A Computational Account of Tone Sandhi Interaction

Jane Chandlee
Haverford College

1 Introduction

This paper presents a computational analysis of three tone sandhi rules in Tianjin Chinese that have received a lot of attention in the previous literature (Chen, 1986; Zhang, 1987; Tan, 1987; Hung, 1987; Chen, 2000; Lin, 2008; Wee, 2010). The interest in these particular rules stems from the seemingly complex manner in which they interact, in that they apply in different directions. As a result, accounting for the interaction in a unified way has been challenging for both rule- and constraint-based approaches.

The computational account presented here offers a way to gauge whether the apparent complexity of the interaction aligns with computational notions of complexity. It also highlights the way each rule's mode of application determines its computational classification. In sum, it will be shown that two of the rules have the computational property of *right output strict locality (ROSL)* while the third is *input strictly local (ISL)*. Furthermore, the combined map of all three rules has the property of being *input-output strictly local (IOSL)*.[1] The terms ROSL and ISL will be defined in the next section; IOSL will be introduced in the analysis that follows. As all three of these properties define properly *subregular* classes of functions, the conclusion is that the tone sandhi rules and their interaction are still relatively simple from a computational perspective.

The paper is organized as follows. Section 2 presents the needed computational background for the analysis to follow, §3 presents the facts from Tianjin Chinese, and §4 surveys the previous analyses of these facts in both rule- and constraint-based approaches. Then §5 presents the computational account, with some discussion of its implications in §6 before §7 concludes.

2 Computational background

The foundation of the computational framework that will be used in the analysis of Tianjin tone sandhi is that rules or processes are functions or *maps* that produce an output string for a corresponding input string according to one or more phonological generalizations. This is still the case even when the rule is described in terms of ranked constraints, as constraint-based grammars likewise map input strings to output strings.

For example, a phonological rule like the epenthesis rule in (1) (insert schwa between two word-initial consonants) represents an infinite set of string pairs, like in (2a), in which the rule has applied to those strings that satisfy the rule's structural description. Another way to think of it is as a function f, that maps a given input to its corresponding output, as in (2b).

(1) $\varnothing \rightarrow$ ə / #C __ C

(2) a. {(ppæt, pəpæt), (kkæt, kəkæt), (pæt, pæt), (kæt, kæt), ... }
b. f(ppæt) = pəpæt

The question of interest is how computationally complex is a function like f? In other words, how much computational power or expressivity is needed to correctly map any possible input string to its output string?

It has been known since Johnson (1972) and Kaplan & Kay (1994) that SPE-style (Chomsky & Halle, 1968) rules like (1) correspond to *regular relations* provided they do not recursively apply to their structural

* Thanks to Chris Oakden and Adam Jardine for bringing this case to my attention, and to the audience at AMP for helpful questions and feedback.

[1] This analysis differs from the one presented at AMP 2018. Thanks to Karthik Durvasula for pointing out an issue with that original analysis.

Proceedings of *AMP 2018*

change. This means that phonological processes require less power than the context-sensitive rewrite rule formalism provides, and that phonological processes are *finite-state describable*.

More recent work (Gainor et al., 2012; Chandlee et al., 2012; Heinz & Lai, 2013; Luo, 2017; Payne, 2017) has pursued the hypothesis that phonological processes require even less power than the regular relations and are in fact describable with properly *subregular* classes of functions. One such class is the *subsequential* functions, which have the added restriction of being *deterministic*.[2] Taking it even further, Chandlee (2014) and Chandlee et al. (2014, 2015) have defined additional classes of functions that are themselves proper subsets of the subsequential functions, in order to better understand the computational nature of various types of phonological phenomena. These classes are the input strictly local (ISL) functions and the output strictly local (OSL) functions. Both are established as proper subsets of the subsequential functions, because in addition to being deterministic they are restricted in the amount of information they can use when deciding what to output. Both of these classes will be introduced first informally, through examples, and then more formally through the finite-state formalism.

Figure 1 presents four examples of phonological maps from an input string (shown on the top of each example) to an output string (shown underneath the input string). The two maps on the left have the property of being ISL while (for contrast) the two on the right do not. The portions of the input strings shown in gray boxes represent the information needed to determine the corresponding portion of the output string (shown in pink). The map ɪNpɑsəbl̩ ↦ ɪmpɑsəbl̩ (i.e., regressive nasal place assimilation) demonstrates that only an input nasal and the following obstruent are relevant to the assimilation; no other information in the string is needed to decide the output at that time. Likewise in the map bɑːd ↦ bɑːt (i.e., final obstruent devoicing) only an obstruent and a following end-of-word marker is needed to decide whether or not to output the obstruent as devoiced. What makes these maps ISL in particular is that 1) this needed information falls into a contiguous 'window' of the input string, and 2) there is a fixed upper bound on the size of that window. In both of these examples, that upper bound is 2, and so the functions are more specifically 2-ISL functions.

Note also in these examples that the input segment that will potentially be altered (the unspecified nasal N and the obstruent 'd') have a λ or 'empty string' underneath them. This represents the fact that if the input string were read one segment at a time starting from the left, at the point these segments are read a decision cannot be made about what their correspondent in the output should be. This is because the information needed to make that decision (the place of the obstruent or the presence/absence of the end-of-word boundary) *follows* the segment in question. The λ then represents a temporary delaying of the output. This is what enables the map to be modeled deterministically, but it can be done only a bounded number of times.

In contrast, in the hypothetical example of final obstruent devoicing shown in the bottom right of the figure, the word must both start *and* end with a voiced obstruent for the devoicing to happen. So the 'window' of information needed to make that decision is no longer contiguous. Likewise, in the top right example of unbounded sibilant harmony, the output for the second sibilant depends on the previous sibilant, which could be an arbitrary number of intervening segments away. So again the window isn't contiguous. Of course all of sɑɑɑʃ and bɑːd could be included in the windows in these examples, but that would violate the second criteria of ISL: that the size of the window has an upper bound. With no principled limit on the length of words, such a bound on the window size cannot be set: a window of size 5 would suffice for sɑɑɑʃ but not for sɑɑɑɑʃ.

The ISL functions have been characterized in a variety of formalisms; in this paper their finite-state representations will be used. A finite-state transducer (FST) for the nasal place assimilation map exemplified by ɪNpɑsəbl̩ ↦ ɪmpɑsəbl̩ is given in Figure 2. For readability, all segments except for {N, B, D} are collapsed to ?, and B, D are abbreviations for bilabial and alveolar obstruents, respectively.[3] FSTs read input strings one symbol at a time and, starting in the start state ⋊, follow the labeled transitions to other states according to the current symbol being read. In the FST in Figure 2, whenever a N is read the FST proceeds to state N and outputs λ (output is shown to the right of the colon on each transition label). From that state, if a bilabial stop is read the output includes the assimilated nasal; likewise if an alveolar is read the output includes the

[2] For more on finite-state models of string-to-string maps and the formal definition of subsequentiality and determinism, see Mohri (1997).

[3] In the complete, unabbreviated FST all segments that could be found in the input string would have their own separate states.

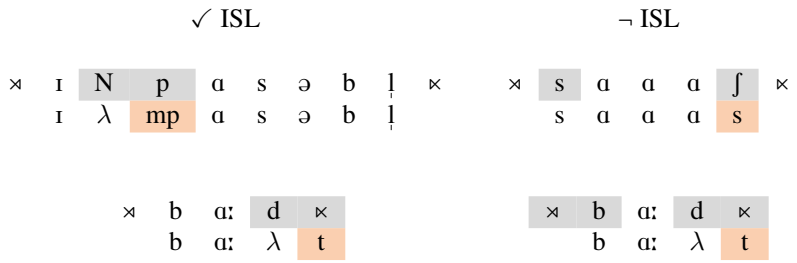

Figure 1: Informal representation of ISL. The examples on the left are 2-ISL functions. The examples on the right cannot be modeled with ISL. ⋊ and ⋉ represent start-of-word and end-of-word, respectively.

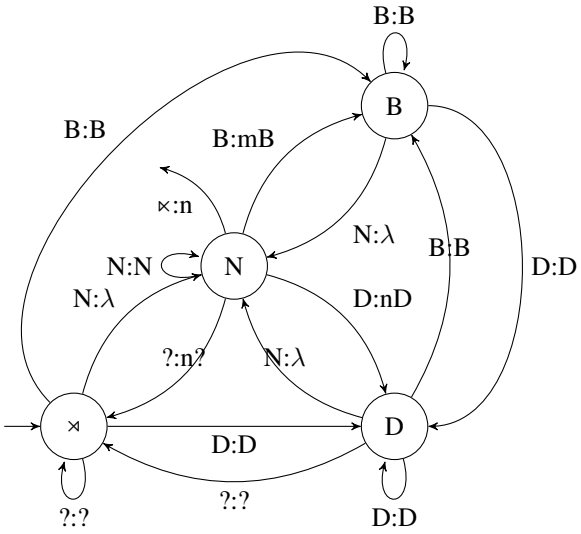

Figure 2: 2-ISL FST for nasal place assimilation. N = unspecified nasal, B = bilabial obstruent, D = alveolar obstruent, ? = all other segments

alveolar nasal.[4] If anything else follows the nasal (i.e., a ?), it is output as 'n' (an assumed default value).

What makes this FST specifically a 2-ISL FST is that the transitions are defined such that the last input segment read exactly determines the state the FST is in (i.e., reading a N the FST goes to the N state, reading a B the FST goes to the B state, etc.). This amounts to a short-term memory restriction in which the FST only 'knows' at any given time what the very last input symbol was. By extension, it can only use that information plus the next input symbol to determine what to output.[5] No more information can be factored in, since the structure of the FST forces any information beyond the previous symbol to be lost.

Generalizing beyond this example, a ISL FST is limited to a portion of the recent input of fixed sized. As already noted, this particular FST is a 2-ISL function, which means the amount of information that can be retained is of length 2-1 or 1 (i.e., the previous symbol). More generally, for a k-ISL function, the states keep track of the previous $k-1$ input symbols, so for example a 4-ISL FST would have states of length up to 3.

Output strict locality (OSL) differs minimally from ISL. Examples to demonstrate the OSL property are progressive (left) and regressive (right) nasal harmony, as shown in Figure 3. The information that can be used to determine what to output is still limited to a window of what came previously, but this time that

[4] The mapping of N to ŋ before velars is also assumed but omitted for readability of the figure, as is the ?:? transition from B to ⋊.

[5] Note the N state has an extra transition with input ⋉ that does not lead to another state. This is the 'final output transition', and its output, in this case 'n', is appended to the current output string iff the end of the input string is reached in the N state. This is needed for inputs that end in N, which otherwise would be missing that final segment because of the N:λ transitions that lead to this state.

Figure 3: Informal representation of OSL. In Left OSL input strings are read from left to right; in Right OSL input strings are read from right to left.

window is split between the most recent output and the currently read input symbol. Thus in progressive nasal harmony the nasality is allowed to perpetuate through the contiguous span of vowels, even though each vowel is getting farther away from the triggering nasal consonant. This is because the most recent output bears the nasal feature. OSL functions differ in whether the input string is read from left-to-right, which is more specifically a Left OSL (LOSL) function, or else the input string is read from right-to-left, which is a Right OSL (ROSL) function.[6] As only the most recently outputted segment is needed to determine whether or not to nasalize an input vowel, both functions in Figure 3 are examples of 2-OSL functions.

Like ISL FSTs, OSL FSTs enforce this limitation on the information that can be retained. The difference is that the state each transition leads to matches its output side, not the input. An example of an OSL FST will be presented in §5.

A last point to raise about the distinction between ISL and OSL is how it relates to the mode of a rule's application. The ability to track the recent output is what allows for 'iterative' or repeated application of the nasalization process in Figure 3. If the function was limited to tracking the recent input, as in ISL, then only one vowel would be nasalized in these examples: the one immediately following the nasal consonant itself. This corresponds to 'simultaneous' rule application.[7] This distinction will greatly inform the computational analysis of Tianjin tone sandhi to follow.

But first, the facts of the Tianjin tone pattern will be presented in the next section, followed by a survey of previous analyses of those facts in §4.

3 Tianjin tone sandhi

Tianjin has four tones: low (L), high (H), rising, and falling. Following Chen (2000) and Wee (2010), in the interest of clarity in what follows R will be used for rising tones and F for falling tones (in lieu of the sequences LH and HL, respectively). In §6 however it will be shown that this notational convenience has a small but inconsequential effect on the conclusions of the computational analysis.

Sequences of two identical tones undergo sandhi, except for HH sequences, which are permitted. Before an identical tone, falling tones become low, low tones become rising, and rising tones become high. These changes are summarized in (3).

(3) a. FF ↦ LF
b. LL ↦ RL
c. RR ↦ HR

In what follows, for convenience, (3a) will be referred to as the 'FF rule', (3b) will be referred to as the 'LL rule', and (3c) will be referred to as the 'RR rule'. An examination of sequences of three tones reveals the distinction in the way these three rules apply.

As shown in (4), the FF and LL rules apply right-to-left. This is seen if we assume a compositional view of rule application, in that the rule applies to a string, and then the resulting string is submitted to the rule a second time. For FFF, for example, starting from the right the middle 'F' meets the conditions for sandhi, and so applying the FF rule gives us FLF. That resulting string no longer satisfies the structural description of the rule and so it is the final output form. Likewise, applying the LL rule right-to-left to LLL gives LRL.

[6] With ISL functions it doesn't matter in which direction the input string is read.

[7] It is worth emphasizing here though that modeling a process with an ISL/OSL-function is not dependent on the rule formalism. As explained above, the function or map is a set of string pairs, so the output of a string like NVVVC in that set is either NṼṼṼC or NṼVVC, which tells us whether the map is OSL or ISL. Noting the mode of 'rule application' is just a way to relate the map to more familiar, theory-internal concepts.

Rule	Direction	FFF	LLL	RRR	RLL	LFF
F→L / __F	right-to-left	FLF	–	–	–	LLF
L→R / __L	right-to-left	–	LRL	–	RRL	RLF
R→H / __R	left-to-right	–	–	HHR	HRL	–
		FLF	LRL	HHR	HRL	RLF

Table 1: Rule-based analysis of Tianjin tone sandhi

(4) a. FFF ↦ FLF
b. LLL ↦ LRL
c. RRR ↦ HHR

The output for RRR, however, does not reflect the same 'alternating' pattern of the other two. This is taken to indicate that the RR rule instead applies left-to-right: first to the first R, giving HRR, and then to the second R, giving HHR. It is this difference in the directionality of the three rules that has caused Tianjin to receive so much attention in the literature, as researchers working in various frameworks have proposed ways to unify the three rules despite the difference in directionality.[8] The next section will present a selection of these accounts.

4 Previous accounts

4.1 *Rule-based accounts* Zhang (1987)'s rule-based analysis of Tianjin sandhi begins with the assumption that (contra the 'implicit assumption' of Chen (1986)), there are multiple rules instead of a single rule that behaves in various ways. From there the analysis is somewhat straightforward, though it relies on the stipulation of each rule's direction of application. Zhang argues that these stipulations in fact follow from general principles, in that right-to-left is the default direction because the sandhi then applies 'away from the determinant' and the left-to-right application of the RR rule is already necessary to account for sandhi in Mandarin.

Tan (1987) likewise assumes there are multiple rules and identifies the same difference in directionality, but also specifies the ordering of the rules based on forms to which multiple rules apply.[9] In particular, as shown in Table 1, the derivation of RLL ↦ RRL ↦ HRL shows that the LL rule feeds the RR rule. The FF and LL rules also interact, but Zhang (1987) and Tan (1987) report different interactions. Zhang (1987) reports the output for LFF to be RLF, which (as shown in Table 1) corresponds to a feeding interaction. Tan (1987) reports the surface form of LFF to be LLF, which instead corresponds to a counterfeeding relation. The computational analysis presented in the next section assumes the feeding relationship to be correct, though nothing hinges on this assumption (i.e., the same computational classification would result if the counterfeeding relation were modeled instead).

Hung (1987) aims to make the rule-based account less stipulative by proposing a phonotactic constraint against adjacent low tones as motivating both the application of the rules and their directionality. Sequences of HH are again permitted, but LL, FF, and RR (the latter two of which contain a low tone component) are targeted. The LL and FF rules then apply the way they do (right-to-left) because the alternative would just recreate the offending sequences (e.g., left-to-right application of the FF rule would give *LLF). The RR rule, however, is free to apply left-to-right, because the resulting HHR does not contain any adjacent low tones. This particular account certainly echoes some of the core assumptions of the constraint-based analyses that followed, several of which are summarized in the next subsection.

4.2 *Constraint-based accounts* The main criticism of rule-based accounts like those presented above is that the stipulation of each rule's direction of application is *ad hoc* and therefore undesirable. Constraint-

[8] These are not the only sandhi rules at work in Tianjin, but accounting for the others has been comparatively straightforward and so they are often put aside. This paper will also put them aside, as incorporating them would not change the computational classification that will be proposed. Readers are referred to the previous accounts surveyed in the next section for more complete descriptions of Tianjin tone alternations.

[9] Zhang (1987) also uses rule ordering, but says that ordering is 'naturally determined' (i.e., it is transparent) and therefore not stipulated as part of the analysis.

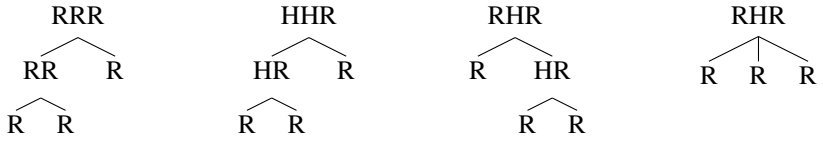

Figure 4: Example candidates from the analysis of Wee (2010)

based frameworks like Optimality Theory (OT) (Prince & Smolensky, 2004) provide a solution in their use of violable constraints: sandhi can be assumed to apply left-to-right by default *unless* that would cause a violation of the Obligatory Contour Principle (OCP), in which case sandhi applies right-to-left.

This is formalized in the analysis of Chen (2000), who proposes a constraint 'Temporal Sequence' that asserts that rules apply left-to-right. This constraint is outranked by the OCP, which is defined such that HH sequences are not subject to it, since they are in fact permitted in Tianjin. Violations of Temporal Sequence are assessed using candidates of the form RLL $\mapsto$ RRL $\mapsto$ HRL that show the derivational histories of each surface form.

Lin (2008) points out that the use of such 'derivational constraints' like Temporal Sequence only reproduces the duplication problem that motivated OT in the first place and instead proposes an account based in prosodic correspondence. Specifically, the constraint IDENT-BO-T requires corresponding tones in prosodically-related bases and outputs to be identical. Since the two-tone sequence RR gets mapped to HR, when that sequence is embedded in a larger constituent like ((RR)R), then IDENT-BO-T prefers the candidate ((HH)R) ('left-to-right application') to ((RH)R) ('right-to-left application') because the mapping of RR to HH is closer to HR than is RH. But as IDENT-BO-T is outranked by the OCP (or specifically the OCP(L, F, R) since high tones are exempt), then again the 'right-to-left application' candidates are preferred in the cases of the LL and FF rules.

Lastly, Wee (2010) proposes a way to do away with the notion of temporality by representing derivation histories as tree structures with inter-tier correspondence. This 'percolative model of phonology' borrows from syntax the mechanism of shared information across nodes in a tree. Violations of faithfulness are the result of imperfect inter-tier correspondence. Thus, candidates like the ones shown in Figure 4 are generated for the input RRR. Perfect inter-tier correspondence is the fully-faithful candidate on the far left. Markedness constraints then lead to candidates with imperfect inter-tier correspondence being favored. In particular, left-branching structures are preferred to right-branching ones, unless the root node includes an OCP(L, F, R) violation as a result. This achieves the difference in directionality among the maps RRR $\mapsto$ HHR, LLL $\mapsto$ LRL, and FFF $\mapsto$ FLF.[10]

The range of proposals put forth to account for Tianjin tone sandhi highlights the reason why it has received so much attention in the literature: accounting for the different sandhi patterns in a unified way given their difference in directionality presents an interesting challenge for a variety of theoretical frameworks. All of the authors mentioned in this section have aimed to explain the pattern through general principles, but the use of added mechanisms like derivational histories certainly takes us away from the original intentions of current phonological theory and, if justified, would suggest that Tianjin tone sandhi truly is exceptional in some way.

The next section presents a computational account of the sandhi pattern using the notions of strict locality introduced in §2. The goal of this exercise is to assess whether the complications Tianjin presents for phonological theory equate to increased computational complexity.

5 Computational account

The computational analysis of the Tianjin facts begins with the observation that the RR rule's application is not necessarily left-to-right, because it is not necessarily iterative at all. Rather, it corresponds to simultaneous application (Chomsky & Halle, 1968), in which all possible targets of the rule are identified in the underlying form and then the rule applies to all of those targets at once.[11] As noted above in §2,

[10] See Wee (2010) for more details of the analysis, including how binary branching is enforced and how the regressive nature of the sandhi is achieved.

[11] Zhang (1987); Tan (1987) also note that left-to-right application of the RR rule is the same as simultaneous application.

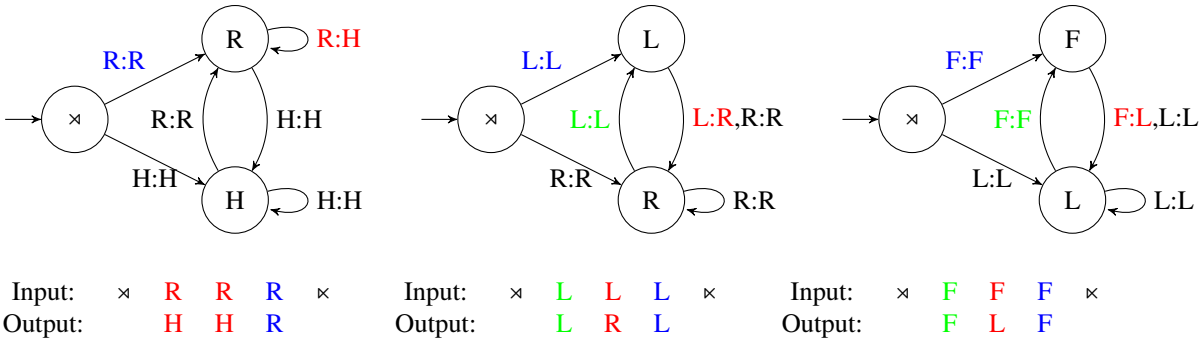

Figure 5: FSTs for the RR rule (left, 2-ISL), LL rule (center, 2-ROSL), and FF rule (right, 2-ROSL). For all three the input is read right-to-left.

simultaneous rule application can be modeled with ISL functions, whereas iterative rule application can be modeled with OSL functions. Below, §5.1 presents this analysis of the individual rules, modeling the RR rule with ISL and the FF and LL rules with OSL. Then, §5.2 offers a means of modeling the 'combined map' of all three rules as a single function.

5.1 *Individual rules* Starting with the FF and LL rules, both are iterative and therefore can be modeled with OSL. And since both are regressive, they are specifically ROSL because the input must be read right-to-left. The FSTs in the center and on the right of Figure 5 model the LL and FF rules, respectively. In the interest of clarity the inputs of these FSTs are limited to strings of L's and R's for the LL rule and strings of F's and L's for the FF rule, but this limitation will be relaxed when the interaction of the three rules is addressed in the next section. The reader can verify that each transition in these FSTs leads to the state matching the output side, which is what makes them OSL instead of ISL. Sample maps are given below each FST.

Turning now to the RR rule, its mode of simultaneous application means it is instead ISL. The 2-ISL FST shown on the left in Figure 5 models this rule. To be consistent with the other two rules (paving the way for the unified analysis in the next subsection) this FST is also assumed to operate by reading the input right-to-left and writing the output right-to-left.[12] A sample derivation for RRR is also given in the figure. Again what makes this FST an ISL one is that each transition leads to the state that matches the input side of its label.

Comparing the three FSTs it can be seen how the OSL ones achieve the 'alternating' iterative application. In the LL rule FST, after one L is read and output faithfully, the next one is changed to R and the FST proceeds to state R. Since the alternation only takes place from the L state, another L must be found in the input in order for the 'rule to apply' again. The FF rule FST works exactly the same way.

In the ISL FST however, since the transitions follow the input, when an alternation takes place (i.e., on the R:H transition), that transition is a loop. As a result, sequences of R's will all change to H, reflecting simultaneous application. Thus it is not the right-to-left direction of the LL and FF rules that necessitates OSL, but rather the alternating (i.e., iterative) pattern. Likewise, the absence of this alternating pattern is why the RR rule is necessarily ISL and not OSL.[13]

The analysis so far accounts for each rule in isolation, but of course the real interest is in capturing the way they interact (i.e., the feeding relations) and in identifying how their individual computational classifications affect the classification of the 'combined map' that models all three at once. One way to achieve this combined map would be by functional composition, where the ordering of the three rules determines the order of composition. So if R is the RR rule FST, L is the LL rule FST, and F is the FF

[12] The difference is that in this case the input doesn't *have to* be read right-to-left, it just *can* be. Regressive patterns like the nasal assimilation example in Figure 2 can be handled by ISL when reading left-to-right by temporarily delaying output using λ. That is not an option for OSL: because the transition leads to the state that matches the output, if the output is λ then the FST must remain in its current state (i.e., loop). A loop with a λ output amounts to unbounded deletion, as there's no way to keep track of how many times that transition was taken.

[13] See Chandlee et al. (in preparation) for formal proofs that the RR rule map isn't OSL and the FF/LL rule maps aren't ISL.

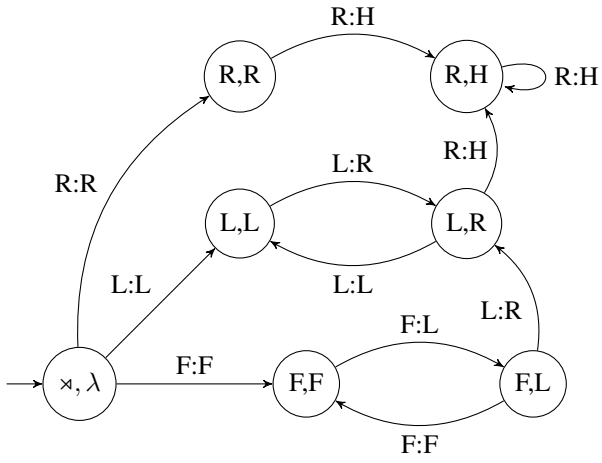

Figure 6: Fragment of (2,2)-RIOSL FST for the combined map of Tianjin tone sandhi.

rule FST, the combined map could be modeled as R ∘ L ∘ F (assuming of course each FST is expanded to accommodate the full alphabet of {R, F, L, H}). The next section offers an alternative approach, which is to model the combined map directly as a single *input-output strictly local function*.

5.2 *Combined map* The necessarily different classifications of the three sandhi rules (ISL for the RR rule and OSL for the FF and LL rules) means the combined map that includes all three patterns is neither ISL nor OSL. It can, however, be modeled using a function class that combines the properties of both ISL and OSL. Such functions are called input-output strictly local (IOSL). The FST characterization of this class will be described and used below. For more on the IOSL functions, include their abstract (formal language) characterization, their relationship with respect to other subregular function classes, and an algorithm for learning them from positive data, see Chandlee et al. (in preparation).

IOSL FSTs combine the properties of the ISL and OSL FSTs, in that the states are now of the form (x, y) where x is the most recent input and y is the most recent output. How much of the input/output is tracked can differ; a given IOSL FST is then (k, ℓ)-IOSL and tracks both the $k-1$ most recent input symbols and the $\ell-1$ most recent output symbols. For Tianjin tone sandhi, it suffices to track just one previous input and one previous output symbol, so its FST is (2,2)-IOSL.

In the interest of readability, the FST is presented in table form in Table 2. Each row represents a transition of the form (q_1, a, x, q_2), where q_1 is the state the transition starts in, a is the input symbol, x is the output string, and q_2 is the state the transition leads to. A fragment of the FST is also presented in Figure 6; the transitions shaded gray in Table 2 are the ones included in the figure. These are the transitions that account for the crucial forms in Table 1 above. As in the individual rule analysis, the FST is RIOSL, so the input string is read right-to-left and the output string is likewise written right-to-left.

The FST fragment highlights the way in which both simultaneous and iterative application can be captured at the same time. The triggering tone of all three rules is read from the start state, leading to either state (R,R) or (L,L) or (F,F). From there, a subsequent R, L, or F, respectively, is subject to sandhi and the FST moves on to a new state that reflects the alternation (either (R,H), (L,R) or (F,L)). Then, the difference in the application types of the three rules is handled by either returning to the 'trigger' state (as in the LL and FF rules) or else looping in the 'target' state (as in the RR rule). Thus all three rules are handled with the same state structure; their difference is enforced in the transitions.

The figure also includes the transitions responsible for the feeding interactions among the rules. From the (F,L) state a subsequent L will become R, reflecting the fact that a 'derived' low tone serves as a trigger for the LL rule. And likewise from the (L,R) state a subsequent R becomes H.

The combined map of the Tianjin tone sandhi rules is then classified as (2,2)-RIOSL. The next section will briefly discuss the implications of this classification.

State	Input	Output	State
(⋊,λ)	L	L	(L, L)
(⋊,λ)	H	H	(H, H)
(⋊,λ)	R	R	(R, R)
(⋊,λ)	F	F	(F, F)
(L, L)	L	R	(L, R)
(L, L)	H	H	(H, H)
(L, L)	R	R	(R, R)
(L, L)	F	F	(F, F)
(H, H)	L	L	(L, L)
(H, H)	H	H	(H, H)
(H, H)	R	R	(R, R)
(H, H)	F	F	(F, F)
(R, R)	L	L	(L, L)
(R, R)	H	H	(H, H)
(R, R)	R	H	(R, H)
(R, R)	F	F	(F, F)
(F, F)	L	L	(L, L)
(F, F)	H	H	(H, H)
(F, F)	R	R	(R, R)
(F, F)	F	L	(F, L)
(L, R)	L	L	(L, L)
(L, R)	H	H	(H, H)
(L, R)	R	H	(R, H)
(L, R)	F	F	(F, F)
(R, H)	L	L	(L, L)
(R, H)	H	H	(H, H)
(R, H)	R	H	(R, H)
(R, H)	F	F	(F, F)
(F, L)	L	R	(L, R)
(F, L)	H	H	(H, H)
(F, L)	R	R	(R, R)
(F, L)	F	F	(F, F)

Table 2: Table representation of the (2,2)-RIOSL FST for the combined map. Shaded transitions are represented in the FST fragment in Figure 6.

6 Discussion

The computational analysis presented above has shown that the difference in application of the three tone sandhi rules in Tianjin amounts to a difference in computational classification, with one rule being ISL and the other two OSL. These classifications also reveal that the real distinction among the rules is not one of directionality, as the RR rule when modeled with an ISL function can either be construed as applying left-to-right *or* right-to-left (i.e., direction doesn't matter). Thus the difference in directionality of the rules is only apparent, a fact that lead to a unified analysis: the combined map that includes all three alternations can be modeled with a single IOSL function that reads strings from right-to-left.

This paper has used Tianjin tone sandhi as a case study to demonstrate how the computational framework of formal language theory (FLT) can provide insights into the nature of phonological phenomena, insights that might be obscured when one is working within a particular phonological theory. For example, the *adirectionality* of the RR rule is revealed once it is represented as a function that generates an output string for an input string by reading the input string one segment at a time. This contrasts with the compositional nature of rule application in rule-based frameworks, in which multiple applications are the result of submitting an output string in its entirety back into the same rule. For example, applying the RR rule right-to-left to RRR in this way gives the incorrect form RHR, leading to the conclusion that the rule is necessarily applied left-

to-right. The computational approach advocated for in this paper instead revealed that the rule's direction doesn't matter.

Of course, FLT is itself a theory, and its application to phonology (i.e., representing phonological processes as string-to-string functions) relies on certain assumptions shared by various phonological theories. But while it is not fully 'neutral' in that respect, it is well-established outside of phonology and as such is more general. It also aims to represent phonological patterns in very simple terms, as functions computing output strings by reading input strings incrementally in a single direction at a time. This enables an analysis of the complexity of a given pattern in terms of the computations involved, using well-defined criteria.

Why analyze Tianjin tone sandhi in particular? The previous accounts surveyed in §4 give the impression that this particular pattern is exceptional in terms of the complexity of the three rules' interaction. Put another way, the fact that Tianjin tone sandhi motivated the use of mechanisms like derivational constraints (Chen, 2000) or a percolative model (Wee, 2010) suggests that it is problematic for the theory, but what does that really mean? Is the pattern inherently difficult? Does its existence indicate that phonology in general is far more complicated than we thought?

In one sense the answer is 'no', because whether the rules are modeled individually or altogether the map is still subregular, and subregular in the larger hierarchy of possible maps is minimally complex. But in another sense, the answer is 'yes', as the classification of the combined map (IOSL) is strictly more complex than that of the individual rules (ISL/OSL) (Chandlee et al., in preparation). Though the difference among the rules is not one of directionality, a difference does exist, and that difference does amount to an increase in complexity when it comes to accounting for all three in a unified way (as a single function). Further study into other sets of phonological processes whose complexity is greater than the sum of their parts could further add to our understanding of how we factor a phonological grammar and how those factors are ultimately combined.

Lastly, to briefly address what would change about the analysis if the tones F and R were represented as the sequences HL and LH, as is more conventional. This would increase the size of the window that the function needs to keep track of, meaning the k values of the ISL and OSL functions and the k and ℓ values of the IOSL function. But increasing those values does not change the computational classification - the function is ISL/OSL/IOSL regardless of the value of k or ℓ. The classification is dependent on the fact that a sufficient value for k or ℓ *exists*, not on what those values are. The increase in the values reflects the increase in the length of the context for the FF and RR rules. The FF rule would now be represented in one of the forms given in (5).

(5) a. HL → L / ___ .HL
b. H → ∅ / ___L.HL

The window size would need to be 5 for either version of the rule, large enough to include both the target and the triggering context. This includes the boundary marker between tones (.), which is necessary to distinguish a sequence of two falling tones HL.HL from the four tone sequence HLHL (assuming only the former is subject to sandhi). Of course, this ambiguity is due to the string representation itself and would be better handled using autosegmental representations (ARs). See Chandlee & Jardine (to appear) for an extension of the ISL property to maps between ARs instead of strings.

7 Conclusion

This paper has presented a computational analysis of three tone sandhi rules in Tianjin Chinese. The results of this analysis show that all three rules are subregular, but differences in their mode of application lead to different computational classifications: two are OSL and one is ISL. Furthermore, the combined map of all three rules (including their feeding interactions) is IOSL.

Though the empirical scope of this paper's analysis is small (three sandhi rules in a single language), the results of that analysis contribute to a larger catalogue of the computational classifications of various types of process interaction. It was shown in Chandlee et al. (2018) that cases of various types of opaque interactions are all ISL (both the individual maps and the combined maps). The case of Tianjin offers another way rules can interact and, unlike the cases of opacity reviewed previously, we see that interaction indeed does increase

complexity.[14]

It is the goal of future work to identify additional ways in which process interaction can affect computational complexity, as a means of understanding the computational nature of process interaction itself. For different but related approaches to studying phonological map interaction see Baković & Blumenfeld (2017) and McCollum et al. (under review). The ultimate goal is to delimit in terms of computation the range of possible phonological maps, which crucially includes maps that traditionally would be represented with multiple rules.

References

Baković, Eric & Lev Blumenfeld (2017). A set-theoretic typology of phonological map interaction. Talk given at the Annual Meeting on Phonology, New York University.

Chandlee, Jane (2014). *Strictly Local Phonological Processes*. Ph.D. thesis, University of Delaware.

Chandlee, Jane & Adam Jardine (to appear). Autosegmental input strictly local functions. *Transactions of the Association for Computational Linguistics* .

Chandlee, Jane, Angeliki Athanasopoulou & Jeffrey Heinz (2012). Evidence for classifying metathesis patterns as subsequential. Choi, Jaehoon, E. Alan Hogue, Jeffrey Punske, Deniz Tat, Jessamyn Schertz & Alex Trueman (eds.), *WCCFL 29: Proceedings of the 29th West Coast Conference on Formal Linguistics*, Cascadilla, Somerville, MA, 303–309.

Chandlee, Jane, Jeffrey Heinz & Rémi Eyraud (2014). Learning strictly local subsequential functions. *Transactions of the Association for Computational Linguistics* 2, 491–503.

Chandlee, Jane, Rémi Eyraud & Jeffrey Heinz (2015). Output strictly local functions. *Proceedings of the 14th Meeting on the Mathematics of Language (MOL 2015)*, Association for Computational Linguistics, Chicago, USA, 112–125.

Chandlee, Jane, Jeffrey Heinz & Adam Jardine (2018). Input strictly local opaque maps. *Phonology* 35:2, 171–205.

Chandlee, Jane, Rémi Eyraud & Jeffrey Heinz (in preparation). Input-output strictly local functions and their efficient learnability. Ms.

Chen, Matthew Y. (1986). The paradox of Tianjin tone sandhi. *Proceedings of Chicago Linguistics Society 22*, 98–114.

Chen, Matthew Y. (2000). *Tone sandhi: Patterns across Chinese dialects*. Cambridge: Cambridge UP.

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

Gainor, Brian, Regine Lai & Jeffrey Heinz (2012). Computational characterizations of vowel harmony patterns and pathologies. Choi, Jaehoon, E. Alan Hogue, Jeffrey Punske, Deniz Tat, Jessamyn Schertz & Alex Trueman (eds.), *WCCFL 29: Proceedings of the 29th West Coast Conference on Formal Linguistics*, Cascadilla, Somerville, MA, 63–71.

Heinz, Jeffrey & Regine Lai (2013). Vowel harmony and subsequentiality. Kornai, Andras & Marco Kuhlmann (eds.), *Proceedings of the 13th Meeting on the Mathematics of Language (MOL 13)*, Association for Computational Linguistics, 52–63.

Hung, Tony T.N. (1987). Tianjin tone sandhi: Towards a unified approach. *Journal of Chinese Linguistics* 15:2, 274–305.

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

Kaplan, Ronald M. & Martin Kay (1994). Regular models of phonological rule systems. *Computational Linguistics* :20, 371–387.

Lin, Hui-Shan (2008). Variable directional applications in Tianjin tone sandhi. *Journal of East Asian Linguistics* 17:3, 181–226.

Luo, Huan (2017). Long-distance consonant agreement and subsequentiality. *Glossa: A Journal of General Linguistics* 2:1, p. 52.

McCollum, Adam, Eric Baković, Anna Mai & Eric Meinhardt (under review). The expressivity of segmental phonology and the definition of weak determinism.

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

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

Prince, Alan & Paul Smolensky (2004). *Optimality Theory: Constraint Interaction in Generative Grammar*. Blackwell.

Tan, Fu (1987). Tone sandhi in the Tianjin dialect. *Journal of Chinese Linguistics* 15:2, 228–246.

Wee, Lian-Hee (2010). A percolative account of Tianjin tone sandhi. *Language and Linguistics* 11:1, 21–64.

Zhang, Zheng Sheng (1987). The paradox of Tianjin: Another look. *Journal of Chinese Linguistics* 15:2, 247–273.

[14] As noted in §4, Tan (1987) reports a counterfeeding order for two of the sandhi rules, which would then be a case of opacity. And Lin (2008) and Wee (2010) describe the RR rule as opaque in that the raising of two R tones looks like overapplication. But neither of these factors is what increases the complexity of the combined map in the present analysis - rather it is the *difference* in the rules' mode of application.