A computational analysis of tone sandhi ordering paradoxes*

Chris Oakden & Jane Chandlee

Rutgers University, Haverford College

1. Introduction

Computational phonology has largely proceeded by analyzing processes in isolation (Kaplan and Kay 1994, Heinz and Lai 2013, among others), with more recent work investigating process interaction (Baković and Blumenfeld 2017, Chandlee et al. 2018). Our more thorough understanding of the range of ways in which processes can co-occur and interact (see Baković 2007) raises the question of whether different types of interactions likewise differ in computational complexity. This paper addresses that question through a computational analysis of multiple tone sandhi processes in Changting, a Hakka dialect of Chinese (Luo 1982, Chen 2004). Tone sandhi is a fitting phenomena to study in this regard, given the overlapping nature of triggers and targets both within and across processes. The analysis indicates that despite the challenges Changting tone sandhi (henceforth CTS) has posed for both rule- and constraint-based theories of phonology, its computational complexity is still quite limited.

The remainder of the paper is structured as follows. In §2 we present the facts of CTS and demonstrate how they create paradoxes in both rule- and constraint-based frameworks. In §3 we present our analysis, first introducing the relevant computational background in §3.1, then giving computational analyses of mutual counterbleeding (§3.2) and mutual bleeding (§3.3) in CTS. In §4 we discuss some implications and open questions raised by the analyses before concluding in §5.

2. CTS and challenges for rule- and constraint-based theories

Changting has five tones: low (L), mid (M), high (H), rising (R), and falling (F). Of the 25 possible two-tone sequences, Chen (2004) reports that 15 undergo tone sandhi. We focus on the four cases in (1) which are relevant to interactions.

(1) a. ‘MR’ rule: M $\rightarrow$ L / __ R

*We thank the audience at NELS 50 for helpful comments.

b. 'RM' rule: R → H / __ M
c. 'LM' rule: L → M / __ M
d. 'ML' rule: M → L / __ L

The targets and triggers of the rules in (1) *overlap* in strings of three or more tones, thus producing interactions of these rules. Relevant mappings between input and output three-tone strings are given below.

(2) a. /MRM/ ↦ [LHM]
b. /RMR/ ↦ [HLR]
c. /MLM/ ↦ [MMM]
d. /LML/ ↦ [LLL]

CTS presents a challenge for a rule-based theory of phonology in the form of ordering paradoxes. The mapping in (2a) of /MRM/ to [LHM] suggests that the MR rule (1a) applies first to derive the intermediate form LRM and then the RM rule (1b) applies to derive [LHM]. But the mapping in (2b) suggests the opposite order: /RMR/ to HMR by rule (1b) and then HMR to [HLR] by rule (1a). Both of these orders correspond to a counterbleeding interaction, illustrated in (3).

(3) a.

	/MRM/	/RMR/
MR rule	LRM	RLR
RM rule	LHM	—
	[LHM]	*[RLR]

b.

	/MRM/	/RMR/
RM rule	MHM	HMR
MR rule	—	HLR
	*[MHM]	[HLR]

Likewise, the mapping in (2c) suggests the LM rule (1c) is ordered before the ML rule (1d): /MLM/ ↦ [MMM], bleeding (1d). But the mapping in (2d) again indicates the opposite order: /LML/ ↦ [LLL], bleeding (1c). The paradox apparent in this mutual bleeding interaction is demonstrated in (4).[1]

(4) a.

	/MLM/	/LML/
LM rule	MMM	MML
ML rule	—	MLL
	[MMM]	*[MLL]

b.

	/MLM/	/LML/
ML rule	LLM	LLL
LM rule	LMM	—
	*[LMM]	[LLL]

As for constraint-based theories, achieving this same set of examples results in ranking paradoxes. For example, in the analysis of Chen (2004), in which candidates are derivations, a constraint like TEMP (scan the string from left to right) outranking ECON (minimize derivational steps) would correctly choose MRM → LRM → LHM over MRM → MHM, but then the reverse ranking is necessary for MLM → MMM to win out over MLM → LLM → LMM. Relevant tableau are given in (5).

[1]The term 'mutual bleeding' is originally due to Kiparsky (1971). Depending on one's interpretation of counterbleeding, however, the interaction in (4) may also be understood as 'bled counterbleeding' (Baković 2011).

(5)

		/MRM/	TEMP	ECON
a.	☞	MRM - LRM - LHM		**
		MRM - MHM	*	*

		/MLM/	ECON	TEMP
b.		MLM - LLM - LMM	**	
	☞	MLM - MMM	*	*

Given these complications, Chen et al. (2004, 1-2) wonder whether existing theoretical models can even account for the 'dauntingly complex' CTS paradigm.

3. Computational account

The issues CTS presents for both theoretical frameworks may imply that CTS is inherently a complex pattern. Our aim is to assess whether that impression of complexity aligns with a more formal and rigorous notion of computational complexity. In particular, we ask whether CTS can be represented with *subregular* functions (i.e., proper subsets of the regular relations) as has been argued for a range of segmental phonological processes (see Chandlee and Heinz 2018, and references therein). The answer is yes, as CTS can be modeled using two of the most restrictive subregular classes of functions, the input strictly local (ISL) and output strictly local (OSL) functions. First, we introduce the notions of mappings between strings as ISL and OSL functions, and then show that CTS interactions fall within these classes.

3.1 Computational background

We formalize tone sandhi process interactions in a computational framework as input-output mappings defined over string models. These models comprise structural nodes which are labeled with symbols from some alphabet. Additionally, nodes relate to one another via a *successor* function (denoted s); this imposes a linear order over nodes by identifying the node which immediately follows it. An example of a string model with alphabet $\{A, B, C\}$ is given in (6).

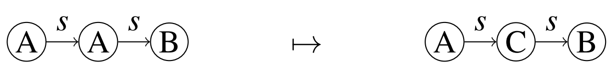
(6) $\text{(A)} \xrightarrow{s} \text{(B)} \xrightarrow{s} \text{(C)}$

This model simply represents the string *ABC*, with the ordering among its characters made explicit via the successor function. Mappings between input and output string models are defined with a set of logical formulas. Such statements determine how string positions are labeled in the output structure. Crucially, they define output labels through bounded reference to the input (ISL) or output (OSL) structure. For example, consider a function which maps A to C when it is immediately followed by B or C in the input. A logical formula thus defines the conditions under which a node is labeled C in the output structure. This formula is denoted C' and given in (7).

(7) $C'(x) \stackrel{\text{def}}{=} (A(x) \wedge (B(s(x)) \vee C(s(x)))) \vee C(x)$

This formula states that an output node (x is a variable ranging over output nodes) will be labeled C in the output when it is labeled A in the input AND its successor in the input is labeled either B (denoted $B(s(x))$) or C (denoted $C(s(x))$). In addition, the $C(x)$ disjunct is included so that nodes already labeled C in the input remain C's in the output. (7) thus describes (part of)[2] an ISL function, because whether a position is labeled C in the output can be determined solely from information in the input structure.

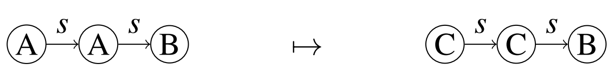
The string mapping $AAB \mapsto ACB$ in (8) satisfies this logical definition; in the input structure, the second node is both labeled A and followed immediately by a B, and so its output correspondent is labeled C. The first and third nodes do not satisfy the definition, and therefore are not labeled C in the output.

(8)


In contrast, the formulas for an OSL function can refer to the output structure. The formula in (9) is similar to (7) except that it maps A to C when an input-specified A is followed immediately by B or C in the *output*. Reference to output structure in the logical formalism is achieved by recursive definitions ($B'(s(x))$ and $C'(s(x))$).

(9) $C'(x) \stackrel{\text{def}}{=} A(x) \wedge (B'(s(x)) \vee C'(s(x)))$

Example (10) shows a different string mapping $AAB \mapsto CCB$ which satisfies (9) but *not* (7).

(10)


In this mapping, the second node in the output is again labeled C, but so is the first node. This is because the first node is labeled A in the input and its successor in the output is labeled C, as required by (9).

Another crucial property of these logical characterizations of ISL and OSL functions is that the formulas that describe them are quantifier-free (QF). This means that the formula can only make *bounded* reference to the input or output structure, meaning it can only refer to a node within a fixed distance of the node identified by x. So multiple applications of the successor function are permitted (e.g., $A(s(s(s(x))))$), but a quantifier cannot be used to verify that a particular node exists *anywhere* in the model or that *every* node has a particular property. This limitation prohibits any kind of global evaluation of the string, such that the computation is necessarily *local* in nature.

QF logical statements—those without existential or universal quantification—are computationally simple and restrictive, yet are sufficient to model many phonological processes (Lindell and Chandlee 2016). The following two sections show that ISL and OSL functions

[2]The complete function would also include formulas for $B'(x)$ and $A'(x)$.

(formalized using QF logical definitions) are also sufficient to model the CTS interactions described above.

3.2 Mutual counterbleeding as an ISL function

The RM/MR rule ordering paradox (2a)-(2b) is resolved once we represent the interaction of the two rules as a single ISL function. Doing so gives both rules access to the UR at the same time, which is necessary to get the correct mappings. Below is a series of QF logical statements, one for each output tone in Changting.

(11)

$$\begin{aligned} L'(x) &\stackrel{\text{def}}{=} L(x) \vee (M(x) \wedge R(s(x))) \\ M'(x) &\stackrel{\text{def}}{=} M(x) \wedge \neg R(s(x)) \\ H'(x) &\stackrel{\text{def}}{=} H(x) \vee (R(x) \wedge M(s(x))) \\ R'(x) &\stackrel{\text{def}}{=} R(x) \wedge \neg M(s(x)) \\ F'(x) &\stackrel{\text{def}}{=} F(x) \end{aligned}$$

For example, the first formula says that an output tone is a low tone if either 1) its input correspondent is already low, or 2) its input correspondent is a mid tone that precedes a rising tone. This is then a logical translation of the 'MR' rule. The remaining formulas likewise identify the conditions for the other tones to appear in the output. Notice that the falling tone F isn't targeted by any of the rules in question, so to be a F tone in the output it suffices to be one in the input.

Notice also that all of these formulas refer only to a bounded contiguous portion of the input structure, and thus describe ISL functions. The relevant three-tone sandhi forms in (2a)-(2b)—repeated as string mappings in (12a)-(12b)—satisfy the logical formulas in (11).

(12)

a. $\text{Ⓜ} \xrightarrow{s} \text{Ⓡ} \xrightarrow{s} \text{Ⓜ} \quad \mapsto \quad \text{Ⓛ} \xrightarrow{s} \text{Ⓗ} \xrightarrow{s} \text{Ⓜ}$

b. $\text{Ⓡ} \xrightarrow{s} \text{Ⓜ} \xrightarrow{s} \text{Ⓡ} \quad \mapsto \quad \text{Ⓗ} \xrightarrow{s} \text{Ⓛ} \xrightarrow{s} \text{Ⓡ}$

To see how this analysis provides a solution, recall that both possible orderings of the RM/MR rules resulted in counterbleeding. This assumes that the output of one rule becomes the input to the next rule. As maps, and particularly as ISL functions, the conditioning environment is restricted to the *input* (i.e., the UR), and not the intermediate forms generated in classic SPE rule derivations. For example, for input /MRM/ in (12a), the second string position satisfies $H'(x)$—the equivalent of an application of the RM rule. However, labeling this string position as H in the output does not bleed the structural environment of the MR rule (governed by the definition $L'(x)$). As an ISL function, it only 'sees' the input string /MRM/, and not any modifications apparent in the output string, including the application of other rules. String position 1 satisfies $L'(x)$, and so it is labeled L in the output.

Therefore, there is no paradox; both rules can 'apply' simultaneously, because their logical translations have equal access to the input/underlying structure.[3]

3.3 Mutual bleeding as an OSL function

Similarly, the mutual bleeding interaction between the ML and LM rules (2c)-(2d) is resolved once we represent their interaction with an OSL function. A series of QF-definable logical statements is given in (13). Note that these formulas contain recursive definitions $L'(s(x))$ and $M'(s(x))$ which refer to a bounded window in the output structure.

(13)
$$\begin{aligned} L'(x) &\stackrel{\text{def}}{=} \big(L(x) \wedge \neg M'(s(x))\big) \vee \big(M(x) \wedge L'(s(x))\big) \\ M'(x) &\stackrel{\text{def}}{=} \big(M(x) \wedge \neg L'(s(x))\big) \vee \big(L(x) \wedge M'(s(x))\big) \\ H'(x) &\stackrel{\text{def}}{=} H(x) \\ R'(x) &\stackrel{\text{def}}{=} R(x) \\ F'(x) &\stackrel{\text{def}}{=} F(x) \end{aligned}$$

According to these definitions, a node is labeled L in the output if it is either: an input L which is not followed by an output M (the first disjunct), or an input M which is followed by an L in the output (the second disjunct). Similarly, for a node to be labeled M in the output it must satisfy one of two structural configurations: an input M not followed by an output L, or an input L followed by an output M, as specified in the two disjuncts of $M'(x)$'s definition, respectively.

The use of recursive definitions is necessary to achieve the mutual bleeding interaction, as opposed to the mutual counterbleeding interaction of the RM/MR rules. For mutual counterbleeding, it was necessary for both rules to access the input structure, which effectively allows them to operate independently of each other. For mutual bleeding, however, each rule needs to factor in the effect of the other applying. This can be accomplished by again representing the two rules as a single map, but one that can reference output structure (i.e., an OSL function).

The crucial mappings for this interaction are shown again in (14).

(14) a. $\text{M} \xrightarrow{s} \text{L} \xrightarrow{s} \text{M} \quad \mapsto \quad \text{M} \xrightarrow{s} \text{M} \xrightarrow{s} \text{M}$

b. $\text{L} \xrightarrow{s} \text{M} \xrightarrow{s} \text{L} \quad \mapsto \quad \text{L} \xrightarrow{s} \text{L} \xrightarrow{s} \text{L}$

According to the formulas in (13), a final L in the input remains an L in the output (by the first disjunct of $L'(x)$), and a preceding L or M is also output as L (by the second disjunct). In terms of the rules in (1), for the input /LML/ this means that the ML rule applies but the LM rule is blocked. Likewise for M: if the final symbol is an input M, it is output as M (by

[3] This does not, however, amount to a general proposal for the simultaneous application of a set of rules, which was argued against in Chomsky and Halle (1968), among others. In fact it is not a proposal for representing processes as rules at all, a point we will return to in §4.

the first disjunct of $M'(x)$), and a preceding L is output as M (by the second disjunct). So for the input /MLM/, the LM rule applies while the ML rule is blocked.

The formulas $L'(x)$ and $M'(x)$ thus mirror each other, enabling the mutual bleeding effect. The ability to reference the ouptut structure is crucial here (i.e., the function is necessarily OSL), because the conditions specified in each formula can factor in the effect of the other formula on the successor of x. As in the previous section, a single set of logical statements (i.e., a single function) captures the map corresponding to a paradoxical interaction between two rules.

4. Discussion

The analyses presented above demonstrate that two problematic tone sandhi rule interactions (mutual counterbleeding and mutual bleeding) can be modeled with subregular (ISL and OSL) functions. The reason these analyses of CTS are possible is because the conditions under which the sandhi rules change a particular tone correspond to unique (bounded and contiguous) substrings in the input or output string alone, meaning the intermediate forms of a derivational representation are not necessary (and in fact, are problematic as the ordering and ranking paradoxes explained above show).

Some earlier work on Chinese tone sandhi—including Changting—gets tantalizingly close to the intuition of ISL functions. Hsu (1994, 2005), for example, introduces the 'One-Step Principle', which prohibits derived tones from serving as inputs to other tone sandhi rules. This cannot be a general principle for all sandhi rules, however, as it prohibits a feeding interaction, which is observed for other rules in Changting. It would also rule out bleeding, which the second case analyzed in this paper requires. The framework used in this paper does not enforce such a strong limitation, as both ISL and OSL functions are available.

The successful treatment of the RM/MR and ML/LM rule interactions relied on representing these pairs of rules as single functions. In other words, the rules in each pair were *not* first represented as individual functions and then combined via some operation like function composition, as was done in earlier computational frameworks for phonology (e.g., Kaplan and Kay 1994, among others). (Indeed, as function composition enforces an order of application, doing so would run into the same paradoxes as the rule-based analysis.) This represents a step away from the serial derivation of SPE towards something closer to the two-level rules proposed by Koskenniemi (1983), in that multiple rules are able to apply simultaneously to a given UR. It should be emphasized, however, that the computational framework used in this paper begins with the *extensions* of rules, rather than rules themselves. The analyses thus treated mappings like /MRM/ $\mapsto$ [LHM] directly, in order to assess what complexity level and type of computation they necessitated. The answer was that their level of computational complexity is quite low, and the distinction between counterbleeding and bleeding (in this case) is one of input- versus output-based computation. More broadly the goal was to show how viewing such phenomena through the lens of computation (as opposed to rules or constraints) can offer additional insights into what is going on with a particular pattern and why it may be problematic for a given framework.

Nonetheless, the analyses presented here indeed raise the question of how different functions interact or are combined in the larger grammar. The necessary distinction between mutual counterbleeding as an ISL function and mutual bleeding as an OSL function means more than one function is needed to model CTS in total. So what role do these two functions play in the larger grammar? Indeed there are more sandhi rules in Changting than the four analyzed here, so more broadly what does the total grammar for CTS look like? These are crucial questions whose answer depends on the results of a larger investigation into how rules can interact, as well as the closure properties of various subregular functions under a range of operations. Such questions are the focus of current and planned future work, and so must remain open for now. The takeaway of this particular case study is that analyses of patterns using subregular functions offer desirable computational restrictiveness alongside useful flexibility in what constitutes a phonological map.

5. Conclusion

This paper presented a computational analysis of two previously troublesome interactions of tone sandhi rules in Changting. The conclusion of the analyses is that mutual counterbleeding in Changting can be represented with an ISL function, while mutual bleeding can be represented with an OSL function. Despite their restricted computational complexity, these subregular function classes were sufficient to model both interactions.

These results build on the findings of previous investigations of the computational complexity of process interactions. Chandlee et al. (2018), for example, showed that the opaque interactions discussed in Baković (2007) are also ISL functions. This paper adds to that catalogue of attested interactions with a case of mutual counterbleeding that is ISL and a case of mutual bleeding that is OSL.

As noted above, a fuller understanding of the computational nature of interactions requires looking outward from this and other case studies to identify the requisite formal conditions for each interaction type to occur. Baković and Blumenfeld (2017) approach that task from a set-theoretic perspective. Studies such as the present one likewise add insights using the tools of first order logic and subregular phonology.

References

Baković, Eric. 2007. A revised typology of opaque generalisations. *Phonology* 24:217–259.

Baković, Eric. 2011. Opacity and ordering. In *The Blackwell handbook of phonological theory*, ed. John Goldsmith, Jason Riggle, and Alan Yu. Wiley-Blackwell.

Baković, Eric, and Lev Blumenfeld. 2017. A set-theoretic typology of phonological map interactions (handout). Paper presented at Phonology Lunch, UC Santa Cruz.

Chandlee, Jane, and Jeffrey Heinz. 2018. Strict locality and phonological maps. *LI* 49:23–60.

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

Chen, M. Y. 2004. Changting Hakka tone sandhi: Analytical challenges. *Language and Linguistics* 5:799–820.

Chen, Matthew Y, Xiuhong Yan, and Lian-Hee Wee. 2004. *Hakka tone sandhi: Corpus and analytical challenges*. Project on Linguistic Analysis, University of California.

Chomsky, Noam, and Morris Halle. 1968. *The sound pattern of english*. New York: Harper & Row.

Heinz, Jeffrey, and Regine Lai. 2013. Vowel harmony and subsequentiality. In *Proceedings of the 13th MOL*, ed. Andras Kornai and Marco Kuhlmann, 52–63.

Hsu, Huichuan. 1994. Constraint-based phonology and morphology: a survey of languages in China. Doctoral dissertation, University of California, San Diego.

Hsu, Huichuan. 2005. Two constraints on tonal derivation in Chinese. *Taiwan Journal of Linguistics* 3:119–147.

Kaplan, R.M., and M. Kay. 1994. Regular models of phonological rule systems. *Computational Linguistics* 20:371–387.

Kiparsky, Paul. 1971. Historical linguistics. In *A survey of linguistic science*, ed. W. O. Dingwall, 576–642. University of Maryland Linguistics Program.

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

Lindell, Steven, and Jane Chandlee. 2016. A logical characterization of strictly local functions. Paper presented at the Fourth Workshop on Natural Language and Computer Science, New York, NY.

Luo, Meizhen. 1982. Fujian Changting Kejiahua de liandu biandiao [Tone sandhi in the Hakka dialect of Changting, Fujian]. *Yuyan Yanjiu* 3:188–197.

Chris Oakden, Jane Chandlee
chris.oakden@rutgers.edu, jchandlee@haverford.edu