma) = v\sigma$.

Assume for the sake of contradiction that $\neg\,\text{faithful}^g_z(\gamma)$. It follows from our $u\sigma \in \text{dom}(f)$ that $\sigma \in \text{dom}(f_u)$, so $\sigma \in \text{dom}(f_v)$ by $u \sim_f v$, and finally $v\sigma \in \text{dom}(f)$. It can now be shown that $z\gamma \in \text{dom}(f \circ g)$ since $g(z\gamma) = v\sigma$. Hence, we may conclude from $f$ being feeding-faithful to $g$ that $\text{faithful}^f_{g(z)}(g_z(\gamma))$. Observe that our $v\sigma = g(z\gamma)$ is equivalent to $v\sigma = g(z) \cdot g_z(\gamma)$, which is itself equivalent to $(g(z)^{-1} \cdot v) \cdot \sigma = g_z(\gamma)$ because $g(z) \preceq v$. Applying the definition of $\text{faithful}^f_{g(z)}(g_z(\gamma))$ to this $(g(z)^{-1} \cdot v) \cdot \sigma \preceq g_z(\gamma)$, we get $f_{g(z)\cdot(g(z)^{-1}\cdot v)}(\sigma) = \sigma$, which simplifies to $f_v(\sigma) = \sigma$. Then, $f_v(\sigma) = \sigma$ is equivalent to $f_u(\sigma) = \sigma$ because $u \sim_f v$, the latter of which yields $\text{faithful}^f_u(\sigma)$ by definition. This is the desired contradiction of our initial hypothesis $\neg\,\text{faithful}^f_u(\sigma)$. We have thus shown by contradiction that $\text{faithful}^g_z(\gamma)$, from which $g_z(\gamma) = \gamma$ follows by definition.

Now, our $v\sigma = g(z\gamma)$ is equivalent to $v\sigma = g(z) \cdot \gamma$, from which we may finally conclude $v = g(z)$ and $\sigma = \gamma$. By substituting the former into our $v \in [u]_{\sim_f}$ and the latter into our $\text{faithful}^g_z(\gamma)$, it turns out that $g(z) \in [u]_{\sim_f}$ is the desired string such that $\text{faithful}^g_z(\sigma)$. ■

## 5. Inferring a Series of Transductions from their Composition

We now present the SFSDLA algorithm for series of total initial-empty sequential transductions $f_1, f_2, \ldots f_n$ that obeys the faithfulness conditions (in the way described by Algorithm 1 specifically). Assuming access to their composition $f_1 \circ f_2 \circ \cdots f_n$ and each of their tail congruences $\sim_{f_i}$, the objective is to infer every transduction $f_i$. Algorithm 1 executes an iterative strategy, repeatedly inferring the next right-hand factor $f_n$ using our *peeling* operation (Theorem 8), and then using $f_n$ to construct a sufficient but maybe-incomplete specification of $f_1 \circ f_2 \circ \cdots f_{n-1}$. By "incomplete specification of $h$", we mean a partial transduction whose domain is a subset of $\text{dom}(h)$ and whose output behavior is the same as $h$ over its domain.

Inferring the right-hand factor is ideal because its transition structure is not obscured by the outputs of other transductions in the composition. To see what is meant by this, observe that in Figure 1, the states in $f \circ g$ have been put into correspondence with the states in the right-hand factor $g$ in such a way that an input traversing to a state $q$ in $g$ will traverse to a state that corresponds with $q$ in $f \circ g$. In contrast, states reached by $u$ in $f \circ g$ are only guaranteed to correspond with states reached by $g(u)$ in $f$, and so we say that the transition structure of $f$ is obscured by the outputs of $g$.

The peeling operation infers this right-hand factor from a maybe-incomplete specification of the composed transduction. The main idea is that if the right-hand factor $g$ realizes a change at state $q$, then $f$ cannot further change its output due to feeding-faithfulness, and thus the same changed output must appear at every state that corresponds to $q$ in $f \circ g$. As an example, in Figure 1, the change of **z** to **az** at the state [s] in $g$ appears equally at the corresponding state $([\text{s}], [\lambda]), ([\text{s}], [\text{t}])$ in $f \circ g$. The converse must hold by transition-witnessing, so we can use this cue to tell whether a change observed in the composed transduction is due to $g$ (case *(a)* in Theorem 8) or due to $f$ (case *(b)* in Theorem 8).

**Theorem 8 (peeling)** *Let $f, g, h : \Sigma^* \to \Sigma^*$ be initial-empty sequential transductions where $g$ is total, $f$ is feeding-faithful to $g$, $f$ transition-witnesses $g \circ \text{img}(h)$, and $g$ is change-witnessed by $h$. Given only the composition $F = f \circ g \circ \text{img}(h)$ and the tail congruence $\sim_g$,*

*one can infer a unique transduction $\hat{g}$, which we will later denote* $\texttt{peel}(F, \sim_g)$*, that is equal to $g$. In particular, $\hat{g}$ is defined by the following character-wise specification of its tail functions $\hat{g}_u$ for all contexts $[u]_{\sim_g}$ and characters $\sigma$.*

1. *If $\forall v \in [u]_{\sim_g}, v\sigma \notin \mathrm{dom}(F)$, then $\hat{g}_u(\sigma) = \sigma$.*
2. *Alternatively, if $\exists v \in [u]_{\sim_g}, v\sigma \in \mathrm{dom}(F)$, then:*
   (a) *If $\forall w \in [u]_{\sim_g}, w\sigma \in \mathrm{dom}(F) \Longrightarrow F_v(\sigma) = F_w(\sigma)$, then $\hat{g}_u(\sigma) = F_v(\sigma)$.*
   (b) *Otherwise, $\hat{g}_u(\sigma) = \sigma$.*

**Proof** We will prove that $\hat{g} = g$ by showing for all strings $u$ and characters $\sigma$ that $u\sigma \in \mathrm{dom}(\hat{g}) \Longleftrightarrow u\sigma \in \mathrm{dom}(g)$ and—if those hold—$\hat{g}_u(\sigma) = g_u(\sigma)$. First, consider case *1.* of the specification, in which $\hat{g}_u(\sigma) = \sigma$ and $\forall v \in [u]_{\sim_g}, v\sigma \notin \mathrm{dom}(f \circ g \circ \mathrm{img}(h))$. It remains to show that $g_u(\sigma) = \sigma$, which is equivalent to $\mathrm{faithful}^g_u(\sigma)$ by definition. Assume for the sake of contradiction $\neg\, \mathrm{faithful}^g_u(\sigma)$. In that case, and because $g$ is change-witnessed by $h$, we may conclude that there is a string $v \in [u]_{\sim_g}$ such that $v\sigma \in \mathrm{img}(h)$. Then, because $f$ and $g$ are total, we have $v\sigma \in \mathrm{dom}(f \circ g \circ \mathrm{img}(h))$, which contradicts our case assumption, as desired.

Now, consider case *2.* of the specification, in which $u\sigma \in \mathrm{dom}(\hat{g})$ and $\exists v \in [u]_{\sim_g}, v\sigma \in \mathrm{dom}(f \circ g \circ \mathrm{img}(h))$. It remains to show that $\hat{g}_u(\sigma) = g_u(\sigma)$. First, consider subcase *(a)* of the specification, in which $\hat{g}_u(\sigma) = (f \circ g \circ \mathrm{img}(h))_v(\sigma)$ and $\forall w \in [u]_{\sim_g}, w\sigma \in \mathrm{dom}(f \circ g \circ \mathrm{img}(h)) \Longrightarrow (f \circ g \circ \mathrm{img}(h))_v(\sigma) = (f \circ g \circ \mathrm{img}(h))_w(\sigma)$. It remains to show that $g_u(\sigma) = (f \circ g \circ \mathrm{img}(h))_v(\sigma)$. We may conclude because $f$ transition-witnesses $g \circ \mathrm{img}(h)$ that there is a string $w \in [u]_{\sim_{g \circ \mathrm{img}(h)}}$ such that $\mathrm{faithful}^f_{(g \circ \mathrm{img}(h))(w)}((g \circ \mathrm{img}(h))_w(\sigma))$. It then follows from Lemma 3 and $u \sim_{g \circ \mathrm{img}(h)} w$ that $(f \circ g \circ \mathrm{img}(h))_w(\sigma) = (g \circ \mathrm{img}(h))_u(\sigma)$. We simplify to $(f \circ g \circ \mathrm{img}(h))_w(\sigma) = g_w(\sigma)$, which is as desired after rewriting according to our subcase assumption.

Now, consider subcase *(b)* of the specification, in which $\hat{g}_u(\sigma) = \sigma$ and $\exists w \in [u]_{\sim_g}, w\sigma \in \mathrm{dom}(f \circ g \circ \mathrm{img}(h)) \wedge (f \circ g \circ \mathrm{img}(h))_v(\sigma) \neq (f \circ g \circ \mathrm{img}(h))_w(\sigma)$. It remains to show that $g_u(\sigma) = \sigma$. Assume for the sake of contradiction that $g_u(\sigma) \neq \sigma$. Because $u \sim_g v \sim_g w$, we also get that $g_v(\sigma) \neq \sigma$ and $g_w(\sigma) \neq \sigma$. These entail $\neg\, \mathrm{faithful}^g_v(\sigma)$ and $\neg\, \mathrm{faithful}^g_w(\sigma)$ by definition. We may conclude from $f$ being feeding-faithful to $g$ that $\mathrm{faithful}^f_{g(v)}(g_v(\sigma))$ and $\mathrm{faithful}^f_{g(w)}(g_w(\sigma))$. It then follows from Lemma 3 and $v \sim_g w$ that $(f \circ g)_v(\sigma) = (f \circ g)_w(\sigma)$, which contradicts our subcase assumption, as desired. ■

For convenience, we will now denote the composition of transductions $f_i \circ f_{i+1} \circ \cdots f_j$ as simply $f_{i:j}$, taking $f_{i:j}$ to be $\texttt{id}$ if $j < i$. Equipped with the peeling operation, Algorithm 1 factorizes a composition of many transductions into given finite-state structures. Its correctness follows easily from Theorem 8 and the invariant $F = f_{1:i} \circ \mathrm{img}(f_{i+1:n})$, which is preserved due to Equation 1. As for the time complexity, the deterministic representation of $F$ grows exponentially in the worst case, and the runtime is only bounded by a polynomial in the sizes of $F$.

**Algorithm 1:** Inferring total transductions $f_1, f_2, \ldots f_n$ from their composition $F$, where for all $i$,
$f_{1:i-1}$ is feeding-faithful to $f_i$,
$f_{1:i-1}$ transition-witnesses $f_i \circ \mathrm{img}(f_{i+1:n})$, and
$f_i$ is change-witnessed by $f_{i+1:n}$.

**Data:** $F$ and the tail congruence $\sim_i$ for each $f_i$
**for** $i \leftarrow n, \ldots 2, 1$ **do**
  $f_i \leftarrow \texttt{peel}(F, \sim_i)$;
  $F \leftarrow F \circ f_i^{-1}$;

## 6. Efficiently Inferring Two Transductions from their Composition

In response to the super-polynomial worst-case time complexity of Algorithm 1, we now present a version of SFSDLA optimized for the case of only two transductions. Unlike Algorithm 1, Algorithm 2 avoids constructing intermediate specifications of composed transductions—whose size could grow exponentially in the worst case—using a *pulling* operation (Theorem 9) to infer the left-hand factor $f$ directly from the inferred right-hand factor $g$ and the composition $f \circ g$.

The pulling operation simply checks $g$ to tell whether a change in the composed transduction is due to $g$ (case *2.* in Theorem 8) or due to $f$ (case *1.* in Theorem 8).

**Theorem 9 (pulling)** *Let $f, g : \Sigma^* \to \Sigma^*$ be initial-empty sequential transductions where $f$ is total, $f$ is feeding-faithful to $g$, and $f$ is change-witnessed by $g$. Given only $g$, the composition $F = f \circ g$ and the tail congruence $\sim_f$, one can infer a unique transduction $\hat{f}$, which we will later denote $\texttt{pull}(F, \sim_f, g)$, that is equal to $f$. In particular, $\hat{f}$ is defined by the following character-wise specification of its tail functions $\hat{f}_u$ for all contexts $[u]_{\sim_f}$ and characters $\sigma$.*

1. *If $\exists g(v) \in [u]_{\sim_f}, \mathrm{faithful}^g_v(\sigma)$, then $\hat{f}_u(\sigma) = F_v(\sigma)$.*
2. *Otherwise, $\hat{f}_u(\sigma) = \sigma$.*

**Proof** We will prove that $\hat{f} = f$ by showing for all strings $u$ and characters $\sigma$ that $\hat{f}_u(\sigma) = f_u(\sigma)$. First, consider case *1.* of the specification, in which $\hat{f}_u(\sigma) = (f \circ g)_v(\sigma)$ and $\exists g(v) \in [u]_{\sim_f}, \mathrm{faithful}^g_v(\sigma)$. It remains to show that $f_u(\sigma) = (f \circ g)_v(\sigma)$. The left-hand side equals $f_{g(v)}(\sigma)$ because $u \sim_f g(v)$, which equals $f_{g(v)}(g_v(\sigma))$ because $\mathrm{faithful}^g_v(\sigma)$, which finally equals the right-hand side by the composition rule, as desired.

Now, consider case *2.* of the specification, in which we have $\hat{f}_u(\sigma) = \sigma$ and $\neg(\exists g(v) \in [u]_{\sim_f}, \mathrm{faithful}^g_v(\sigma))$. It remains to show that $f_u(\sigma) = \sigma$, which is equivalent to $\mathrm{faithful}^f_u(\sigma)$ by definition. Since $f$ is feeding-faithful to $g$, we can use the Lemma 7 characterization of the fact that $f$ change-witnesses $g$. In fact, $\mathrm{faithful}^f_u(\sigma)$ follows exactly from its contrapositive applied to our case assumption, as desired. ■

Equipped with the peeling and pulling operations, Algorithm 2 factorizes a composition of two transductions into given finite-state structures in polynomial time. Its correctness follows easily from Theorem 8 and Theorem 9.

**Algorithm 2:** Inferring total transductions $f$ and $g$ from their composition $F$, where
$f$ is feeding-faithful to $g$,
$f$ transition-witnesses $g$, and
$f$ is change-witnessed by $g$.

**Data:** $F$ and the tail congruences $\sim_1$ and $\sim_2$ for $f$ and $g$ respectively
$g \leftarrow \texttt{peel}(F, \sim_2)$;
$f \leftarrow \texttt{pull}(F, \sim_1, g)$;

## 7. Empirical Coverage and a Linguistic Analysis of Feeding-Faithfulness

In this section, we argue informally that our main feeding-faithfulness condition is related to phonological *feeding* (resp. *bleeding*) *interactions* (Kiparsky, 1968), where the realization of one phonological rule permits (resp. prevents) the realization of another rule. Specifically, we propose that—with the exception of rules with right-hand contexts, which present a challenge to be discussed later—feeding-faithfulness is guaranteed to hold for phonological rules that exhibit a bleeding interaction, whereas the failure of feeding-faithfulness may be stated as a special case of a feeding interaction.

We provide the following arguments in the context of two phonological rules called $A$ and $B$, where $B$ acts on the output of $A$.

1. *The presence of a bleeding interaction between $A$ and $B$ implies that $B$ is feeding-faithful to $A$:* When the realization of $A$ prevents the realization of $B$, changes due to either cannot co-occur on the same input form. In that case, it is impossible for a change due to $B$ to act on the output of a change due to $A$.

2. *$B$ failing to be feeding-faithful to $A$ implies the presence of a feeding interaction between $A$ and $B$:* If $B$ realizes a change on the output of a change due to $A$, then the realization of $A$ was necessary for $B$ to act on its changed output. In other words, the realization of $A$ permits the realization of $B$.

3. Given that $A$ and $B$ exhibit a feeding interaction, all that remains for $B$ to fail to be feeding-faithful to $A$ is for the permitted change due to $B$ to act on the output of the change due to $A$. *Hence,* (in conjunction with argument 2.) *the failure of feeding-faithfulness is exactly that special case of a feeding interaction.*

Note that argument 2. requires one to accept the realization of a change as a necessary condition for acting on its output, an assumption that could be reasonably disputed. In this case, we still propose argument 3. as evidence that failure of feeding-faithfulness is closely related to the presence of feeding interactions.

Another challenge to the empirical coverage of our results is phonological rules with right-hand contexts. The sequential transducers that model such rules tend to realize large amounts of *parasitic change*, that is, transitions that do not produce the change realized by the rule, but that are still not faithful in the sense of Definition 1. As an example, the phonological rule in Figure 2 (Appendix B) must buffer its output when reading **t** at the state [a] until the next character is read, and it is determined whether to output **t** or **ɾ**.

For those transitions where **t** is output, faithfulness does not hold even though the input string undergoes no change overall. Such parasitic change can challenge our faithfulness conditions, which enforce the presence of faithful transitions.

## 8. Conclusion

This paper studies the decomposition learning (Hua and Jardine, 2021) of series of sequential transductions under conditions motivated by linguistic faithfulness, and assuming a priori structural information in the sense of Jardine et al. (2014). Our main contribution is the decomposition learning algorithm SFSDLA, which takes advantage of our proposed faithfulness conditions to uniquely decompose sequential transductions into given finite-state structures. SFSDLA is easily combined with an in-the-limit learner for sequential transductions such as OSTIA (Oncina et al., 1993) to produce an in-the-limit learner for series of sequential transductions given their individual finite-state structures and data from their composition (the precise construction is detailed in Appendix C). We suggest application of our results to generative phonology and linguistic reconstruction, where identification of phonological rules from natural language data is a primary objective.

The main limitations for empirical application of our results are the faithfulness conditions and the need for structural information. Our qualitative analysis of the feeding-faithfulness condition suggests that our results are best fit for phonological processes that exhibit bleeding interactions, but may be limited for processes that exhibit feeding interactions or have right-hand contexts. Our work would benefit from a quantitative analysis of the coverage of feeding-faithfulness in phonological processes attested in world languages, but it is beyond the scope of this paper and left as a future direction.

While this work focuses on the inference of output strings given structural information, we see inference of such structural information as a compelling orthogonal direction. In this case, it may be helpful to examine a generalization of the right-factorization problem addressed by the peeling operation in this paper, namely the problem of inferring a *common* right factor of multiple sequential transductions. We predict that the presence of information from multiple transductions would assist in the inference of ideal finite-state structures for decomposition. Fortunately, this generalization is relevant to the task of linguistic reconstruction, where multiple transductions with a common right factor can model the evolution of a common ancestral language into multiple daughter languages.

## Acknowledgements

We thank the reviewers for their thoughtful comments, which informed revision and strengthened the paper in its current form.

## References

Jane Chandlee and Jeffrey Heinz. Strict locality and phonological maps. *Linguistic Inquiry*, 49(1):23–60, 01 2018. ISSN 0024-3892. doi: 10.1162/LING_a_00265. URL https://doi.org/10.1162/LING_a_00265.

Jane Chandlee, Rémi Eyraud, and Jeffrey Heinz. Learning strictly local subsequential functions. *Transactions of the Association for Computational Linguistics*, 2:491–503, 2014. doi: 10.1162/TACL_A_00198. URL https://doi.org/10.1162/tacl_a_00198.

Colin de la Higuera. *Grammatical Inference: Learning Automata and Grammars*, chapter 7, pages 143–172. Cambridge University Press, 2010. ISBN 9780521763165. URL https://doi.org/10.1017/CBO9781139194655.

E. Mark Gold. Language identification in the limit. *Information and Control*, 10(5): 447–474, 1967. doi: 10.1016/S0019-9958(67)91165-5. URL https://doi.org/10.1016/S0019-9958(67)91165-5.

Yiding Hao and Samuel Andersson. Unbounded stress in subregular phonology. In Garrett Nicolai and Ryan Cotterell, editors, *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 135–143, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-4216. URL https://aclanthology.org/W19-4216/.

Jeffrey Heinz, Chetan Rawal, and Herbert G. Tanner. Tier-based strictly local constraints for phonology. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA - Short Papers*, pages 58–64. The Association for Computer Linguistics, 2011. URL https://aclanthology.org/P11-2011/.

Wenyue Hua and Adam Jardine. Learning input strictly local functions from their composition. In Jane Chandlee, Rémi Eyraud, Jeff Heinz, Adam Jardine, and Menno van Zaanen, editors, *Proceedings of the 15th International Conference on Grammatical Inference, 23-27 August 2021, Virtual Event*, Proceedings of Machine Learning Research, pages 47–65. PMLR, 2021. URL https://proceedings.mlr.press/v153/hua21a.html.

Adam Jardine, Jane Chandlee, Rémi Eyraud, and Jeffrey Heinz. Very efficient learning of structured classes of subsequential functions from positive data. In Alexander Clark, Makoto Kanazawa, and Ryo Yoshinaka, editors, *Proceedings of the 12th International Conference on Grammatical Inference, ICGI 2014, Kyoto, Japan, September 17-19, 2014*, JMLR Workshop and Conference Proceedings, pages 94–108. JMLR.org, 2014. URL http://proceedings.mlr.press/v34/jardine14a.html.

Paul Kiparsky. Linguistic universals and linguistic change. In Emmon W. Bach and Robert Thomas Harms, editors, *Universals in Linguistic Theory*, pages 170–202. Holt, Rinehart, and Winston, 1968.

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

José Oncina, Pedro García, and Enrique Vidal. Learning subsequential transducers for pattern recognition interpretation tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(5):448–458, 1993. doi: 10.1109/34.211465. URL https://doi.org/10.1109/34.211465.

Marcel Paul Schützenberger. Sur une variante des fonctions sequentielles. *Theoretical Computer Science*, 4(1):47–57, 1977. doi: 10.1016/0304-3975(77)90055-X. URL https://doi.org/10.1016/0304-3975(77)90055-X.

Odile Vaysse. Addition molle et fonctions p-locales. *Semigroup Forum*, 34:157–176, 1986-1987. URL http://eudml.org/doc/134836.

## Appendix A. Coherence of Right Inverses and Proof of Equation 1

In Section 2, we defined $h \circ g^{-1}$ to be $f \circ \mathrm{img}(g \circ L)$ whenever $h = f \circ g \circ L$. It remains to show that this operation is single-valued. That is, if $h$ has multiple valid decompositions,

$$h = f \circ g \circ L \qquad\qquad h = f' \circ g \circ L'$$

then the result of $h \circ g^{-1}$ should not depend on which decomposition is used:

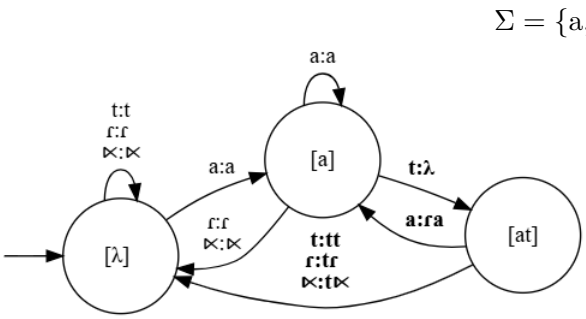
$$f \circ \mathrm{img}(g \circ L) = f' \circ \mathrm{img}(g \circ L')$$

It suffices to fix $u \in \mathrm{dom}(f \circ \mathrm{img}(g \circ L))$ and prove $(f \circ \mathrm{img}(g \circ L))(u) = (f' \circ \mathrm{img}(g \circ L'))(u)$ since the reverse direction is symmetric. Well, $u \in \mathrm{dom}(f)$ and $u \in \mathrm{img}(g \circ L)$ by definition. Then, there exists a $v \in L$ such that $u = g(v)$. Thus, $v \in \mathrm{dom}(f \circ g \circ L)$ and also $v \in \mathrm{dom}(f' \circ g \circ L')$. As a result, $v \in L'$ and hence $u \in \mathrm{img}(g \circ L')$. This is all we need to complete the desired calculation.

$$\begin{aligned} & (f \circ g \circ L)(v) = (f' \circ g \circ L')(v) \\ \implies \quad & f(g(v)) = f'(g(v)) \\ \implies \quad & f(u) = f'(u) \\ \implies \quad & (f \circ \mathrm{img}(g \circ L))(u) = (f' \circ \mathrm{img}(g \circ L'))(u) \end{aligned}$$

We now prove Equation 1, which amounts to showing that $\mathrm{img}(f \circ \mathrm{img}(g)) = \mathrm{img}(f \circ g)$. As for the forward direction, assume $u \in \mathrm{img}(f \circ \mathrm{img}(g))$. That is, there is a $v \in \mathrm{img}(g)$ such that $u = f(v)$. Well, the former implies that there is a $w$ such that $v = g(w)$. Thus, $u = f(g(w))$ and $u \in \mathrm{img}(f \circ g)$ as desired. As for the backward direction, assume $u \in \mathrm{img}(f \circ g)$. That is, there is a $w$ such that $u = f(g(w))$. Clearly, $g(w) \in \mathrm{img}(g)$, so $u \in \mathrm{img}(f \circ \mathrm{img}(g))$ as desired.

## Appendix B. A Phonological Rule with a Right-Hand Context




$h$: "**t** becomes **ɾ** between **a** and **a**"

| | $h$ | |
|---|---|---|
| tat⋉ | ↦ | tat⋉ |
| tata⋉ | ↦ | taɾa⋉ |
| tatta⋉ | ↦ | tatta⋉ |

$h$ on various input forms

Figure 2: A simplified model of English flapping, a phonological rule with a right-hand context, presented using its canonical transducer, as described by Equation 4. ⋉ is the end-of-word marker. Transitions that realize a change have their labels bolded. The transitions labeled **t:**$\lambda$, **t:tt**, **ɾ:tɾ**, **⋉:t⋉** realize parasitic change in the sense of Section 7.

## Appendix C. The Learning Result

We briefly recall in-the-limit identification in the sense of Gold (1967). Consider a set of representations $\mathbb{R}$, a set of target functions $\mathbb{F} \subseteq \{f : X \to Y\}$, and a surjective naming function $\mathcal{L} : \mathbb{R} \to \mathbb{F}$. A sample of a function $f \in \mathbb{F}$ is a set $S \subseteq X \times Y$ for which $f(u) = v$ for all $(u, v) \in S$. A learning algorithm is a computable function $\mathfrak{U}$ that infers for any set of data $S \subseteq X \times Y$ a representation $\mathfrak{U}(S) \in \mathbb{R}$. A sample $C$ of $\mathcal{L}(r)$ is said to be (strongly) characteristic of representation $r$ if and only if $\mathfrak{U}(S) = r$ for any sample $S$ of $\mathcal{L}(r)$ for which $C \subseteq S$. Finally, a learning algorithm $\mathfrak{U}$ is said to be a (strong) in-the-limit learner if and only if every $r \in \mathbb{R}$ has a characteristic sample with respect to $\mathfrak{U}$.

We now describe the learning problem that SFSDLA intends to solve. Assume access to an in-the-limit learner $\check{\mathfrak{U}}$ for initial-empty sequential transductions represented as canonical initial-empty sequential transducers. It is well known that these transducers are canonical in the sense that the naming function $\check{\mathcal{L}}$ is a bijection, not merely a surjection. Next, fix a distinguished series of tail congruences $\sim_1, \sim_2, \cdots \sim_n$, which will be abbreviated as $\sim$ for convenience. In our learning problem, the target functions $\mathbb{F}$ will consist of all $F$ of the form $F = f_1 \circ f_2 \circ \cdots f_n$, where each $f_i$ is an initial-empty sequential transduction whose tail congruence is $\sim_i$, and with the series $f_1, f_2, \ldots f_n$ satisfying the faithfulness conditions outlined by Algorithm 1. The representations $\mathbb{R}$ will be series of canonical initial-empty subsequential transducers $r_1, r_2, \ldots r_n$, with the naming function $\mathcal{L}$ acting on them by way of

$$\mathcal{L}(r_1, r_2, \ldots r_n) = \check{\mathcal{L}}(r_1) \circ \check{\mathcal{L}}(r_2) \circ \cdots \check{\mathcal{L}}(r_n)$$

Note that $\mathcal{L}$ is indeed a surjection because the domain is defined so that any $F$ may be decomposed into an expression of the form above, and because $\check{\mathcal{L}}$ is a surjection. Furthermore,

$\mathcal{L}$ is an injection because decompositions of the form above are guaranteed to be unique by Theorem 8, and because $\check{\mathcal{L}}$ is an injection. Hence, $\mathcal{L}$ is a bijection in the same way as $\check{\mathcal{L}}$.

Lastly, we define a learning algorithm for this problem. Here, SFSDLA may refer to either of Algorithm 1 or Algorithm 2, depending on the size of $n$.

$$\mathfrak{U}(S) \triangleq \text{SFSDLA}(\check{\mathfrak{U}}(S), \sim)$$

To establish that $\mathfrak{U}$ is indeed an in-the-limit learner, it suffices to show that any sample characteristic of $\check{\mathcal{L}}^{-1}(F)$ with respect to $\check{\mathfrak{U}}$ is also characteristic of $\mathcal{L}^{-1}(f_1 \circ f_2 \circ \cdots f_n)$ with respect to $\mathfrak{U}$. Well, suppose that $C$ is characteristic of $\check{\mathcal{L}}^{-1}(F)$. Fixing another sample $S$ of $F$ for which $C \subseteq S$, it remains to show that $\mathfrak{U}(S) = \mathcal{L}^{-1}(f_1 \circ f_2 \circ \cdots f_n)$. By the assumption that $C$ is characteristic, we have $\check{\mathfrak{U}}(S) = \check{\mathcal{L}}^{-1}(F)$, and then the desired $\mathfrak{U}(S) = \mathcal{L}^{-1}(f_1 \circ f_2 \circ \cdots f_n)$ follows from the correctness of SFSDLA.