Weakly deterministic transformations are subregular

Andrew Lamont
University of Massachusetts Amherst
alamont@linguist.umass.edu

Charlie O'Hara
University of Southern California
charleso@usc.edu

Caitlin Smith
Johns Hopkins University
csmit372@jhu.edu

Abstract

Whether phonological transformations in general are subregular is an open question. This is the case for most transformations, which have been shown to be subsequential, but it is not known whether weakly deterministic mappings form a proper subset of the regular functions. This paper demonstrates that there are regular functions that are not weakly deterministic, and, because all attested processes so far studied are weakly deterministic, supports the subregular hypothesis.

1 Introduction

Phonological transformations, i.e., mappings from underlying representations onto surface representations, are computationally regular (Johnson, 1972; Kaplan and Kay, 1994). Most phonological transformations have further been shown to belong to the *subsequential* classes, which form a proper subset of the regular relations (Oncina et al., 1993; Mohri, 1997). These include *strictly local* transformations (Chandlee, 2014; Chandlee and Heinz, 2018), and long-distance transformations where target segments may depend on information arbitrarily far away in one direction (Chandlee and Heinz, 2012; Chandlee et al., 2012; Gainor et al., 2012; Heinz and Lai, 2013; Payne, 2014, 2017; Luo, 2017).

Two classes of phonological transformations have been shown not to be subsequential. The first, *weakly deterministic* transformations (Heinz and Lai, 2013), comprise mostly long-distance bidirectional processes such as root-controlled vowel harmony. For such transformations, some targets depend on information arbitrarily far to their left, such as suffix vowels assimilating to root vowels, and others depend on information arbitrarily far to their right, such as prefix vowels assimilating to root vowels. These transformations have phonologically intuitive decompositions into a left-subsequential transformation and a related right-subsequential transformation, such as treating regressive and progressive harmony from the root as distinct processes.

Here, we make use of the original definition of weakly deterministic transformations as proposed by Heinz and Lai (2013). Note that this definition differs crucially from the revised definition proposed by McCollum et al. (2018), which properly captures the bidirectional harmony cases discussed above, without capturing any of the second class of mappings.

The second class of non-subsequential transformations are the *unbounded circumambient* transformations (Jardine, 2016a; McCollum et al., 2018), which are characterized by target segments depending on information both arbitrarily far to their left and arbitrarily far to their right. Unlike bidirectional harmonies, these processes do not decompose into phonologically intuitive transformations, and are conjectured not to be weakly deterministic (Heinz and Lai, 2013; Jardine, 2016a). However, every unbounded circumambient process studied so far has been shown to be weakly deterministic either by taking advantage of the alphabet (Graf, 2016) or by using predictable substrings as markup (McCollum et al., 2018; O'Hara and Smith, 2018, 2019; Lamont, 2019; Smith and O'Hara, 2019).

To make this concrete, we illustrate the latter strategy for unbounded tonal plateauing (UTP). In UTP, all tone-bearing units must surface with high tone if there is a high tone somewhere to their left and a high tone somewhere to their right. For example, in Luganda, the input /mu-tém-a-bi-sikí/ maps onto [mùtémábísíkí] 'log-chopper' (Hyman and Katamba, 2010; Jardine, 2016a), with all vow-
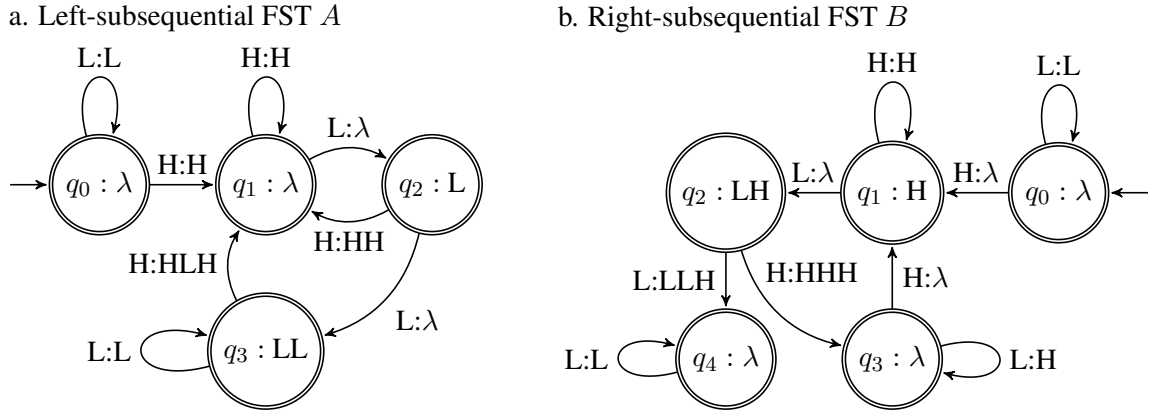
a. Left-subsequential FST A

b. Right-subsequential FST B

Figure 1: Weakly deterministic UTP: $B(A(i)) = UTP(i)$.

els between the high tones of /-tém-/ 'to chop' and /-sikí-/ 'log' surfacing with high tones.

Because there is no bound on how far the left or right triggering high tones can be from a given target, UTP cannot be modeled by a subsequential finite state transducer (FST) (Karttunen (2003);[1] see Jardine (2016b) for a formal proof). While no subsequential FST can identify which tone-bearing units are *targets*, one can identify which high tones are *triggers*. Marking triggering high tones makes it possible for a second subsequential FST to read the string in the opposite direction and correctly identify the targets.

This markup strategy is implemented by the FSTs in Figure 1. The left-subsequential FST A (1a) marks which high tones are triggers, and the right-subsequential FST B (1b) uses this information to complete the transformation. Because the structural description for UTP is symmetrical, it is arbitrary that the first machine reads left-to-right. A machine reading right-to-left would do equally well to markup inputs. Figure 2 illustrates a derivation mapping an input i /HLHHLLLLH/ onto an output with all high tones; the symbols ⋊ and ⋉ are used to explicitly mark the left and right

word boundaries, respectively.

A makes the first pass through i, removing and inserting the substring HLH. It maps HLH onto HH̲H (2a), modeling UTP in a local context. Following the first high tone, A prefixes every high tone span with HLH: H...LLH → H...HLH (2b). Because all HLH substrings that were present in the input have been removed, HLH substrings now only appear in contexts where another high tone is arbitrarily far to their left. This unambiguously encodes the unbounded context for UTP: H...H.

B makes the second pass through i, interpreting the markup left by A: HLH is an instruction to start or continue spreading high tone, and LLH is an instruction to stop. B maps HLH onto HH̲H, and spreads the high tone leftwards until another high tone: $HL^nHLH \to HH̲^nHH̲H$ (2c). This repeats until B reaches the left end of the string.

We present this analysis only to demonstrate that UTP is weakly deterministic according to the definition given by Heinz and Lai (2013): UTP is a regular function that can be decomposed into a left-subsequential transformation and a right-subsequential transformation, where the first mapping is both length- and alphabet-preserving. Encoding instructions in intermediate representations is strikingly unphonological and is not intended as a plausible interpretation of the process.

Similar analyses have shown that all unbounded circumambient processes studied so far are in fact weakly deterministic: high tone spreading in Copperbelt Bemba (McCollum et al., 2018; O'Hara and Smith, 2018, 2019; Smith and O'Hara, 2019), vowel harmony in Tutrugbu (McCollum et al., 2018), and Sour Grapes spreading (Lamont, 2019). At present, then, there are no exceptions to

---

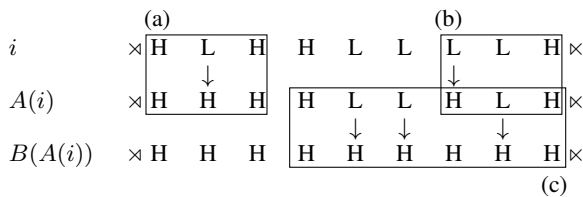[1] We are grateful to an anonymous reviewer for drawing our attention to this paper.

Figure 2: Example of weakly deterministic UTP.

the hypothesis that phonological transformations are weakly deterministic (Heinz and Lai, 2013). However, it is not known whether the weakly deterministic class is a proper subset of the regular functions. If it is, then there are no exceptions to the hypothesis that phonological transformations are *subregular* (Heinz, 2018). If it is not, then some weaker hypothesis, such as phonological transformations being regular, holds.

In this paper, we show that there are regular functions that are not weakly deterministic, supporting the subregular hypothesis. Section 2 presents two such mappings, variations of attested transformations. Section 3 generalizes the class of weakly deterministic unbounded circumambient transformations. Section 4 concludes.

2 Non-weakly deterministic regular functions

This section presents two regular functions that are not weakly deterministic: *first-last UTP* and *double-edged spread*. Both are variations on attested phonological transformations analyzed by Jardine (2016a), and both are defined over a binary alphabet of high tones H and low tones L. If the hypothesis that phonological transformations are weakly deterministic is correct, then neither should exist in natural language phonology.

2.1 First-last UTP

First-last UTP is a variation on UTP where plateauing only occurs if the two high tone triggers are at the word edges.[2] That is, inputs that begin and end with high tones surface with all high tones, e.g., /HLLLH/ → [HHHHH], and inputs that begin or end with low tones surface faithfully, e.g., /HLLHL/ → [HLLHL].

First-last UTP is a regular function, which is modeled by the non-deterministic FST in Figure 3. From the start state q_0, if the first symbol read is a low tone, then the FST transitions to q_1, and writes out the rest of the input faithfully. If the first symbol read is a high tone, then the machine transitions to q_2, where it must decide what to do with following low tones. It can either take the upper path to q_4, where low tones are replaced with high tones, or take the lower path to q_3, where low tones are written faithfully. If the upper path is taken, the input must end with a high tone for the
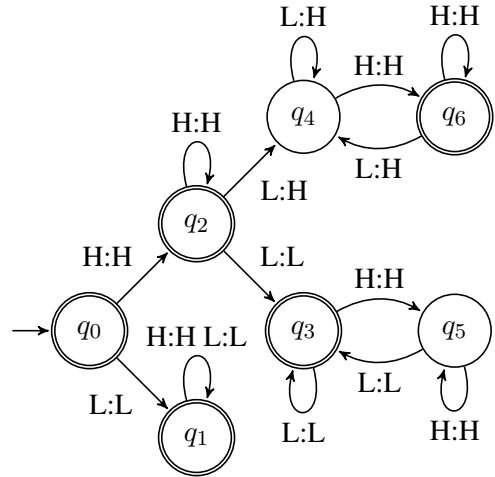
Figure 3: FST for first-last UTP.

FST to accept the mapping, and if the lower path is taken, the input must end with a low tone.

First-last UTP is not subsequential. In strings with initial and final high tones, the triggers circumscribe an unbounded number of targets, which no subsequential FST can identify. We showed in the previous section that a subsequential FST can not only identify the context for spreading in UTP, but one can also unambiguously mark it. This is not the case for first-last UTP: while a subsequential FST can identify the context for spreading, i.e., ⋊H…H⋉, marking it up is impossible.

The markup strategy for UTP exploits the fact that the string HLH never surfaces; it is always mapped onto HHH. Because the first FST removes all instances of HLH that were present in the input, the second FST knows that any remaining HLH strings encode the context for spreading. Furthermore, because the markup string overwrites segments that will be neutralized, there is no harm in changing them to HLH. In first-last UTP, every string surfaces faithfully in some context, such as ⋊L{H, L}*L⋉. There is no string that always neutralizes, and there is no guarantee that changes introduced by the first FST can be undone by the second FST. Thus, some strings that should surface faithfully will instead surface with markup.

This boils down to a pigeonhole argument. Because subsequential FSTs can only model functions that are unbounded on one side, the context for spreading must be identified within k segments from one of the word edges: ⋊H…{H, L}$^{k-1}$H⋉ reading left-to-right, or ⋊H{H, L}$^{k-1}$…H⋉ reading right-to-left. This leaves k segments for markup, and, over the binary alphabet $\Sigma = \{$H,

L}, a total of 2^k possible strings. There are not enough strings to unambiguously encode inputs that should surface faithfully. Setting aside one string to encode spreading leaves $2^k - 1$ strings for non-spreading contexts. Assuming the first FST is left-subsequential, it has to encode 2^k suffixes from faithful inputs with initial low tones $\rtimes$L...{H, L}$^k\ltimes$ and 2^{k-1} suffixes from those with initial high tones $\rtimes$H...{H, L}$^{k-1}$L$\ltimes$. Some faithful input that ends with the designated markup string will have to be changed so that the second FST does not overwrite it with all high tones. However, it is impossible for the second FST to know that it was changed. Thus, at least two inputs that should surface unchanged will be incorrectly mapped onto the same output. The markup strategy is impossible, and first-last UTP is not weakly deterministic.

2.2 Double-edged spread

In Copperbelt Bemba, the rightmost high tone spreads unboundedly far to the right edge of the word, and other high tones undergo local spreading (Bickmore and Kula, 2013; Kula and Bickmore, 2015; Jardine, 2016a). For example, the high tone of the subject marker /bá-/ spreads to the end of a word when no high tone follows, as in /bá-ka-mu-londolol-a/ → [bákámúlóóndólólá] 'they will introduce him/her'. When the locative enclitic /=kó/ is added, its high tone blocks unbounded spreading, and other high tones only spread locally, as in /bá-ka-londolol-a=kó/ → [bákálóòndòlòlàkó] 'they will introduce'. Inputs without high tones surface with all low tones, such as /u-ku-tul-a/ → [ùkùtùlà] 'to pierce'.

As mentioned in Section 1, this mapping is weakly deterministic (McCollum et al., 2018; O'Hara and Smith, 2018, 2019; Smith and O'Hara, 2019), and follows along the same lines as UTP. When the context for unbounded spreading is met, the first FST marks up one of the triggers, either the rightmost high tone or the right edge of the input, or, when the triggers are within a bounded window, performs the entire mapping. For example, if a left-subsequential FST makes the first pass, it markups the right edge because it cannot identify the rightmost high tone. In bounded spreading contexts, where only one segment is targeted, the FST completes the mapping: HL$\ltimes$ → HH̲$\ltimes$. In unbounded spreading contexts, it leaves the string HL as markup: H...LL$\ltimes$ → H...H̲L$\ltimes$.

The right-subsequential FST that makes the second pass interprets a final high tone as an instruction to do nothing, and a final low tone as an instruction to replace low tones until it reaches another high tone: HL^{n}HL$\ltimes$ → HH̲^{n}HH̲$\ltimes$.

As with the string HLH in UTP, HL never surfaces word-finally in Copperbelt Bemba, and is an effective markup string. Similarly, because local spreading guarantees that HL does not surface word-internally, it can be used to markup the triggering high tone when the first FST is right-subsequential, provided it also removes all instances of HL from the input.

Intuitively, the reverse mapping, where the leftmost high tone spreads unboundedly to the left edge of the word and other high tones spread locally, is also weakly deterministic. We are not aware of an attested example in tonal phonology, but Tutrugbu vowel harmony presents a case of unbounded leftward spread (McCollum and Essegbey, 2018; McCollum et al., 2018).

Double-edged spread is a variation on the Copperbelt Bemba mapping where the leftmost high tone spreads unboundedly to the left edge of the word, the rightmost high tone spreads unboundedly to the right edge of the word, and no other tone spreads, e.g., /LLLHLLHLLHLLL/ → [H̲H̲H̲HLLHLLHH̲H̲H̲]. Inputs without high tones surface faithfully, e.g., /LLLL/ → [LLLL].

Double-edged spread is a regular function, which is modeled by the non-deterministic FST in Figure 4. From the start state q_0, if the first symbol read is a low tone, then the FST must decide what to do. It can either transition to state q_1 to anticipate an all low-toned input, or it can transition to q_2 to begin spreading an anticipated
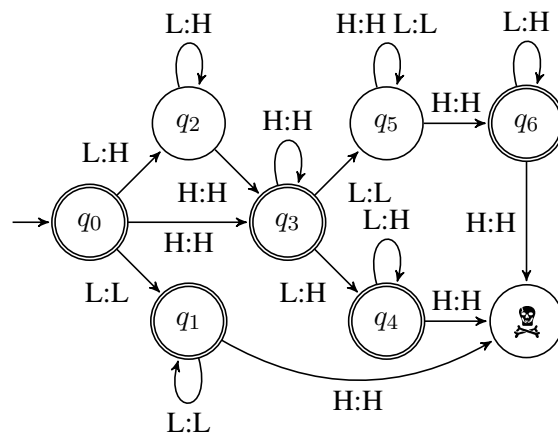

Figure 4: FST for double-edged spread.

high tone. If the FST encounters any high tones in q_1, it transitions into the sink state indicated with the skull and crossbones ☠, and the mapping is not accepted. From q_2, the FST transitions to q_3 once it has identified the leftmost high tone. If the leftmost high tone is word-initial, then the FST transitions directly to q_3 from the start state. Having identified the leftmost high tone, the FST must now identify the rightmost high tone. From q_3, it can either transition to q_4, expecting the string to contain only one high tone span, or transition to q_5 to anticipate a later high tone. From q_5, the FST must eventually transition to q_6 to accept the mapping. Encountering an unexpected high tone in q_4 or q_5 leads to the sink state.

Double-edged spread is not weakly deterministic, but the argument is different than the one for first-last UTP. In double-edged spread, only the leftmost and rightmost high tones spread. Thus, as in first-last UTP, every string surfaces faithfully in some context, such as $\rtimes LH\{H, L\}^*HL\ltimes$. This means that there is no unambiguous way to markup the high tone triggers; only the word edges can be marked up. Because the high tone triggers may be arbitrarily far away, each word edge must be marked up by a subsequential FST that starts at the opposite end of the string: a right-subsequential FST is required to mark the left edge of the string, and a left-subsequential FST is required to mark the right edge of the string. After an edge is marked as the context for spreading, a subsequential FST reading in the opposite direction can complete the mapping. Thus, to spread the leftmost high tone, a right-subsequential FST must make the first pass, marking up the left edge of the string as an instruction to the left-subsequential FST that makes the second pass. Spreading the rightmost high tone requires a left-subsequential FST to make the first pass, and a right-subsequential FST to make the second pass. For both high tones to spread, the two FSTs are paradoxically ordered, and a weakly deterministic analysis is impossible. Modeling double-edged spread with markup requires a third pass by a subsequential FST.

We conjecture that the weakly deterministic class forms a hierarchy as in Figure 5. We call functions where it does not matter whether the first pass is made a left-subsequential FST or a right-subsequential FST *bi-weakly deterministic*. To our knowledge, all attested phonological transforma-

Regular relations
|
Non-deterministic regular functions

Left-weakly deter-   Right-weakly deter-
ministic functions   ministic functions

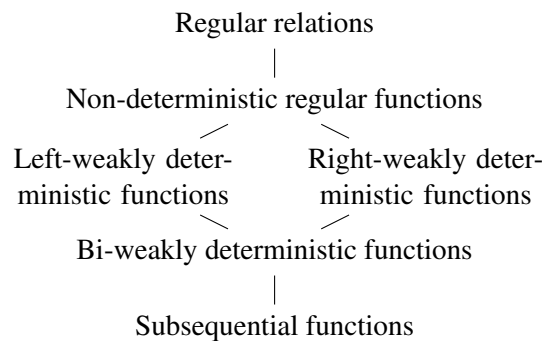Bi-weakly deterministic functions
|
Subsequential functions

Figure 5: Subregular hierarchy of functions.

tions belong to this class. For other mappings, the order is crucial. For example, to model a hypothetical variant of Copperbelt Bemba without local spreading, markup must be placed at the right edge of the word, and so the first pass must be made by a left-subsequential FST. We call such functions *left-weakly deterministic*. The reverse, where the first pass must be made by a right-subsequential FST, we call *right-weakly deterministic*.

3 Weakly deterministic unbounded circumambient mappings

The previous section demonstrated that there are regular functions that are not weakly deterministic. Like the attested transformations analyzed by Jardine (2016a), first-last UTP and double-edged spread are both unbounded circumambient mappings. Alternative definitions of weak determinism have been proposed which properly exclude all unbounded circumambient mappings (Graf, 2016; McCollum et al., 2018), but it is also important to characterize exactly which unbounded circumambient mappings are weakly deterministic because this makes falsifiable predictions. In this section, we present an initial characterization.

Jardine (2016a, 249) defines unbounded circumambient mappings as in Definition 3.1.

Definition 3.1. An UNBOUNDED CIRCUMAMBIENT MAPPING is a mapping for which:

1. Its application is dependent on information (i.e., the presence of a *trigger* or *blocker*) on *both* sides of the target, and

2. *On both sides,* there is no bound on how far this information may be from the target.

Unbounded circumambient mappings can be represented with rewrite rules such as (1). Some

target X is mapped onto Y when circumscribed by triggers L and R. An unbounded number of non-blocking segments N may intervene between the target and the triggers; this set includes targets and transparent segments, and may include other triggers. We assume L and R are not empty strings and are bounded; that is, there exists a k such that k is larger than $|L| + |R|$.

$$X \to Y / L\ N^*\ \_\ N^*\ R \qquad (1)$$

For example, UTP can be represented as in (2). The left and right triggers are both high tones, and the set of non-blocking segments N includes high tones and low tones.

$$L \to H\ /\ H\{H, L\}^*\_\{H, L\}^*H$$
$$\Sigma = \{H, L\},\ N = \{H, L\} \qquad (2)$$

Mappings that involve blocking, such as unbounded spreading in Copperbelt Bemba, can also be represented this way as in (3). The left trigger is a high tone and the right trigger is the right edge of the word. High tones block unbounded spreading, so the set N only contains low tones.

$$L \to H\ /\ HL^*\_L^*\ltimes$$
$$\Sigma = \{H, L\},\ N = \{L\} \qquad (3)$$

Mappings of this type are weakly deterministic if they meet the criteria in Theorem 3.1.

Theorem 3.1. An unbounded circumambient mapping with bounded triggers is weakly deterministic if and only if:

1. There exists at least one bounded substring LXR that is banned from all licit output strings, that is made up of only symbols in the alphabet (no word-boundaries), or

2. There exists at least one bounded substring $\rtimes LXR$ (or $LXR\ltimes$) that is banned from all licit output strings, that contains just one word-boundary, and the triggering substring R (or L) is a blocker, i.e., $R \notin N$ (or $L \notin N$)

The first criterion is exemplified by UTP, where the substring HLH is banned from ever surfacing. The second criterion is exemplified by unbounded spreading in Copperbelt Bemba. The substring HL$\ltimes$ is banned from ever surfacing, and the left

trigger H blocks preceding high tones from undergoing unbounded spreading.

As an aside, we note an intriguing connection between banned substrings and infinite rule schemata proposed by Chomsky and Halle (1968), where rules with unbounded structural descriptions are understood as infinitely many rules with finite contexts. Under that approach, the representation of UTP in (2) would be broken down into the list of rules in (4). The first rule ensures that HLH does not surface, guaranteeing that there is a banned bounded substring.

$$L \to H\ /\ H\_H$$
$$L \to H\ /\ HL\_H$$
$$L \to H\ /\ H\_LH$$
$$L \to H\ /\ HLL\_H$$
$$\cdots \qquad (4)$$

The rest of this section proves Theorem 3.1. Section 3.1 identifies the contexts where a subsequential FST requires more information than it has access to. Section 3.2 discusses the conditions under which those contexts can be disambiguated by another subsequential FST. Sections 3.2.1 and 3.2.2 demonstrate that when the conditions in Theorem 3.1 are met, it is possible for the first FST to smuggle disambiguating information to the second FST. Section 3.2.3 sketches the inverse, that mappings that do not meet the criteria in Theorem 3.1 are not weakly deterministic.

For simplicity, we assume throughout this section that the first pass is made by a right-subsequential FST, and the mapping is completed by a left-subsequential FST. A similar argument can be made by symmetry for the opposite order.

3.1 Identifying ambiguous contexts

Subsequential FSTs cannot model unbounded circumambient mappings (Karttunen, 2003; Heinz and Lai, 2013; Jardine, 2016a). Following the proofs given by Heinz and Lai (2013) and Jardine (2016b), in a left-subsequential mapping, the realization of any target X in the input is predictable from material that may be unboundedly far to its left or at most k segments to its right. In an unbounded circumambient mapping, contexts as simple as LXN^{k+1}, where X maps to Y if an R follows and maps to X otherwise, cannot be identified by a left-subsequential FST. We refer

to these contexts as *ambiguous* because a subsequential FST does not have enough information to correctly determine the output for X.
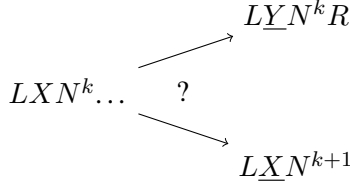
$$LXN^k\ldots \quad \overset{\displaystyle L\underline{Y}N^kR}{\underset{\displaystyle L\underline{X}N^{k+1}}{?}}$$

Figure 6: Ambiguous context for the mapping $X \to Y \mathbin{/} LN^* \_ N^*R$.

There are two types of contexts that may appear in an input, which are defined by the behavior of a target X. We refer to contexts where X is mapped onto itself as *faithful* contexts, and contexts where X is mapped onto some other segment Y as *unfaithful* contexts. In subsequential mappings, every position in the string is unambiguously a faithful or unfaithful context, and changes between contexts depend only on material boundedly far ahead of where the FST is currently printing. For example, in a left-subsequential mapping represented by the rewrite rule $X \to Y \mathbin{/} LN^*\_$, any X that follows an L is mapped onto Y. Thus, the beginning of the input is a faithful context, and everything following an L is an unfaithful context. The first L in the input unambiguously signals the change from a faithful to an unfaithful context. Blocking segments B unambiguously signal the change back to a faithful context.
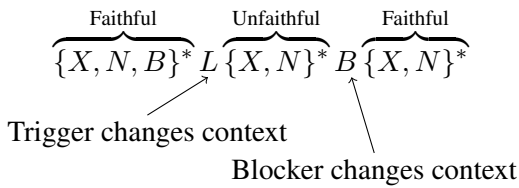
$$\overbrace{\{X,N,B\}^*}^{\text{Faithful}} L \overbrace{\{X,N\}^*}^{\text{Unfaithful}} B \overbrace{\{X,N\}^*}^{\text{Faithful}}$$

Trigger changes context

Blocker changes context

Figure 7: Faithful and unfaithful contexts for the mapping $X \to Y \mathbin{/} LN^* \_$, where $B \notin N$.

Unbounded circumambient mappings are not subsequential because contexts may not be unambiguously faithful or unfaithful. On its own, a trigger L does not unambiguously signal the change from a faithful context to an unfaithful context; it only does so when there is an R somewhere to its right. For a subsequential FST to process such contexts, they must first be *disambiguated*. Thus, for an unbounded circumambient mapping to be weakly deterministic, all ambiguous contexts must

be marked up by a subsequential FST that makes the first pass through the input.

We define ambiguous contexts in Lemma (3.2).

Lemma 3.2. In an unbounded circumambient mapping, ambiguous contexts occur:

1. After any left trigger (L): $L\ldots\_$, and

2. After any right trigger (R) that follows a left trigger without an intervening blocker (unless R itself is a blocker): $L\ldots R\ldots\_$.

Proof. First, we show that ambiguous contexts appear after Ls. The behavior of X in the context LN^*XN^*, depends on following segments. If an R follows, the context is unfaithful: LN^*XN^*R is precisely the structural description of the rule. Otherwise, the context is faithful: LN^*XN^*. Clearly, if $R = L$, R creates ambiguous contexts in the same way.

Next, we show that if $R \neq L$, R creates ambiguous contexts following L if and only if R is not a blocker. The context following an R that is not preceded by an L is unambiguously faithful. If R is a blocker, the context following an R that is preceded by an L is also unambiguously faithful. Because R is a blocker, it is not in N, and therefore, LN^*R cannot meet the structural description of the rewrite rule in (1) regardless of the following context. If R is not a blocker, it is a part of N. Thus, the string $LN^*RN^*\_$ can be rewritten as the prefix of the structural description $LN^*\_$ and segments unboundedly far to the right determine the context. $\square$

3.2 Markup strategies

For an unbounded circumambient mapping to be weakly deterministic, the right-subsequential FST that makes the first pass must disambiguate all potentially ambiguous contexts. Reading right-to-left, this FST is able to use information arbitrarily far to the right, and so it can identify the ambiguous contexts.

Regardless of whether an ambiguous context comes from an L or R, the right-subsequential FST knows the context is an unfaithful context if it consists of a string of non-blocking segments followed by an R, i.e., LN^*R. To provide this information to the left-subsequential FST, Ls must be somehow marked up in these contexts (as must Rs if Rs are not blockers). In order for the marked up information to be useful to the left-subsequential

FST it must either be within k segments of the trigger or arbitrarily far to its left.

There are two crucial cases: either L contains a word-boundary and is therefore unique, or L does not contain a word-boundary, and there may be arbitrarily many of them in the input. We show first that if L does not contain a word-boundary, then there is a substring that can act as a markup. If L does contain a word-boundary, we show that there is a markup substring only if R is a blocker.

3.2.1 L does not contain word-boundaries

Lemma 3.3. If there exists an L and an R that do not contain word-boundaries, the unbounded circumambient mapping is weakly deterministic.

Proof. First, we show that such a mapping can contain a potentially unbounded number of ambiguous contexts, requiring an unbounded number of potential markup locations. If L and R do not contain word-boundaries, for any k, and any m, a word exists containing a substring of the form $(LT^kXT^kRB)^m$, where B is some blocker, and where T is the set of "**T**ransparent segments and **T**argets"—segments that are not left triggers, right triggers, or blockers. The behavior of the Xs in such a word cannot be predicted by a subsequential FST regardless of its direction. The presence of the blocker B means that even if the triggers are non-blockers, the behavior of each X is independently based on the most local L and R.[3] Each ambiguous context requires 1 bit of information indicating whether it is a faithful or unfaithful context. Thus, for the subsequential FST making the first pass to disambiguate ambiguous contexts, at least m bits of markup are required. Because m can grow to become unboundedly large, the amount of markup must also grow as m becomes larger.

As illustrated in Section 1, the right-subsequential FST that makes the first pass must do two things. First, it must map all substrings up to length k of the form LT^*XT^*R to $LT^*\underline{Y}T^*R$ (as well as if L is non-blocking, any substrings up to length k of the form LT^*XT^*L that are followed by R). Given this first action, any substring underlyingly containing LXR will be changed, and, as a result, LXR can be used as markup. In order to transmit the needed m bits of information, markup must be placed after any L

that is in an unfaithful context (that is, followed by a string of non-blockers and then an R), and (if R is a non-blocker) any R unambiguously followed by a faithful context: that is, any R that is not followed by another R. However, if a blocker or other trigger appears within k segments of such a trigger, no markup is needed because the presence of a blocker is sufficient to show that there is a faithful context, and any unfaithful context between two triggers within k segments is handled by the first part of this function.

This leaves L or R followed by an unbounded number of segments that are either targets or transparent segments. We define a k-SUBSTRING OF UNCERTAINTY as any substring in this context, of the shape $(L, R)T^k$. Following an L, the number of possible k-substrings of uncertainty depends on the number of segments in T (specifically, $|T|^k$). A successful markup strategy would replace some of these k segments, but must still contain enough information to reconstruct the string. Any underlying substring of length k that starts with LXR, has been changed already, allowing any substring of that sort to be used as an intermediate markup. LXR is some finite length $j \leq k$. Thus, there are a number of potential markup substrings of the form $LXR\Sigma^{k-j}$, equal to the number of symbols in the alphabet to the $k - j$ power.

For the markup strategy to be successful there need to be at least $|T|^k$ possible markup strings, so that all contrastive k-substrings of uncertainty can be reconstructed, therefore if $|\Sigma|^{k-j} \geq |T|^k$ the process is weakly deterministic. Since the triggers L and R and any blockers B are not in the set T, the non-blocker segments must be less than the full alphabet Σ.[4] If T is a proper subset of Σ, since $|\Sigma| > |T|$,

$$\frac{|\Sigma|}{|T|} > 1 \qquad (5)$$

$|\Sigma|^j$ is bounded by definition, and α^x grows unboundedly if $\alpha > 1$, so there exists a k such that:

$$(\frac{|\Sigma|}{|T|})^k \geq |\Sigma|^j \qquad (6)$$

$$|\Sigma|^{k-j} \geq |T|^k \qquad (7)$$

Therefore, for some k, there are more banned substrings of length k that begin with LXR ($|\Sigma|^{k-j}$), than there are contrastive k-substrings of uncertainty that must be reconstructed ($|T|^k$). □

---

[3]If the process in question lacks blocking segments, the amount of markup necessary can only *decrease*, simplifying the proof of weak determinism.

[4]If all triggers and blockers are longer than one segment, the same logic holds using substrings rather than segments.

3.2.2 L contains word-boundaries

If a word-boundary is included in all potential markup substrings, only processes where there is at most one ambiguous context can be weakly deterministic.

Lemma 3.4. If all left (or right) triggers include a word-boundary, but there exists a right (or left) trigger from the other side that does not include a word-boundary, and it blocks application of the process, the process is weakly deterministic.

Proof. If all left triggers L include the left boundary of the word $\rtimes$, the substring LXR cannot be used freely as a markup. However, this can greatly decrease the number of potential k-substrings of uncertainty in the word, as there can only be one L per word. If right triggers block, then there is no k-substring of uncertainty local to any R in the word. Thus the markup strategy must only encode one additional bit of information not present in the original string, so we can make use of the one potential markup location, the k-substring of uncertainty local to the left edge of the word. In this case, LXR substrings could be used to markup the beginning of the word, as long as an R exists that does not include the end of the word $\ltimes$. This is simply a reversed case of the Copperbelt Bemba markup used by McCollum et al. (2018) and demonstrated in Section 2.2, as only the first R in the word would spread all the way to the beginning of the word.

Like the Copperbelt Bemba case, if the reverse is true – that is, if all right triggers R include the right edge of the word – it is impossible for a right-subsequential FST to markup any information in the string, because the only markup location is at the right edge of the word, before the FST is aware of any left triggers in the word; but as above, if there exists an L that does not include a word-boundary, and L are blockers, the process is weakly deterministic using a left-subsequential FST to markup information on the right side of the word first. $\square$

3.2.3 Summary

In both of the weakly deterministic cases, the number of potential locations for markup strings is at least as many as the number of k-substrings of uncertainty in a word. The cases of non-weakly deterministic unbounded circumambient mappings have a limited number of possible markup locations because all banned substrings include at least one word boundary.

In the first-last UTP case in Section 2.1, there is at most one k-substring of uncertainty possible in a word ($\rtimes$H L^k), but no banned substring that can be placed as markup in that position because all banned substrings include both word-boundaries.

The other types of non-weakly deterministic unbounded mappings can be seen in the double-edged spreading in Section 2.2, or the true sour grapes mapping described in O'Hara and Smith (2019). In each of these mappings, there are potentially unbounded numbers of k-substrings of uncertainty (for double-edged spreading both $\rtimes$L^k and any HL^k), but all banned substrings include a word boundary, restricting the number of possible markup locations to one.

4 Conclusion

This paper demonstrated that the class of weakly deterministic mappings as defined by Heinz and Lai (2013) forms a proper subset of the class of regular functions. This was shown by examining two hypothetical mappings, first-last UTP and double-edged spread, that are regular but not weakly deterministic. The lack of non-weakly deterministic phonological transformations may be demonstrative of an upper bound on the complexity of phonological mappings.

We have also characterized the necessary and sufficient conditions by which an unbounded circumambient mapping is weakly deterministic. This characterization reveals that the set of non-weakly deterministic unbounded circumambient mappings are those that make crucial reference to both edges of the word.

References

Lee S. Bickmore and Nancy C. Kula. 2013. Ternary spreading and the OCP in Copperbelt Bemba. *Studies in African Linguistics*, 42(2):101–132.

Jane Chandlee. 2014. *Strictly local phonological processes*. Ph.D. thesis, University of Delaware.

Jane Chandlee, Angeliki Athanasopoulou, and Jeffrey Heinz. 2012. Evidence for classifying metathesis patterns as subsequential. In *Proceedings of the 29th West Coast Conference on Formal Linguistics*, pages 303–309.

Jane Chandlee and Jeffrey Heinz. 2012. Bounded copying is subsequential: Implications for metathesis and reduplication. In *Proceedings of SIGMORPHON 2012*, pages 42–51.

Jane Chandlee and Jeffrey Heinz. 2018. Strict locality and phonological maps. *Linguistic Inquiry*, 49(1):23–59.

Noam Chomsky and Morris Halle. 1968. *The Sound Pattern of English*. Harper & Row, New York.

Brian Gainor, Regine Lai, and Jeffrey Heinz. 2012. Computational characterizations of vowel harmony patterns and pathologies. In *Proceedings of the 29th West Coast Conference on Formal Linguistics*, pages 63–71.

Thomas Graf. 2016. Weak determinism is not too weak: A reply to Jardine (2016). Unpublished manuscript, Stony Brook University.

Jeffrey Heinz. 2018. The computational nature of phonological generalizations. In Larry M. Hyman and Frans Plank, editors, *Phonological Typology*, pages 126–195. Mouton.

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

Larry M. Hyman and Francis X. Katamba. 2010. Tone, syntax and prosodic domains in Luganda. *ZAS Papers in Linguistics*, 53:69–98.

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

Adam Jardine. 2016b. Computationally, tone is different. *Phonology*, 33(2). Supplemental materials.

C. Douglas Johnson. 1972. *Formal Aspects of Phonological Description*. Mouton, The Hague.

Ronald Kaplan and Martin Kay. 1994. Regular models of phonological rule systems. *Computational Linguistics*, 20(3):331–378.

Lauri Karttunen. 2003. Finite-state technology. In Ruslan Mitkov, editor, *The Oxford Handbook of Computational Linguistics*, pages 339–357. Oxford University Press, Oxford.

Nancy C. Kula and Lee S. Bickmore. 2015. Phrasal phonology in Copperbelt Bemba. *Phonology*, 32(1):147–176.

Andrew Lamont. 2019. Sour Grapes is phonotactically complex. Paper presented at LSA 2019 Annual Meeting.

Huan Luo. 2017. Long-distance consonant agreement and subsequentiality. *Glossa*, 2(52).

Adam G. McCollum, Eric Baković, Anna Mai, and Eric Meinhardt. 2018. The expressivity of segmental phonology and the definition of weak determinism. Unpublished manuscript, University of California San Diego. Available at https://ling.auf.net/lingbuzz/004197.

Adam G. McCollum and James Essegbey. 2018. Unbounded harmony is not always myopic: Evidence from Tutrugbu. In *Proceedings of the 35th West Coast Conference on Formal Linguistics*, pages 251–258.

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

Charlie O'Hara and Caitlin Smith. 2018. Weakly deterministic characterizations of unbounded tonal and featural spreading. Talk at SCAMP 2018.

Charlie O'Hara and Caitlin Smith. 2019. Computational complexity and sour-grapes-like patterns. In *Supplemental Proceedings of the Annual Meeting on Phonology 2018*.

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

Amanda Payne. 2014. Dissimilation as a subsequential process. In *Proceedings of the 44th Meeting of the North East Linguistic Society*, volume 2, pages 79–90, Amherst, MA. Graduate Linguistics Student Association.

Amanda Payne. 2017. All dissimilation is computationally subsequential. *Language*, 93(4):e353–e371.

Marcel-Paul Schützenberger. 1961. A remark on finite transducers. *Information and Control*, 4:185–196.

Caitlin Smith and Charlie O'Hara. 2019. Formal characterizations of true and false sour grapes. In *Proceedings of the Society for Computation in Linguistics 2019*, volume 2, pages 338–341.