MODELING PHONOLOGICAL INTERACTIONS USING RECURSIVE SCHEMES

By

CHRISTOPHER DONAL OAKDEN

A dissertation submitted to the

School of Graduate Studies

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Graduate Program in Linguistics

written under the direction of

Adam Jardine

and approved by

New Brunswick, New Jersey

October, 2021

ABSTRACT OF THE DISSERTATION

Modeling phonological interactions using recursive schemes

by CHRISTOPHER DONAL OAKDEN

Dissertation Director:

Adam Jardine

This dissertation pursues a computational theory of phonological process interactions whereby individual processes are formalized as input-output mappings (i.e. functions), and interactions are the combinations of those functions using a set of two *operators*: one previously defined in the literature and another defined in this dissertation. Building on hypotheses regarding the computational complexity of phonological processes in isolation (Heinz and Lai, 2013), the primary novel contribution of this dissertation is to extend these insights to interactions within larger phonological grammars, but in a systematic way. Specifically, it shows that the *subsequential* class of functions, sufficient to describe a great majority of phonological generalizations in isolation, also provides a well-motivated upper bound on the complexity of phonological interactions. Analyses developed in this work offer a straightforward solution to a number of outstanding cases of interactions in the Chinese tone sandhi literature. Crucially, this includes sandhi paradigms for which traditional generative phonological theories (rule-based SPE (Chomsky and Halle, 1968a) and Optimality Theory (Prince and Smolensky, 2004)) fail to account. Thus this novel approach permits an explicit, restrictive theory of phonological interactions whose predictions more closely align with attested data.

The formal apparatus for defining functions and operators used in this work is *boolean monadic recursive schemes* (BMRS; Bhaskar et al., 2020; Chandlee and Jardine, 2020). BMRS are a logical formalism rooted in theoretical computer science, and have been recently applied to computational analyses of phonology. Thus another important contribution of this dissertation is that it represents the first major work using BMRS to explore a specific type of linguistic phenomenon. In addition to demonstrating its application to specific tone sandhi paradigms, this study identifies advantages to BMRS in modeling interactions more generally, especially in comparison to other computational formalisms. The dissertation also leverages the phenomenon-*independent* nature of this logical formalism by applying BMRS to questions of phonological representation. Specifically, it is shown

how operations over BMRS contribute to recent computational work using model theory and logic to explore notational equivalence across representational theories (Strother-Garcia and Heinz, 2015; Danis and Jardine, 2019; Oakden, 2020).

Acknowledgments

Acknowledging all the people whose help and support have led me here—and to fully express my gratitude to them—would result in a section that is inappropriate in both its length and effusiveness, as far as these sections go. So I will keep it short and do my best to capture the essence of it.

First, this dissertation would not have happened without the unwavering support, optimism, and dedication of my advisor Adam Jardine. Thank you for enabling me to pursue something that I am passionate about, and in a way that moves the field forward.

I wish also to express thanks to my committee members. To Akin Akinlabi, thank you for encouraging my growth as a phonologist ever since you taught the first phonology course I took at Rutgers. To Jane Chandlee, thank you for lighting my path toward computational work on tone sandhi, and for agreeing to collaborate on some of that work. And to Adam McCollum, thank you for the opportunity to explore other avenues of my research, even if it doesn't always lead to satisfying conclusions (like Liko!).

If it weren't for Paul de Lacy, I might still be cleaning loos for a living, so I thank him for giving me the chance to do a PhD at Rutgers, and for all the wisdom and advice he's shared over the years. I am indebted to the entire Linguistics family at Rutgers: faculty, grad students past and present, friends. It has been a pleasure to share these past five years with you. Special thanks go to Marilyn and Ann for holding it all together.

I would also like to thank those who advised me during my time at the University of Arizona. In particular I thank Diana Archangeli, who first introduced me to phonology and was a generous host to me in both Tucson and Hong Kong. To Liu Laoshi (Liu Feng-hsi), my first advisor, thank you for teaching me perseverance in the face of journal rejections, though it still hasn't gotten easier.

While the pandemic has prevented me from seeing my family and friends in Arizona, I have felt their support nonetheless. To my parents Ray and MJ, thank you for believing in me, encouraging me, and always reminding me to "stop and smell the roses." I also thank my sister Kim and best friends David and Sean for brightening even my darkest days with humor and perspective.

Finally, and most importantly, I thank my loving wife Luo Xue for her support and patience. Thank you for keeping me sane during this process, and for forcing me to eat healthy and exercise; I am likely to have not survived this dissertation without you.

Contents

1 Introduction

1.1 Overview

This dissertation pursues a computational theory of phonological process interactions. It does so by formalizing phonological processes as mappings (i.e. functions) from input structures to output structures, adding to a growing body of work within the Subregular Hypothesis of phonology (Heinz and Lai, 2013). Specifically, the current study defines functions using *boolean monadic recursive schemes* (BMRS; Bhaskar et al., 2020; Chandlee and Jardine, 2020), a logical formalism recently applied to computational analyses of phonological transformations. The main thrust of this theory is that process interactions are the combination of functions (defining individual phonological processes) via a set of two BMRS-definable *operators*: one previously defined in the literature and another defined in this dissertation. This approach permits an explicit, restrictive theory of phonological interactions, crucially in ways unavailable to other grammatical formalisms in traditional generative phonology (rule-based SPE (Chomsky and Halle, 1968a) and Optimality Theory (Prince and Smolensky, 2004)), as well as to other computational formalisms.

1.1.1 Interactions in phonology

Broadly speaking, an interaction occurs when some individual process influences or is influenced by another process, caused by an overlap of the processes' targets or triggers. Four basic types of interactions in a serial rule-based framework are summarized below (adapted from McCarthy, 2007). An implicit one-to-one correspondence between *process* and *rule* is assumed.

(1.1)
a. Rule A **feeds** rule B when A creates additional inputs to B

b. Rule A **bleeds** rule B when A eliminates potential inputs to B

c. Rule B **counterfeeds** rule A when B creates additional inputs to A and B is ordered before A

d. Rule B **counterbleeds** rule A when B eliminates potential inputs to A and B is ordered before A

A substantial body of work over the past 40 years has been devoted to studying process interactions, and especially *opaque* interactions—(1.1c) and (1.1d), as contrasted with *transparent* interactions (1.1a) and (1.1b)—for which certain processes appear on the surface to have either *over*applied or *under*applied (McCarthy, 1999). Early attempts at explaining opaque interactions (Kenstowicz and Kisseberth, 1971; Kean, 1974; Kiparsky, 1976, 1982; Mascaró, 1976) proposed various principles by which separate 'rules' apply, either in an ordered, serial fashion, simultaneously, or cyclically. Opaque interactions have proven difficult to capture in classic parallel OT (Prince and Smolensky, 2004), but subsequent constraint-based work (McCarthy, 1999, 2003, 2007) has proposed extensions to this formalism to accommodate opacity, and has identified other interactions (Baković, 2007) beyond the traditional four. This includes opaque interactions which are problematic for rule-based accounts but have an intuitive explanation in OT (Baković, 2011).

1.1.2 Rules, ordering, and their relation to functions

Traditionally, interaction between individual processes in rule-based phonology is formalized via rule ordering. The first rule in a pairwise ordering applies to the input form, and its output—an *intermediate* representation—becomes the input to the following rule; the surface form is the output of the final ordered rule. The creation/elimination of (potential) inputs in (1.1) thus refers to the intermediate representations generated by the application of non-final rules. A hypothetical example from Baković (2011) illustrates: a system in which a Deletion process is ordered before a Palatalization process.

(1.2)

		a. /tue/	b. /tio/
Deletion:	$V \rightarrow \emptyset$ / ___ V	[te]	[to]
Palatalization:	t → tʃ / ___[-back]	[tʃe]	—
Output:		[tʃe]	[to]

For inputs /tue/ and /tio/, Deletion produces the respective intermediate forms [te] and [to]. These serve as inputs to Palatalization. Deletion thus *feeds* Palatalization for the form /tue/ by creating an intermediate representation that satisfies the structural description of the latter. The opposite is true for /tio/—a bleeding interaction. The opposite ordering (Palatalization ordered before Deletion) would yield counterfeeding and counterbleeding, respectively.

Individual processes in an interaction scenario can be conceived as input-output mappings (i.e. functions). Given Kaplan and Kay (1994)'s result connecting regular relations and their closure properties to rewrite rules in SPE, ordering of these individual rules within a grammar corresponds to function *composition.* Using the example above, let $f(x)$ and $g(x)$ denote functions that describe the mappings produced by Deletion and Palatalization, respectively. The composition $g \circ f$ or $g(f(x))$ defines the result of applying g to the output of f. This composite function models the interaction of Deletion and Palatalization given the ordering above. Order of composition is also consequential as in ordering of rules; the opposite composition $f \circ g$ would predict the same counterfeeding/counterbleeding outputs as the inverse rule ordering.

A great many process interactions receive a straightforward account via rule-ordering (composition). Composition alone, however, offers an incomplete picture of the typology of rule interaction, especially with respect to opaque patterns. Some interactions present ordering *paradoxes*; no pairwise rule ordering—therefore no order of composition—will derive attested output forms (see §2 of this Chapter, as well as Chapter 2). Additionally, Baković (2011) cites several examples of rule *blocking* which require additional mechanisms beyond ordering. These cases are amenable to an OT analysis, and their existence suggests that other operations exist by which individual rules combine in interaction contexts.

Unfortunately, extracting such operations from a satisfactory OT account of interaction is impossible. Parallel evaluation over a total order of constraints selects a *single* output candidate from a *single* input representation. An interaction is 'one jump' from an input to an output, without distinct sub-evaluations or intermediate forms. This means that individual processes have no ontological status in an OT framework. In other words, OT provides a glimpse into $g(f(x))$ as a single function, but not $g(x)$ or $f(x)$ as individual functions. A rule-based framework provides the opposite perspective—decomposability. Neither formalism permits *both* decomposable and undecomposable perspectives on process interaction.

1.1.3 A computational approach

A theoretical computational approach—i.e. one that focuses on the nature of *mappings* between input and output structures—is well-suited in this regard. This is because individual processes and their interactions (referred to as the 'combined map') can be formalized as single functions, thereby providing a vantage point unavailable to SPE and OT. And while advances in computational theories of phonology have focused primarily on single processes, recent studies have turned their attention toward interactions (Chandlee et al., 2018; Chandlee, 2019; Oakden and Chandlee, 2020).[1] The main thrust of this work is to show that certain types of interactions are describable

[1] In addition to work in the subregular paradigm, work by Baković and Blumenfeld (2017, 2018, 2020) provides formal characterizations of input-output map relations and interactions.

by subregular function classes; that is, that theories of the computational complexity of individual processes also extend to their interactions. Chandlee et al. (2018), for example, demonstrate that a number of opaque interactions are *input strictly local* (ISL; Chandlee, 2014) functions as combined maps. Individual generalizations comprising interactions are also ISL, with the implicit understanding that the two relate via function composition. A later analysis by Oakden and Chandlee (2020) echoes this result, identifying a different ISL-definable opaque interaction, as well as a transparent *output strictly local* (OSL; Chandlee, 2014) one. Likewise, they do not relate individual maps with combined maps, but it is noted that composition would be insufficient to capture these interactions. This also gels with Baković (2011)'s conclusion introduced above, that rule ordering is insufficient in capturing process interactions in phonology.

The computational approach furnishes a framework for combining individual and combined (interaction) maps; however previous studies do not do so explicitly. This dissertation capitalizes on this feature; it builds on recent computational studies by proposing a theory whereby interactions are the combination of functions (describing individual processes) via a set of *operators*.[2] This includes composition as well as a novel operator, acknowledging that composition alone is insufficient.

1.2 Tone sandhi in Chinese dialects and their interaction

The empirical focus of this dissertation is the set of tonal processes known as *tone sandhi* attested in, among other groups, Chinese (Sinitic) dialects. Broadly speaking, tone sandhi can be understood as a process affecting tones which appear together within a domain. A vast and diverse set of alternations have been termed 'tone sandhi' in the literature (see Chapter 2 for in-depth discussion). This dissertation focuses on *right-dominant* tone sandhi (Yue-Hashimoto, 1987; Chan, 1995a; Zhang, 2007); that is, processes which preserve the tone on the final syllable in a domain and alternate non-final syllables. A well-known example of right-dominant sandhi comes from Standard Mandarin (see Wang and Li, 1967, for an early discussion) and is given in (1.3); when two low-dipping tones in isolation (represented as 'L') appear adjacently, the first surfaces as a rising 'R' tone.

(1.3)			
	xiao		'small'
	L		isolation form
	xiao	*gou*	'small dog; puppy'
	R	L	sandhi form

[2]The phonological grammar then is a single function that comprises combinations of functions.

A sandhi paradigm comprises a set of individual alternations like in (1.3). Some dialects have only several such alternations, while others have many. For example, the Hakka dialect Changting (Li, 1965; Luo, 1982; Rao, 1987; Chen, 2004) exhibits 15 distinct sandhi alternations out of 25 possible disyllabic combinations of five lexical tones: H(igh), M(id), L(ow), R(ising), and F(alling). In many dialects, patterns observed in disyllabic environments generalize to longer sequences such that tonal changes in strings of three or more tones (within a single domain) can be understood in terms of the disyllabic patterns. Given the restricted set of phonological primitives—that is, lexical tones—over which sandhi processes operate, overlap of the targets and triggers of these processes is inevitable in longer sequences. This gives rise to sandhi paradigms with complex *interactions*, both transparent and opaque.

Consider an example from Changting. Two of its 15 attested disyllabic sandhi changes are as follows: a falling tone becomes a rising tone before a mid tone (FM → RM), and a high tone becomes a falling tone before a rising tone (HR → FR). In a trisyllabic sequence /HFM/, these patterns interact in the following way. First, the FM sequence in the input triggers the FM → RM sandhi process. Then, the resulting R satisfies the structural description of (HR → FR), triggering it and yielding the surface form [FRM]. This overlap of target and trigger in the latter case constitutes a feeding interaction, as shown in (1.4).

(1.4)

	/HFM/
FM → RM	H**R**M
HR → FR	**F**RM
	[FRM]

Overlapping targets and triggers in Changting trisyllabic sequences gives rise to a rich and complex sandhi paradigm, which includes feeding, bleeding, counterfeeding, and counterbleeding, all within a single system (Chen, 2004). And Changting is not the only such case; the descriptive literature on Chinese tone sandhi abounds with interactions in trisyllabic constructions. Tone sandhi is thus well-suited to a comprehensive study of phonological process interactions given the abundance of complex paradigms from which to draw, each with a distinct set of interactions playing out over a relatively small inventory of lexical tones. Additionally, tone sandhi patterns tend to be arbitrary in nature, meaning that they seldom evince general markedness pressures or phonetic grounding (see Chapter 2 and Chapter 4 for more details). In a sense, they are pure interaction.

Further justification for adopting tone sandhi as the focus of this dissertation comes from the existence of outstanding cases of sandhi interaction in the literature. These have proven difficult for current theories of phonology to explain, and each paradigm presents distinct analytical challenges to rule-based and optimization-based theories. Two examples are introduced briefly here, but the reader is directed toward Chapters 2, 5, and 6 for more discussion.

One challenge posed by sandhi interactions is in determining the so-called ‘directionality’ of rule application; in other words, how strings of tones ought to be parsed in mapping underlying forms to surface forms. To illustrate, consider Tianjin (Chen, 1986, 2000), which has a four-tone system: H(igh), L(ow), R(ising), F(alling). Among its attested sandhi patterns is an RR ‘rule’—R surfaces as H before another R (RR → HR)—and an LL ‘rule’—L surfaces as R before another L (LL → RL). In the mappings in (1.5), /RRR/ seems to require a left-to-right parse to yield the attested [HHR], while /RLL/ seems to require a right-to-left parse to yield the attested [HRL].

(1.5)				
	$\underline{\text{RR}}\text{R}$		$\text{R}\underline{\text{LL}}$	
	|		|	
	$\text{H}\underline{\text{RR}}$	by RR rule	$\underline{\text{RR}}\text{L}$	by LL rule
	|		|	
	HHR	by RR rule	HRL	by RR rule

Previous SPE (Zhang, 1987; Tan, 1987; Hung, 1987) and OT (Chen, 2000; Ma, 2005; Lin, 2008; Wee, 2010) analyses struggle to account for these directionality affects using general principles, and must either stipulate a parsing direction *ad hoc* or propose questionable extensions to the theory.

The Changting sandhi paradigm presents an even greater challenge given the seemingly paradoxical interaction of four disyllabic sandhi patterns in (1.6).

(1.6)
a. MR rule: M becomes L before R (MR → LR)
b. RM rule: R becomes H before M (RM → HM)
c. LM rule: L becomes M before M (LM → MM)
d. ML rule: M becomes L before L (ML → LL)

Changting sandhi presents challenges for rule-based theories in the form of ordering paradoxes. In the trisyllabic form mappings in (1.7), opposite rule orderings are necessary to derive each output: MR < RM in (1.7a) but RM < MR in (1.7b).

(1.7) 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 in (1.8), contradicting orders of LM and ML rules is necessary to derive the attested output forms of /MLM/ and /LML/.

(1.8) a.

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

b.

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

Achieving this same set of examples in a constraint-based theory results in ranking paradoxes. For example, Chen (2004) posits two constraints—Temp (scan the string from left to right) and Econ (minimize derivational steps)—which evaluate sets of derivations as candidates. Selecting the attested outputs requires conflicting orders of these constraints, as illustrated in the tableau in (1.9).

(1.9) a.

	/MRM/	Temp	Econ
☞	MRM - LRM - LHM		**
	MRM - MHM	*	*

b.

	/MLM/	Econ	Temp
	MLM - LLM - LMM	**	
☞	MLM - MMM	*	*

Given such complications, Chen et al. (2004) capitulate to the ‘dauntingly complex’ Changting paradigm, and argue that current theories of phonology are incapable of capturing it.

Recent computational analyses of both paradigms, however, abstract away from formalism-specific assumptions about how the interactions are represented, focusing instead on the computational power necessary to compute the attested input-output maps. Using this approach, analyses of both Tianjin (Chandlee, 2019) and Changting (Oakden and Chandlee, 2020) interactions have shown that the computational complexity of these interactions aligns with well-attested and uncontroversial phonological processes, despite the difficulty posed to traditional theories. These studies serve as a launching point for this dissertation.

1.3 Modeling interactions using *boolean monadic recursive schemes*

This dissertation models process interactions in phonology via logical transduction over strings. It borrows from logical techniques in model theory and graph theory (Engelfriet and Hoogeboom, 2001; Courcelle, 1994) and builds on a growing body of recent work in computational phonology that defines phonological processes as functions (Lindell and Chandlee, 2016; Strother-Garcia, 2018;

Chandlee and Jardine, 2019a; Koser et al., 2019; Mamadou and Jardine, 2020; Koser and Jardine, 2020; Oakden, 2020; Chandlee and Jardine, 2020). In particular, it adopts a formalism known *boolean monadic recursive schemes* (BMRS; Bhaskar et al., 2020) to model tone sandhi interactions. BMRS are based on the well-studied notion of recursive program schemes, a tool used in computer science to study the complexity of algorithms (Moschovakis, 2019). Chapter 3 introduces the formalism in detail, but here I give an intuitive introduction.

Broadly speaking, the BMRS formalism describes mappings from input structures to output structure. It does so using a series of unary, boolean functions that take input string positions as their domain. Functions define the conditions under which some part of the input structure maps to a particular output. This is achieved using an `if .. then .. else ..` syntax which operates over structures in the input or output. To demonstrate with a simplified example, recall Mandarin 3rd tone sandhi in (1.3): a low tone becomes a rising tone before another low tone. The crucial question here is whether a low tone maps to L (doesn't undergo sandhi) or R (undergoes sandhi) in the output. Two BMRS functions below in (1.10) describe how L and R outputs ($L_o(x)$ and $R_o(x)$ respectively) are computed in Mandarin tone sandhi. Variable x refers to any input position, $\top$ denotes 'true' and $\perp$ denotes 'false'.

(1.10)
$$\begin{aligned} L_o(x) &= \texttt{if } \underline{L}L(x) \texttt{ then } \perp \texttt{ else } L(x) \\ R_o(x) &= \texttt{if } \underline{L}L(x) \texttt{ then } \top \texttt{ else } R(x) \end{aligned}$$

The definition of $L_o(x)$ states the following: if x is an input L tone and is followed immediately by another input L tone (given by $\underline{L}L(x)$), $L_o(x)$ evaluates to $\perp$ (false). This means that x does *not* map to L in the output. Otherwise, if x is an input L tone (that is, not followed by another L), it maps directly to an output L tone.[3] For output L, the configuration $\underline{L}L(x)$ is a *blocking structure*, because it prevents L from surfacing in the output.

Definition $R_o(x)$, on the other hand, states that if x is an input L tone and is followed immediately by another input L tone, $R_o(x)$ evaluates to $\top$ (true), meaning that x *does* map to R in the output. The same configuration $\underline{L}L(x)$—that blocks output L—licenses output R, and is therefore referred to as a *licensing structure* for output R. Otherwise, if x is an input R tone, it maps directly to R in the output. Taken together, definitions $L_o(x)$ and $R_o(x)$ describe exactly the sandhi alternation in (1.3).

BMRS are a useful tool for phonological analysis. Chandlee and Jardine (2020) highlight some advantages of BMRS compared to other computational formalisms. This includes the ability to

[3] For the same reason, if it is not input-specified L, it will *not* map to output L.

intensionally express phonological generalizations in a way that gels with a traditional generative outlook on phonology. One example is blocking structures; in identifying marked structures that trigger phonological transformations, they are not unlike markedness constraints in OT. BMRS also provides a means to represent phonological substance, and this is something unavailable to other computational approaches such as finite-state models. Importantly, the BMRS formalism preserves crucial insights regarding the complexity of phonological processes, given the result of Bhaskar et al. (2020), who show that BMRS-definable transductions describe the subsequential class of functions.

A primary goal of this dissertation is to advance a computational theory of phonological process interactions, and demonstrate how analyses using this theory outperform rule- and optimization-theoretical accounts of tone sandhi interactions. However, it will also evaluate BMRS relative to other theoretical computational formalisms used in phonology. Bhaskar et al. (2020)'s result means that BMRS systems have the same extensional properties as, for example, deterministic finite-state transducers. And it is also the case that both formalisms can model process interactions either as single maps or as combinations of maps describing individual processes (see more discussion in Chapter 3, §5.1). However, this dissertation will highlight additional benefits of BMRS—compared to other computational approaches—that are specific to interactions. One key advantage of BMRS is that it provides an intuitive means of defining new *operations* over functions. This is crucial to the theory of interactions advocated in this dissertation. Additionally, the result of combining functions via operations is considerably more interpretable than equivalent combinations in other computational approaches, and the BMRS formalism furthermore clarifies the contribution of each individual process in such combinations. This clarity is especially useful in modeling interactions of multiple processes. Chapter 7 explores these issues in greater detail, building on the analyses developed in Chapters 5 and 6.

1.4 Main contributions of this dissertation

This dissertation makes three main contributions to phonological theory.

One contribution of this dissertation is that it represents the first major work using boolean monadic recursive schemes to explore a specific type of phenomenon (tone sandhi in Chinese dialects), building on recent formal and theoretical work on BMRS (Bhaskar et al., 2020; Oakden et al., 2020; Chandlee and Jardine, 2020; Oakden and Jardine, 2020). This dissertation demonstrates how the formalism can be fruitfully applied to sandhi interactions, and identifies its advantages, especially compared to other computational approaches. Ideally, it serves as a proof of concept for future work that utilizes BMRS to examine other phenomena in phonology, its interfaces, and beyond.

Second, this dissertation develops a novel computational theory of phonological process interactions: as operations over BMRS systems of equations. This provides for a straightforward solution to a number of outstanding cases of interaction in the tone sandhi literature. Stated simply, the theory pursued here captures Chinese tone sandhi in a way that previous rule- and optimization-based approaches cannot. This is true for directionality issues raised by Tianjin sandhi mentioned above, and importantly for certain paradoxical cases such as Changting and Xiamen addressed in detail in Chapter 6. Given the challenges that these tone sandhi interactions pose to SPE and OT, they are of broad theoretical concern. However, interest in them has waned within theoretical discussions of phonology in the past 20 years or so. A main contributor to this trend is the commonly-held attitude that these interactions are beyond the scope of current phonological theories. For example, Chen (2004, 818)'s assessment of Changting is as "a limiting case that severely test[s] the adequacy of conceptual tools at our disposal." Indeed, a full monograph devoted to Changting by Chen et al. (2004, 1-3) is prefaced with a grudging admission of failure to "render a satisfactory account of the Hakka facts, either in rule-based generative framework or in constraint-based OT terms." The intractability of the Xiamen paradigm, similarly, has led to its dismissal as irrelevant to questions of tonal phonology (Anderson, 1987; Ballard, 1988), and assertions that it is "neither learnable, nor productive, in fact 'not a part of the speakers' grammars, but historical artifacts.'" (Chen, 2000, 42). This has stymied serious inquiry into the properties of sandhi interactions—and their potential contributions to phonological theory more broadly—yet remains the consensus in the Chinese linguistics literature (Zhang, 2014). In proposing a computational theory of process interactions, this dissertation will demonstrate the continuing relevance of tone sandhi to phonological theory, and future work can further refine this theory by applying it to a broader set of interactions.

A third contribution of this dissertation comes in the form of BMRS applications to questions of phonological representation, building on recent computational work using model theory and logic to explore representation (Strother-Garcia and Heinz, 2015; Danis and Jardine, 2019; Oakden, 2020). These studies extend the notion of logical transduction between input and output structures—for modeling phonological processes—to mappings between structures that describe distinct theories of phonological representation. Under certain conditions, these mappings constitute formally-rigorous arguments that two representational schema are *notationally equivalent*. That is, that there are no substantive differences between the theories from a formal perspective. The dissertation leverages this tool to adopt one string-based representation of tone over another (see Chapter 4 for more details); it does so in two ways. First, it presents BMRS-definable transductions that map from one string-based structure to another (and vice versa). More importantly, however, it applies the composition operator defined in Chapter 5 to demonstrate the *bi-interpretability* (in the sense of Friedman and Visser, 2014) of these representations (see Chapter 7), a formal measuring stick of notational equivalence adopted in previous studies. The ability to compose transductions is a crucial component to proving bi-interpretability, but previous work in the area has fallen short (see especially Oakden, 2020) given the lack of an established composition procedure for logical transductions. Thus, future work can adapt BMRS-definable transductions—and their composition—to the notational equivalence of other reprsentational schema.

1.5 Outline of the dissertation

Chapter 2 provides a broad introduction to the empirical focus of the dissertation: interactions among tone sandhi processes in Chinese dialects. It identifies the subset of sandhi processes which form the main focus, presenting sandhi paradigms from four dialects to be analyzed in Chapters 5 and 6: Tianjin (Li and Liu, 1985; Chen, 1986; Shi, 1990; Chen, 2000), Changting (Li, 1965; Luo, 1982; Rao, 1987; Chen, 2004), Xiamen (Dong, 1960; Cheng, 1968; Chen, 1987), and Nanjing (Fei and Sun, 1993; Liu and Li, 1995; Ma and Li, 2014). Previous analyses of these sandhi patterns—if extant in the literature—are also summarized, along with issues of theoretical interest raised by each paradigm. The main goal of this chapter is to demonstrate the suitability of tone sandhi for analysis of phonological process interactions.

Chapter 3 establishes the formal framework for the computational analysis of interactions pursued in the dissertation. This includes outlining foundational concepts such as string models (the adopted representational formalism for tonal structures), the conception of phonological processes as input-ouput maps or functions, and the Subregular Hypothesis for phonology (Heinz and Lai, 2013; Heinz, 2018). This chapter also introduces *boolean monadic recursive schemes* (BMRS), the adopted formalism for describing functions. It describes the basic properties of BMRS and discusses some advantages of this formalism in capturing phonological generalizations (to be further developed in Chapter 7).

Chapter 4 leverages the computational framework to motivate the adoption of *syllabic* string models as the preferred representational schema for modeling tone sandhi interactions in Chinese dialects. Syllabic strings are string representations whereby the tone on each syllable is represented with a single symbol, including contours. Both conceptual and formal arguments are presented to motivate this type of representation specifically over *melodic* string representations; that is, those which decompose tonal contours into sequences of level tones. This chapter argues that not only do melodic representations not provide a more restrictive theory of tone sandhi than syllabic representations, but also that analyses utilizing melodic representations can make paradoxical generalizations about sandhi data. Two case studies of sandhi are presented—one from Hakha Lai (Hyman and VanBik, 2004) and another from Nanjing—to demonstrate the pitfalls of adopting melodic representations, and highlight the benefits of adopting syllabic representations. The conceptual motivation is then bolstered by a rigorous formal demonstration that syllabic representations and melodic representations (crucially those enriched with syllable boundary symbols) are notationally equivalent, following recent computational work on phonological representations (Strother-Garcia and Heinz, 2015; Danis and Jardine, 2019; Oakden, 2020). Given these results, syllabic string models are well-motivated as the representational scheme of choice.

Chapters 5 and 6 present a formal theory of interactions whereby phonological processes are defined as individual BMRS systems of equations, and process interactions are combinations of these systems using a set of two operators. Chapter 5 builds on Kaplan and Kay (1994)'s observation that *composition* of string relations models pairwise rule ordering—that is, the effect of one rule operating on the output of an earlier-ordered rule. It defines a composition operator over BMRS systems and relates this operator to pairwise rule ordering. A broad typology of *strictly-local* (Chandlee, 2014) function compositions is also presented. Drawing upon case studies of three dialects, each composition type is manifested in a specific interaction that is amenable to a BMRS-composition (and thus a rule-ordering) analysis. This includes: feeding in Tianjin, feeding in Changting, and counterbleeding in Nanjing. Composite systems are defined, and a series of evaluation tables are given to both familiarize the reader with the formalism, as well as to confirm that the analyses do in fact make the purported predictions. The analyses presented in Chapter 5 build on earlier insights about directionality effects in sandhi interactions (Chandlee, 2019; Oakden and Chandlee, 2020), and the chapter discusses benefits of this approach, in particular as they relate to previous attempts to explain directionality effects in SPE and OT frameworks. The chapter ends by presenting an ordering paradox in Changting sandhi; this interaction cannot be captured by BMRS composition, thus setting the stage for the next chapter.

Given that composition alone is insufficient to model sandhi interactions in this way, an additional operator—termed *parallel satisfaction* (PS)—is defined in Chapter 6, and added to the set of operators in the theory of interactions. Its formal properties are introduced, and it is shown that combinations of BMRS systems using PS model parallel evaluation of more than one BMRS-definable string function over a single input and output string. This type of operation bears some similarity to previous approaches such as simultaneous application of rules (Chomsky and Halle, 1968a; Postal, 1968; McCawley, 1968; Harms, 1968) and two-level phonology (Koskenniemi, 1983), and the PS operator is distinguished from these approaches. Using the enriched set of operators, analysis of a broader range of tone sandhi interactions is thus possible, including mutual counterbleeding and mutual bleeding in Changting, as well as the Xiamen tone circle. As in Chapter 5, the BMRS PS analyses outperform rule- and optimization-based accounts of both Changting and Xiamen. Alternative analyses of counterbleeding and counterfeeding in Nanjing are presented, which demonstrates an overlap in the mappings that can be captured by composition and PS operators respectively. The discussion section of this chapter introduces interactions which are not formalized by the PS operator, and poses the question of the relative order of systems using PS, to be discussed in further detail in Chapter 7.

Chapter 7 interprets the results of the preceding chapters, considers their ramifications, and identifies new avenues for research. First, it continues the discussion introduced in Chapter 2, which highlights the benefits of the BMRS formalism in formalizing phonological process interactions. It does so specifically by comparing the BMRS analyses to other formalisms in theoretical computational phonology. Representational issues are also examined, including how modification of representational assumptions permits alternative analyses to the Xiamen data. Additionally, the discussion of notational equivalence in Chapter 4 is revisited, and a more rigorous demonstration of the equivalence of syllabic and melodic string representations is presented, this time using the composition operator. Further formal properties of the PS operator are then presented, including its closure properties and non-commutativity. The chapter also presents an in-depth comparison of PS with other formalisms. Finally, areas for future research using BMRS are identified, including process-specific constraint effects (Davis, 1995; McCarthy, 1997), and learnability.

Chapter 8 summarizes the results of the dissertation and concludes.

2 Empirical Focus and Review

2.1 Introduction

This chapter introduces the empirical focus of the dissertation—tone sandhi in Chinese dialects—and motivates this focus as it is relevant to theoretical investigation of phonological process interactions. Tone sandhi is a tonal process attested in, among other groups, Chinese (or Sinitic) languages. This family consists of roughly ten mutually-unintelligible dialect groups: Mandarin (Guanhua), Jin, Wu, Hui, Gan, Xiang, Min, Yue, Pinghua, and Hakka (Kejia) (Yuan, 1960; You, 1992). Individual dialects within groups can be mutually-unintelligible as well (Norman, 1988), underlying the high degree of linguistic diversity among Chinese languages. Sandhi systems attested across these dialects reflect that diversity.

A sandhi system can contain multiple alternations, and in such systems interactions among individual patterns are commonplace. Interactions can be highly complex, and a number of outstanding cases of interaction exist in the literature which have evaded a straightforward analysis using current theoretical tools. For these reasons, tone sandhi is an appropriate phenomenon to examine within a study of phonological process interactions.

Introducing the class of alternations labeled tone sandhi, identifying relevant cases, and motivating their relevance to the theoretical questions broached by this dissertation are the primary goals of this chapter. To achieve these goals, this chapter is organized as follows. §2 introduces the diverse set of processes that have been described as tone sandhi, and identifies the subset that forms the main empirical focus of this dissertation. In particular, these are sandhi paradigms containing interacting processes. Four case studies from among this set—including outstanding cases in the literature—are discussed in detail in §3. Each subsection presents the basic facts and outlines the results of previous analyses, with a focus on the issues of theoretical interest. Finally, §4 summarizes and offers further justification for adopting tone sandhi as the empirical focus of the dissertation.

2.2 Tone Sandhi

Generally speaking, tone sandhi can be understood as a process affecting tones based on their context. A classic example is 3rd tone sandhi in Mandarin (an early discussion is due to Wang and Li, 1967); when two low-dipping tones—aka 3rd tone or 'L' below—appear adjacent to one another, the first surfaces as a rising tone ('R' below). This is shown in (2.1).[1]

(2.1)

xiao		'small'
L		citation form
xiao	*gou*	'small dog; puppy'
L	L	base form
R	L	sandhi form

The terms *citation*, *base*, and *sandhi form* are widely adopted in the tone sandhi literature (Chen, 2000; Yip, 2002; Zhang, 2014), but I explain them briefly here, following an overview by Chen (2000). Citation tones (Chinese *danzidiao* 'single word tone') refer to tones on syllables when pronounced in isolation, as in the low tone on *xiao* 'dog' in (2.1). In Chinese dialects, most tonal contrasts are preserved in this environment. This has led most analysts to conflate the citation tone with the *base* tone (Chinese *jidiao* 'base tone'), which in traditional generative thinking is the underlying form. The *sandhi* tone (Chinese *biandiao* 'changed tone'), then, is the tone on the surface form. Differences in citation and sandhi forms are taken as indication that a sandhi process has occurred.

Tonal contrasts are typically preserved in citation form and may neutralize in a sandhi environment. In Fuzhou (Chan, 1985, 1989), for example, a three-way contrast between H(igh), F(alling), and L(ow) tones is neutralized to H before another F tone.

(2.2)

a.			b.			c.		
sing		'new'	*sing*		'grown'	*sing*		'holy'
H		citation	F		citation	L		citation
sing	*ing*	'newlywed'	*sing*	*ing*	'grownup'	*sing*	*ing*	'sage'
H	F	base	F	F	base	L	F	base
H	F	sandhi	**H**	F	sandhi	**H**	F	sandhi

This situation is common in Chinese tone sandhi, and supports positing citation forms as underlying. However, there are cases where the opposite situation obtains; that is, where tonal contrasts only arise in a sandhi environment. Under such circumstances, it incorrect to assume that contrasts are preserved in isolation. An example comes from Wenling (Li, 1984); the contrast between a R(ising) and F(alling) tone is neutralized (to F) in isolation, only surfacing before another falling tone, as in (2.3).

[1] A common convention is to give phonetic values for tones using Chao tone letters (Chao, 1930); that is, where 1 denotes the low end of the pitch range and 5 the highest. Thus the low-dipping tone is given the value '214' and the rising tone '35'. This dissertation adopts syllabic string representations, i.e. 'L' and 'R' for low and rising tones. See chapter 3 for motivation and discussion.

(2.3) a.

bi		‘skin’
F		citation
bi	*li*	‘inside the skin’
R	F	sandhi

b.

bi		‘blanket’
F		citation
bi	*li*	‘inside the blanket’
F	F	sandhi

Sandhi interactions explored in this dissertation do not display such characteristics, and thus I join the general consensus in the field that citation forms are the basic (underlying) forms for tone.

2.2.1 What counts as tone sandhi?

Tone sandhi as a class of alternations has been defined in a variety of ways in the literature, and most descriptions are purposefully vague. Yip (2002, xx) defines sandhi as a “phonological process [usually tonal changes] which happens between words”, while Zhang (2014, 443) describes the alternations as “complex patterns of tone alternation caused by adjacent tones or the prosodic/morphosyntactic environment in which a tone appears”. Perhaps the most authoritative source on Chinese tone sandhi of the last twenty years, (Chen, 2000), describes sandhi thusly (19): “processes which often drastically alter the phonetic shape of adjacent tones, when they come into contact with each other in connected speech”.

The vagueness of these definitions underlies the breadth of the range of phenomena labeled tone sandhi. While patterns like 3rd-tone sandhi in (2.1) can be understood simply as phonological dissimilation via substitution based on a local tonal environment, this is by no means characteristic of tone sandhi in general, either in terms of tonological operation or the context in which the operation occurs. Tone sandhi—whether understood as assimilatory, dissimilatory (especially in terms of contour (Chang, 1992)), or something else—can manifest in the form of local substitutions, tonal metathesis, neutralization, and tonal deletion, among others (Chen, 2000). An example from Pingyao (Hou, 1980) illustrates metathesis as contour dissimilation in (2.4); a rising contour tone becomes a falling tone before another rising tone.[2] Note that this generalization holds whether the process is formalized over melodic string representations by permuting the order of single tones, or autosegmental representations by permuting the order of terminal tonal nodes.

(2.4)

hai	*bing*	‘become ill’
LH.	LH	base form
HL.	LH	sandhi form

[2] This dissertation adopts syllabic string representations (see Chapter 4), thus giving a non-metathetic account of Pingyao. The examples in this section are presented using the tonal notation of the original sources which differ from the representational assumptions adopted in the dissertation.

These changes are not limited to local environments of two adjacent tones, either. Tone sandhi in some Wu dialects often involves deletion of all but the initial tone in a phrase, followed by the extension (or spread) of the initial tone over the entire phrase. The result is extensive contrast neutralization in non-initial syllables. Data from Shanghai (Sherard, 1972; Yip, 1980; Zee and Maddieson, 1980) in (2.5) provide an illustration; in lexical compounds of three or more syllables, non-initial tones are deleted, and the initial tone spreads to the right edge of the domain (assuming that the lexical compound forms a domain, but see more below).[3]

(2.5)	sɨ	sang	cɨ	'bastard'	sɨ	ka	dha	cœ	'world war'
	HL	HL	LH	base tones	HL	LH	LH	LH	base tones
	HL	.	.	Deletion	HL	.	.	.	Deletion
	H.	L	L	Spread	H.	L	L	L	Spread

Additionally, the locus of sandhi application—often referred to as the 'sandhi domain'—is seldom definable by a local phonological environment alone. Instead, these domains commonly interact with morphological and syntactic structure, and even interface with stress and higher prosodic structure. A recent acoustic study by (Shih, 2017), for example, finds that the domain of 3rd tone sandhi in Taiwan Mandarin is the Major Phrase (Selkirk, 1984; Nespor and Vogel, 1987), with Minor Phrases being separated by glottal stops. In Xiamen (see §3), the sandhi domain has been claimed to be influenced by syntactic, morphological, and prosodic factors (Chen, 1986; Lin, 1994; Du, 1983). Some sandhi processes are even parameterized for specific types of constructions. Old Chongming, a Wu dialect, presents such a case (Chen and Zhang, 1997). Not only do identical tonal sequences behave differently in single lexical items (2.6a) vs phrasal environments (2.6b), but also differ depending on the type of morphosyntactic construction in which they occur (2.6c-d).

(2.6)	a.	*fang*	*xing*	'to let pass through customs' (lexical)
		M	LM	base tone
		HMH	H	sandhi form
	b.	*fang*	*ping*	'to lay flat' (phrasal)
		M	LM	base tone
		M	H	sandhi form
	c.	*si*	*dun*	'four meals' (number + measure word)
		M	M	base tone
		M	H	sandhi form
	d.	*ci*	*ci*	'every time' (reduplicated nouns)
		M	M	base tone
		HMH	M	sandhi form

[3] See (Xu et al., 1981; Selkirk and Shen, 1990; Duanmu, 1991) for more discussion of this phenomenon, but also acoustic work by Chen (2008) for a dissenting viewpoint and alternative explanation.

Crucially, this is not a property of the individual morphemes, in spite of the fact that sandhi is determined in part by morphological structure; *any* sequence of /M.M/ tones in a number+measure construction surfaces as [M.H], and in a reduplicated noun construction as [HMH.M].

Given the breadth of attested tone sandhi operations, it is also the case that these patterns vary in terms of tonological naturalness (in the sense of Hyman and Schuh, 1974; Hyman, 2007). Often, sandhi systems evade a straightforward characterization using rules that target natural classes, or a basic set of operations over an enriched autosegmental representation (e.g. feature-geometric representational models proposed by Yip, 1980, 1989; Bao, 1990; Duanmu, 1990, 1994, among many others).[4] Consider as an example the Southern Min dialect Zhangping (Zhang, 1985; Chen, 2000), in which disyllabic combinations of seven lexical tones surface as one of three surface forms in an essentially arbitrary manner. Below in (2.7), 'q' denotes a checked tone and 'T' any lexical tone.[5]

(2.7)

		second σ	
		R, LL, Hq, F, Fq	ML, L
first σ	R LL L Hq	**M**-T	**H**-T
	ML F Fq	**L**-T	

Sometimes, clear patterns only emerge when one examines the diachronic development of sandhi in related dialects. Zhang (2014, citing Mei (1977)) points out that 3rd tone sandhi in Mandarin originates from the Middle Chinese pattern *shang* → *yang ping* / ___ *shang*, which has phonetically-distinct (but identical in terms of MC *category*) exponents in other Northern dialects, as in (2.8).

(2.8) a. Jinan (Qian and Zhu, 1998)

Tone	MC category	
H.H	*shang.shang*	base form
F.H	*yang ping.shang*	sandhi form

b. Taiyuan (Wen and Shen, 1999)

[4]See chapter 4 for more discussion of representation.

[5]Originally given in Chao numbers: LL = '11', ML = '31', L = '21'. The characterization here is simplified somewhat as a result, but it begs the question of how to distinguish these three using only L and M. See Chapter 4 on representation for more discussion.

Tone	MC category	
F.F	*shang.shang*	base form
L.F	*yang ping.shang*	sandhi form

The arbitrary nature of numerous sandhi patterns presents a challenge both for synchronic analysis and in securing a robust typological characterization of the process itself.

Perhaps a better to question to ask, then, is: what *doesn't* count as tone sandhi? Answering this question is non-trivial, and is beyond the scope of this dissertation, but Chen (2000) identifies certain distinctions as a point of departure. First, categorical tone sandhi alternations can be compared to (oftentimes) gradient tonal coarticulation effects. A commonly-cited example of one such effect is: a falling tone in Beijing Mandarin [HL] does not fall to the bottom of the pitch range when followed by another tone, being produced instead as [HM] (Shih, 1988). Shen (1990, 1992) distinguishes tonal coarticulation from tone sandhi based on their basic properties (assimilatory only vs. assimilatory or dissimilatory) and their conditioning factors (largely language-independent and based on production vs. language-specific and based on morphological/phonological factors). The picture is considerably more complicated than this, and disagreement abounds regarding which patterns are regulated by the phonology and which patterns the phonetics.[6] Traditional accounts (Chao, 1968) categorize the [HL]/[HM] alternation described above as a proper sandhi pattern. The same is true for other phenomena as well—e.g. a 'sandhi' rule changing a rising tone to a high tone within trisyllabic sequences, but that subsequent work claims is actually a coarticulation effect (Shen, 1992; Shih and Sproat, 1992; Xu, 1994). Other tonal variations mentioned by Chen (2000) are less controversially distinguishable from tone sandhi. These include intonational effects like declination (Pierrehumbert and Beckman, 1988) and catathesis ('automatic downstep'; Liberman and Pierrehumbert, 1984), as well as morphologically-conditioned tonal modifications (called *bianyin* 'tone change') such as the tonal morpheme in Cantonese (see (Yip, 1980, 60-65) and (Bao, 1990, 182-193) for data and autosegmental analyses).

2.2.2 Interaction, directionality, and the focus of this dissertation

A sandhi paradigm comprises a set of individual alternations like those described above. Some dialects have only a few such alternations, but others contain many. Changting (Li, 1965; Luo, 1982; Rao, 1987), a Hakka dialect with five lexical tones—H(igh), M(id), L(ow), R(ising), F(alling))—

[6] More recent work has addressed this question in other dialects including Tianjin (Zhang and Liu, 2011), Nanjing (Sun and Huang, 2015), and Fuzhou Min (Li, 2015).

exhibits 15 sandhi changes among 25 possible disyllabic combinations.[7] In many dialects, disyllabic sandhi patterns generalize to longer sequences such that tonal changes in strings of three, four, or more tones within a sandhi domain can be understood in terms of the basic set of two-syllable patterns. Given that processes within a paradigm operate over a single, self-contained set of phonological primitives (i.e., lexical tones), sandhi patterns often interact in such environments as targets and triggers overlap. To illustrate, the Changting paradigm has two disyllabic sandhi changes: a falling tone becomes rising before a mid tone (FM → RM), and a high tone becomes falling before a rising tone (HR → FR). In a trisyllabic sequence /HFM/, these two patterns interact in the following way. Input F tone becomes R as a result of FM → RM. This R then triggers HR → FR, as its structural description is satisfied. Overlap in the target of the former and the trigger of the latter constitutes a feeding interaction, illustrated in the derivation in (2.9).

(2.9)

	/HFM/
FM → RM	H**R**M
HR → FR	**F**RM
	[FRM]

Examples such as the one above are widely attested in sandhi paradigms. Many systems demonstrate a complex interplay of both opaque and transparent interactions. This is the case for Changting trisyllabic sequences; overlapping targets and triggers between disyllabic changes gives rise to feeding, bleeding, counterfeeding, and counterbleeding within a single system (see Chen, 2004, for more details). The arbitrary nature of some disyllabic patterns also extends to longer sequences as well, meaning that interactions seldom evince general markedness pressures or phonetic grounding. They can be thought of as "pure interaction" in the sense that they offer a view into phonological computation of interactions, but divorced from synchronic, phonetically-grounded markedness effects. Tone sandhi paradigms are thus well-suited to studying the abstract properties of phonological process interactions. Rich and complex interactions play out over relatively small tonal inventories, and the descriptive literature abounds with such cases. As the following sections will show, many interactions have proven difficult for current theories of phonology to explain.

Much of this interaction is concentrated in sandhi systems with so-called 'right-dominant' directionality. This refers to an early but influential classification of sandhi systems into 'left-dominant' and 'right-dominant' groups based on which tones are preserved in sandhi and which alternate (Yue-Hashimoto, 1987; Chan, 1995a; Zhang, 2007). Put simply, left-dominant sandhi preserve the base tone on the initial syllable while alternating non-initial syllables, and right-dominant sandhi

[7] See §3.2 for more details.

preserve the base tone on the final syllable while alternating non-final syllables.[8] Zhang (2007) and others note an asymmetry between left- and right-dominant systems. The former are typified by rightward extension of the initial tone across the entire sandhi domain. A number of Northern Wu dialects exhibit classic left-dominant behavior, such as the Shanghai example in (2.5), but also Wuxi (Chan and Ren, 1989), Changzhou (Wang, 1988), and Tangxi (Kennedy, 1953). The latter are not, however, characterized by leftward extension onto non-final tones. Instead, right-dominant sandhi usually manifests as default insertion and paradigmatic neutralization. 3rd-tone Mandarin sandhi (2.1), neutralization in Zhangping (2.7), and substitution in Changting (2.9) are prototypical cases of right-dominant sandhi.

Interactions are less common in left-dominant systems because rightward extension obliterates all non-initial tonal contrasts 'in one sweep' (Chen, 2000, 98). Insertion and substitution in right-dominant systems, by contrast, impart local, incremental changes on input tonal strings. Given this and the tendency of targets and triggers to overlap (described above), interactions are common. The empirical focus of this dissertation, then, is on right-dominant sandhi systems in Chinese dialects, particularly those for which interactions are attested. The following section formally introduces four right-dominant sandhi paradigms to be analyzed in detail in subsequent chapters. Each exhibits a unique set of interactions among individual alternations.

2.3 Empirical Focus

Interactions within four right-dominant sandhi paradigms form the empirical focus of this dissertation. The purpose of this section is to introduce each study case. Each subsection includes the basic facts of respective paradigms as well as a summary of previous analyses, with a focus on the issues of theoretical interest that the patterns highlight. They are presented in the following order: Tianjin (§3.1), Changting (§3.2), Xiamen (§3.3), and Nanjing (§3.4).

2.3.1 Tianjin

The Mandarin dialect Tianjin (Chen, 1986, 2000) has garnered much attention in the literature for its seemingly paradoxical sandhi system.

[8] An alternative typology based on the behavior of targets and triggers is given in (Bao, 2004).

2.3.1.1 Basic paradigm

Chen (2000) assumes four lexical tones for Tianjin based on phonetic descriptions by Li and Liu (1985) and Shi (1990): two level tones H(igh) and L(ow), and two contour tones R(ising) and F(alling). Disyllabic sandhi targets sequences of identical tones, with the exception of HH sequences. Relevant data are presented below (Chen, 2000, p. 106):[9]

(2.10) a. FF $\mapsto$ LF e.g. $jing^F zhong^F \rightarrow jing^L zhong^F$ 'net weight'

b. LL $\mapsto$ RL e.g. $fei^L ji^L \rightarrow fei^R ji^L$ 'air plane'

c. RR $\mapsto$ HR e.g. $xi^R lian^R \rightarrow xi^H lian^R$ 'wash one's face'

A falling tone becomes low before another falling tone (as in (2.10a), hereafter 'FF rule'); a low tone becomes rising before another low tone (as in (2.10b), hereafter 'LL rule'); and a rising tone becomes high before another rising tone ((2.10c), hereafter 'RR rule').

The mode of application of these rules in sequences of three or more syllables has been a source of substantial debate. It centers on the observation that the rules appear to apply in different directions. Consider the tritonal sequences in (2.11).

(2.11) a. FFF $\mapsto$ FLF e.g. $suo^F liao^F bu^F \rightarrow suo^F liao^L bu^F$ 'plastic cloth'

b. LLL $\mapsto$ LRL e.g. $tuo^L la^L ji^L \rightarrow tuo^L la^R ji^L$ 'tractor'

c. RRR $\mapsto$ HHR e.g. $li^R fa^R suo^R \rightarrow li^H fa^H suo^L$ 'barber shop'

The forms above suggest that FF and LL rules apply right-to-left. For example, in the FF rule application above, the second and third Fs satisfy the triggering environment of the rule, and the middle F surfaces as L. The string no longer contains the conditioning environment (a sequence of two Fs), so FLF is the output of the rule. In (2.11b), the same generalization applies. The RR rule, however, does not apply in the same manner, that is, the output is not *RHR. Instead, the rule is said to apply left-to-right, to the first R, then the second R, yielding HHR.

Motivating and accounting for this discrepancy poses a challenge in and of itself. A further complication to the pattern is two feeding relationships which obtain between rules, where the output of one rule (a 'derived' tone) provides the conditioning environment for the application of another rule.

(2.12) a. LL rule feeds RR rule: R$\underline{\text{LL}}$ $\rightarrow$ $\underline{\text{R}\mathbf{R}}$L $\rightarrow$ **H**RL

b. FF rule feeds LL rule: L$\underline{\text{FF}}$ $\rightarrow$ $\underline{\text{L}\mathbf{L}}$F $\rightarrow$ **R**LF

[9] This is not the full disyllabic sandhi paradigm; there is also a FL $\rightarrow$ HL rule described as tonal absorption (Hyman and Schuh, 1974). However, I follow Chandlee (2019) in putting this rule aside, as accounting for it is straightforward and is not directly relevant to interactions.

A full account of Tianjin sandhi should explain the interactions in (2.12) in light of the directionality paradox of individual rule application. Unfortunately, morphological structure does not clarify the issue; the full paradigm is summarized with relevant data below (adapted from (Chen, 2000, p. 107)), which the reader will notice applies in a variety of morpho-syntactic (left-branching, right-branching, and unstructured) contexts. Chen et al. (2004) term this type of sandhi application 'structure neutrality':

(2.13)

Input	Output	[x x] x	x [x x]	[x x x]
RRR	HHR	*[li.fa] suo* "barber shop"	*mu [lao.hu]* "tigress"	*ma.zu.ka* "mazurka"
FFF	FLF	*[su.liao] bu* "plastic cloth"	*ya [re.dai]* "subtropical"	*yi.da.li* "Italy"
LLL	LRL	*[tuo.la] ji* "tractor"	*kai [fei.ji]* "pilot a plane"	
RLL	HRL	*[bao.wen] bei* "thermos cup"	*da [guan.qiang]* "speak in a bureaucratic tone"	
LFF	RLF	*[wen.du] ji* "thermometer"	*tong [dian.hua]* "make a phone call"	

For strings of Ls and Rs which may or may not satisfy the structural environment for either—that is, LL or RR—rule, directionality does not seem to matter. The mapping /LRR/ $\mapsto$ [LHR], for example, is achieved regardless of a leftward or rightward parse.

(2.14)

$\underline{\text{LR}}\text{R}$		$\text{L}\underline{\text{RR}}$	
|		|	
$\text{L}\underline{\text{RR}}$	n/a	$\underline{\text{LH}}\text{R}$	by RR rule
|		|	
LHR	by RR rule	LHR	n/a

Complications arise, however, when reconciling directionality of rule application in isolation and in situations of rule *interaction*. To illustrate: while the mapping /RRR/ $\mapsto$ [HHR] suggests a left-to-right parse, the mapping /RLL/ $\mapsto$ [HRL] (where the LL rule feeds the RR rule) requires a right-to-left parse.

(2.15)

$\underline{\text{RR}}\text{R}$		$\text{R}\underline{\text{LL}}$	
|		|	
$\text{H}\underline{\text{RR}}$	by RR rule	$\underline{\text{RR}}\text{L}$	by LL rule
|		|	
HHR	by RR rule	HRL	by RR rule

As Chen (2000, p. 109) observes: "the crux of the problem...is to find some general principle or principles which govern the traffic of sandhi operations". That is, how to determine the direction in which strings are parsed. A unified analysis of Tianjin reconciles the difference in directionality to account for the full set of trisyllabic forms, ideally using basic principles of the theory.

2.3.1.2 Previous approaches

Following Chen (1986), a sizable body of literature endeavored to identify the single (or set of) directionality principles underlying Tianjin. Early attempts couched in a rule-based framework stipulate directionality *ad hoc*. This is something of a necessity given that consistent rightward or leftward iterative application fails to derive the full paradigm. Similarly, cyclic application is a nonstarter, as sandhi application is structure-neutral (as in (2.13)). Zhang (1987), therefore, claims right-to-left application as default, as application happens 'away from the determinant', and that the left-to-right directionality of the RR rule is a special case attested in other dialects. Tan (1987) assumes the same. Hung (1987) instead proposes a phonotactic constraint to motivate both the rules and the directionality of their application, resulting in a less stipulative analysis. According to Hung's account, Tianjin bans sequences of adjacent low tones. This explains why HH sequences are well-formed, but LL, FF (=H**L**H**L**), and RR (=**L**H**L**H) sequences are not. FF and LL rules then apply right-to-left because doing so avoids the marked structures. For comparison, left-to-right application of the FF rule for a sequence FFF would yield *LLF which contains the banned sequence. Since left-to-right application of the RR rule does not yield adjacent low tones—e.g. RRR → HHR—it is free to do so. This early attempt advances output markedness as a condition driving directionality of application in Tianjin.

The advent of OT brought about a renewed interest in the paradox, as evaluation of output candidates using violable constraints promised a less stipulative explanation of directionality. Chen (2000, 110) proposes a constraint 'temporal sequence' (Temp) which "makes the default assumption that we apply rules left to right, in tandem with the planning and execution of speech." Thus in Chen's formalism, candidates are derivational histories and constraints target both output forms and derivations. Temp ranks below a set of general well-formedness conditions driven by the Obligatory Contour Principle (OCP), specifically an instantiation of the OCP that does not target HH sequences. Pressure to apply rules in a particular direction can therefore be overridden by a stronger surface markedness stricture—the OCP. This captures the difference in directionality of FF, LL, and RR rules in tritonal sequences in a very similar manner as Hung (1987)'s analysis. However, it falls short in its fidelity to classic parallel OT.

Ma (2005) points out that Chen's approach merely recasts a serial derivation in an optimization framework, and instead offers an account of Tianjin using basic principles of OT: parallel evaluation of a set of surface candidates against a ranked hierarchy of markedness and faithfulness constraints. Ma's solution leverages markedness constraints over tonal melodies—*L.LL militates against adjacent low tones LL.LL[10] while *XY.XY (essentially OCP for contours) covers rising LH.LH and falling HL.HL sequences—coupled with positional faithfulness privileging the right edge of the sandhi domain (Beckman, 1999). Ranking positional faithfulness >> markedness >> general faithfulness correctly selects optimal outputs for the tritonal mappings in (2.11) and (2.12) argued to require a right-to-left parse. That is, all but the form RRR $\mapsto$ HHR. This is not a coincidence; Ma maintains that right to left application results from the high ranking of a right-edge-preserving positional faithfulness constraint. To explain the left-to-right form RRR $\mapsto$ HHR, Ma claims that such forms are opaque and therefore amenable to an analysis using sympathy (McCarthy, 1999). While this analysis is more orthodox than Chen's, recent work (Hsiao, 2015) has criticized the combination of OCP and positional faithfulness for producing potential paradoxes in typology. More importantly, though, Ma's analysis still retains the basic assumptions about directionality—and crucially derivation—implicit in earlier work.

Other accounts reconcile directionality by appealing to correspondence in prosodic structure (Lin, 2008), while Wee (2010) uses tree structures to represent derivational histories, introducing the notion of inter-tier correspondence. This analysis, however, gives rise to questionable extensions to the theory, underlying the inherent evasiveness of an explanation of Tianjin using basic principles. Similar to Chen (2000)'s serialist account, the notion of derivational histories as candidates is unorthodox (Chen (2004) admits this in later work) in OT as it abandons the central tenet of parallel evaluation, coercing a serial outlook into a non-serial framework.

A computational approach to Tianjin abstracts away from how the sandhi patterns are represented within a particular grammatical formalism, and instead focuses on the type of computation necessary to a) determine the well-formedness of a given surface string or b) map a set of inputs to the set of attested outputs. An early paper by Jansche (1998) investigates the Tianjin paradox in a finite-state framework (Johnson, 1972; Karttunen, 1993; Kaplan and Kay, 1994), and shows that the pattern is eloquently captured as a finite-state acceptor.[11] This work lay the foundation for future work on the functional characterization of the paradox.

[10] as well as a falling+low (FL = H**L.LL**) pattern not discussed here

[11] Jansche also claims that, when modeled as transducers, the FF/LL rules cannot be computed deterministically, and thus are not (sub)-sequential (Mohri, 1997). Future work has shown that this is not the case, so I abstract away from this detail.

More recent computational work by Chandlee (2019) shows that the Tianjin data are describable using strictly-local functions, a class of functions that model a great number of attested phonological processes (Chandlee, 2014; Chandlee et al., 2015a; Chandlee and Heinz, 2018).[12] This shows that the 'complex' nature of Tianjin sandhi does not correlate to an increase in computational complexity, despite the issues posed to rule- and optimization-based frameworks. The main contribution here is how the computational perspective bears on the 'paradox' of directionality. Chandlee shows that the apparent difference in directionality between RR and FF/LL rules can be formalized in terms of input strictly-local (ISL) and output strictly-local (OSL) functions over string models (Chandlee, 2014, see also Chapter 3). Such a distinction mirrors the difference between simultaneous and iterative rule application in an SPE framework. Specifically, FF and LL rules describe Right-OSL (ROSL) functions, computed using bounded reference to the output string, and formalized as a automaton reading a string from the right. The RR rule, on the other hand, describes an ISL function, computed using bounded reference to the input string only. It does so regardless of whether the string is read from the left or the right. This asymmetry elegantly captures Tianjin sandhi, including feeding interactions.

Chandlee's analysis formalizes each of the three 'rules' as a finite-state machine, and the full paradigm as a single 'combined map' machine, but does not explore how the former relate to the latter in any great detail. Chapter 4 builds on (Chandlee, 2019), offering a computational account of the Tianjin data, and leveraging the BMRS framework to explore the relationship between individual 'rules' as functions and the combined map functions that comprise them.

2.3.2 Changting

Like Tianjin, the Hakka dialect Changting (Li, 1965; Luo, 1982; Rao, 1987) exhibits sandhi alternations that appear paradoxical. Sources vary somewhat on the details of disyllabic and trisyllabic sandhi; this dissertation follows the generalizations put forth in (Chen, 2004) (as well as (Chen et al., 2004), considered the most authoritative work on Changting).

2.3.2.1 Basic paradigm

Changting is a five-tone system comprising L(ow), M(id), H(igh), R(ising), and F(alling) tones. Out of 25 possible two-tone combinations, Chen (2004, 800) reports 15 combinations which undergo sandhi, as in the table in (2.16), where rows represent the leftmost tone in a disyllabic sequence and

[12] Oakden (2019a) provides an account of the same data using logical transduction.

columns the rightmost. Empty cells indicate no alternation, and 'x' stands for the tone that does not alternate.

(2.16)

	M	**R**	**F**	**H**	**L**
M		Lx			Lx
R	Hx				xF
F	Rx	Lx	Mx	Lx; xM	Rx
H	Fx	Fx			Fx
L	Mx		Mx	Mx	

Of these only a subset are relevant to the interactions explored in this dissertation. These are given in (2.17) with example forms; as with Tianjin, this section refers to each transformation as a 'rule' named based on its conditioning environment.

(2.17) a. LF $\mapsto$ MF e.g. $dai^L biao^F \rightarrow dai^M biao^F$ 'represent'

b. RM $\mapsto$ HM e.g. $han^R leng^M \rightarrow han^H leng^M$ 'cold'

c. MR $\mapsto$ LR e.g. $hua^M qian^R \rightarrow hua^L qian^R$ 'spend money'

d. LM $\mapsto$ MM e.g. $shi^L zhai^M \rightarrow shi^M zhai^M$ 'vegetarian'

e. ML $\mapsto$ LL e.g. $gan^M yuan^L \rightarrow gan^L yuan^L$ 'willing'

These basic patterns extend to sequences of three or more tones, with overlapping targets and triggers giving rise to interactions. Trisyllabic data in (2.18) illustrate the following feeding relationships: the LF rule (2.17a) feeds the RM rule (2.17b), and the MR rule (2.17c) feeds the LM rule (2.17d).

(2.18) a. LF rule feeds RM rule: R<u>LF</u> → <u>R**M**</u>F → **H**MF

b. MR feeds ML rule: M<u>MR</u> → <u>M**L**</u>R → **L**LR

Changting feeding interactions exhibit similar features to those in Tianjin, namely their structure neutrality. This is illustrated in (2.19).

(2.19)

Input	Output	[x x] x	x [x x]
RLF	HMF	*[wen.xue] shi*	*lan [mo.shui]*
		"history of literature"	"blue ink"
MMR	LLR	*[deng.xin] rong*	*zuo [ban.fang]*
		"corduroy"	"sit in the office"

The root of the controversy in Changting is not the feeding interactions described above. Rather, it stems from interactions between two pairs of alternations summarized in (2.20): the RM/MR interaction (2.20a-b) and the LM/ML interaction (2.20c-d).

(2.20) a. /MRM/ $\mapsto$ [LHM]

b. /RMR/ $\mapsto$ [HLR]

c. /MLM/ $\mapsto$ [MMM]

d. /LML/ $\mapsto$ [LLL]

Formalized in a rule-based framework, these interactions represent ordering paradoxes. To illustrate: the mapping in (2.20a) suggests the rule ordering MR >> RM. This derives the intermediate form LRM to which RM applies, yielding surface form [LHM]. In (2.20b), however, the opposite ordering is necessary; RM first applies to /RMR/ to generate intermediate representation HMR, then MR applies to derive the output [HLR]. Derivations in (2.21) show that the interaction is a kind of *mutual counterbleeding*, as each rule counterbleeds the other.[13]

(2.21) a.

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

b.

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

As Chen (2004) notes, these mappings are also structure-neutral.

(2.22)

Input	Output	[x x] x	x [x x]
MRM	LHM	*[hua.qian] duo*	*xin [bei.jing]*
		"spending a lot of money"	"new Beijing"
RMR	HLR	*[yi.jin] you*	*chu [feng.tou]*
		"one catty of oil"	"to show off"

Likewise, (2.20c) seems to indicate the ordering LM >> ML such that the former applies to /MLM/ to generate the surface form [MMM] directly—thus bleeding the latter. But (2.20d) again suggests ML >> LM and the reverse bleeding relationship. The LM/ML interaction represents a *mutual bleeding* paradox (originally due to Kiparsky, 1971, but also see (Baković, 2011)), summarized in the derivations in (2.23).

(2.23) a.

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

b.

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

The LM/ML interaction mappings are also structure-neutral based on data from (Chen, 2004).

[13] This terminology is used more for its convenience than its appropriateness, as no order of these rules correctly generates all attested surface forms.

(2.24)

Input	Output	[x x] x	x [x x]
MLM	MMM	*[gan.yuan] jiao*	*wo [shi.zhai]*
		"willing to teach"	"I am a vegetarian"
LML	LLL	*[ren.zhen] du*	*shi [xi.yao]*
		"seriously study"	"take western medication"

2.3.2.2 Previous approaches

Ordering paradoxes described above indicate that an analysis of Changting using serially-ordered rules is untenable. Chen (2004) demonstrates that no single order can derive outputs from input forms. Additionally, he argues in this paper and elsewhere (Chen, 2000; Chen et al., 2004) that ordering paradoxes are merely a side effect of directional rule application (Howard, 1972; Kisseberth and Kenstowicz, 1977)—and so the fundamental challenge in capturing Changting sandhi is of the same nature as that of Tianjin. This is based on the observation that the tritonal strings evincing the RM/MR interaction require a left-to-right parse, while the LM/ML interaction cases require a right-to-left parse, illustrated in (2.25).

(2.25) a. MR/RM: left-to-right

/RMR/:	<u>RM</u>R	→	**H**<u>MR</u>	→	H**L**R	[HLR]
/MRM/:	<u>MR</u>M	→	**L**<u>RM</u>	→	L**H**M	[LHM]

b. LM/ML: right-to-left

/LML/:	L<u>ML</u>	→	L**L**L	[LLL]
/MLM/:	M<u>LM</u>	→	M**M**M	[MMM]

Couched in similar terms as the earlier Tianjin OT analysis Chen (2000)—that is, where constraints are derivations—Chen (2004, 806) proposes a set of general principles which may bear on predicting directionality of tone sandhi application, operationalized as violable constraints. These are summarized in (2.26).

(2.26)
a. Structural affinity (SA): cyclicity of rule application
b. Temporal Sequence (Temp): apply rules from left to right
c. Derivational economy (Econ): minimize derivational steps
d. Transparency (Transp): maximize transparency (i.e. feeding and bleeding)
e. Simplicity (Simp): avoid contour (i.e. complex) tones on the surface
f. Wellformedness (WF): avoid marked tonal combinations on the surface

Despite the reasonably complete analysis this approach provides for Tianjin, it falls short of a unified analysis for Changting. Evidence comes in the form of ranking paradoxes for the same set of cases (2.20) that confounded the rule-based approach. In Chen (2004)'s account, ranking Temp above Econ is motivated as it correctly chooses the derivation MRM → LRM → LHM over MRM → MHM, as in (2.27).

(2.27)

	/MRM/	Temp	Econ
☞	MRM - LRM - LHM		**
	MRM - MHM	*!	*

But then the opposite ranking—Econ>>Temp in (2.28)—is necessary to select the derivation MLM → MMM over a different candidate containing an unattested surface form MLM → LLM → LMM.

(2.28)

	/MLM/	Econ	Temp
	MLM - LLM - LMM	**!	
☞	MLM - MMM	*	*

Given these complications, Chen et al. (2004) capitulate to the 'dauntingly complex' pattern of Changting sandhi, arguing that current phonological theory is incapable of capturing it.

Other work on Changting has made headway by focusing attention on how rules apply to intermediate representations in a derivation. Extending an early observation by Hung (1985), Hsu (1994, 1995, 2002, 2005) promotes an analysis based in part on the interaction of the One Step Principle (OSP), defined in (2.29)[14], with general wellformedness.

(2.29) **One Step Principle** (OSP): A tone that has undergone change cannot change again; a derived tone must not serve as input to another tone sandhi rule.

This principle disfavors derivations in which an input tone is operated upon twice by different sandhi rules. As an illustration, consider a trisyllabic input /RLL/. It has (at least) two derivational paths to an output (recall disyllabic rules from (2.17)): RLL → RFL → R**R**L and RLL → R**FL**.

(2.30) a.

RLL
|
R**FL** (RL → RF)
|
*R**R**L (FL → RL)

b.

RLL
|
RLL n/a
|
R**F**L (RL → RF)

Derivation (2.30a)—representing the unattested output—modifies the middle tone twice in the derivation, from input L, to F, to output R, violating OSP. The output of derivation (2.30b) contains

[14] Hsu has defined these principles differently in different papers. I include definitions from both (Hsu, 1994) and (Hsu, 2005).

an ill-formed surface substring FL (the locus of the RL → FL) rule, but does not make multiple modifications to the same input tone. According to Hsu's analysis, the OSP takes priority over surface well-formedness in selecting outputs from possible derivations. In later OT accounts (Hsu, 2002), these are expressed as violable constraints with the ranking OSP >> WFC. Further motivation for the OSP comes from its applicability to other sandhi such as Tianjin, Dongshi Hakka (Chiang, 1985), Yaoping (Yang, 1963; Zhan, 1993; Hsu, 2002), and Taiwanese secret languages based on Ilan (Li, 1985, 1997). In spite of being couched in an optimizational framework, Hsu's clearly derivational analyses bear the same burden as Chen (2000)'s Tianjin account with respect to unconventional extensions to the theory.

As was the case with Tianjin, sandhi interactions in Changting receive a straightforward account in a computational framework, falling well within established complexity bounds for phonological processes despite what rule- and optimization-based accounts might suggest. Oakden and Chandlee (2020), for example, show that the RM/MR interaction (2.20a-b) is describable using an ISL function, and that the LM/ML (2.20c-d) interaction is describable using an OSL function (see Chapters 4 and 5 for more details). A main contribution of that study is its demonstration that intermediate representations—i.e. within a derivation in a rule-based analysis—are not necessary to determine how an input tone will be mapped to an output tone within the sandhi paradigm. In fact, it is precisely the reference to these intermediate forms that creates the reported ordering/ranking paradoxes. Also like the computational analysis of Tianjin, this work represents 'rule'-pair interactions as single functions. In doing so, it foregoes an investigation of how the each 'rule' relates to the other within a pair. Chapter 5 of this dissertation builds on Oakden and Chandlee (2020)'s result, emphasizing how these relationships can be formalized in the BMRS framework.

2.3.3 Xiamen

The Southern Min tone circles are a notorious set of sandhi alternations sharing a distinct characteristic: mapping from base tones to sandhi tones in a circular chain shift pattern. This section introduces the oft-cited Xiamen tone circle (Dong, 1960; Cheng, 1968; Chen, 1987) as a representative case. While a substantial amount of literature has been devoted to identifying the sandhi domain in these dialects and their relationship to syntactic (Chen, 1987, 1992; Lin, 1994) and prosodic (Hsiao, 1991; Hsu, 1992; Ku, 1993; Duanmu, 1995) structure, this dissertation will focus instead on the circular nature of the mapping itself.

2.3.3.1 Basic paradigm

Xiamen is a five tone system comprising R(ising), H(igh), M(id), L(ow), and F(alling) tones:[15] Chen (1987) gives the following disyllabic sandhi data, where sandhi tones are given in bold.

(2.31)	a.	*p'ang*	H	"fragrant"
		p'ang tsui	**M**F	"perfume" (lit. fragrant + water)
	b.	*we*	R	"shoes"
		we tua	**M**L	"shoe laces"
	c.	*pĩ*	M	"sick"
		pĩ lang	**L**R	"patient" (lit. sick + person)
	d.	*ts'u*	L	"house"
		ts'u ting	**F**F	"roof" (lit. house + top)
	e.	*hai*	F	"ocean"
		hai kĩ	**H**R	"ocean front"

The above generalization is not limited to sequences of two tones; any non-final tone within a sandhi domain surfaces as a sandhi tone, and domains may contain arbitrarily long strings of sandhi tones. Two examples (also from Chen, 1987) demonstrate.

(2.32)	a.	*pang*	*hong*	*ts'e*	'fly a kite'
		L	H	H	Base form
		F	**M**	H	Surface form

b.	*yi*	*kiong*	*kiong*	*kio*	*gua*	*ke*	*k'uah*	*puah*	*tiam*		*tsing*	*ku*	*ts'eq*
	H	R	R	L	F	H	L	L	F	H	F	M	Base
	M	**M**	**M**	**F**	**H**	**M**	**F**	**F**	**H**	**M**	**H**	M	Surface

'He insisted that I read for another half an hour'

Curiously, the mapping from base (i.e. isolation) tone to sandhi variant proceeds in a circular manner, which Chen (2000, 42) describes as "a musical-chair pattern produced by the replacement of tone A by tone B, which is in turn replaced by tone C, and so forth." This is shown diagrammatically below in (2.33).

(2.33)

$$\begin{tikzcd}
& & L \arrow[dr] & \\
R \arrow[r] & M \arrow[ur] & & F \arrow[dl] \\
& & H \arrow[ul] &
\end{tikzcd}$$

Intuitively, a citation tone is realized as the 'next' tone in the circle when it appears in non-final position. Another way to state these generalizations is in the form of rewrite rules. A set of five

[15] Chen gives Chao letter equivalents 24, 44, 22, 21, and 53 respectively. In addition, this analysis follows the conventional assumption that citation forms are underlying.

rules in (2.34) describe the individual sandhi transformations in (2.31), where 'T' denotes any lexical tone.

(2.34) a. R → M / __ T

b. H → M / __ T

c. M → L / __ T

d. L → F / __ T

e. F → H / __ T

The mapping thus forms a type of chain shift. In a rule-based framework, linear chain shifts produce counterfeeding orders (Kisseberth and Kenstowicz, 1977). However, given the circular nature of the mappings in (2.33), Xiamen sandhi outputs cannot be produced by ordering these individual rewrite rules. To see why, consider the ordering $e < d < c < b < a$ in (2.35) as an example; this produces some correct surface forms but not others.

(2.35)

Input	RT	MT	LT	FT	HT
e	–	–	–	HT	–
d	–	–	FT	–	–
c	–	LT	–	–	–
b	–	–	–	MT	MT
a	MT	–	–	–	–
Output	MT	LT	FT	*MT	MT

Indeed, *any* permutation over the rules will result in an unwanted feeding relationship between two rules; above, rule *e* feeds rule *b* over input /FT/ producing *[MT] when [HT] is attested. Xiamen tone sandhi can therefore be described as a type of circular counterfeeding. And like mutual counterfeeding and bleeding patterns in Changting, the data present an ordering paradox.

2.3.3.2 Previous approaches

The past 50 years have witnessed numerous accounts of Xiamen, approaching the problem from various theoretical viewpoints. Examining even a modest portion of that work in any detail is beyond the scope of the current dissertation, so the discussion here shall proceed as follows. Rule-based and autosegmental analyses are introduced briefly. The main focus, however, is on the recalcitrant nature of circular chain shifts in an OT framework. This section introduces the fundamental challenge that Xiamen poses to classic OT, summarizes attempts to capture the paradigm using certain extensions to the theory, and examines some criticisms of those methods.

Despite the paradoxical nature of a rule-based analysis sketched in the previous section, some of the earliest accounts of Xiamen were rule-based (Wang, 1967; Cheng, 1968, 1973; Shih, 1986). A famous analysis posited by Wang (1967) casts the tone circle as switching operations over two tonal features: high and falling. Five Xiamen tones are represented as matrices of these features, as in (2.36).[16]

(2.36) a. High: $\begin{bmatrix} +\text{high} \\ -\text{falling} \end{bmatrix}$

b. Mid: $\begin{bmatrix} -\text{high} \\ -\text{falling} \end{bmatrix}$

c. Low: $\begin{bmatrix} -\text{high} \\ +\text{falling} \end{bmatrix}$

d. Falling: $\begin{bmatrix} +\text{high} \\ +\text{falling} \end{bmatrix}$

e. Rising: $\begin{bmatrix} +\text{high} \\ -\text{falling} \end{bmatrix}$

A single alpha-switching rule (2.37) derives the circular mapping—H→ M → L → F → H—between those four tones.

(2.37) $$\begin{bmatrix} \alpha \text{ high} \\ \beta \text{ falling} \end{bmatrix} \rightarrow \begin{bmatrix} \beta \text{ high} \\ -\alpha \text{ falling} \end{bmatrix}$$

This analysis, though approximating the Xiamen facts, is not without criticism. Chen (2000, 43-44) remarks that it is essentially arbitrary and lacks any explanatory power. In particular, the focus on the phonetic substance of Xiamen tones fails to explain the more basic issue wanting attention: the arbitrary nature of the circular mapping. Wang's analysis might work for Xiamen, but fails with other historically-related Southern Min tone circles. A related dialect Longxi exhibits a different tone circle in terms of surface phonetic values, but when one strips away this level of representation, we find that the mappings are identical in terms of their Middle Chinese categories. The diachronic relatedness of the two patterns notwithstanding, a superior analysis ideally pinpoints the basic properties of the circle such that it could apply to the Southern Min circles in general.

Other researchers apply an autosegmental approach (Yip, 1980; Wright, 1983; Du, 1983), capturing the tonal chain shift as operations over autosegmental representations. Yip (1980)'s analysis hinges on Mid tone's ability to assume two distinct surface representations (albeit being pronounced the same); the transformation R → M is a tonal node deletion process producing one representation

[16] The phonetic value of L is 21 in Chao tone letters, thus is +falling. Also, H and R have the same featural specifications.

while H → M is the result of a switch in register (± upper in her model; see Chapter 4) producing a different representation. These are summarized in (2.38a) and (2.38b) respectively.

(2.38) a. $$\begin{array}{ccc} \text{[-upper]} & & \text{[-upper]} \\ \wedge & \rightarrow & | \\ \text{L H} & & \text{H} \end{array} \quad = \text{R} \rightarrow \text{M}$$

b. $$\begin{array}{ccc} \text{[+upper]} & & \text{[-upper]} \\ \wedge & \rightarrow & \wedge \\ \text{H H} & & \text{H H} \end{array} \quad = \text{H} \rightarrow \text{M}$$

This effectively 'breaks' the circle. Positing two phonological representations for phonetically-indistinguishable tones transforms a circular chain shift into a linear one. And while it successfully accounts for the data (including dialectical variations on the same pattern), both Moreton (2004) and Barrie (2006) criticize this approach as ad hoc and arbitrary. It also lacks the same explanatory adequacy as the rule-based accounts by side-stepping the core issue of circularity, instead reducing the pattern to a representational issue, despite being motivated more generally.

It is also the circularity of the pattern that provides the greatest challenge for optimization-based theories of phonology. Linear chain shifts are problematic for classic OT given that they are neither output-driven (Tesar, 2014) nor idempotent (Magri, 2018). Circular chain shifts such as Xiamen are especially vexing, then. Moreton (2004) proves that such mappings cannot be computed by any conservative OT grammar—that is, a grammar consisting of markedness and faithfulness constraints only. To provide a simple illustration of why, consider a circular mapping A → B → A. The submapping /A/ → [B] implies that surface form [B] is less marked than the fully-faithful candidate [A]. But the other submapping /B/ → [A] also implies that surface [A] is less marked than fully-faithful [B], leading to a contradiction—[B] must be less marked than itself. Xiamen thus poses an empirical challenge to the predictions about phonological maps made by OT. Moreton salvages his main point—that conservativity is a property of phonological processes expressible by OT grammars—by claiming that since Xiamen is conditioned by a non-homogeneous representational factor (phrase-level prosody), it does not provide a counterexample to the claim that conservative OT grammars cannot compute circular chain shifts. Instead, Xiamen is simply an example of paradigmatic substitution, irrelevant to the phonological generalizations captured in OT.

This early result did not stifle further exploration into Xiamen within an optimization framework, as other scholars aimed to bring the tone circle under the umbrella of phonology proper. Extension of the basic set of constraint types to include anti-faithfulness (Alderete, 1999, 2001), for example, paved the way for subsequent analyses using variants of OT enriched with these constraints (Hsieh, 2005;

Thomas, 2008). Mortensen (2002, 2004) cautions, however, that anti-faithfulness analyses contravene the basic principles of OT, that is, that phonological alternations are the product of the competing pressures to avoid marked surface structures and preserve underlying contrasts. More concerning is the danger that anti-faithfulness is too powerful, and thus limits the valuable restrictions classic OT places on the space of possible grammars.

Other analyses of Xiamen in OT are of a similar vein. Barrie (2006), for example, offers a model using contrast preservation and tokenized markedness (based on Łubowicz, 2003). In this framework, EVAL comprises two stages—one that applies preserve contrast and markedness constraints and another that applies generalized faithfulness constraints. Additionally, candidates are not surface representations, but rather 'scenarios' illustrating transformation patterns. This begs the question of whether such augmentations—like Chen (2000)'s derivational-histories-as-candidates—are desirable for the theory. And while this approach does model the circular chain shift using independently-motivated constraints, it only applies in cases where some degree of neutralization obtains. As Mortensen (2006) and Hsiao (2015) point out, however, there exist numerous non-neutralizing tone circle patterns, both in Southern Min and elsewhere.

Circular chain shifts, exemplified by the Xiamen tone circle, pose significant challenges to rule-based and optimizational theories of phonology. This dissertation will argue that much like Tianjin and Changting, Xiamen tone sandhi enjoys a straightforward characterization in a computational framework.

2.3.4 Nanjing

Nanjing is a Jianghuai Mandarin dialect (Liu and Li, 1995; Fei and Sun, 1993). Sandhi interactions in trisyllabic sequences have only recently been reported in the literature, following a production study by Ma and Li (2014). This dissertation will thus offer one of the first formal analyses of this data.

2.3.4.1 Basic paradigm

Nanjing has five lexical tones: H(igh), L(ow), R(ising), F(alling), and a C(hecked) tone, which is described as high, short, and containing a glottal stop, and which is distinct from the high tone in terms of its phonological—i.e. tone sandhi—behavior. Liu and Li (1995) report six right-dominant disyllabic tone sandhi alternations out of 25 possible combinations of five lexical tones, summarized in (2.39)[17]. Nanjing differs from other dialects surveyed so far in that it exhibits a sandhi variant of the checked tone—denoted C′ below—that is not found among the set of citation tones. C′ is defined as having a lower pitch than C (3 in Chao letters compared to 5).

(2.39) a. FF ↦ HF e.g. $bing^F xiang^F \rightarrow bing^H xiang^F$ 'refrigerator'

b. LF ↦ RF e.g. $lao^L shi^F \rightarrow lao^R shi^F$ 'teacher'

c. LL ↦ RL e.g. $hen^L hao^L \rightarrow hen^R hao^L$ 'very good'

d. HC ↦ FC e.g. $shu^H xue^C \rightarrow shu^F xue^C$ 'mathematics'

e. RC ↦ LC e.g. $tong^R xue^C \rightarrow tong^L xue^C$ 'classmate'

f. CC ↦ C′C e.g. $qi^C shi^C \rightarrow qi^{C'} shi^C$ 'seventy'

Note also that the checked high tone (denoted C) and the non-checked high tone (denoted H) are distinct entities within this sandhi system. For example, a rising tone undergoes sandhi when it appears before a checked high tone (2.39e); the same alternation does *not* occur when a rising tone appears before a non-checked high tone. That is, /LH/ maps directly to [LH].

Traditional sources provide little information about tone sandhi in sequences of more than two syllables. A recent production study by Ma and Li (2014), however, provides experimental evidence for sandhi interactions in trisyllabic forms and explores the issue of directionality in sandhi application. The authors perform an acoustic analysis on production data elicited from four native speakers of Nanjing. To investigate sandhi interactions, target stimuli constitute interaction contexts; that is, sequences of three syllables where targets and triggers of the disyllabic alternations in (2.39) overlap. Additionally, to test whether Nanjing sandhi is sensitive to morpho-syntactic structure, tonal combinations with different constituent structures—e.g. [xx]x, x[xx] as in (2.13,2.19,2.22)—were used. Their analysis produces a set of ten interaction mappings in (2.40).[18]

(2.40) a. /FFF/ ↦ [HHF] f. /LLL/ ↦ [LFL]

b. /LFF/ ↦ [RHF] g. /CCC/ ↦ [CC′C]

c. /LLF/ ↦ [RRF] h. /LRC/ ↦ [LLC]

d. /RCC/ ↦ [LC′C] i. /FHC/ ↦ [FFC]

e. /HCC/ ↦ [FC′C] j. /LHC/ ↦ [LFC]

[17] Liu and Liu and others also report a pattern triggered by the combination of a rising tone and a neutral tone (*qingsheng*) by which the latter surfaces as a low tone. This dissertation will not address this issue, but the reader is referred to (Sun, 2003) for more details. Additionally, there are differences in the phonetic realization of lexical tones between older and younger speakers, though the sandhi paradigm is robust. See (Song, 2006), (Liu and Li, 1995) and especially (Chen and Wiltshire, 2013) for more information.

[18] Two surface forms are reported for (2.40f), LFL and FFL, with the former being produced by female participants and the latter produced by male participants. This dissertation adopts the former, but the computational characterization developed in the following chapters could accommodate either.

Like Tianjin and Changting, these mappings are structure-neutral; morphological structure does not affect sandhi application. In addition, they are opaque; (2.40b-e) exhibit counterbleeding on environment and (2.40h-j) counterfeeding on environment when disyllabic sandhi patterns in (2.39) are conceived as rewrite rules. Consider two derivations of /LFF/ ↦ [RHF] in (2.41a) and /FHC/ ↦ [FFC] in (2.41b).

(2.41) a.

$\underline{\text{LF}}\text{F}$		$\text{L}\underline{\text{FF}}$	
|		|	
$\text{R}\underline{\text{FF}}$	by LF rule	LHF	by FF rule
|		|	
RHF	by FF rule	*LHF	LF rule n/a

b.

FHC		$\text{F}\underline{\text{HC}}$	
|		|	
$\text{F}\underline{\text{HC}}$	by FF rule n/a	$\underline{\text{FF}}\text{C}$	by HC rule
|		|	
FFC	by HC rule	*HFC	by FF rule

In their discussion of the results, Ma and Li (2014) categorize the interaction mappings in terms of their directionality, demonstrating that the majority of the patterns exhibit *left-to-right* application of sandhi in Nanjing (note that the attested forms in (2.41) are derived with a left-to-right parse). It is not completely uniform, however; the one exception is a sequence of three checked tones /CCC/, for which speakers produce [CC′C]. Schematized below in (2.42) are leftward and rightward parses of the same string which show that this sequence requires a right-to-left parse:

(2.42)

Input	Right-to-left	Left-to-right
/CCC/	$\text{C}\underline{\text{CC}} \rightarrow \text{C}\mathbf{C'}\text{C}$	$\underline{\text{CC}}\text{C} \rightarrow \mathbf{C'}\underline{\text{CC}} \rightarrow {}^{*}\text{C}'\mathbf{C'}\text{C}$

Only by scanning the string from the right edge and proceeding leftward can the attested output [CC′C] be derived from application of the CC rule. Scanning the string in the opposite direction—i.e. from the left and proceeding rightward—yields an unattested form. Thus much like Tianjin and Changting, the authors reduce the question of sandhi interactions to direction of application.

2.3.4.2 Previous approaches

Very few analyses of Nanjing sandhi exist in the literature, and no systematic attempt has been made to explain the data in (Ma and Li, 2014). Ma (2009)'s OT analysis of disyllabic sandhi is similar in nature to his 2005 account of Tianjin: surface markedness constraints target ill-formed submelodies, and interact with positional faithfulness (specifically to the right edge of a sandhi domain) to select attested surface forms. Oakden (2012) adopts Duanmu (1990, 1994)'s autosegmental model to analyze disyllabic sandhi as a mixture of OCP effects and constraint conjunction over these representations. Since disyllabic sandhi patterns are uniformly right-dominant, directionality is a non-issue in those analyses.

Ma and Li (2014) offer a functional—but not formal—explanation of the discrepancy, noting that the single mapping which requires a rightward parse consists entirely of checked tones, a category which has a special status in the dialect: its realization includes segmental (glottal coda) and durational (shorter than other tones) information in additional to pitch modulation, and it is the only tone whose sandhi variant is not also a citation tone.[19] Their assessment is as follows (106; my translation):

> In general, trisyllabic tone sandhi in Nanjing tends toward rightward directionality. [The mapping /CCC/ $\mapsto$ [CC′C]] is a special case; this is very likely related to the nature of checked tone. When checked tone occupies the first position of a trisyllabic sequence, speakers likely lean toward preserving the citation tone form at the beginning of a phrase in order to highlight checked tonal features. Thus, the result is a trisyllabic sequence in which the initial checked tone does not undergo sandhi.

How this tendency might be formalized is not addressed in their study. This dissertation will not attempt such an undertaking, but instead will demonstrate how the Nanjing interaction data—along with Tianjin, Changting, and Xiamen—are accounted for in a computational framework.

2.4 Discussion

The preceding sections demonstrate the suitability of tone sandhi to the study of phonological process interactions. Sandhi paradigms in these four dialects comprise rich systems of (oftentimes) arbitrary tonal changes, confined to a limited set of targets and triggers. Overlap of targets and triggers is common in longer tonal sequences where patterns can be understood in terms of the basic two-syllable rules. Many of these paradigms provide clear challenges for existing theories of phonology, both rule-based and optimization based, and so tone sandhi provides a unique opportunity to study the properties of interactions. This section provides further justification for adopting tone sandhi as the empirical focus of this dissertation.

Zhang (2014) notes that tone sandhi has played an important role in the development of phonological theory by providing further evidence for the autosegmental nature of tone and their feature-geometric representation. To the extent that the sandhi interactions summarized above pose signif-

[19] There is much to be said about checked tone in Nanjing. Another possibility is the typological uniqueness of checked tone among Mandarin dialects, and its current status as a mid-merger tonal category; see, for example, (Gu, 2015; Tang, 2019) for more information.

icant challenges to SPE and OT, they are also of broad theoretical concern. However, interest in these systems has waned within theoretical discussions of phonology in the past 20 years or so. One contributor to this is the commonly-held attitude that these interactions are beyond the scope of current phonological theories. Chen (2004, 818)'s assessment of Changting, for example, is a "limiting case that severely test[s] the adequacy of conceptual tools at our disposal." An entire monograph devoted to Changting by Chen et al. (2004, 1-3) is prefaced with a grudging admission of failure to "render a satisfactory account of the Hakka facts, either in rule-based generative framework or in constraint-based OT terms." This fatalism has pointed the field away from Changting and similar paradigms for which attempts at formal analysis are considered futile. How to correctly represent the tonal primitives undergoing sandhi (and thus which mechanisms drive it) introduces yet another complication. Depending on one's representational assumptions, a single sandhi pattern can be interpreted to support multiple, sometimes conflicting, analyses.[20] This has stymied serious inquiry into the basic properties of sandhi and their interactions because they are impossible to fruitfully analyze, either due to their complexity or the arbitrariness of representational choice.

To a certain extent this attitude is long-standing. The arbitrary nature of Southern Min tone circle patterns like Xiamen—not traditionally considered an interaction in the same vein as Changting and Tianjin—has led to doubts concerning its psychological reality and its presence in the synchronic grammars of speakers. According to Chen (2000, 42), Xiamen and similar alternations "often strike the analyst as bizarre because they seem to relate or map one tone to another in an essentially arbitrary and whimsical manner." In a footnote, he echoes earlier dismissals by Anderson (1987) and Ballard (1988), who assert that such patterns are irrelevant to questions of tonal phonology (at least a feature-based theory of tone), and go as far to say that the synchronic rules comprising them are "neither learnable, nor productive, in fact 'not a part of the speakers' grammars, but historical artifacts.'" An early claim by Schuh (1978) casts Xiamen as an instance of 'paradigmatic replacement', and Moreton (2004) echoes this sentiment. Tsay and Myers (1996) advance an 'allomorph selection hypothesis', relegating Xiamen tone sandhi to the morphological component of the grammar. Thus in addition to potentially insurmountable analytical challenges, there is suspicion that sandhi is irrelevant to theoretical discussions of phonology because it is not truly phonological.

Recent work has perpetuated this viewpoint by calling into question the veracity of traditional impressionistic descriptions. For example, phonetic studies of Tianjin (Zhang and Liu, 2011; Li and Chen, 2012) suggest that patterns reported in earlier literature either apply inconsistently

[20] Zhang (2014) summarizes the contentious issue of tonal representation and its effect on analysis of tone sandhi, offering several diagnoses and possible solutions. See also (Chen, 2010) for a relevant discussion of the representation of contour and its application to Danyang (Lü, 1980) tone sandhi.

or not at all, or that some are better understood as tonal coarticulation and not tone sandhi. The slimmed-down paradigm is thus no longer paradoxical in the original sense, allowing for a straightforward analysis (Wang and Lin, 2017). Additionally, psycholinguistic studies of reduplication and wug-tests in Xiamen indicate that the tone circle is not productive (Zhang et al., 2006, 2009, 2011), thus lending support to the notion that it does not form part of the speaker's phonological competence. This also indirectly insinuates that the impressionistic accounts were describing impossible patterns to begin with.

Results of phonetic/psycholinguistic studies contradicting impressionistic accounts has become increasingly common (see for example (Bowern et al., 2013) and (Shih, 2018) for work on stress), and justifiably raises questions about the reliability of impressionistic field work descriptions, particularly from non-native speaker investigators. As they apply to tone sandhi interactions, however, these studies have not been without criticism. Both Hsieh (2005) and Barrie (2006) point out weaknesses in wug-test paradigms used in earlier work on Xiamen (Hsieh, 1970; Wang, 1995); results of such tests might not be reliable because they ask participants to apply tone sandhi to phonotactically ill-formed words. The tone circle is, however, productive in loan words, so the pattern does apply to novel forms in more naturalistic contexts. This is explainable under the assumption that the tone circle is a phonological transformation, but perhaps less so with an allomorph selection hypothesis. Furthermore, Mortensen (2006, 103-107) takes issue with the philosophical underpinnings of such arguments and rails against 'psychological reality' as a measuring stick for what processes phonological theory is tasked with explaining. Of Min tone circles he says (105-6):

> In practice, in fact 'psychological reality' is usually evoked as a means of dismissing a pattern or phenomenon for which one's theory of choice is unable to account... Investigators, presented with two generalizations, A and B, where A is compatible with the existing theory and B is not, are likely to investigate the psychological reality of B (e.g. Min tone sandhi) rather than A. The result of this practice is simply the reinforcement of existing ideas in the face of potentially disconfirmatory facts.

Work by Mortensen (2006) and Hsiao (2015) casts additional doubt on the possibility that Min tone circles are a historical fluke. This is because circular tonal shifts are attested outside the Min group, including the Mandarin dialect Laoling (Cao, 2007) and the Tibeto-Burman language Jingpho (Matisoff, 1974; Dai, 1990; Lai, 2002). And while careful instrumental studies of sandhi enrich the descriptive literature, they should not be used as evidence that such processes cannot exist in a phonological grammar. In fact, experimental work of the same vein shows that such interactions can be a part of phonological competence; Ma and Li (2014) offer rigorous description of a sandhi system in Nanjing that is not unlike Tianjin in exhibiting directionality effects. Some sandhi appear to apply left to right and others right to left, with no general principle adequately governing their application. It is this quality—not the specifics of the rules or even the phonetic values of tones themselves—that provides a challenge to existing theories, and thus why sandhi interactions are of interest to phonological theory in general. Taking all the facts into account, then, I adopt the traditional descriptions of Tianjin and Xiamen, as the *nature* of the paradoxes they represent are important to address.

Recent computational work has shown that, despite the issues some recalcitrant sandhi cases pose to existing theories of phonology, their computational complexity is still very limited, and importantly aligns with that of well-attested phonological processes (Chandlee, 2019; Oakden and Chandlee, 2020). These preliminary results suggest that further inquiry into the abstract properties of sandhi interactions is a potentially fruitful venture despite the lack of enthusiasm in the broader literature. The computational perspective is well-suited to this undertaking. Because it examines the nature of the interaction mappings themselves, it can strip away assumptions particular to a grammatical formalism or representational theory (recalling Zhang (2014)'s discussion of this issue). This dissertation will expand on these earlier results in the form of a more comprehensive investigation. Among other things, it will argue that patterns like Min tone circles are predicted by a restrictive computational theory of phonology, and are part of speakers' phonological knowledge—that is, these processes are psychologically real. A related goal is to show that—as it was in the past—tone sandhi continues to be relevant to phonological theory in general. It has the capacity to inform our theory of interaction provided the relevant patterns are analyzed in a particular way.

3 Formal Foundations

3.1 Introduction

This chapter lays the formal foundation for the computational analysis of tone sandhi interactions to be pursued in the dissertation. It is organized as follows. §2 introduces string models, the representational formalism for tonal structures that undergo sandhi transformations. §3 outlines the computational perspective on phonological processes—whereby processes are formalized as input-output mappings—introducing the subregular hypothesis adopted in the dissertation and several key concepts within that hypothesis. In §4, I present a formalism for describing input-output mappings known as boolean monadic recursive schemes, and explain its basic properties. §5 discusses some of the advantages of this formalism as it pertains to a formal investigation of process interactions in phonology.

3.2 String models

In this dissertation, I adopt a model-theoretic approach (Courcelle, 1994; Enderton, 2001; Libkin, 2013) to provide a mathematically-rigorous definition of syllabic strings of tones (see discussion in Chapter 4 on motivating representations). The goal is to establish a precise characterization of the phonological structures that undergo transformations. In the computational literature, this approach has been applied to various phonological representations beyond simple strings, including autosegmental representations (Chandlee and Jardine, 2019a), syllabic structure (Strother-Garcia, 2018), segmental features (Strother-Garcia, 2019), and feature-geometric models of tone (Oakden, 2020) among others. Such definitions are valuable for formalizing transformations as functions or mappings from some input structure to a corresponding output structure.

Under this approach, strings of lexical tones are defined as models. A model is a mathematical object comprising a *domain* of elements and a set of their *relations* defined over some universe of elements, for example an alphabet Σ which corresponds to an inventory of lexical tones. To illustrate, consider a dialect like Standard Mandarin (Li and Thompson, 1989) with four lexical tones: high, low, rising, and falling. Example (3.1) provides string representations of these lexical tones with phonetic correspondents in terms of Chao (1930) tone letters: 1-5 where 1 corresponds to the lowest pitch in a speaker's register and 5 corresponds to the highest.

(3.1)

	Chao tone letter	String	Example
a.	55	H	*ma* 'mom'
b.	35	R	*ma* 'hemp'
c.	214	L	*ma* 'horse'
d.	51	F	*ma* 'scold'

Suppose we wanted to represent the string of input tones L+L that undergoes Mandarin 3rd tone sandhi (3TS), recalling example (2.1) from the previous chapter, repeated as (3.2) below.

(3.2)

xiao		'small'
L		citation form
xiao	*gou*	'small dog; puppy'
L	L	base form
R	L	sandhi form

This can be achieved with a relational string model defined over an alphabet $\Sigma = \{\text{H, L, R, F}\}$ corresponding to Standard Mandarin's four-tone inventory. Let such a model $\mathcal{M}_{sm}$ be defined $\mathcal{M}_{sm} \stackrel{\text{def}}{=} \langle \mathcal{D}; P_H, P_L, P_R, P_F; p, s\rangle$. The domain $\mathcal{D}$ contains a set of integers denoting individual elements in the model (string positions). Their linear order is determined by unary predecessor p and successor s functions; these take a string position as input and output the position immediately preceding or following it.[1] P_H, P_L, P_R, and P_F are instantiations of $(P_\sigma)_{\sigma\in\Sigma}$); this set of unary relations labels elements with a particular feature—that is, the property of being a high, low, rising, or falling tone.

The models considered are restricted such that a given domain element is in no more than one unary relation, a common assumption in traditional model-theoretic work (Büchi, 1960).[2] Thus it is not possible for a single element to be labeled as high *and* low. A string model of an L+L tonal sequence is defined in (3.3a). Its graphical equivalent is in (3.3b).

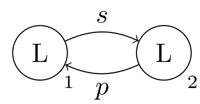
(3.3) a.

$$\mathcal{D} = \{1,2\} \qquad P_H = \{\} \qquad P_L = \{1,2\} \qquad P_R = \{\}$$
$$P_F = \{\} \qquad p(x) = \begin{cases} 1 & x = 2 \end{cases} \qquad s(x) = \begin{cases} 2 & x = 1 \end{cases}$$

b.

[1] These functions can be defined as total functions such that the final element in the string is its own successor and the first element in the string is its own predecessor. This dissertation uses partial predecessor/successor functions as in the following example.

[2] But see (Strother-Garcia et al., 2016; Strother-Garcia, 2019; Chandlee et al., 2019) for related work which relaxes this assumption.



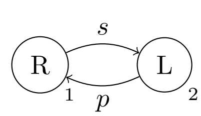
The model contains two elements in its domain, one for each lexical tone segment. Natural numbers are used to denote the domain elements for clarity, but this choice is arbitrary and does not suggest any intrinsic ordering on or relationship between the elements; this is defined entirely in the relations and functions. Unary relations label string positions with one of the tones from the alphabet, and are denoted by sets containing the domain element bearing that label. P_L is the set containing elements 1 and 2, meaning that both string positions are labeled L. Since none of the other three lexical tones show up on string positions, all other relations—P_H, P_R, P_F—are the empty set $\{\}$. Unary predecessor and successor functions define ordering relations over these positions, such that 1 is the immediate predecessor of 2, and 2 the immediate successor of 1 (indicated with labeled arrows). Taken as a whole, this model offers an explicit characterization of a disyllabic LL sequence in Standard Mandarin.

The unary relations and functions described in the model above are not limited to the description of a single string. They comprise a general model *signature* $\zeta = \{P_H, P_L, P_R, P_F; s, p\}$ over which any tonal string in Standard Mandarin may be represented. The sandhi form (output) of (3.2) is represented in much the same way in (3.4); the different sets denoting unary relations reflects the difference between input and output structures.

(3.4) a.

$$\begin{array}{llll} \mathcal{D} = \{1, 2\} & P_H = \{\} & P_L = \{2\} & P_R = \{1\} \\ P_F = \{\} & p(x) = \begin{cases} 1 & x = 2 \end{cases} & s(x) = \begin{cases} 2 & x = 1 \end{cases} & \end{array}$$

b.



Models provide a means for explicit representation of phonological structures. Relating input structures to output structures models phonological transformations, the focus of the next section.

3.3 Transformations as functions

That phonological processes like tone sandhi are psychologically-real *transformations* from an underlying form to a surface form is a foundational idea in generative perspectives on phonology, including SPE (Chomsky and Halle, 1968a) and OT (Prince and Smolensky, 2004). In the former, transformations are formalized using a potentially-ordered set of rewrite rules. In the latter, a transformation is the result of an evaluation over a set of possible output candidates against a ranked hierarchy of constraints. Phonological transformations are also amendable to a more abstract characterization: a function. A function of the form $f : A \rightarrow B$ pairs members of set A (the set of underlying representations) to set B (the set of surface realizations) such that each member of A is paired to at most one member of B. Sandhi processes in Chinese dialects may be formalized as functions which relate the set of input tonal strings to corresponding output strings that have—or have not—undergone sandhi. Using Standard Mandarin 3TS as an example:[3]

(3.5)

Input	Output
L	L
LL	RL
LLL	RRL
LLLL	RRRL
⋮	⋮

A function modeling Mandarin 3TS maps any input string to a corresponding output string regardless of length. That is, it is a relation between infinite sets.

The current section frames this view of phonological transformations as it applies to modeling tone sandhi and their interactions in this dissertation. It begins with an introduction to computational complexity (and the important notion of classifying functions based on their complexity), outlining early results relevant to phonology (§3.1). Chief among these is the subregular hypothesis, which argues (in part) that phonological transformations are sufficiently described by a subsequential class of functions (Heinz and Lai, 2013; Heinz, 2018). §3.2 and §3.3 introduce subclasses of the subsequential class, the input strictly-local and output-strictly local functions, respectively. Sandhi processes and interactions examined in this dissertation are describable by functions of these two classes.

[3]For simplicity, this assumes that strings of tones belong to a single domain. Deriving sandhi domain formation is not a primary objective of this dissertation; see (Shih, 1986, 1997) and (Chen, 2000, ch. 9) for a prevailing explanation in terms of prosody. The generalization for sandhi in strings of 2+ syllables reflects data found in (Zhang, 1997), e.g. *gou*^{R} *bi*^{R} *ma*^{R} *xiao*^{L} 'a dog is smaller than a horse'.

3.3.1 Complexity and the (sub)regular hypothesis

What is the nature of the function in (3.5)? More generally, what is the nature of functions which describe phonological transformations such that they can be distinguished from functions which describe *non*-phonological transformations? For example, we may also imagine a function for the set of Standard Mandarin input strings, but instead of producing sandhi outputs, it simply produces a mirror image of the string by reversing its linear order.

(3.6)

Input	Output
H	H
RH	HR
FHRL	LRHF
HLRRFL	LFRRLH
⋮	⋮

Another example is a function which takes symbols in a string and puts them in alphabetical order such that HLRRFL $\mapsto$ FHLLRR. These functions are definable but are clearly unnatural as phonological processes and are likely unattested.[4] A means to restrict functions such that they describe (ideally) all and only phonological transformations, and do not describe transformations like the one above, becomes necessary.

One avenue is to categorize functions based on their computational complexity. A foundational measuring stick for the complexity of formal languages—crucially *sets* of strings and not mappings between strings as explored in this dissertation—generated by grammars and expressing linguistically-significant generalizations is the Chomsky hierarchy (Chomsky, 1956), given in (3.7). This hierarchy is a set of nested classes of increasing complexity; e.g. every regular language is also a context-free language but not the other way around.

(3.7) finite $\subset$ regular $\subset$ context-free $\subset$ context-sensitive $\subset$ recursively enumerable

The Chomsky hierarchy has been applied to natural language phenomena, and it has been noted that linguistic patterns differ in terms of the expressivity of the grammars that generate them (Chomsky, 1959; Shieber, 1985; Heinz and Idsardi, 2011, 2013). For example, syntactic patterns fall within the context-free/context-sensitive region, but no phonological pattern has been shown to do so.

Instead, an early result relevant to phonology (Johnson, 1972; Kaplan and Kay, 1994) determined that SPE-style rewrite rules of the form $A \rightarrow B\ /\ C\_\_\ D$ correspond to the *regular* class of relations

[4] However, Lamont (2018) shows that alphabetization is possible in OT and Harmonic Grammar, even using simple constraints.

(that is, they generate regular languages) leading to the hypothesis that phonology is regular.[5] In intuitive terms, regular relations are those for which a fixed amount of memory is required to compute an output; the needed memory does not increase as the size of the string increases. Thus the Mandarin 3TS pattern is regular in that the rule can be computed using a fixed amount of memory. Scanning through the input string, if an input L is followed by an input L, the first L is output as an R. This generalization thus requires a scanning window of length 2: the current symbol and the following symbol. Importantly, this description holds true for strings of any length—including potentially arbitrarily long strings—using the same fixed window, as in the example below.

(3.8)

Input	Output	Steps
...LLLLL...	...**R**LLLL...	1
...LLLLL...	...R**R**LLL...	2
...LLLLL...	...RR**R**LL...	3
...LLLLL...	...RRR**R**L...	4

The restriction that phonological transformations must be regular correctly rules out phonologically unnatural functions like the mirroring function in (3.6) whose memory requirements increase proportionately to the size of the input string.

(3.9)

Input	Output
H	**H**
RH	**HR**
FHRL	**LRHF**
HLRRFL	**LFRRLH**
⋮	⋮

To produce a mirror image of a string, the requisite memory window is the length of the string itself. As the size of the string increases, so does the window. This pattern is therefore not regular.

Refinements were needed to rule out functions that were regular in their complexity but unattested for phonology. This includes pathological spreading patterns such as sour grapes.[6] Sour grapes is a logically-possible but unattested progressive vowel harmony pattern whereby vowels harmonize, but only if there are no opaque vowels later in the word. Example (3.10), using notation from (Heinz and Lai, 2013), offers some examples where '+' indicates a vowel containing the harmony trigger, '−' a vowel containing the target, and '⊟' an opaque vowel.

[5]With the important caveat that the rule does not apply to its own output within the same cycle.

[6]The term is originally due to Padgett (1995), who uses it to refer to spreading of an entire bundle of feature-geometric representations or none. The sense of sour grapes used by Heinz and Lai (2013) in this section is that of Wilson (2003).

(3.10)

Input	Output
$+ + -$	$+ + +$
$+ - +$	$+ + +$
$+ - \boxminus$	$+ - \boxminus$
$+ - - \boxminus - -$	$+ - - \boxminus - -$
$\vdots$	$\vdots$

The problem with patterns like sour grapes is that they require unbounded lookahead; a vowel needs access to information at a potentially arbitrary distance to determine whether or not it will harmonize. Restricted subclasses of the regular region therefore must capture the *myopic* nature (in the sense of Wilson, 2003) of spreading patterns and phonological processes in general, to distinguish them from regular—yet pathological—patterns like sour grapes.

In terms of functions, Heinz and Lai (2013) identify a well-known restriction on regular relations that captures this generalization: the subsequential (SEQ) class (Schützenberger, 1977; Mohri, 1997). The subsequential class is properly regular, and is divided into two incomparable subclasses, left-subsequential (LSEQ) and right-subsequential (RSEQ). Importantly, when parsing a string, subsequential functions impose a bound on the lookahead in either direction. The Subsequential Hypothesis—the hypothesis that phonological functions must be subsequential functions (Heinz and Lai, 2013; Heinz, 2018)—thus presents a more restrictive theory of phonological transformations. Subsequent work demonstrates that a variety of phonological processes can be modeled with subsequential functions: vowel and consonant harmony (Heinz and Lai, 2013; Luo, 2013, 2017), dissimilation (Payne, 2014, 2017), and metathesis (Chandlee and Heinz, 2012), among others.[7] Subsequentiality is also an attractive complexity bound for phonology because functions of this class has also been shown to be learnable (Oncina et al., 1993; Chandlee et al., 2014; Jardine et al., 2014).

3.3.2 Input strictly-local (ISL) functions

One of the most restrictive classes of subsequential functions is the *input strictly-local* class (ISL, Chandlee, 2014). Functions of this class, simply put, compute outputs using only local, bounded reference to input structure. Mandarin 3TS is regular, subsequential, and also ISL. A diagram in (3.11) shows the mapping /LL/ ↦ [RL].

(3.11)

Input:	L	L
Output:	**R**	L

[7]Other work has identified non-subsequential tonal (Jardine, 2016) and vowel harmony (McCollum et al., 2017; McCollum and Essegbey, 2018) patterns. This dissertation acknowledges these results but notes that the tone sandhi patterns and their interactions investigated here are properly subsequential, along with a significant amount of phonological processes.

Whether input /L/ maps to its base form [L] or sandhi form [R] (the gray cell) can be determined solely from a bounded window of information in the input (the pink cells). In this case, the size of the window is two string positions: the tone on the current position and on the position immediately to its right. An ISL-2 function—indicating a window of size 2—is sufficient to describe not merely the mapping in (3.11), but the infinite set of maps {(/LLL/, [RRL]), (/LLLL/, [RRRL]), (/HLLH/, [HRLH]), ...}.

Chandlee (2014) finds that ISL functions model the simultaneous application of rules of the form $A \to B$ / C__ D. Written as a rule L → R / __ L, the ISL-ness of Mandarin 3TS is apparent in the mappings in (3.5). 'Application' of the rule is triggered by satisfaction of the structural description in the input string, and crucially not the output string (but see the next section).

As a class, ISL functions are sufficient to model a wide range of segmental and autosegmental phonological processes, despite their restrictiveness (Chandlee, 2014; Chandlee and Jardine, 2019a). Maps describable by this class encompass those corresponding to individual processes, as well as multiple phenomena applying to the same input structure (Chandlee and Heinz, 2018). Chandlee et al. (2018) find that a number of opaque interactions are also formalizable as ISL functions. Interactions of sandhi processes explored in subsequent chapters also share this quality.

3.3.3 Output strictly-local (OSL) functions

Chandlee (2014) also identifies a class of output strictly-local functions (see also Chandlee et al., 2014, 2015a; Chandlee and Heinz, 2018). As the name suggests, functions of this type compute outputs using a bounded window in the output structure. OSL is split into two intersecting subclasses, L(eft)-OSL and R(ight)-OSL—proper subsets of Left- and Right-subsequential classes respectively—named so based on the orientation of the output window. To provide an illustration, recall the LL rule in the Tianjin dialect (Li and Liu, 1985) introduced in the previous chapter. On its face, it resembles Standard Mandarin 3TS; both map input /LL/ to [RL]. The patterns diverge, however, in strings of length greater than two, with Tianjin resembling iterative right-to-left rule application. Compare the respective mappings in (3.12).

(3.12)

Standard Mandarin		**Tianjin**	
Input	Output	Input	Output
LLL	RRL	LLL	LRL
LLLL	RRRL	LLLL	RLRL
LLLL	RRRRL	LLLLL	LRLRL

OSL functions model iterative rule application in the same way ISL functions model simultaneous application. This difference is a reflection of the formal difference between OSL and ISL functions: OSL functions scan a local output window and ISL functions a local input window. As Chandlee (2019) argues, the Tianjin LL pattern describes a ROSL function—outputs are computed using a bounded output window *to the right* of the current input string position under evaluation. The diagram in (3.13) demonstrates the computation of /LLL/ $\mapsto$ [LRL] in Tianjin.

(3.13) a.

Input:	L	L	L
Output:		**R**	L

b.

Input:	L	L	L
Output:	**L**	R	L

The triggering environment for the Tianjin LL rule is still a sequence of two L tones, much like 3TS, with the difference being that the second L in the sequence is an *output* L. In (3.13a), the second string position (in gray) is output as R because it is an input L and its immediate successor in the output is L. Evaluation of the first string position in (3.13b) does not produce an output R; its immediate successor in the output is R. That it is input-specified as L is immaterial.

OSL functions model iterative application like Tianjin LL sandhi above, as well as output-oriented spreading processes not captured by ISL functions like nasal spreading in Johore Malay (Onn, 1980). Subsequent chapters in this dissertation build on the result in (Chandlee and Heinz, 2018) by showing that, like ISL functions, both individual processes and their interactions are formalizable as OSL functions.

3.4 Boolean monadic recursive schemes (BMRS)

This dissertation follows a growing body of work that defines phonological transformations using logical transduction (Lindell and Chandlee, 2016; Strother-Garcia, 2018; Chandlee and Jardine, 2019a; Koser et al., 2019; Mamadou and Jardine, 2020; Koser and Jardine, 2020; Oakden, 2020). Simply put, logical transduction is a method for representing mappings from input structures to output structures. A transduction comprises a set of logical formulae, one for each relation and function in an output model, to be interpreted in terms of the structure of the input model (Courcelle, 1994; Engelfriet and Hoogeboom, 2001; Filiot, 2015).

In this section, I introduce a type of logical transduction using *Boolean Monadic Recursive Schemes* (henceforth BMRS; Bhaskar et al., 2020; Chandlee and Jardine, 2020) to be utilized in the following chapters. BMRS have their basis in recursive program schemes, which are used to study the complexity of algorithms (Moschovakis, 2019). They are a useful tool for two reasons. First, Bhaskar et al. (2020) show that BMRS describe exactly the subsequential class of functions, so they have utility in an analysis couched within the subregular hypothesis. Second, they capture both input-based and output-based mappings, which will be crucial in accounting for the full extent of tone sandhi processes and interactions explored here.[8] This section begins with an introduction to the structure of BMRS and BMRS logical transductions (§4.1), then introduces classes of BMRS corresponding to ISL (§4.2) and OSL (§4.3) functions.

3.4.1 Structure of BMRS

This section introduces the basic structure of BMRS and BMRS transductions. The discussion presented here is intuitive, and focuses on application of BMRS to modeling phonological transformations over string structures.[9]

3.4.1.1 Sorts and Terms

BMRS operate over string models using two types (sorts) of data structures: boolean values $\top$ (true) and $\bot$ (false) and string positions (also called indices). The basic units of BMRS are called *terms* (T), and are formed with the following grammar:

(3.14) $$T \rightarrow \mathtt{x} \mid T = T \mid \top \mid \bot \mid \mathtt{f}(T) \mid p/s(T) \mid \sigma(T)(\sigma \in \Sigma) \mid \mathtt{if}\ \ T_1\ \ \mathtt{then}\ \ T_2\ \ \mathtt{else}\ \ T_3$$

Terms can be of sort boolean or sort index. Sorts are determined inductively with the following rules.

(3.15)

a. For terms $\mathtt{f}(T)$, $\sigma(T)$, $p(T)$, and $s(T)$, T must be of sort index.

b. Sorts T_1 and T_2 for term $T_1 = T_2$ must match.

c. For terms $\mathtt{if}\ \ T_1\ \ \mathtt{then}\ \ T_2\ \ \mathtt{else}\ \ T_3$, T_1 must be sort boolean, sorts T_2,T_3 must match.

Terms of sort index range over positions in a string model. This includes the variable x, and the terms $p(T)$ and $s(T)$. As in section 2 of this chapter, p and s identify the immediate predecessor and successor of a given string position. That is, when given an index term T denoting a string position, they return either the position that comes immediately before or after it.

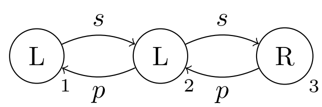
Unlike index terms which return a string index, boolean terms return a boolean *value*: true $\top$ or false $\bot$. This includes $T = T$ (the equality of two terms), $\sigma(T)$ (which I focus on in this section), $\mathtt{if}\ \ T_1\ \ \mathtt{then}\ \ T_2\ \ \mathtt{else}\ \ T_3$ (see §4.1.2), and $\mathtt{f}(T)$ (see §4.1.3). Terms $\sigma(T)$—one for each symbol σ in a string model's alphabet Σ—denote monadic predicates. These take a single index

[8] See also (Chandlee and Jardine, 2019b,c; Oakden, 2019a; Oakden and Chandlee, 2020).

[9] For a more formal introduction, see (Bhaskar et al., 2020; Oakden et al., 2020). For an introduction of BMRS structure using feature-based representations, see (Chandlee and Jardine, 2020)

term T as an argument and return a boolean value ($\top/\bot$) based on whether or not the string index is labeled with that particular symbol σ. Importantly, index and boolean terms can be embedded within one another, provided they conform to the inductive rules defined in (3.15). Thus $p(p(x))$ ('the predecessor of the predecessor of x') is a well-formed term, as is $\sigma(s(x))$ ('the index that is the successor of x is labeled with σ').

To illustrate with an example, consider a string model of a sequence of three tones: LLR. The table below shows, for each string position, the *index* values returned for $p(x)$ and $s(x)$, as well as the *boolean* values returned for predicates $L(x)$ and $R(x)$. It also shows the values returned for $L(s(x))$—that is, whether the successor of some position is labeled L.[10]

(3.16) 

Label:	L	L	R
Position:	1	2	3
$p(x)$		1	2
$s(x)$	2	3	
$L(x)$	$\top$	$\top$	$\bot$
$R(x)$	$\bot$	$\bot$	$\top$
$L(s(x))$	$\top$	$\bot$	$\bot$

When the variable x is interpreted as string position 1 or 2, the predicate $L(x)$ returns a true value, because those positions are labeled L. Likewise, $R(x)$ returns $\top$ when x is interpreted as position 3 (but importantly not positions 1 or 2). Only position 1 is true for the predicate $L(s(x))$, because it is the only position in the string followed immediately by an index labeled L.

3.4.1.2 `if`...`then`...`else` and local structures

Boolean terms of the form `if` T_1 `then` T_2 `else` T_3—which I refer to as 'if-then-else statements'—are the logical workhorse of the BMRS formalism as they are utilized in this dissertation. They take three terms T_1, T_2, T_3, as arguments, and return a boolean value depending on the *evaluation* of those terms. Evaluation of these terms follows that of similar statements in programming languages, and crucially *not* material implication in logic. That is, if T_1 (a boolean term) returns true $\top$, then T_2 is evaluated (i.e. returning its boolean value). If T_1 returns false $\bot$, then T_3 is evaluated. Graphically:

[10] Recall that a partial successor function is assumed, meaning that $(s(x))$ is undefined on the final string position and that $(p(x))$ is undefined on the initial string position. I adopt the convention, like Chandlee and Jardine (2020), that $\sigma(T)$ and $\mathtt{f}(T)$ evaluate to false whenever T returns undefined.

(3.17)
$$\begin{tikzcd} & T_1 \arrow[dl, "\top"'] \arrow[dr, "\bot"] & \\ T_2 & & T_3 \end{tikzcd}$$

Consider, in (3.18), an if-then-else statement that takes monadic predicates $L(x)$ and $R(x)$ as terms, as well as a graphical representation of its evaluation.

(3.18) `if` $L(x)$ `then` $\top$ `else` $R(x)$

$$\begin{tikzcd} & L(x) \arrow[dl, "\top"'] \arrow[dr, "\bot"] & \\ \top & & R(x) \end{tikzcd}$$

The evaluation of this statement can be described as follows. If x is interpreted as a string index labeled L, return a true ($\top$) value. Otherwise, return a value based on evaluation of R(x): $\top$ if x is interpreted as a string index labeled R, and $\bot$ otherwise. Note that all three string positions in the LLR example in (3.16) would return a true value upon being evaluated by the statement in (3.18). This is consistent with the intuition that the above if-then-else statement describes the state of being labeled as either L or R.

This dissertation will use if-then-else statements to describe *local* structures over strings. For example, consider the following statement (and its graphical evaluation):

(3.19) `if` $L(x)$ `then` $L(s(x))$ `else` $\bot$

$$\begin{tikzcd} & L(x) \arrow[dl, "\top"'] \arrow[dr, "\bot"] & \\ L(s(x)) & & \bot \end{tikzcd}$$

Evaluation first checks whether a certain string position is labeled L ($L(x)$), and if it is, evaluation then checks whether its immediate successor is labeled L ($L(s(x))$). A true $\top$ value is returned only when these two conditions are met; otherwise a false $\bot$ value is returned. In other words, this statement *identifies*—i.e. it only returns a $\top$ value for—a sequence of adjacent L symbols in the string, that is, a *local* structure. Statements of this type are also well-formed BMRS terms, and so I will denote them with a shorthand notation, such that the term `if` $L(x)$ `then` $L(s(x))$ `else` $\bot$ is written $\underline{L}L(x)$. To briefly illustrate, consider the evaluation of different positions in the string LLR against the term $\underline{L}L(x)$, as in the evaluation table below.

(3.20)

Label:	L	L	R
Position:	1	2	3
$\underline{L}L(x)$	$\top$	$\bot$	$\bot$

String position 1 evaluates to true $\top$ for $\underline{L}L(x)$ because it is labeled L *and* so is its immediate successor. This is not the case for the second and third positions, which therefore return false $\bot$ for $\underline{L}L(x)$. In other words, the only LL sequence in this string is the one that begins with string position 1.

The ability to identify local string structures is a crucial property of the BMRS formalism, and will be utilized extensively in this dissertation. This is important because it captures the intuition that the majority of tone sandhi patterns are computationally local (Chandlee, 2019; Oakden, 2020; Oakden and Chandlee, 2020, see also the analyses in chapters 5 and 6).

3.4.1.3 Systems of equations and BMRS transductions

Having described the basic structure of BMRS, I now turn to how the formalism is used to model phonological transformations, that is, as mappings (i.e. functions) from inputs to outputs using logical transduction. In this section, I use the example of Mandarin 3rd tone sandhi to illustrate.

In intuitive terms, logical systems like BMRS can formalize input/output mappings by defining the output structure in terms of the input structure (Engelfriet and Hoogeboom, 2001; Courcelle, 1994; Enderton, 2001).[11] This is achieved in BMRS via a *system of equations.* A system of equations is a set of function names $\mathtt{f}(T)$ of sort boolean (recall the grammar in (3.14)), where for each function name $\mathtt{f}_i$, there is a corresponding term $\mathtt{f}_i(x) = T_i$, where T_i is a term of sort boolean. I will refer to this term T_i as the 'definition' of the function.

Over tonal string models, the set of output functions in a system of equations corresponds to the set of output tones. Thus, a system of equations modeling Mandarin 3TS comprises four function names $H_o(x)$, $L_o(x)$, $R_o(x)$, $F_o(x)$—one for each output tone in Mandarin—along with the definition for each function. Note that these function names are denoted with a subscripted '$_o$' to distinguish them from the monadic predicates (e.g. $L(x)$, $R(x)$) introduced in the previous subsection.

Function definitions describe the conditions under which some input string position (interpreted as x) is mapped to a particular tone in the output, and are built using well-formed BMRS terms. If the function definition evaluates a $\top$ (true) value for some input string position interpreted as x, it is mapped to that tone in the output. Likewise, if the term evaluates to a $\bot$ (false) value, that position does *not* map to the given output tone. Each input position in a string will evaluate to $\top$ for exactly one output function (and $\bot$ for all the others), and so the output structure is 'built' from the input structure based on the function definitions.[12]

[11] For more formal details on the semantics of these transductions, see (Bhaskar et al., 2020; Oakden et al., 2020).

[12] This is not guaranteed by the syntax that I define here. Instead, I only consider BMRS systems with this quality. See (Chandlee and Jardine, 2020) for BMRS systems for which a single input position evaluates to true for multiple

Function definitions can be any boolean term, including an if-then-else statement. Many of the definitions provided in this dissertation are of this type. Continuing with the Mandarin 3TS example, a definition for $L_o(x)$, given in (3.21), provides one such example.

(3.21) $L_o(x) \quad = \quad \texttt{if} \ \underline{L}L(x) \ \texttt{then} \ \bot \ \texttt{else} \ L(x)$

The definition of this function is an if-then-else statement. In (3.21), if the string position under evaluation is true for term $\underline{L}L(x)$—a local structure denoting a sequence of adjacent input L tones, recalling (3.19) from the previous subsection—then the function $L_o(x)$ evaluates to false $\bot$, meaning that the string position will not be labeled L in the output. If it is false for $\underline{L}L(x)$, it evaluates the term $L(x)$, that is, whether it is labeled as L.

Importantly, $\underline{L}L(x)$ can be understood as a *blocking structure* for output L, because it blocks it from surfacing on x. In general, any structure denoted by T_1 in a term $\texttt{if} \ T_1 \ \texttt{then} \ \bot \ \texttt{else} \ \ldots$ is a blocking structure for some output function name.

A definition for $R_o(x)$ is provided in (3.22).

(3.22) $R_o(x) \quad = \quad \texttt{if} \ \underline{L}L(x) \ \texttt{then} \ \top \ \texttt{else} \ R(x)$

It also contains a single if-then-else statement, and references the same local structure as its T_1. But instead of a false $\bot$ value, this structure causes $R_o(x)$ to evaluate to $\top$, meaning that the string position will be labeled R in the output. Again like $L_o(x)$, if it is false for $\underline{L}L(x)$, evaluation proceeds to the third term $R(x)$.

For output R, $\underline{L}L(x)$ is a *licensing structure*, because it allows the tone to surface on x. Similarly *any* structure denoted by T_1 in a term $\texttt{if} \ T_1 \ \texttt{then} \ \top \ \texttt{else} \ \ldots$ is a licensing structure for some output function name.

A full system of equations modeling Mandarin 3TS comprises (3.21) and (3.22), as well as the function names $H_o(x)$ and $F_o(x)$. Since H and F tones do not participate in 3TS, function definitions for output High and Falling tones are defined such that input H/R tones map directly to corresponding outputs. The complete definition is in (3.23).

(3.23)
$$\begin{aligned}
H_o(x) &= H(x) \\
L_o(x) &= \texttt{if} \ \underline{L}L(x) \ \texttt{then} \ \bot \ \texttt{else} \ L(x) \\
R_o(x) &= \texttt{if} \ \underline{L}L(x) \ \texttt{then} \ \top \ \texttt{else} \ R(x) \\
F_o(x) &= F(x)
\end{aligned}$$

---

output functions—modeling various segmental and suprasegmental processes.

This system defines a logical transduction over string models. It maps input string structures to output string structures such that for any sequence of two input L tones, the first tone surfaces as R. In other words, it models a Mandarin 3TS function. The table in (3.24) shows how this system evaluates an input string /LLL/ and maps it to [RRL] (i.e. consistently with the characterization in (3.5)).

(3.24)

Input:	L	L	L
	1	2	3
$\underline{L}L(x)$	$\top$	$\top$	$\bot$
$R(x)$	$\bot$	$\bot$	$\bot$
$L(x)$	$\top$	$\top$	$\top$
$H_o(x)$	$\bot$	$\bot$	$\bot$
$L_o(x)$	$\bot$	$\bot$	$\top$
$R_o(x)$	$\top$	$\top$	$\bot$
$F_o(x)$	$\bot$	$\bot$	$\bot$
Output	R	R	L

Each string position returns a true $\top$ value for exactly one function name from the system, meaning that output positions are labeled with one of the four lexical tones. Positions 1, and 2 evaluate to true for the structure $\underline{L}L(x)$—an input L tone with input L tone as its immediate successor. Output R is therefore licensed on these positions in the output, and output L is blocked. These two positions evaluate $\top$ for $R_o(x)$, and appear as R in the output string. Position 3 does not return a true value for this structure. As a result, it returns a false $\bot$ value for $R_o(x)$. This also means that, in its evaluation by $L_o(x)$, it passes on to the final term $L(x)$. Since position 3 is an input L, it returns a true value for $L_o(x)$ and is labeled L in the output. Note that all three positions return false for $H_o(x)$ and $F_o(x)$ because none are input-specified as High or Falling tones.

Thus the system of equations models Mandarin 3TS as a mapping from input structures to output structures. It does so using local structures in the input.

3.4.1.4 BMRS with recursive definitions

Recall that function names $\mathtt{f}(T)$ are boolean terms. Well-formed function definitions—themselves well-formed boolean terms—can therefore contain function names. In other words, output functions in a system of equations can be defined *recursively*. As an example, consider a new system of equations with the same four output function names as in the Mandarin 3TS example: $H_o(x)$, $L_o(x)$, $R_o(x)$, $F_o(x)$. Their definitions are different from those presented in (3.23), namely that they contain recursive definitions. The new definition of $L_o(x)$ is given in (3.25).

(3.25) $\quad L_o(x) \quad = \quad \texttt{if} \;\; \underline{L}L_o(x) \;\; \texttt{then} \;\; \perp \;\; \texttt{else} \;\; L(x)$

As in the system for Mandarin 3TS, the function $L_o(x)$ is defined using a single if-then-else statement indicating a blocking structure. Note that the blocking structure $\underline{L}L_o(x)$ is shorthand for

$$\texttt{if} \;\; L(x) \;\; \texttt{then} \;\; L_o(s(x)) \;\; \texttt{else} \;\; \perp$$

This term returns a true value if and only if the input string index under evaluation (interpreted as x) is specified as L and its immediate successor returns true for $L_o(x)$—that is, if it is *output* as L. Thus the definition of $L_o(x)$ is recursive: $L_o(x)$ is part of the definition of $L_o(x)$.

Recursive statements like the above also identify local structures. As part of a function definition, they determine which string positions are output with a particular label using the current string position and a local, bounded window of output structure.

Example (3.26) gives the equivalent full system, but using the recursively-defined function definition.

(3.26)
$$\begin{array}{lcl} H_o(x) & = & H(x) \\ L_o(x) & = & \texttt{if} \;\; \underline{L}L_o(x) \;\; \texttt{then} \;\; \perp \;\; \texttt{else} \;\; L(x) \\ R_o(x) & = & \texttt{if} \;\; \underline{L}L_o(x) \;\; \texttt{then} \;\; \top \;\; \texttt{else} \;\; R(x) \\ F_o(x) & = & F(x) \end{array}$$

Despite its similarity to the Mandarin 3TS system in (3.23), this system of equations describes a different function. Whereas the function described by (3.23) maps /LLL/ to [RRL], this function maps /LLL/ to [LRL], as in (3.27).

(3.27)

Input:	L	L	L
	1	2	3
$\underline{L}L_o(x)$	$\perp$	$\top$	$\perp$
$R(x)$	$\perp$	$\perp$	$\perp$
$L(x)$	$\top$	$\top$	$\top$
$H_o(x)$	$\perp$	$\perp$	$\perp$
$L_o(x)$	$\top$	$\perp$	$\top$
$R_o(x)$	$\perp$	$\top$	$\perp$
$F_o(x)$	$\perp$	$\perp$	$\perp$
Output	L	R	L

The crucial difference here lies in evaluation of position 1. Note that since position 2 satisfies the structure $\underline{L}L_o(x)$, it is output as R. But because of that, position 1 does *not* evaluate to true for the same structure—that is, the structure that licenses R. Instead, it is mapped to L by not conforming to the (same) blocking structure $\underline{L}L_o(x)$, and being input-specified as L.

3.4.1.5 Summary

The BMRS formalism developed here is a form of logical transduction over strings. Specifically, BMRS transductions are defined as systems of function names of sort *boolean*, one for each output symbol. The function names take single string indices as arguments—and hence are *monadic*. They can also be defined *recursively*.

Despite the introduction of recursion, BMRS transductions are restrictive in ways that are relevant to phonology. Bhaskar et al. (2020) show that, when recursion on $\mathtt{f}(T)$ is limited to terms $p(T)$ or $s(T)$, the resulting transductions correspond to exactly the left- and right-subsequential functions (respectively).

As shown above, BMRS transductions compute outputs using reference to local structures in either in input or output. The next sections connect classes of BMRS systems to subclasses of subsequential functions based on these properties, and which are relevant to the analyses of tone sandhi interactions explored in this dissertation.

3.4.2 ISL in BMRS

BMRS systems of equations describe subsequential functions. Subclasses of BMRS systems describe subclasses of subsequential functions. In particular, a class of *non-recursive* BMRS (NR-BMRS) describes ISL functions.

Definition 1 *NR-BMRS are the class of well-formed BMRS systems of equations where no term* T_i *in list* $(\mathtt{f}_1(x_1) = T_1, \ldots, \mathtt{f}_k(x_k) = T_k)$ *contains a term of the type* $\mathtt{f}(T)$*.*

Like the name suggests, non-recursive BMRS systems are those whose definitions do not contain any recursive function calls. Intuitively, it restricts computation to a bounded window in the input, as the remaining set of model-relational terms upon which licensing/blocking structures can be built is limited to $\sigma(T)(\sigma \in \Sigma)$—that is, terms that reference input structures. This is precisely the restriction that defines ISL functions. Oakden et al. (2020) prove that the set of NR-BMRS transductions corresponds to the ISL class of functions. The discussion below uses 3TS to illustrate.

Recall that an ISL function models Mandarin 3TS. That function is described by the NR-BMRS system in (3.23). Below in (3.28), a list of terms for each function definition verifies this.

(3.28)

Function definition	Terms
$H_o(x)$	$H(x)$
$L_o(x)$	$L(x)$
	$L(s(x))$
$R_o(x)$	$L(x)$
	$L(s(x))$
	$R(x)$
$F_o(x)$	$F(x)$

None of the definitions contain recursive function calls. They are limited to instantiations of terms $s(T)$ and $\sigma(T)$; the transduction thus computes outputs using bounded reference to input structure only.

3.4.3 OSL in BMRS

OSL functions can also be modeled using a class of *output-restricted* BMRS (OR-BMRS).

Definition 2 *OR-BMRS are the class of well-formed BMRS systems of equations where no term of the form $\sigma(T)$ takes index sort terms of the type $s(T)$ or $p(T)$.*

OR-BMRS systems are those for which terms $\sigma(T)$ can only take the index variable x. Index terms $s(T)$ and $p(T)$ can form recursive function terms $\mathtt{f}(T)$. Thus $\sigma(s(x))$ is not a well-formed term in a OR-BMRS system, but $\mathtt{f}(s(x))$ is. In intuitive terms, functions described by OR-BMRS transductions restrict reference to the current input string, and a bounded window in the output structure to either the right or left of the current input under evaluation. I conjecture that OR-BMRS describe the OSL functions, but forego a formal proof here.[13] Using the same intuition, LOSL and ROSL classes are describable by OR-BMRS systems for which recursive function terms $\mathtt{f}(T)$ take only terms $p(T)$ and $s(T)$, respectively. This is in line with Bhaskar et al. (2020)'s generalization regarding LSEQ and RSEQ classes.

[13]This proof will require restrictions to OR-BMRS beyond the limitation of using either p or s in recursive calls, as below. To illustrate, well-formed BMRS systems can describe long-distance agreement patterns, which have been shown to be non-OSL. Chandlee and Jardine (2019b) provide a schematic definition—for $\Sigma = \Gamma = \{a, b, c\}$ and b spreads to a with c as a blocker—using QFLFP:

$$b'(x) = [\mathtt{lfp}(b(y) \vee A(p(y)))](x) \wedge \neg c(x)$$

This definition models long-distance agreement by building a set of string positions following a b, plucking out the input cs (the blockers), and outputting the resulting set as b. A BMRS system describing this function is possible (with the use of helper functions), and is OR-BMRS-definable:

$$\begin{array}{rcl} b'(x) & = & \mathtt{if}\ bs(x)\ \mathtt{then} \\ & & \quad \mathtt{if}\ c(x)\ \mathtt{then}\ \bot\ \mathtt{else}\ \top \\ & & \mathtt{else}\ \bot \end{array}$$

where $bs(x)$ is defined as $\mathtt{if}\ \ b(x)\ \ \mathtt{then}\ \ \top\ \ \mathtt{else}\ \ bs(p(x))$ (and hence is OR-BMRS). Restricting recursion to rule out such cases is a crucial step in proving equivalence with OSL, and is left for future work.

Recall in §3.3 that an OSL function models the iterative Tianjin LL sandhi pattern. This function can be described as an OR-BMRS system. Its definition is given in (3.29).

(3.29)
$$
\begin{aligned}
H_o(x) &= H(x) \\
L_o(x) &= \texttt{if } \underline{L}L_o(x) \texttt{ then } \bot \texttt{ else } L(x) \\
R_o(x) &= \texttt{if } \underline{L}L_o(x) \texttt{ then } \top \texttt{ else } R(x) \\
F_o(x) &= F(x)
\end{aligned}
$$

In the system above, all terms of type $\sigma(T)$ (i.e. $H(T)$, $L(T)$, $R(T)$, $F(T)$) take only the variable x. The only instance of recursion—$L_o(T)$ in definitions $L_o(x)$ and $R_o(x)$—takes $s(x)$ as its term. It is OR-BMRS given that it conforms to Definition 2, and intuitively describes an ROSL function. Note that this system is nearly identical to (3.23), the exception being that the structure that licenses output R and blocks output L is defined recursively. Like the ISL-equivalent, $\underline{L}L_o$ is a shorthand for the term

$$\texttt{if } L(x) \texttt{ then } L_o(s(x)) \texttt{ else } \bot$$

and describes precisely the pink highlighted structure in (3.13a): an input L followed immediately by an output L. In (3.30), an evaluation table shows that the transduction maps /LLLL/ to [RLRL], consistently with attested Tianjin data (see (3.12)).

(3.30)

	L 1	L 2	L 3	L 4
$\underline{L}L_o(x)$	$\top$	$\bot$	$\top$	$\bot$
$R(x)$	$\bot$	$\bot$	$\bot$	$\bot$
$L(x)$	$\top$	$\top$	$\top$	$\top$
$H_o(x)$	$\bot$	$\bot$	$\bot$	$\bot$
$L_o(x)$	$\bot$	$\top$	$\bot$	$\top$
$R_o(x)$	$\top$	$\bot$	$\top$	$\bot$
$F_o(x)$	$\bot$	$\bot$	$\bot$	$\bot$
Output	R	L	R	L

This generalization extends to all mappings from strings in $\Sigma = \{H, L, R, F\}$ that undergo Tianjin LL sandhi, regardless of the size of the string. A bounded window in the output structure—the hallmark of an OSL function and formalized as a OS-BMRS transduction—is sufficient to compute correct outputs.

3.5 Discussion

So far, this chapter has framed the formal apparatus to be utilized in exploring phonological process interactions. This section highlights some of the benefits of this approach, and in particular the BMRS formalism. First, it provides a means to study both individual processes and their interactions as single functions (§4.1). Additionally, the BMRS formalism provides an intuitive means to explore process interaction compared to other computational formalisms, among its other advantages (§4.2). Subsequent chapters will clarify this point in more detail, but it is prefaced here.

3.5.1 Individual map vs combined map

One advantage of the computational characterization advocated here is that it provides a vantage point unavailable to either rule-based or optimization-based formalisms. That is, both individual processes and their interactions can be modeled as single functions. Consider as an illustration a hypothetical counterfeeding interaction between Mandarin 3TS and another sandhi processes given by the following rule: R → H / __ R, denoted 'RR rule'. In this toy example, the former counterfeeds the latter such that /LLLL/ maps to [RRRL] and not *[HHRL]. This is shown in the derivations in (3.31).

(3.31) a.

	/LLLL/
RR rule	—
3TS	RRRL
	[RRRL]

b.

	/LLLL/
3TS	RRRL
RR rule	HHRL
	*[HHRL]

A separate BMRS system defines each individual 'rule'. The 3TS system in (3.23) is repeated below as (3.32a); a NR-BMRS system for the RR rule is given in (3.32b).

(3.32) a.

$$\begin{aligned} H_o(x) &= H(x) \\ L_o(x) &= \texttt{if } \underline{L}L(x) \texttt{ then } \bot \texttt{ else } L(x) \\ R_o(x) &= \texttt{if } \underline{L}L(x) \texttt{ then } \top \texttt{ else } R(x) \\ F_o(x) &= F(x) \end{aligned}$$

b.

$$\begin{aligned} H_o(x) &= \texttt{if } \underline{R}R(x) \texttt{ then } \top \texttt{ else } H(x) \\ L_o(x) &= L(x) \\ R_o(x) &= \texttt{if } \underline{R}R(x) \texttt{ then } \bot \texttt{ else } R(x) \\ F_o(x) &= F(x) \end{aligned}$$

When each pattern is modeled as a distinct entity, it is possible to study their individual properties. For example, both sandhi patterns can be characterized as ISL functions, but compute outputs using distinct sets of licensing/blocking structures. In this way, the computational framework offers a view of the grammar not unlike an SPE-style analysis, where each transformation is defined as a separate rewrite rule.[14] This is unavailable to OT, which conceptualizes the grammar as a single mapping from input to output.

A 'combined map' description of the grammar—modeling the interaction as a single step—can also be defined using BMRS. In (3.33), a single BMRS system describes the counterfeeding order in (3.31a).

(3.33)
$$\begin{aligned}
H_o(x) &= \texttt{if } \underline{R}R(x) \texttt{ then } \top \texttt{ else } H(x)\\
L_o(x) &= \texttt{if } \underline{L}L(x) \texttt{ then } \bot \texttt{ else } L(x)\\
R_o(x) &= \texttt{if } \underline{L}L(x) \texttt{ then } \top \texttt{ else}\\
&\qquad \texttt{if } \underline{R}R(x) \texttt{ then } \bot \texttt{ else } R(x)\\
F_o(x) &= F(x)
\end{aligned}$$

As this system is NR-BMRS, it also serves as verification that the combined map (and crucially the counterfeeding interaction) is ISL. This cannot be deduced simply by looking at (3.32a) and (3.32b) in isolation. Thus it is possible to understand the properties of the *interaction* itself by defining it as a single function (like an OT grammar), properties unavailable to a formalism that decomposes interactions into discrete units (like SPE).

3.5.2 Benefits of BMRS

Chandlee and Jardine (2020), responding to earlier criticisms lodged by Pater (2018), outline a number of advantages to the BMRS formalism compared to other computational formalisms (especially automata-theoretic characterizations of phonological transformations). One is its ability to intensionally express phonologically-significant generalizations more directly than previous approaches. For example, blocking structures represent marked surface structures that trigger a phonological transformation, not unlike markedness constraints in OT. They capture the motivation for the process itself, crucially in a way that aligns with traditional phonological analysis, but that might be obscured if it were represented as an automaton. Additionally, BMRS provide a means to implement phonological substance, something that is unavailable to previous approaches as well.[15] It does so all while maintaining the important insights about subsequential complexity.

[14] One crucial difference is how BMRS models pairwise *ordering* of rules (or non-ordering). The next several chapters explore this in detail.

[15] This dissertation does not explore the issue of substance in detail. BMRS analyses using phonological features can be found in (Chandlee and Jardine, 2020). Thus, a benefit of this formalism is that it introduces one of Pater (2018)'s senses of 'substance' into the theory—phonetic substance. The other sense, i.e. restrictions on combining formal primitives, is not *a priori* guaranteed by representing phonological transformations as functions over string models, whether as finite-state machines or logical transduction.

This dissertation will add another advantage of the BMRS formalism to the list: an intuitive means to explore the relationship between individual maps and combined maps (in the terms described in the previous section), and therefore a formal method for investigating phonological process interactions. BMRS will be leveraged to determine how functions defined in (3.32a) and (3.32b) relate to an extensionally-equivalent combined map function in (3.33). In particular, this dissertation highlights BMRS' ability to clearly define sets of *operations* over systems of equations. This is the focus of chapters 5 and 6.

4 Motivating Representations

4.1 Introduction

The purpose of this chapter is to provide motivation for a particular string-based representation of tone. There exist a variety of options—both linear string models and non-linear representations—for representing tone and tonal processes. String models vary from phonetic descriptions of pitch height (as in Chao (1930)'s tone letters), to syllable-level symbols, to melodic representations (e.g. falling is represented as 'HL', rising as 'LH', etc.), to the more abstract characterizations in terms of Middle Chinese register (*yin* and *yang*) and category (*ping*, *shang*, *qu*, *ru*) distinctions (see Mei (1970); Pulleyblank (1978) for discussion but also Yip (1980); Pulleyblank (1986); Bao (1990) for a generative view). A falling tone can thus be represented in any of the following ways as strings:

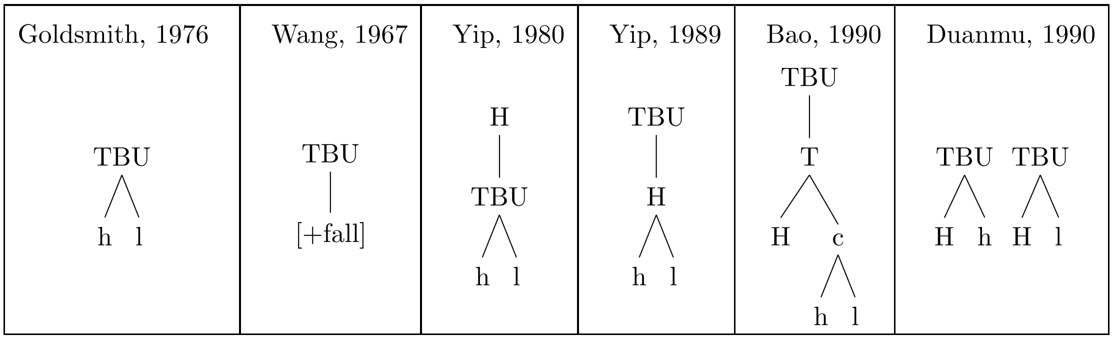
(4.1)

Chao Tone Letter	Syllable String	Melody
53	F	HL

Non-linear representations comprise classic autosegmental representations (ARs; Goldsmith, 1976) and any of the numerous feature-geometric models that have been proposed for tone. Differences among the latter type focus primarily on the set of features relevant to tone, as well as how to represent hierarchical relationships between TBU, register, tonal root, and contour (see Chen, 2000; Yip, 2002, for in depth discussion). Contours can be represented as single features (Wang, 1967), or as a constituency of level tones with an additional binary (upper/high vs. lower/low) register feature. Contour may be independent from register (Yip, 1980), dominated by it (Yip, 1989), or a sister to it (Bao, 1990). Other models reject constituency of contour (Duanmu, 1990, 1994), instead representing contours as sequences of level tone/register complexes. The same falling tone in (4.1) therefore corresponds to any one of the following non-linear representations, where 'TBU' denotes the tone-bearing unit (syllable, mora, or vowel depending on the theory), 'H' denotes upper register, 'T' a featureless root node, 'c' a featureless contour node, and 'h'/'l' terminal tonal nodes corresponding to 'high' and 'low':[1]

[1] This is only a small number of available non-linear theories. Others include (Clements, 1983; Hyman, 1986; Snider, 1990) for a tier-based approach and (Shih, 1986) for a prosodic approach.

(4.2)



Given this wealth of options, the choice of which representational scheme to adopt—linear or non-linear—is non-trivial. One representation may be preferable to another on conceptual grounds, for example if it can distinguish natural and unnatural tonal patterns in a way unavailable to other representations. Adopting one representation may also be preferable if it makes relatively clearer and more straightforward generalizations about tonal processes in a given language.

Another important question to address is whether a representational choice made for expository clarity has *theoretical* consequences, as different theories of representation may or may not make different predictions. Chen (2000, 56-7), for example, argues that the string representations (as in (4.1)) are equivalent, and can thus be used interchangeably. The non-linear representations presented above are claimed to constitute distinct theories as they make different empirical predictions (Chen, 2000; Yip, 2002). One's choice of a feature geometric model of tone ostensibly commits them to the particular predictions of that theory.

Recent computational work (Oakden, 2019b) has called this claim into question by showing that feature geometric models proposed by Yip (1989) and Bao (1990) are notationally equivalent in a mathematically rigorous way (see more discussion in §4). The computational framework can therefore meaningfully inform choices of representation by verifying (or refuting) that the selection between two representations carries theoretical ramifications. It also provides a means to test Chen (2000)'s claim of equivalence of string representations.

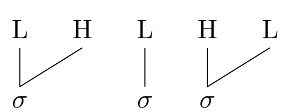
The focus of this chapter is on the choice between syllable string and melodic string representations (in (4.1)) as it pertains to tone sandhi processes. It shall provide conceptual and formal motivation for adopting syllabic string representations over melodies, and is organized as follows. §2 introduces these representations and sketches some general advantages of melodies from local and non-local tonal domains which are unavailable to syllabic string representations. Despite this, §3 provides a series of conceptual arguments in favor of syllabic representations particular to tone sandhi. The first is that melodies do not provide a more restrictive theory of tone sandhi than syllabic strings. The two thus make the same claims about natural and unnatural patterns in the sandhi typology. Two case studies are then presented. They show that, on the level of individual sandhi paradigms, melodic analyses make stipulative, sometimes paradoxical generalizations about this data. Assuming syllabic representations, by contrast, permits straightforward and unproblematic generalizations and thus allows greater focus on computation. §4 uses BMRS transductions to verify Chen (2000)'s claim that these string representations are notationally equivalent, and therefore that the choice between the two is theoretically non-binding. However, this is contingent on the crucial assumption that melodic representations are enriched with syllable boundaries. §5 interprets these results together and speculates on possible extensions.

4.2 Syllabic Strings and Melodic Strings

This section introduces two forms of tonal representation over strings—syllabic strings and melodic strings—as well as their properties, as they relate to tonal patterns. These properties are relevant to the choice of which string representation to adopt for tone sandhi pattern analysis. For simplicity, this chapter limits melodies to strings of H(igh) and L(ow) tonal segments and syllabic strings to H(igh), L(ow), R(ising), and F(alling) symbols. It does not explicitly discuss M(id) tone as a melodic/syllabic element, its possible permutations with H and L (low-rising/falling [LM/ML] and high-rising/falling [MH/HM]), or its relevance to motivating representations. In-depth examination of the issues presented in this chapter incorporating Mid tone is thus left for future work, but it is important to note that the characterization of restricted melodies and syllabic strings is sufficient to motivate one representation over the other.

In traditional autosegmental theory (Goldsmith, 1976), individual [H] and [L] tonal autosegments populate a separate *tier* from a tier of timing units (e.g. syllables or morae) and relate to them via an association relation (denoted by lines below). Assuming a timing tier of syllables, a trisyllabic sequence of a rising tone followed by a low tone followed by a falling tone is thus represented:

(4.3)



As a string, the melodic representation of the structure above is [LHLHL]. Syllabic tier string representation, on the other hand, can be thought of as a projection of the tonal information on the melodic tier, but condensed such that the number of symbols in the string is equal to the number of syllables. Contour tones, which necessarily comprise two autosegments on the melodic tier, are denoted with a single symbol in syllabic representation. Rising tones [LH] are denoted [R] and falling tones [HL] are denoted [F]. Keeping with the projection analogy, level high and low tones project directly to [H] and [L]. The same sequence in (4.3) can be represented over syllabic strings as [RLF]. Both [LHLHL] and [RLF] describe the same tonal structure. Thinking of the projection analogy in a different manner, the key difference between these two representations is that syllabic strings represent contour tones as single symbols while melodic strings *decompose* contours into sequences of level tones.

While both representations have been utilized in previous studies of tonal phonology, one benefit of melodic representations (not available to syllabic representations) is that they can capture tonal patterns with non-local dependencies. For example, Jardine (2018, 2020) posits a class of *melody-local* grammars which capture long distance tonal phenomena such as H-tone spreading patterns in Copperbelt Bemba (Bickmore and Kula, 2013). In this pattern, final H tones spread rightward to the end of the word (in phrase-final position), and any preceding (non-final) H tones only spread one additional TBU. This is summarized below where acute accents denote high tone:[2]

(4.4)

	Example	Gloss	Syllabic string
a.	tu-léé-pát-á	'we are hating'	LHHH
b.	bá-ká-fík-á	'they will arrive'	HHHH
c.	tú-lúb-ul-ulé	'we should explain'	HHLLH
d.	*tú-lúb-úl-ulé		*HHHLH
e.	*bá-ká-fik-a		*HHLL

This pattern cannot be captured using string representations with a one-to-one correspondence between tones and syllables. Crucially, spread of the first H tone is determined by the presence of another H tone later in the string, but the two may be separated by an arbitrary number of intervening L tones. This means that there is no way to *locally* determine whether a string is wellformed with respect to this pattern. Distinguishing grammatical and ungrammatical surface string representations would therefore require an arbitrarily large number of constraints (Jardine, 2018, 4):

(4.5)

	a.		b.
	HHLLH		*HHLLL
	HHLLLH		*HHLLLL
	HHLLLLH		*HHLLLLL
	...		...

[2] Jardine's analysis uses TBU strings for which the tone-bearing unit is the mora. The same generalization about locality applies regardless of whether syllables or morae are the TBU (though some details about the Copperbelt Bemba analysis differ). For consistency with the discussion in this chapter, syllabic strings are used instead.

This non-local dependency can be described, however, as a constraint on the melodic tier. Because association from autosegments to timing tier segments can be one-to-many, the simple constraint *HL#—that is, a prohibition on a 'H+L+word edge' melody—captures the (potentially) long-distance dependency in Copperbelt Bemba. It thus distinguishes grammatical (4.5a) and ungrammatical (4.5b) tonal structures, importantly those where an arbitrary number of Ls intervene between two H tones. This is shown below using traditional autosegmental representations.

(4.6) a. Satisfies *HL# b. Violates *HL#

Given that tone sandhi does not exhibit long-distance dependencies as in Copperbelt Bemba, this feature of melodies may not be directly relevant to the issue of string representations of sandhi processes. Does the difference between melodic and syllabic string representations also extend to local phenomena such that it bears on sandhi representation?

There is reason to believe that it does, as much previous work has adopted melodic representations (Chen, 2000; Hyman and VanBik, 2004, and many others) enriched with a '.' syllable boundary symbol to analyze tone sandhi. These representations are discussed in detail in the following sections, but an intuitive introduction about possible conceptual and computational differences between these representations is provided here. It focuses on the observation that melodies 'zoom in' on sub-syllabic tonal environments, a perspective which is unavailable to syllabic representations.

Consider the example of Tianjin tone sandhi (Li and Liu, 1985; Chen, 2000; Chandlee, 2019). In sequences of disyllables, three sandhi alternations are attested: a falling tone surfaces as a low tone before another rising tone (FF → LF or the 'FF rule'), a low tone surfaces as a rising tone before another low tone (LL → RL or the 'LL rule'), and a rising tone surfaces as a high tone before another rising tone (RR → HR or the 'RR rule').

(4.7)

	Input	Output	Gloss
a.	jing$^{\mathrm{F}}$zhong$^{\mathrm{F}}$	jing$^{\mathrm{L}}$zhong$^{\mathrm{F}}$	'net weight'
b.	fei$^{\mathrm{L}}$ji$^{\mathrm{L}}$	fei$^{\mathrm{R}}$ji$^{\mathrm{L}}$	'airplane'
c.	xi$^{\mathrm{R}}$lian$^{\mathrm{R}}$	xi$^{\mathrm{H}}$lian$^{\mathrm{R}}$	'wash one's face'

Both syllabic and melodic representations can describe the FF, LL, and RR rules. Note that the melodic tier string in the example below utilizes the enriched representation to include syllable boundaries (§3 and §4 offer conceptual and formal support for the enriched representation):

(4.8)

	Syllable Representation	Melodic Representation
FF rule	F → L / __ F	H → ∅ / __ L.HL
LL rule	L → R / __ L	∅ → R / L__ .L.
RR rule	R → H / __ R	L → ∅ / __ H.LH

Two differences are apparent here. The first deals with characterizing the nature of each process, and in particular the repair strategy employed to resolve an illicit tonal structure. Notice that over syllabic representations, Tianjin tone sandhi rules are uniformly substitution patterns; an [F] tone is substituted for an [L] tone, an [L] tone is substituted for an [R] tone, etc. Over melodies, however, the rules can be described as either deletion (FF and RR rules) or epenthesis (LL rule). This level of representation distinguishes the rules' repair strategies in a way that the syllabic representation does not, and therefore provides more information with which to generalize the paradigm. In other words, decomposing contour tones into sequences of level tones provides an additional sub-syllabic local environment at the syllable boundary (i.e. 'edge-effect' environments; see the next section for more discussion). To the extent that sandhi grammars are sensitive to this level of representation, melodies may be preferable over syllabic tier strings in analyzing tone sandhi processes.

The second difference is computational in the nature: the size and structure of the *window* (i.e. the value of k) needed to identify the triggering environment and thus compute the mapping from input to output. For syllabic representations, the value of k for all three rules is 2: they can be computed by referencing the current input symbol and one input symbol immediately to the right. No other information is needed. Representation over melodies entails a uniform increase of the k-value to 5 when syllable boundaries constitute a distinct symbol (consistent with analysis in Chandlee, 2019). This is shown for the RR rule in the example below.

(4.9) a.

Input:	...	R	R	...
Output:	...	**H**	R	...

b.

Input:	...	L	H	.	L	H	...
Output:	...		H	.	L	H	...

Using syllabic representations (4.9a), the sandhi output (the grey cell) can be determined solely by a window of size two in the input, whereas melodic representations (4.9b) require a window of five string positions to identify the same sequence: two adjacent rising tones.

There is also a difference in the nature of the conditioning environment. As illustrated in (4.8), the FF and RR rules contain triggers to the right of the target only, but the LL rule must be described with both rightward and leftward environments. Melodic representations thus divide the Tianjin rewrite rules into distinct categories along two dimensions. FF and RR rules are deletion rules with right environments, and the LL rule is an epenthesis rule with both right and left environments. These generalizations are absent from syllabic tier string representations, which collapse all three rules into a single category: substitution rules with a right environment. Whether these intuitions provide substantial motivation for adopting one representation over the other to analyze tone sandhi is the topic of the next two sections.

4.3 Conceptual Motivation

Syllables and melodies offer two distinct levels of representation over which tonal strings are defined. As the previous section illustrates, melodies capture tonal phenomenon at the non-local level (unavailable to syllabic representations) while also allowing 'decomposition' of contour tones at the local level (also unavailable to syllabic representations). A reasonable assumption following from the latter observation is that melodic string representations are a good candidate for analyzing local tone sandhi processes, and are perhaps preferable to syllabic representations. The purpose of this section, however, is to attack that assumption. Three conceptual arguments presented here indicate that melodic tier string representations are ill-suited to tone sandhi processes, and that their syllabic string counterparts are in fact preferable. The first argument weakens the conceptual appeal of melodies by illustrating that they do not permit a more restrictive theory of tone sandhi than is available to syllabic strings. Two case studies of attested sandhi paradigms are then presented. These case studies show that while adopting melodic representations present numerous complications, syllabic string representations allow for straightforward and focused computational analysis of the data. They are thus preferable to adopt in this dissertation.

4.3.1 Phonetically-arbitrary Sandhi

Recall that the apparent nature of the process changes when switching from a syllable-level string representation to a melodic representation: substitution over syllabic strings and deletion or epenthesis over melodies. Does this suggest that melodies capture the more basic mechanisms of sandhi that are obscured in a syllabic-representation? On the surface, it would seem that melodies 'zoom in' on tonal information by representing a local environment below the syllable level. This allows for a more fine-grained description of both triggering environments and repair strategies. A reasonable conjecture, then, might be that a melodic representation is preferable because it may separate natural and unnatural sandhi patterns, and therefore it offers a more restrictive theory of tone sandhi in language. The general consensus in phonological theory is that natural patterns are typologically frequent, phonetically grounded, and describable with simple rules or constraints (Halle, 1962; Chomsky and Halle, 1968b; Hyman, 1975). In a computational sense, this simplicity might correspond to the value of k (Danis et al., 2017), where natural patterns have a lower k value and unnatural patterns have a higher k value. Though the Tianjin sandhi rules defined over melodies in §2 did not differ in the value of k, a wider search of attested patterns would likely yield differing values, whereas syllabic representations would not. Might this mean something?

One way to test this conjecture is to determine whether melodies' fine-grained descriptions can explain systematic gaps in the typology. Examining the typology indicates that melodic representations do not distinguish natural and unnatural sandhi patterns. This is due in part to the fact that many tone sandhi patterns are phonetically *unnatural* and seemingly arbitrary (see, e.g., Zhang, 2014), and have therefore evaded a straightforward characterization in terms of one or more basic mechanisms.[3] To illustrate, consider the following table which exhausts the logically-possible extensions of disyllables with the same tone in so-called 'right-dominant' systems of four lexical tones (high, falling, low, rising), and includes whether these extensions are attested in Chinese dialects.

(4.10)

Input	Output	Attested?
H.H	H.H	yes (Tianjin; Li and Liu, 1985)
H.H	L.H	yes (Hefei; Kong, 2008)
H.H	HL.H	yes (Rizhao; Wu and Liu, 1981)
H.H	LH.H	yes (Boshan; Chen, 2000)
HL.HL	HL.HL	yes (Standard Mandarin; Li and Thompson, 1989)
HL.HL	L.HL	yes (Tianjin; Li and Liu, 1985)
HL.HL	H.HL	yes (Nanjing; Liu and Li, 1995)
HL.HL	LH.HL	yes (Pingyao; Hou, 1980)
L.L	L.L	yes (Laowu Zhoujia; Yan, 1981)
L.L	H.L	yes (Yantai; Qian, 1981)
L.L	LH.L	yes (Standard Mandarin; Li and Thompson, 1989)
L.L	HL.L	no
LH.LH	LH.LH	yes (Nanjing; Liu and Li, 1995)
LH.LH	H.LH	yes (Tianjin; Li and Liu, 1985)
LH.LH	HL.LH	yes (Pingyao; Hou, 1980)
LH.LH	L.LH	yes (Ningbo; Chan, 1995b)

The table suggests that using the types of environments distinguishable by melodic representations to split up the typology into natural and unnatural patterns is not a good fit for tone sandhi. This is because nearly every combination is attested in the typology.

Another possibility is that melodies' advantages are best exploited not on the meta-level of sandhi typology, but in providing natural explanations of sandhi paradigms within individual languages.

[3]This is not to say that all tone sandhi are arbitrary or do not lend themselves to analyses in terms of sequences of melodic units. Previous arguments for feature-geometric representations of tone (Yip, 1989) demonstrate how, in some sandhi processes, contour tones display edge effects similar to affricates, but also function as units (Clements, 1985; Sagey, 1986). In this chapter I argue that, while this may be true for isolated cases, adopting melodic representations for tone sandhi more broadly lacks conceptual and formal motivation.

The primary difference between melodies and syllable-level representations is that melodies can represent so-called 'edge effect' environments. Several examples below illustrate.

(4.11)

Syllable	Melody
FR	H**L.L**H
LL	**L.L**
FL	H**L.L**
LR	**L.L**H

Melodies demonstrate that FR is the same edge environment as LL, FL, and LR: adjacent L-tones abutting a syllable boundary. The same principle applies to RF and HH, HF, and RH. If these sequences pattern together in some sandhi paradigm, the melodic representation offers a potential explanation of a trigger that is not available to syllabic representation.

In phonological analyses of tone sandhi processes, however, the vantage point afforded by melodic representation does not return such results. Again, this is because tone sandhi is phonetically arbitrary. Attempts to impose a melodic framework on tone sandhi often produce stipulative analyses. For example, Chen (2000, 123)'s analysis of Tianjin tone sandhi posits two types of OCP constraints as the active markedness strictures; one operates over syllables (4.12a) and the other operates over *melodies* (4.12b-c).

(4.12) a. OCP- no adjacent identical tones (except HH)

b. OCP′- no *FL (=HL.L) sequences

c. OCP″- no adjacent partially-identical tones (*L.LH, *H.HL, *HL.LH, etc.)

The ranking OCP, OCP′ >> Faithfulness >> OCP″ explains why FF maps to LF and not HF. The string HF violates OCP″ [H.HL] in addition to a single faithfulness violation, and so LF wins out via an *Emergence of the Unmarked* effect (TETU Prince and Smolensky, 1993). This appears to be an argument in favor of melodic representations, but the necessarily specific definitions of these constraints suggest otherwise. OCP′ is posited as a melodic-tier constraint which prohibits the substring [HL.L]. It does not apply to all sequences containing [L.L] as in (4.11), and thus does not describe an edge effect or an OCP effect in the conventional sense of those terms. In other words, it stipulates an environment that cannot be derived from general principles. Additionally, the constraint does not apply to all sequences of [HL.L], either. One example is a disyllabic sequence of a falling tone followed by a rising tone [FR]. According to Chen's analysis, [FR] does not violate OCP′, however it is unclear why this should be the case. These tones do, in fact, exhibit the prohibited substring—[**HL.L**H]. Chen's OCP′ actually targets the illicit substring [HL.L.], which is to say it targets syllable-level strings [FL]. The only reason to posit it as a melodic constraint is to subsume the pattern under the OCP edge effects; as the above discussion has shown, however, it is neither.

The above example illustrates how representing tones as H/L melodies does not clarify restrictions on tone sandhi in any way that is unavailable to syllabic strings. The only real limitation on sandhi seems to be that patterns are local, a generalization captured by strictly-local functions regardless of representation. In other words, the most relevant restriction is not on representation, but rather on *computation*.

The following sections present two case studies: Hakha Lai tone sandhi and Nanjing tone sandhi. These underlie the phonetic arbitrariness of tone sandhi and illustrate that imposing melodic representations on sandhi data results in failed analyses, whereas equivalent syllable-based representations are straightforward. These thus provide conceptual support for adopting syllable-level representations of sandhi processes.

4.3.2 Case Study 1: Hakha Lai Sandhi

The first case study analyzes representational issues emerging in an OT account of Hakha Lai tone sandhi (Hyman and VanBik, 2004). On the disyllabic level, a conspiracy presents itself when representing the language's contours as sequences of level tones—that is, when the scope of analysis is on the melodic tier. This generalization collapses upon expansion to trisyllabic sequences, as the driving force of the conspiracy fails to hold for 3-syllable forms. By assuming syllable-level tonal representation, however, a straightforward analysis of 2- and 3-syllable sandhi forms is possible. The Hakha Lai data thus present a cautionary tale for sandhi analysis: appealing to the melodic tier may present a convincing explanation of a subset of the data, but it may also lead the analyst *away* from a unified account of the full paradigm.

4.3.2.1 An Ostensible Conspiracy

Given a tonal inventory of three tones R(ising), F(alling), and L(ow), three basic rules describe disyllabic sandhi patterns in Hakha Lai:

(4.13)

Rule	Description
FL rule	F → L / {F,L} ___
RL rule	R → L / ___ L
RR rule	R → F / R ___

This means that of the 9 possible disyllabic combinations of three lexical tones, four combinations trigger sandhi and five do not. This information is summarized in the table below, where the first syllable is indicated in rows and the second syllable in columns. Sandhi forms are given in bold.

(4.14)

	F	R	L
F	F**L**		
R		R**F**	**L**L
L	L**L**		

A natural question arises from these data: why only the four combinations above? From the vantage point of syllabic representations, these changes are arbitrary. However, decomposing these tones into their melodic representation presents a possible explanation which is unavailable to syllable-level representations. To see this, consider the table below which presents the sandhi paradigm in terms of melodic tier strings; F(alling) tones are represented as [HL] and R(ising) tones as [LH]. All melodies are presented, and alternating sandhi forms are again given in bold.

(4.15)

	HL	LH	L
HL	HL.**L**	HL.LH	HL.L
LH	LH.HL	LH.**HL**	**L**.L
L	L.**L**	L.LH	L.L

When contours are defined as sequences of level tones, the difference between alternating and non-alternating tones is remarkably clear: only inputs whose melodic-tier tones differ across a syllable boundary undergo sandhi. Additionally, sandhi produces forms with identical tonal units across the syllable boundary.

(4.16)

	Non-alternating		*Alternating*		*Sandhi Output*
a.	LH-HL	b.	HL-HL	c.	→ HL-L
	HL-LH		LH-LH		→ LH-HL
	L-LH		LH-L		→ L-L
	HL-L		L-HL		→ L-L
	L-L				

Given this striking consistency, it would appear that a melodic-tier conspiracy is at work. In their analysis, Hyman and VanBik (2004) describe the situation as: "The end-tone of one syllable should be the same as the beginning tone of the next (i.e., do not change tone levels between syllables!)." They formalize this stricture as a markedness constraint termed the No Jumping Principle, or No-Jump. It prohibits a structure whereby some melodic unit (αH) associating to a TBU is immediately followed by a melodic unit of different specification ($-\alpha$H) and associates to an adjacent TBU. In other words, tonal changes (from H to L or L to H) must take place *within* syllables, not across them.

Sequences of tones in violation of NoJump are precisely those which undergo sandhi, and those which do not violate the constraint do not. This consistency lends support to the conspiracy approach and the activeness of NoJump in the grammar of Hakha Lai. An OT analysis with undominated NoJump explains the constraint's interaction with basic tonal faithfulness and other markedness strictures to predict optimal forms in disyllabic sequences. The fundamental markedness-over-faithfulness ranking is NoJump >> Ident(T); that is, the pressure to avoid intrasyllabic tonal changes outweighs faithfulness to underlying tonal specification. Both constraints outrank a universal, "phonetically-grounded" markedness scale *R >> *F >> *L, which is condensed into a single constraint Markedness. Its effect in the grammar is to privilege the least marked repair, e.g. FF → FL and not the more marked *FR. To capture the paradigm's (mostly) left-dominant nature—sandhi affects the rightmost tone—Hyman and VanBik propose LeftProm. This is violated by RL → LL, which motivates the ranking Markedness >> LeftProm. The full hierarchy is thus NoJump >> Ident(T) >> Markedness >> LeftProm. An example tableau for the input /RR/ is given below to illustrate evaluation with this hierarchy.

(4.17)

/ka R-R/	NoJump	Ident (T)	Markedness	LeftProm
F-F	*!	*_*		*
F-R		*_		*!
F-L		*_*!		*
→ R-F		_*		
R-R	*!			
R-L	*!	_*		
L-F	*!	*_*		*
L-R		*_		*!
L-L		*_*!		*

The ordering over four constraints predicts observed outputs for all disyllabic forms. Hakha Lai therefore presents a strong case for appealing to the melodic tier; seemingly arbitrary tonal changes become principled changes by 'zooming in' on fine-grained tonal structure.

Compared to Chen (2000)'s melodic analysis of Tianjin introduced in the previous section, the Hakha Lai data and their explanation by Hyman and VanBik (2004) are much more compelling. One important reason is that—unlike Chen's OCP′—NoJump describes a true edge effect: the illicit substrings are [L.H] and [H.L]. In addition, the disyllabic data pattern around these edge environments consistently such that every disyllabic sequence with one of these substrings undergoes sandhi, and no others. Representing Hakha Lai tones as syllable-level tonal strings obscures this generalization, and as such provides no explanation for why four disyllables undergo sandhi to the exclusion of the other five.

4.3.2.2 A Non-Conspiracy

The conspiracy approach, despite its clear and compelling nature, only applies to a subset of the data. An OT analysis with NoJump as the primary markedness pressure fails when applied to tonal strings longer than two syllables. This is because simultaneous application of the RF rule in sequences of *three* or more input R tones yields outputs which themselves violate NoJump:

(4.18) a. RRR ↦ RFF

b. RRRR ↦ RFFF

For inputs /RRR/ and /RRRR/, surface forms [RFF] and [RFFF] garner multiple violations of NoJump— e.g. RFF = LH.H$\underline{\text{L.H}}$L. However, these outputs are preferred over the unattested *[RFR] and *[RFRF] which incur no violations. What is remarkable here is that the repair strategy for a NoJump conspiracy also violates NoJump! Thus, while the disyllabic data seem to indicate a straightforward conspiracy, expanding the scope to three or more syllables problematizes that analysis (in addition to the pattern being opaque).

Confronted with this, one way to salvage the conspiracy analysis is to propose that some other constraint in the grammar or general principle compels violation of NoJump for tri-syllabic sequences, but is inactive for disyllables. Hyman and VanBik (2004) do precisely this, and attribute the anomaly to directionality. The offending RFF and RFFF sequences are the result of right-to-left application of the RR rule: left-to-right application would predict the non-NoJump-violating—albeit unattested—*[RFR].

(4.19) a. right-to-left

R$\underline{\text{RR}}$
|
$\underline{\text{RR}}$F
|
RFF

b. left-to-right

$\underline{\text{RR}}$R
|
*RFR

The authors cite Chen (2000, 2004)'s general principles on sandhi interaction—the same are employed in Chen's accounts of Tianjin and Changting—in an attempt to redeem the conspiracy analysis:

(4.20) a. Temporal Sequence

b. Well-Formedness Conditions

c. Derivational Economy

d. Transparency

e. Structural Affinity

f. Simplicity (=Markedness)

However, they illustrate that no appeal to these principles can save the conspiracy approach with NoJump as a driving force. It is not simply the case that the conspiracy is active in the grammar but obscured in trisyllables as a result of other markedness or faithfulness pressures. To provide a unified analysis of the full dataset—crucially including trisyllabic forms—an alternative non-conspiracy account is needed. In other words, it is necessary to propose a completely different account.

One alternative analysis starts with the assumption that tone sandhi tends to be phonetically arbitrary and is thus unlikely to exhibit edge effect conspiracies. Melodic representations are therefore unnecessary as the extra information they provide may be irrelevant; syllable-level representations are posited instead. In the next section, I sketch such an analysis using BMRS and illustrate that it accounts for 2- and 3-syllable data equally well.

4.3.2.3 A Syllabic Analysis

Assuming syllable-level strings, the following BMRS system of equations captures the basic sandhi facts in (4.13):

(4.21)
$$\begin{aligned}
R_o(x) &= \texttt{if } R\underline{R}(x) \texttt{ then } \bot \texttt{ else} \\
&\quad\ \texttt{if } \underline{R}L(x) \texttt{ then } \bot \texttt{ else } R(x) \\
F_o(x) &= \texttt{if } R\underline{R}(x) \texttt{ then } \top \texttt{ else} \\
&\quad\ \texttt{if } F\underline{F}(x) \texttt{ then } \bot \texttt{ else} \\
&\quad\ \texttt{if } L\underline{F}(x) \texttt{ then } \bot \texttt{ else } F(x) \\
L_o(x) &= \texttt{if } F\underline{F}(x) \texttt{ then } \top \texttt{ else} \\
&\quad\ \texttt{if } L\underline{F}(x) \texttt{ then } \top \texttt{ else} \\
&\quad\ \texttt{if } \underline{R}L(x) \texttt{ then } \top \texttt{ else } L(x)
\end{aligned}$$

In intuitive terms, this system describes the following relationships between input and output lexical tones in Hakha Lai. First, an input [R] tone will map to an output [R] tone provided it is not immediately preceded by another input [R] tone or immediately followed by an input [L] tone. In the former case, it is output as [F] and in the latter case as [L]. An input [F] tone will map to an output [F] tone provided it is not preceded by an input [L] or [F] tone, in which case it surfaces as [L]. Finally, an input [L] will always map to [L] in the output. This system describes an ISL2 function.

This system of equations makes the correct predictions regarding disyllabic sandhi. More importantly, the system describes the mapping /RRRR/ $\mapsto$ [RFFF] which proved fatal for the conspiracy analysis using melodic representations. The table below gives the evaluation of the input /RRRR/

against the system of equations in (4.21):

(4.22)

Input:	R	R	R	R
	1	2	3	4
$R_o(x)$	$\top$	$\bot$	$\bot$	$\bot$
$F_o(x)$	$\bot$	$\top$	$\top$	$\top$
$L_o(x)$	$\bot$	$\bot$	$\bot$	$\bot$
Output:	R	F	F	F

Analyzing sandhi alternations in this way—that is, as potentially arbitrary substitutions of syllable-level lexical tones in local environments—has the advantage of accounting for both 2- and 3-syllable forms equally well. This is unavailable to the conspiracy account, which fails in spite of providing a compelling picture of melodic edge effects for disyllables.

4.3.2.4 Discussion

One of the advantages of the NoJump account and of conspiracies in general is that they explain more data with fewer constraints. A potential criticism of the analysis above, then, is that it merely lists out the changes without condensing the generalizations. It is possible to define a BMRS system of equations over melodies which recreates the NoJump conspiracy effect. A simplified system is defined below.[4]

(4.23)
$$
\begin{aligned}
L_o(x) &= \texttt{if } \underline{H}.L_o(x) \texttt{ then } \top \texttt{ else} \\
&\quad\ \texttt{if } \underline{L}.H_o(x) \texttt{ then } \bot \texttt{ else } L(x) \\
H_o(x) &= \texttt{if } \underline{H}.L_o(x) \texttt{ then } \bot \texttt{ else} \\
&\quad\ \texttt{if } \underline{L}.H_o(x) \texttt{ then } \bot \texttt{ else } H(x)
\end{aligned}
$$

With only two licensing/blocking structures, this system accounts for the same patterns for which the syllabic string system in (4.21) requires four. It presents a concise generalization about the data—a prohibition on output H-L and L-H melodic sequences across a syllable boundary—using fewer constraints. However, it will fail to account for the trisyllabic data for the same reason that doomed the OT analysis: the full dataset cannot be explained by a melodic conspiracy. The opaque trisyllabic mapping /LH.LH.LH/ $\mapsto$ [LH.HL.HL] (that is, /RRR/ $\mapsto$ [RFF] as above) illustrates this point.

(4.24)

Input:	L	H	.	L	H	.	L	H
	1	2	3	4	5	6	7	8
$L'(x)$	$\top$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$	$\top$	$\bot$
$H'(x)$	$\bot$	$\top$	$\bot$	$\top$	$\bot$	$\bot$	$\bot$	$\top$
Output:	L	H	.	H	L	.	L	H

[4] This system describes a 3-ROSL function.

Much like the corresponding OT analysis, a conspiracy account driven by a NoJump-like restriction will predict the unattested /LH.LH.LH/ $\mapsto$ *[LH.HL.LH] (/RRR/ $\mapsto$ *[RFR]), as it maximally obeys the conspiracy restriction represented by $\underline{H}.L'(x)$ and $\underline{L}.H'(x)$. This is inconsistent with the attested forms in Hakha Lai.

Interestingly, the final conclusion Hyman and VanBik (2004) reach is not that there might be no conspiracy in Hakha Lai after all. Rather, the output-drivenness of OT is the main culprit of the paradox. The authors present multiple alternative analyses, including a "direct-mapping" approach—similar to the system in (4.21)—which they select as the most simple and revealing as it contains none of the difficulties inherent in a strictly output-driven account. However, they retain the crucial assumption that the language's contours are indeed sequences of level tones—that is, that melodic representations offer the best analysis. This presents multiple options for how to describe the basic disyllabic sandhi rules, and the picture becomes rather complicated (Hyman and VanBik, 2004, 856):

> In §6 we then considered different interpretations of the tone rules. Up until this point we had assumed that both of the changes F $\rightarrow$ L and R $\rightarrow$ L involved deletion of a H tone feature (HL, LH $\rightarrow$ L), and that R $\rightarrow$ F involved tonal metathesis (LH $\rightarrow$ HL). In §6 we considered two possibilities involving intermediate stages; (i) input /LH/ first becomes H by delinking the L, and then HL by insertion of L on the other side of H (since Hakha Lai does not permit H level tone); (ii) all three tones sandhi are spreading rules that produce HLH and LHL contours, which then simplify by H-delinking to HL and L, respectively. We stipulate that a three-level approach can certainly be made to work, whether by rules, direct mapping, or output constraints (whose ranking could vary between levels 1,2 vs. levels 2,3).

Assuming syllable-level (that is, non-melodic and undecomposable) tonal representations for Hakha Lai and positing the potentially arbitrary nature of tone sandhi obviates the need to consider any of the complications described above. Sandhi processes are local substitutions, and the same analysis for disyllabic forms extends automatically to trisyllabic forms.

Two claims follow from this analysis. One is that the grammar is not sensitive to—or has no access to—the melodic tier for these patterns, and the other is that the 'conspiracy' Hyman and VanBik (2004) observe is merely a coincidence. Given a melodic tier with only H and L tone segments (a binary distinction) and a lexical inventory of three tones (R,H,L), such a configuration is inevitable. The authors present a convincing case for the conspiracy, but the appearance of this configuration of tones over a subset the data is not an automatic guarantee that a conspiracy is afoot. In fact, strong evidence *against* a conspiracy comes from the trisyllabic data and subsequent failure to account for these forms incorporating Chen's general principles. This case shows that decomposing contours into melodic elements can lead away from a unified account of the data. Syllable-level representations, especially for phonetically-arbitrary tone sandhi processes, provide no such danger, and are thus preferable in the analyses of tone sandhi conducted in this dissertation.

4.3.3 Case Study 2: Nanjing Sandhi

The second case study demonstrates that melodic representations are problematic in an account of Nanjing dialect tone sandhi. Not only do these patterns not exhibit conspiracy-like behavior as in Hakha Lai, but an exclusively melodic explanation of the data yields paradoxical generalizations about the sandhi paradigm. The crux of the issue is the presence of checked tones in this language's lexical tone inventory; these are defined both in terms of tonal realization and syllable structure. This tone is phonologically-distinct (as evinced through sandhi) from another tone with the same melody, so distinguishing the two representationally requires adding segmental information to the melodic tier, an unconventional and undesirable assumption. In anticipation of the formal discussion in the next section, it is also shown that syllable boundaries are an equally crucial component of the melodic tier to derive the Nanjing paradigm. Thus the 'melody' is a tier necessarily containing tones, segments, and syllable boundaries. By contrast, an alternative syllable-level analysis distinguishes tones simply by positing different symbols; it avoids these complications and offers a straightforward account of the data.

4.3.3.1 Sandhi Paradigm and Melodic Interpretation

The Nanjing dialect is a five-tone system; it contains L(ow), H(igh), R(ising), and F(alling) tones, as well as a high 'checked' tone (Liu and Li, 1995). These tones are defined by the presence of an occlusive coda, typically [p, t, k, ʔ]. The Nanjing checked tone contains a glottal stop coda, though recent work suggests it may be disappearing from the language (Gu, 2015; Oakden, 2017). Despite this, the checked tone is active in the tone sandhi paradigm of the language. Disyllabic sandhi is summarized below, where "Hʔ" indicates the checked tone, and alternating tones are in bold. Note that the sandhi patterns for 'H' and 'Hʔ ' are different. H triggers sandhi on a preceding L, and undergoes sandhi before Hʔ , while Hʔ triggers sandhi before R and H, and itself does not undergo sandhi alternation.

(4.25)

	F	R	L	H	Hʔ
F	**H**F				
R					**L**Hʔ
L	**R**F		**R**L	**R**H	
H					**F**Hʔ
Hʔ					

As with Hakha Lai, appealing to the melodic tier may shed light on why certain concatenations of lexical tones trigger sandhi and others do not. It has the potential to condense generalizations about illicit tonal surface representations across processes by identifying conspiracy-like behavior. Within the Nanjing sandhi paradigm, the melodic edge-effect approach at first glance appears fruitful, given that five of the six disyllabic sandhi processes can be described in terms of only two illicit submelodies.

(4.26)

Sandhi Rule	Melodic Representation	Illicit Submelody
FF → HF	HL.HL → H.HL	**L.H**
LF → RF	L.HL → LH.HL	
LH → RH	L.H → LH.H	
RH? → LH?	LH.H? → L.H?	**H.H**
HH? → FH?	H.H? → HL.H?	

The illicitness of the 'L.H' submelody may bear some resemblance to the NoJump-driven conspiracy in Hakha Lai tone sandhi. A 'jump' from a low tone to a high tone across a syllable boundary triggers sandhi in three distinct cases involving different lexical tones, though this sequence is acceptable within a syllable (i.e. rising tones). If such a conspiracy were active in Nanjing, the expectation is that 'H.L' would be similarly ill-formed. However no sandhi is attested for disyllables with this submelody, as the following table illustrates. Note that none of the disyllables shown in bold undergo sandhi.

(4.27)

	HL	LH	L	H	H?
HL					
LH		L**H.L**H	L**H.L**		
L					
H		**H.L**H	**H.L**		
H?					

The pattern /LL/ ↦ [RL] offers more evidence against the Hakha Lai-style conspiracy analysis; its surface form violates NoJump [L**H.L**] to resolve an input configuration [L.L] which does not violate this constraint. Thus the data do not enjoy a straightforward characterization in terms of a conspiracy. This does not automatically preclude a melodic analysis, however. It may still be possible to explain the paradigm using a tier of H and L segments. The next section addresses this more generally.

4.3.3.2 Issues with the Melodic Approach

A melodic analysis is in general ill-suited to the Nanjing sandhi data. It is unattractive for two reasons, one conceptual and the other formal. This section addresses each separately.

First, the melodic interpretation of this sandhi paradigm offers no explanatory or predictive power. Not only is there no conspiracy that can be gleaned from the melodic tier, framing the problem in terms of illicit submelodies—crucially as edge-effects—fails to yield any meaningful generalization of the data. To see why, recall that the three disyllabic sandhi rules in (4.26) containing the 'L.H' submelody all resolve to 'H.H'. Under a melodic tier analysis, then, an H segment can be said to trigger melody-local assimilation. The other two sandhi patterns contain the illicit submelody 'H.H' and resolve to 'L.H' suggesting local *dis*similation.

(4.28)

Input	Output	Process
H**L.H**L	**H.H**L	Local Assimilation
L.HL	L**H.H**L	Local Assimilation
L.H	L**H.H**	Local Assimilation
L**H.H**ʔ	**L.H**ʔ	Local Dissimilation
H.Hʔ	H**L.H**ʔ	Local Dissimilation

Thus the generalization from the melodic tier is that high tone segments drive assimilation *and* dissimilation. Given a size two inventory of tonal segments {H,L}, this exhausts the logically-possible options for transformations; in other words, this analysis makes no predictions because everything is possible (and in fact is attested).

From the perspective of SPE, attempting to define melodic rewrite rules for these patterns illustrates that such an approach is untenable. Because each rule creates the structural environment of the other rule, no ordering between rules will account for the subset of the data under consideration. In other words, represented over melodies, Nanjing tone sandhi presents an ordering *paradox*. This is presented schematically below, where some rule $\mathbb{R}1$ maps submelody 'L.H' to 'H.H' and a different rule $\mathbb{R}2$ performs the opposite operation.

(4.29)

UR	/...L.H.../	/...H.H.../
$\mathbb{R}1$	...H.H...	—
$\mathbb{R}2$	...L.H...	...L.H...
SR	*[...L.H...]	[...L.H...]

UR	/...L.H.../	/...H.H.../
$\mathbb{R}2$	—	...L.H...
$\mathbb{R}1$	...H.H...	...H.H...
SR	[...H.H...]	*[...H.H...]

The result is a melodic Duke-of-York-type effect over a subset of the data for each ordering. Positing melodic edge environments for Nanjing sandhi is clearly a nonstarter. The only alternative—similar to Chen's analysis of Tianjin—is to specify a larger window of tonal information such that the conditioning environments are no longer edge-effects in the sense of Hakha Lai. In the case of Nanjing, this requires including the glottal stop in the relevant window so as to distinguish the lexical high and high checked tones.

Doing so, however, underlies a central issue facing the melodic approach. This is because the level of representation necessary to capture Nanjing sandhi is not the conventional melodic tier, that is, sequences of H and L tonal segments. Expanding the prohibited 'H.H' submelody and its repaired melody to include the glottal stop—H.H? and L.H? —resolves the issue stated above; the triggers and repairs for L.H and H.H? no longer overlap. It furthermore prevents the repair from overapplying to well-formed tonal strings with the H.H submelody, namely disyllabic sequences of rising + falling L**H.H**L, rising + high L**H.H**, high + falling **H.H**L and high + high **H.H** tones.[5] In other words, it disambiguates the phonologically-distinct checked and non-checked high tones which, in terms of the melody, are simply 'H'. While presenting a more complete account of the data, this assumption is problematic because it requires segmental information, in this case a glottal stop coda, to be represented on the melodic tier. This confounds basic assumptions about the autosegmental nature of tone.

Relatedly, the presence of syllable boundaries '.' implicit in the analyses of Nanjing and Hakha Lai are both equally necessary (see the next section for more discussion). From a formal perspective, a melodic tier analysis *without* syllable boundaries is impossible for the Nanjing data. This is because syllable boundaries crucially distinguish licit and illicit substrings. For example, consider the two pairs of mappings in (4.30). The first pair comprises the disyllabic sequence [FF] and the quadrisyllabic sequence [HLHL]. Without syllable boundaries, their melodic representation is identical: [HLHL]. But recall that their outputs are distinct; [FF] undergoes sandhi to yield [HHL] and [HLHL] surfaces unchanged. The same is true of [R] and [LH] in the second pair.

(4.30)

Intended Structure	Melody	Output
Disyllabic FF $\mapsto$ HF	HLHL	HHL
Quadrisyllabic H+L+H+L	HLHL	HLHL
Monosyllabic rising tone R	LH	LH
Disyllabic LH $\mapsto$ RH	LH	LHH

Lack of explicit reference to syllable boundaries produces multiple distinct outputs for the same input. In formal terms, this mapping cannot be described by a function. The representation therefore must include the syllable boundary (and the glottal stop) for these mappings to be described as functions.

[5] This also indicates that the lexical tonal *category* of checked tone—represented in its entirety on the melodic tier—is what triggers sandhi, not just the presence of a high-toned melodic segment.

To summarize, both syllable boundaries and glottal stop coda—neither of which are 'melodic' elements in the conventional sense—are necessary to account for the Nanjing data. The following discussion sketches an alternative account using syllabic representations. The phonologically-relevant distinction between high and high checked tones is achieved simply by positing different symbols, resulting in a straightforward analysis of the disyllabic data.

4.3.3.3 A Syllabic Analysis

A syllable-level representational analysis posits different input symbols for each lexical tone in the Nanjing dialect. One possibility is $\Sigma = \{\text{H, L, F, R, C}\}$ where 'H, L, F, R' refer to tones in the conventional way and 'C' denotes checked tone syllables. High and high checked tones are thus distinguished in the input alphabet. A BMRS system of equations describing Nanjing sandhi contains five output boolean functions (one for each lexical tone). Separate definitions of $H_o(x)$ and $C_o(x)$ underlie the distinction between these lexical tones in the phonological grammar. A fragment of such a system illustrates.

(4.31)
$$\begin{aligned}
H_o(x) &= \texttt{if } \underline{H}C(x) \texttt{ then } \bot \texttt{ else} \\
&\quad\ \texttt{if } \underline{F}F(x) \texttt{ then } \top \texttt{ else } H(x) \\
F_o(x) &= \texttt{if } \underline{F}F(x) \texttt{ then } \bot \texttt{ else} \\
&\quad\ \texttt{if } \underline{H}C(x) \texttt{ then } \top \texttt{ else } F(x) \\
C_o(x) &= C(x)
\end{aligned}$$

The definitions above relate input tones to output tones in Nanjing using two input-local environments as blocking/licensing conditions. When [HC] is read in the input, only the output [FC] (the input high tone is realized as falling) will satisfy this system. Similarly, when [FF] is read, only the output [HF] (the input falling is realized as high) will satisfy this system. High tones therefore undergo sandhi when immediately followed by a high checked tone (and function as the sandhi tone for the input [FF]), but do not trigger sandhi. This is distinct from high checked tones, which trigger sandhi in the aforementioned environment but do not undergo it. The mappings /HC/ $\mapsto$ [FC] and /FF/ $\mapsto$ [HF] satisfy the transduction in (4.31) as below.

(4.32) a.

Input:	H	C
	1	2
$H_o(x)$	$\bot$	$\bot$
$F_o(x)$	$\top$	$\bot$
$C_o(x)$	$\bot$	$\top$
Output:	F	C

b.

Input:	F	F
	1	2
$H_o(x)$	$\top$	$\bot$
$F_o(x)$	$\bot$	$\top$
$C_o(x)$	$\bot$	$\bot$
Output:	H	F

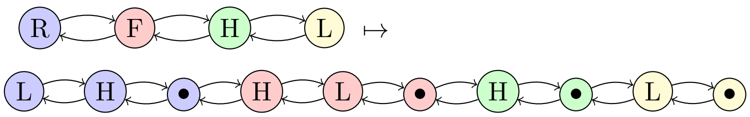
Each string position in the above examples evaluates to true for only one output boolean function (to the exclusion of the other two). The mappings accepted by this system are consistent with the two disyllabic sandhi patterns described. A full system modeling all six patterns is definable using input-local environments.

Syllabic representations thus garner additional conceptual support from the Nanjing sandhi case. Representing lexical tones as single symbols provides a degree of abstraction from their tonal (and potentially segmental) content. It captures the important intuition that tones can be phonologically distinct—that is, they pattern as distinct entities in phonological processes—despite similar phonetic realization. Adopting syllable-level representations where such distinctions are reflected in the input alphabet circumvents the complications apparent in a melodic approach and centers the focus on computation; the same processes which fail to be characterized even as functions when defined over melodies are simple, input 2-local substitutions over syllabic strings.

The following section examines the formal properties of these representations, and finds that, under certain crucial assumptions, they are indistinct. This means that adopting one representation over another does not impel any theoretical commitment. Interpreted in tandem with the results of this section, then, syllabic representations are not only more conceptually desirable than melodies (to analyze tone sandhi), the choice between the two is demonstrably agnostic in terms of a theory of representation.

4.4 Formal Motivation

Building on recent model-theoretic studies (Strother-Garcia and Heinz, 2015; Danis and Jardine, 2019; Oakden, 2019b, 2020) of notational equivalence in phonological representation, this section provides formal verification of Chen (2000, 56-7)'s claim that melodic and syllable string representations are equivalent. In particular, it adopts Oakden (2020)'s requirement of *bi-interpretability* (Friedman and Visser, 2014) to confirm that two models represent the same set of abstract properties and differ only superficially. In general terms, bi-interpretability imposes two conditions: that the representations are *intertranslatable*—one can be translated into the other and vice versa—and that these translations are *contrast-preserving*—any contrast expressible in one representation is not lost as a result of translation into a different representation. This section's primary purpose, then, is to demonstrate bi-interpretability for syllabic and melodic string representations. In §4.1, a transduction defines a translation from a model-theoretic syllable-representation string of tones (from the alphabet $\Sigma = \{\text{H, L, R, F}\}$) to an equivalent melody. §4.2 presents a transduction for the opposite translation, that is, from any model-theoretic melody string (from the alphabet $\Sigma = \{\text{H, L}, \bullet\}$) to an equivalent syllabic representation. The two representations are intertranslatable such that one can be derived from the other and vice versa. This is schematized below, where a model of a four-syllable string [RFHL] is translated to its equivalent melody [LH.HL.H.L]. Each circle node represents a string position, and arrows denote immediate successor and predecessor.

(4.33) 

§4.3 argues that these translations are maximally contrast-preserving. Formally, this means that applying the transductions in succession (that is, their composition) to the representations generates an equivalent structure to an identity mapping on either one. The details of the composition are revisited in Chapter 7 (after a BMRS composition operator is defined in Chapter 5). Thus this section illustrates a conjecture that the second condition of bi-interpretability is met, and will be explored in a later chapter.

Note that, in keeping with the generalizations about sandhi environments introduced in the previous section, melodic representations under consideration include a syllable boundary symbol '•'. §4.4 proves that deriving a syllable representation from a melody requires this structure, thus complimenting the previous section's conceptual arguments from a rigorous formal perspective (see §5 for more discussion). With this crucial assumption, then, it is possible to prove bi-interpretability of syllabic and melodic representations; any sequence of tones using the former representation can be translated into the latter representation and vice versa, and these translations maximally preserve all contrasts. Given this equivalence, the conclusion is that adopting one representation over the other is non-binding theoretically.

The next two subsections provide BMRS definitions for these transductions. This is thus the first application of the BMRS formalism to translating between representations.

4.4.1 Deriving Melodies from Syllables

Let $\mathcal{T}^{sm}$ be a BMRS transduction from syllable-level representations of the signature $\zeta_s = \{H, R, L, F; p, s\}$ to melodic representations of the signature $\zeta_m = \{H, L, \bullet; p, s\}$. It is defined over a copy set of size three. The intuition here is that F will map to H in the first copy and L in the second copy, R will map to L in the first copy and H in the second copy, H and L will map to themselves in the first copy, and the syllable boundary will map to the third copy. Below, superscripts denote output boolean function definition over some copy.

$$
(4.34)\quad
\begin{array}{lcl}
H^1(x) & = & \texttt{if}\ \ F(x)\ \ \texttt{then}\ \ \top\ \ \texttt{else}\ \ H(x) \\
H^2(x) & = & R(x) \\
H^3(x) & = & \bot \\
\hline
L^1(x) & = & \texttt{if}\ \ R(x)\ \ \texttt{then}\ \ \top\ \ \texttt{else}\ \ L(x) \\
L^2(x) & = & F(x) \\
L^3(x) & = & \bot \\
\hline
\bullet^1(x) & = & \bot \\
\bullet^2(x) & = & \bot \\
\bullet^3(x) & = & \top
\end{array}
$$

The table below shows an example derivation of $\mathcal{T}^{sm}$ applied to the syllabic string [FRHL]. Values returned by output boolean function evaluation for each input string position are shown, as well as the predicted output string (over each copy).

(4.35)

Input:	F	R	H	L
	1	2	3	4
$H^1(x)$	$\top$	$\bot$	$\top$	$\bot$
$H^2(x)$	$\bot$	$\top$	$\bot$	$\bot$
$H^3(x)$	$\bot$	$\bot$	$\bot$	$\bot$
$L^1(x)$	$\bot$	$\top$	$\bot$	$\top$
$L^2(x)$	$\top$	$\bot$	$\bot$	$\bot$
$L^3(x)$	$\bot$	$\bot$	$\bot$	$\bot$
$\bullet^1(x)$	$\bot$	$\bot$	$\bot$	$\bot$
$\bullet^2(x)$	$\bot$	$\bot$	$\bot$	$\bot$
$\bullet^3(x)$	$\top$	$\top$	$\top$	$\top$
Output:				
Copy 1 :	H	L	H	L
Copy 2 :	L	H		
Copy 3 :	$\bullet$	$\bullet$	$\bullet$	$\bullet$

In the mapping above, string position 1 evaluates to true for $H^1(x)$ and $L^2(x)$ by virtue of being specified as [F] in the input, and evaluates to true for $\bullet^3(x)$ vacuously. Positions 2-4 evaluate in a similar manner. Note that positions 3 and 4 do not satisfy any definitions over the second copy set and so contain no output in the transduction. As a result, $\mathcal{T}^{sm}$ maps [FRHL] to [HL.LH.H.L.] as is the desired effect. $\mathcal{T}^{sm}$ will in fact map any syllabic string representation to an equivalent string of tones in melodic representation, and its outputs are fully-specified with respect to syllable boundaries. The system representing the inverse transduction is presented below.

4.4.2 Deriving Syllables from Melodies

Let $\mathcal{T}^{ms}$ be a BMRS transduction from melodies of the signature $\zeta_m = \{H, L, \bullet; p, s\}$ to syllable-level string representations of the signature $\zeta_s = \{H, R, L, F; p, s\}$. It is defined over a single copy. The intuition here is that F maps from some input string 'HL.', R from 'LH.', H from '.H.', and L from '.L.'.

(4.36)
$$\begin{array}{lcl} R^1(x) & = & \texttt{if } L(x) \texttt{ then} \\ & & \quad \texttt{if } H(s(x)) \texttt{ then } \bullet(s(s(x))) \texttt{ else } \bot \\ & & \texttt{else } \bot \\ F^1(x) & = & \texttt{if } H(x) \texttt{ then} \\ & & \quad \texttt{if } L(s(x)) \texttt{ then } \bullet(s(s(x))) \texttt{ else } \bot \\ & & \texttt{else } \bot \\ H^1(x) & = & \texttt{if } H(x) \texttt{ then} \\ & & \quad \texttt{if } \bullet(p(x)) \texttt{ then } \bullet(s(x)) \texttt{ else } \bot \\ & & \texttt{else } \bot \\ L^1(x) & = & \texttt{if } L(x) \texttt{ then} \\ & & \quad \texttt{if } \bullet(p(x)) \texttt{ then } \bullet(s(x)) \texttt{ else } \bot \\ & & \texttt{else } \bot \end{array}$$

The series of embedded if-then-else statements in output boolean function definitions above ensures prescribed mappings between melodic substrings and syllable-level symbols. To illustrate, consider the melody [LH.HL.H.].

(4.37)

Input:	L	H	$\bullet$	H	L	$\bullet$	H	$\bullet$
	1	2	3	4	5	6	7	8
$R^1(x)$	$\top$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$
$F^1(x)$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$	$\bot$	$\bot$	$\bot$
$H^1(x)$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$
$L^1(x)$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$
$H(s(x))$	$\top$	$\bot$	$\top$	$\bot$	$\bot$	$\top$	$\bot$	$\bot$
$L(s(x))$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$	$\bot$	$\bot$	$\bot$
$\bullet(s(s(x)))$	$\top$	$\bot$	$\bot$	$\top$	$\bot$	$\bot$	$\bot$	$\bot$
$\bullet(s(x))$	$\bot$	$\top$	$\bot$	$\bot$	$\top$	$\bot$	$\top$	$\bot$
$\bullet(p(x))$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$	$\bot$	$\top$	$\bot$
Output:								
Copy 1 :	R	$-$	$-$	F	$-$	$-$	H	$-$

In the mapping above, string position 1 satisfies the definition of $R^1(x)$. It does so by evaluating to true for the hypothesis in the first if-then-else statement, as well as the hypothesis and consequent of the embedded statement; it is input-specified as L ($L(x)$), and is followed by an input-specified H ($H(s(x))$) and then a syllable boundary ($\bullet(s(s(x)))$). Note that both if-then-else statements contain $\perp$ as their 'else' condition; this means that the function returns a false $\perp$ value when any of these conditions fail to be met. String positions 4 and 7 evaluate to true for $F^1(x)$ and $H^1(x)$ by the same principles. The remaining positions—2, 3, 5, 6, and 8—do not satisfy any definition as they are the pieces of input structure that are 'deleted' in the transformation. The output of the transduction is [RFH], as is the desired effect.

4.4.3 Contrast Preservation between Representations

Syllabic and melodic string representations are intertranslatable because some transduction exists by which the former can derive the latter and vice versa. The second condition on bi-interpretability states that these transductions must preserve all contrasts. That they do so is intuitively apparent; for any tonal contrast available to syllabic strings as defined in this chapter—R(ising), F(alling), H(igh) and L(ow)—there is a distinct sequence of melodic elements that will map to it via the transduction. The same condition holds in the opposite direction. To validate this explicitly, it is necessary to show that both orders of *composition* of the transductions are isomorphic to the identity map on either model. In Chapter 5, a composition operator is defined for BMRS systems, and the formal demonstration of bi-interpretability is pursued further in Chapter 7. At this stage, however, I will illustrate the *conjecture* that, given any melodic string representation with syllable boundaries, applying $\mathcal{T}^{ms}$ to that string and then applying $\mathcal{T}^{sm}$ to its output will produce a structure that is equivalent to the original melody. Similarly, for any syllabic string representation, applying $\mathcal{T}^{sm}$ to that string and then applying $\mathcal{T}^{ms}$ to its output will produce a structure that is equivalent to the original syllabic-tier representation. The strings [RFHL] and [LH.HL.H.L.] provide illustration as a general case, with the understanding that this example can be extended to all possible permutations and strings of any size.

First, applying $\mathcal{T}^{sm}$ to [RFHL] yields [LH.HL.H.L.] over a copy set of size three. When $\mathcal{T}^{sm}$ is applied to that output, it returns [RFHL] over one copy.

(4.38)

Input:	R	F	H	L
	1	2	3	4
$H^1(x)$	$\bot$	$\top$	$\top$	$\bot$
$H^2(x)$	$\top$	$\bot$	$\bot$	$\bot$
$H^3(x)$	$\bot$	$\bot$	$\bot$	$\bot$
$L^1(x)$	$\top$	$\bot$	$\bot$	$\top$
$L^2(x)$	$\bot$	$\top$	$\bot$	$\bot$
$L^3(x)$	$\bot$	$\bot$	$\bot$	$\bot$
$\bullet^1(x)$	$\bot$	$\bot$	$\bot$	$\bot$
$\bullet^2(x)$	$\bot$	$\bot$	$\bot$	$\bot$
$\bullet^3(x)$	$\top$	$\top$	$\top$	$\top$
Output:				
Copy 1 :	L	H	H	L
Copy 2 :	H	L		
Copy 3 :	$\bullet$	$\bullet$	$\bullet$	$\bullet$

$\rightarrow$

Input:	L	H	$\bullet$	H	L	$\bullet$	H	$\bullet$	L	$\bullet$
	1	2	3	4	5	6	7	8	9	10
$R^1(x)$	$\top$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$
$F^1(x)$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$
$H^1(x)$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$	$\bot$	$\bot$
$L^1(x)$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$
Output:										
Copy 1 :	R	–	–	F	–	–	H	–	L	–

The output [RFHL] is equivalent to the result of applying the identity map to the original string [RFHL]. Thus the translations preserve all string representation contrasts.

In a similar manner, first applying $\mathcal{T}^{ms}$ to [LH.HL.H.L.] yields [RFHL] over a single copy. When $\mathcal{T}^{sm}$ is applied to the output, it returns [LH.HL.H.L.] over three copies.

(4.39)

Input:	L	H	$\bullet$	H	L	$\bullet$	H	$\bullet$	L	$\bullet$
	1	2	3	4	5	6	7	8	9	10
$R^1(x)$	$\top$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$
$F^1(x)$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$
$H^1(x)$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$	$\bot$	$\bot$
$L^1(x)$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$
Output:										
Copy 1 :	R	–	–	F	–	–	H	–	L	–

$\rightarrow$

Input:	R	F	H	L
	1	2	3	4
$H^1(x)$	$\bot$	$\top$	$\top$	$\bot$
$H^2(x)$	$\top$	$\bot$	$\bot$	$\bot$
$H^3(x)$	$\bot$	$\bot$	$\bot$	$\bot$
$L^1(x)$	$\top$	$\bot$	$\bot$	$\top$
$L^2(x)$	$\bot$	$\top$	$\bot$	$\bot$
$L^3(x)$	$\bot$	$\bot$	$\bot$	$\bot$
$\bullet^1(x)$	$\bot$	$\bot$	$\bot$	$\bot$
$\bullet^2(x)$	$\bot$	$\bot$	$\bot$	$\bot$
$\bullet^3(x)$	$\top$	$\top$	$\top$	$\top$
Output:				
Copy 1 :	L	H	H	L
Copy 2 :	H	L		
Copy 3 :	$\bullet$	$\bullet$	$\bullet$	$\bullet$

The output [LH.HL.H.L.] is equivalent to the result of applying the identity map to the original string [LH.HL.H.L.]. Thus the translations preserve all melodic representation contrasts.

Results from this section and the previous two form the conjecture that syllabic and melodic string representations are bi-interpretable and are therefore notationally-equivalent in a strict formal sense.

4.4.4 On the Necessity of Syllable Boundaries

Bi-interpretability likely holds between syllabic string and melodic string representations. Importantly, the latter assumes an enriched representation with syllable boundaries, resembling melodies posited in previous analyses of tone sandhi (as in §3.1-2), but unlike those used in Jardine (2018, 2020)'s work on non-local dependencies in tone. This section proves that equivalence between these representations hinges on the *necessary* assumption that melodies contain syllable boundaries. Recall the stringent conditions imposed by $\mathcal{T}^{ms}$ using embedded if-then-else statements. These guarantee a *univalent* mapping between melodies and syllable string representations. That is, for any possible string of melodic-tier elements, the transduction will map it to one distinct syllabic-tier string. Relaxing this condition results in non-univalence and bi-interpretability becomes untenable.

To see why, suppose that the transduction $\mathcal{T}^{sm}$ in (4.34) is redefined such that outputs do not contain syllable boundaries. The resulting function is equivalent to Jardine (2018, 2020, 43)'s $\texttt{cntr}(x)$ function—it 'expands contour-toned TBUs (but leaves H and L-toned TBUs as is).' The new transduction $\mathcal{T}'^{sm}$ is defined over two copies as in (4.40).

(4.40)
$$
\begin{aligned}
H^1(x) &= \texttt{if } F(x) \texttt{ then } \top \texttt{ else } H(x)\\
H^2(x) &= R(x)\\
L^1(x) &= \texttt{if } R(x) \texttt{ then } \top \texttt{ else } L(x)\\
L^2(x) &= F(x)
\end{aligned}
$$

Much like the original $\mathcal{T}^{sm}$, when this transduction reads [F], it outputs [H] to copy 1 and [L] to copy 2. Similarly, reading an [R] segment yields [L],[H] across both copies. [H] and [L] segments are left *as is*, as identity maps over the first copy. Application to the string [FRHL] produces [HLLHHL]; this is identical to the output form in (4.35), with the exception that there is no third copy with a single syllable boundary for each input.

(4.41)

Input:	F	R	H	L
	1	2	3	4
$H^1(x)$	$\top$	$\perp$	$\top$	$\perp$
$H^2(x)$	$\perp$	$\top$	$\perp$	$\perp$
$L^1(x)$	$\perp$	$\top$	$\perp$	$\top$
$L^2(x)$	$\top$	$\perp$	$\perp$	$\perp$
Output:				
Copy 1 :	H	L	H	L
Copy 2 :	L	H		

Similarly, the redefined $\mathcal{T}'^{ms}$, as the inverse of $\mathcal{T}'^{sm}$—i.e. $\texttt{cntr}^{-1}(x)$—outputs [HL] sequences as [F], [LH] sequences as [R], but leaves [H] and [L] alone. Updating the definition in (4.36) entails removing terms which reference the syllable boundary '•'. This is given below.

(4.42)
$$
\begin{aligned}
R^1(x) &= \texttt{if } \underline{L}H(x) \texttt{ then } \top \texttt{ else } \bot \\
F^1(x) &= \texttt{if } \underline{H}L(x) \texttt{ then } \top \texttt{ else } \bot \\
H^1(x) &= H(x) \\
L^1(x) &= L(x)
\end{aligned}
$$

Defined in this way, however, the resulting transduction represents a *non-univalent* function; for the input [HLLHHL], there are multiple outputs which satisfy the system of equations. Eight possible mappings are shown below.

(4.43)

Input:	H	L	L	H	H	L
	1	2	3	4	5	6
$R^1(x)$	$\bot$	$\bot$	$\top$	$\bot$	$\top$	$\bot$
$F^1(x)$	$\top$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$
$H^1(x)$	$\top$	$\bot$	$\bot$	$\top$	$\top$	$\bot$
$L^1(x)$	$\bot$	$\top$	$\top$	$\bot$	$\bot$	$\top$
Output 1 :	F	–	R	–	F	–
Output 2 :	F	–	R	–	H	L
Output 3 :	F	–	L	H	F	–
Output 4 :	H	L	R	–	F	–
Output 5 :	H	L	R	–	H	L
Output 6 :	F	–	L	H	H	L
Output 7 :	H	L	L	H	F	–
Output 8 :	H	L	L	H	H	L

Without syllable boundaries, then, melodies and syllable representations cannot be bi-interpretable.

An informal proof by contradiction shows why this is the case. Assume that models $\mathcal{S}$ of syllabic strings and models $\mathcal{M}$ of melodies (without syllable boundaries) are bi-interpretable. By the definition of bi-interpretability, there exists some *interpretation* of $\mathcal{S}$ in $\mathcal{M}$, a surjective map from a $\mathcal{M}$-signature to a $\mathcal{S}$-signature (Hodges, 1997). Let $\mathcal{T}'^{ms}$ define this interpretation. However, as shown in (4.43), $\mathcal{T}'^{ms}$ defines a non-univalent function. Since it is not univalent, it cannot be surjective. Therefore, syllabic and melodic string representations (without syllable boundaries) cannot be bi-interpretable.

It is possible in principle to coerce $\mathcal{T}'^{ms}$ into univalence, but this does not lead to bi-interpretability. As an example, consider the re-updated transduction $\mathcal{T}''^{ms}$ defined below.

$$
\begin{array}{llcl}
(4.44) & R^1(x) & = & \texttt{if}\ \underline{L}H(x)\ \texttt{then}\ \top\ \texttt{else}\ \bot \\
 & F^1(x) & = & \texttt{if}\ \underline{H}L(x)\ \texttt{then}\ \top\ \texttt{else}\ \bot \\
 & H^1(x) & = & \texttt{if}\ \underline{H}H(x)\ \texttt{then}\ \top\ \texttt{else}\ \bot \\
 & L^1(x) & = & \texttt{if}\ \underline{L}L(x)\ \texttt{then}\ \top\ \texttt{else}\ \bot
\end{array}
$$

Despite making intuitively incorrect predictions, it *does* describe a surjective map onto all logically-possible combinations of syllabic-tier symbols. Applied to the melody [HLLHHL], it yields a single syllabic representation [FLRHF]:

(4.45)

Input:	H	L	L	H	H	L
	1	2	3	4	5	6
$R^1(x)$	$\bot$	$\bot$	$\top$	$\bot$	$\bot$	$\bot$
$F^1(x)$	$\top$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$
$H^1(x)$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$	$\bot$
$L^1(x)$	$\bot$	$\top$	$\bot$	$\bot$	$\bot$	$\bot$
Output :	F	L	R	H	F	–

Combined with $\mathcal{T}'^{sm}$, this transduction satisfies the first condition of bi-interpretability: intertranslatability. Ultimately, however, it fails, because it does not satisfy the second condition of contrast preservation.

Another informal proof by contradiction shows why this is the case. Assume that models $\mathcal{S}$ of syllabic strings and models $\mathcal{M}$ of melodies (without syllable boundaries) are bi-interpretable. By the definition of bi-interpretability, given two surjective maps G and F and any $\mathcal{M}$-structure $\mathtt{K}$, $G(F(\mathtt{K}))$ produces a structure which is homomorphic to the identity map on $\mathtt{K}$. Let G be denoted by $\mathcal{T}''^{ms}$, and let F be denoted by $\mathcal{T}'^{sm}$. Also, let $\mathtt{K}$ be the $\mathcal{M}$-structure [HLLHHL]. As in (4.45), $\mathcal{T}''^{ms}$([HLLHHL]) yields [FLRHF]. By the definition of $\mathcal{T}'^{sm}$ in (4.40), $\mathcal{T}'^{sm}$([FLRHF]) yields [HLLLHHHL]. Since [HLLLHHHL] $\neq$ [HLLHHL], it cannot be the case that $G(F(\mathtt{K}))$ produces a structure which is homomorphic to the identity map on any $\mathtt{K}$. Therefore, syllabic and melodic string representations (without syllable boundaries) cannot be bi-interpretable. The following section relates these results to the conceptual arguments presented in §3.

4.5 Discussion

Adopting syllabic-tier string representations over melodies for tone sandhi analysis is motivated on both conceptual and formal grounds. Despite providing a fine-grained description of edge environments, melodic tier representations do not provide a more restrictive theory of tone sandhi than is available to syllabic strings. This is due primarily to the phonetic arbitrariness of tone sandhi.

Attempts to impose a melodic framework on tone sandhi data often result in stipulative or paradoxical analyses. They may even guide the analyst away from a unified account of the full paradigm by identifying non-existent conspiracies over a subset of the data. The case studies presented here thus serve as a cautionary tale, and underlie the importance of considering especially tone sandhi data that contain longer sequences of tones. Often, the full paradigm only emerges in trisyllabic sequences (or longer), as was seen with Hakha Lai.

As the case studies illustrate, syllabic string representations circumvent these issues and provide straightforward accounts of sandhi patterns. They permit a degree of abstraction in positing input alphabets (recall Nanjing checked tone) which is well-suited to the arbitrary nature of attested sandhi. Using these strings, it is possible to focus directly on the computational properties of sandhi processes which are of interest to this dissertation.

Fortunately, the choice of one representation over the other does not impel any theoretical commitment. The reason for this is that melodic and syllabic representations of tone are notational variants of one another. It is possible to prove this in formal terms using a restrictive definition of equivalence from model theory and implementing the definition within the BMRS framework. As §4 illustrates, syllable and melody strings can be freely translated into and from one another, and translation does not entail any loss of contrast.

This is contingent on the crucial assumption that melodic representations contain syllable boundaries; without them, melodic tier and syllabic tier representations are no longer notationally equivalent. Interestingly, this formal result is directly related to the conceptual results discussed in §3. Recall that in the proposed melodic analysis of Nanjing sandhi, syllable boundaries are required to differentiate licit and illicit substrings. For example, a sequence of two falling tones [HL.HL.] (which undergoes sandhi to produce [H.HL.]) can only be distinguished from a quadrisyllabic sequence [H.L.H.L.] (which does not undergo sandhi) with the use of syllable boundaries. When syllable boundaries are deleted, the two melodies are identical: [HLHL]. This means that no function can describe the sandhi data, because applying the hypothetical sandhi function to [HLHL] would produce *two* outputs (as in (4.30); partially repeated in (4.46)):

(4.46)

Intended Structure	Melody	Output
Disyllabic FF $\mapsto$ HF	HLHL	HHL
Quadrisyllabic H+L+H+L	HLHL	HLHL

The core of this paradox is precisely the complication facing the transduction $\mathcal{T}'^{ms}$ defined in (4.42). Applied to the melodic string [HLHL], this transduction accepts both the 2-syllable [FF] and 4-syllable [HLHL] as well-formed outputs.

(4.47)

Input:	H	L	H	L
	1	2	3	4
$R^1(x)$	$\perp$	$\perp$	$\perp$	$\perp$
$F^1(x)$	$\top$	$\perp$	$\top$	$\perp$
$H^1(x)$	$\top$	$\perp$	$\top$	$\perp$
$L^1(x)$	$\perp$	$\top$	$\perp$	$\top$
Output 1 :	F	–	F	–
Output 2 :	H	L	H	L

In other words, string representations over melodies are ambiguous with respect to the syllabic tier unless syllable boundaries are explicitly stated. This underlies the important observation that the locus of the tone sandhi alternations discussed in this chapter is the syllable.

Melodic representations without syllable boundaries are ill-suited to tone sandhi pattern analysis, but they are the key to analyses of other tonal phenomena. In particular, boundary-free melodic-tier strings successfully capture long-distance tonal dependencies by recreating one-to-many associations between tonal segments and timing tier segments (Jardine, 2018). As §2 demonstrates, syllabic-tier strings fail to capture this level of representation. This raises an important conceptual question about the autosegmental nature of tone sandhi. That syllables are a required component of the representation seems to indicate that tone sandhi is fundamentally connected—or tethered—to the syllable itself. However, perhaps the most basic and crucial insight of autosegmental tonology is that tones are "semi-autonomous from the tone-bearing units on which they are realized" (Goldsmith, 1976; Hyman, 2011, 2014). Is it possible, then, that tone sandhi is somehow *less* autosegmental than other tonal phenomena such as Copperbelt Bemba H-tone spreading? And if so, is this a dimension along which tonal systems can be distinguished typologically? This question is not within the scope of the current dissertation and is therefore left for future work, but the discussion in this chapter can serve as a foundation for later inquiry.

Given the conceptual and formal results presented here, syllabic-tier string representations are adopted for computational analysis of tone sandhi patterns in the dissertation. In the next chapter (and in Chapter 6), the theory is formally introduced and analyses of tone sandhi interactions—formalized using syllabic-tier representations—are presented.

5 Interactions as Composition with $\otimes$

5.1 Introduction

BMRS systems of equations describe subsequential functions. They can therefore be used to formalize a broad range of phonological transformations. A single BMRS system of equations can model an individual process, or it can model a 'combined map' which represents interactions of two or more individual processes. This includes transparent interactions such as feeding and bleeding, as well as interactions traditionally labeled as 'opaque', e.g. counterfeeding and counterbleeding. To briefly illustrate, consider two context-free rules (a $\rightarrow$ b and b $\rightarrow$ c) which transform inputs in isolation as in (5.1).

(5.1)	a. a $\rightarrow$ b	Input:	/aba/	b. b $\rightarrow$ c	Input:	/aba/
		Output:	[bbb]		Output:	[aca]

When applied to an input string /aba/, the context-free rules in (5.1a) and (5.1b) produce different output strings. An interaction occurs between the rules when placed in a particular order.

(5.2)	a $<$ b	Input:	/aba/
		a:	[bbb]
		b:	[ccc]
		Output:	[ccc]

In (5.2), the rules are ordered such that the output of (5.1a) becomes the input to (5.1b), yielding a distinct output. This represents a feeding relationship; the first rule provides additional inputs to the second rule that were not present in the input string.

Each rule in (5.1) also has a functional characterization. Assuming sets of strings over an alphabet $\Sigma = \{a, b, c\}$, the rule in (5.1a) describes a function which maps every a to b, every b to b, and every c to c. Similarly, (5.1b) as a function maps a to a, b to c, and c to c. Example (5.1c) also describes a single function; it maps a, b, and c to c. The three functions are properly subsequential (more specifically, 1-ISL), which means that they may be characterized as BMRS systems of equations. Let systems $\mathcal{T}_1$, $\mathcal{T}_2$, $\mathcal{T}_3$ denote these functions, respectively. Their definitions are given in (5.3a-c).

(5.3) a.
$$\begin{aligned} a_1(x) &= \bot \\ b_1(x) &= \texttt{if}\ a(x)\ \texttt{then}\ \top\ \texttt{else}\ b(x) \\ c_1(x) &= c(x) \end{aligned}$$

b.
$$\begin{aligned} a_2(x) &= a(x) \\ b_2(x) &= \bot \\ c_2(x) &= \texttt{if}\ b(x)\ \texttt{then}\ \top\ \texttt{else}\ c(x) \end{aligned}$$

c.
$$\begin{aligned} a_3(x) &= \bot \\ b_3(x) &= \bot \\ c_3(x) &= \top \end{aligned}$$

Both individual rules and 'combined maps' of interactions can be defined as individual BMRS systems of equations. But suppose that we wanted to *relate* (5.3a-b) to (5.3c) in a way that reflects the ordering relationship apparent in (5.1c). Following the observation that the composition of string relations models the effect of having one rule operating on the output of another rule (Johnson, 1972; Kaplan and Kay, 1994), we could compose the functions described by systems $\mathcal{T}_1$ and $\mathcal{T}_2$, keeping in mind that (5.1a) applies before (5.1b). The resulting composite function would be extensionally-equivalent to $\mathcal{T}_3$ (5.3c)—that is, it maps inputs to the same outputs as does $\mathcal{T}_3$—with the added benefit of clarifying the modifications to input strings contributed by each function. In other words, it captures the notion that "individual rules of a grammar are meant to capture independent phonological generalizations", and that interactions are the effect of overlap in targets/triggers of independent generalizations (Kaplan and Kay, 1994, 364). Additionally, the computational properties of the composite function can be understood along with that of its component parts.

The purpose of this chapter is to define a composition operation over BMRS systems that relates individual processes to 'combined map' interactions in the manner described above. It applies this framework to attested tone sandhi interactions in three Chinese dialects (Tianjin, Changting, and Nanjing), specifically those which may be derived via pairwise rule ordering. This includes transparent feeding interactions as well as opaque counterbleeding interactions. By defining individual BMRS systems modeling sandhi 'rules' and then composing them, the approach advocated here clarifies the distinct contributions made by each part. This is unavailable to analyses that only consider a combined map function. Furthermore, this chapter expands on earlier insights about so-called 'directionality' effects in sandhi interactions (Chandlee, 2019; Oakden and Chandlee, 2020) by applying them to new data. These insights are rooted in Chandlee (2014)'s finding that processes formalized as simultaneous or iterative rule application can be described using ISL or OSL functions, respectively (recall §3.2 and §3.3 of Chapter 3). Analyses in terms strictly-local (SL) function compositions clarify the issue of directionality in a way that is not apparent in rule-based or optimization-based theories.

This chapter is organized as follows. §2 offers a formal definition of the composition operator (denoted $\otimes$). This section outlines the operator's basic properties, relates it to pairwise rule ordering, and develops a broad typology of SL function compositions. Building on the aforementioned connections between ISL/OSL functions and rule application (simultaneous vs iterative), the typology makes explicit predictions about so-called 'directionality' effects from the sandhi literature. §3, §4 and §5 present case studies of tone sandhi interactions in Tianjin, Changting, and Nanjing dialects, respectively. Across three case studies, each composition type in the typology manifests in a specific interaction, and the predictions regarding directionality contribute to a straightforward account of interaction paradigms. Attention is also paid to the individual contribution of each 'rule' in the interaction, and how the composition of individual systems recapitulates rule-ordering generalizations. A discussion section (§6) summarizes the results of the chapter. It offers further support for composition/ordering equivalence by showing that reversing the order of composition produces the same effect as reversing rule order. Benefits of the SL functional approach are discussed, in particular as they relate to previous attempts to explain directionality effects in SPE or OT frameworks. This chapter ends with an ordering paradox from the sandhi data, demonstrating the failure of rule ordering (and thus composition) to account for it, thus setting the stage for the next chapter.

5.2 Definition and Formal Properties

This section defines a syntactic operator $\otimes$, the analog of function composition over BMRS transductions. Oakden et al. (2020) provide a proof that BMRS composition is identical to function composition, but the discussion in this chapter focuses on the relationship between the $\otimes$ operator and pairwise rule ordering. Primarily, it will show that composition recreates the effect of ordering in a serial rule-based framework. That is, given two rules A and B and the ordering $A < B$, the output of A serves as the input of B. Similarly, given two BMRS systems of equations denoted A and B, and which model the separate processes formalized as rules, the composite function $B \otimes A$ accepts the same input-to-output mappings as the crucial order $A < B$, and is thus an analog of pairwise rule ordering. This syntactic operator shall be used to formalize attested tone sandhi interactions in three case studies.

Additionally, this section sketches a typology of composition orders based on strictly-local functions and examines their features. A broad typology includes composition of two ISL functions (ISL $\otimes$ ISL), two OSL functions (OSL $\otimes$ OSL), and OSL function with an ISL function (OSL $\otimes$ ISL), and an ISL function with an OSL function (ISL $\otimes$ OSL). All four composition orders are attested, as the subsequent sections demonstrate.

5.2.1 A syntactic operator $\otimes$

What follows is a definition of the syntactic operator $\otimes$ over BMRS systems of equations. This dissertation focuses on compositions of *length-preserving* transductions, that is, those defined over a single copy set. See (Oakden et al., 2020) for definition of this operator in the general case, and a proof that it produces composition.

Definition 3 *For a BMRS transduction $\mathcal{T}_1$ from strings in Σ^* to strings in Δ^* and a transduction $\mathcal{T}_2$ from strings in Δ^* to strings in Γ^*, let $\mathcal{T}_2 \otimes \mathcal{T}_1$ be a syntactic operation on BMRS transductions as follows. Let*

$$\mathcal{T}_1 = \{\ \mathtt{f}_1(x_1) = T_1,\ \ \ldots\ ,\ \mathtt{f}_n(x_n) = T_n\}$$

be a system of equations over a signature of strings in Σ^*: a set of recursively-defined, boolean-type function symbols F *where each $\mathtt{f}_i$ corresponds to some δ_i in the output alphabet Δ, where $n = |\Delta|$. Similarly, let*

$$\mathcal{T}_2 = \{\ \mathtt{g}_1(x_1) = T_1,\ \ \ldots\ ,\ \mathtt{g}_m(x_m) = T_m\}$$

be a system of equations over a signature of strings in Δ^*: a set of recursively-defined, boolean-type function symbols G *where each $\mathtt{g}_j$ corresponds to some γ_j in the output alphabet Γ, where $m = |\Gamma|$.*

Let $\mathcal{T}_2'$ then be identical to $\mathcal{T}_2$ with the exception that any occurrence of boolean term $\delta_i(x)$ on the right-hand side of an equation in $\mathcal{T}_2$ is replaced with $\mathtt{f}_i(x)$ (corresponding to $\delta_i(x)$) from $\mathcal{T}_1$, for each function symbol $\mathtt{f}_i(x)$ in F*. Then, let $\mathcal{T}_2 \otimes \mathcal{T}_1 := \mathcal{T}_2' \cup \mathcal{T}_1$.*

The composition of two transductions $\mathcal{T}_2 \otimes \mathcal{T}_1$ is the union of $\mathcal{T}_1$ with a *modified* $\mathcal{T}_2$ (denoted $\mathcal{T}_2'$), the result of replacing every *non-recursively defined* function name in $\mathcal{T}_2$'s function definitions with the corresponding function definition from $\mathcal{T}_1$. Recursively-defined function names remain unchanged.

Transduction compositions explored in this dissertation are primarily those for which input and output alphabets across both transductions are the same, that is, $\Sigma = \Delta = \Gamma$. There are some minor exceptions, however (e.g. in §5). Additionally, I restrict compositions of certain function types to guarantee that the complexity of composite functions does not exceed subsequential (see more discussion in §2.3.2).

To illustrate the functionality of $\otimes$, consider a BMRS system denoted $\mathcal{T}_1$, defined over an alphabet $\Sigma = \{a, b, c\}$ (with the assumption that the input and output alphabet are the same). $\mathcal{T}_1$ describes a function that maps an a to a b whenever its immediate successor is an a. As a rewrite rule, this would be: a → b / __ a (henceforth the 'aa rule'). The system is defined below in (5.4); it contains three output boolean function definitions, one for each symbol in the output alphabet (function names are subscripted with '1' for $\mathcal{T}_1$, etc., for clarity).

(5.4)
$$\begin{aligned} a_1(x) &= \texttt{if } \underline{a}a(x) \texttt{ then } \bot \texttt{ else } a(x) \\ b_1(x) &= \texttt{if } \underline{a}a(x) \texttt{ then } \top \texttt{ else } b(x) \\ c_1(x) &= c(x) \end{aligned}$$

Definition $a_1(x)$ evaluates input string positions in the following way: if the current string position under evaluation is an a and its immediate successor in the input is also an a (the term $\underline{a}a(x)$), a false $\bot$ value is returned. A sequence of consecutive as constitutes a *blocking structure* for a in this system. If the structure is not satisfied, evaluation moves to the final term, where a truth value is returned based on whether that position is input-specified as a. Note that the same structure (two consecutive a's) constitutes a *licensing structure* for b; the same evaluation will cause $b_1(x)$ to return a true $\top$ value, otherwise it proceeds to the 'default' evaluation (that is, input specification as b). The definition of $c_1(x)$ states that a symbol maps to c in the output only if it is specified as such in the input. Applied to input string /aaaa/, the system accepts output [bbba], as in (5.5).

(5.5)

Input:	a	a	a	a
	1	2	3	4
$a_1(x)$	$\bot$	$\bot$	$\bot$	$\top$
$b_1(x)$	$\top$	$\top$	$\top$	$\bot$
$c_1(x)$	$\bot$	$\bot$	$\bot$	$\bot$
Output:	b	b	b	a

The first three positions each return a true value for $b_1(x)$ by satisfying the first term—the licensing structure $\underline{a}a(x)$. String position 4 is false for $\underline{a}a(x)$; although the current position is an input a, its successor is undefined, and evaluates to false by default. It proceeds to the final term $a(x)$, for which it is true. None of the positions are true for $c_1(x)$.

Now, let another BMRS system of equations be denoted $\mathcal{T}_2$ and defined over the same alphabet $\Sigma = \{a, b, c\}$. This system describes a function that maps b to c whenever its immediate successor is a. It is describable by the rewrite rule b → c / __ a, henceforth that 'ba rule'. The definition of $\mathcal{T}_2$ is given in (5.6):

(5.6)
$$
\begin{aligned}
a_2(x) &= a(x) \\
b_2(x) &= \texttt{if } \underline{b}a(x) \texttt{ then } \bot \texttt{ else } b(x) \\
c_2(x) &= \texttt{if } \underline{b}a(x) \texttt{ then } \top \texttt{ else } c(x)
\end{aligned}
$$

Like (5.4), this set of output boolean function definitions contains a pair of licensing/blocking structures; a *ba* sequence—represented by the term $\underline{b}a(x)$ in $b_2(x)$ and $c_2(x)$—blocks an output b but licenses an output c. Applied to an input string /bbba/, this system accepts the output [bbca], as illustrated by the evaluation table in (5.7):

(5.7)

Input:	b	b	b	a
	1	2	3	4
$a_2(x)$	$\bot$	$\bot$	$\bot$	$\top$
$b_2(x)$	$\top$	$\top$	$\bot$	$\bot$
$c_2(x)$	$\bot$	$\bot$	$\top$	$\bot$
Output:	b	b	c	a

These individual systems may be combined using the $\otimes$ operator. For clarity, the transduction to the left of the operator is termed the 'outer' function, while the transduction to the right of the operator is termed the 'inner' function. The result is a new system of equations describing the composition of the individual transductions. To illustrate, the composite system $\mathcal{T}_2 \otimes \mathcal{T}_1$ is the union of $\mathcal{T}_1$ with a *modified* $\mathcal{T}_2$ (denoted $\mathcal{T}_2'$), the result of replacing every *non-recursively defined* function name in $\mathcal{T}_2$'s function definitions—that is, every occurrence of $a(x)$, $b(x)$, or $c(x)$—with the corresponding function definition from $\mathcal{T}_1$. Put another way, combination via $\otimes$ modifies $\mathcal{T}_2$ (the outer function) such that the right side of every output boolean function definition is indexed with definitions from $\mathcal{T}_1$ wherever a labeling predicate appears and is not defined recursively. Truth values returned from evaluating $\mathcal{T}_2'$ determine the output string. $\mathcal{T}_2 \otimes \mathcal{T}_1$ is defined below in (5.8); the example contains the modified $\mathcal{T}_2'$ as well as the original $\mathcal{T}_1$ for reference (all subsequent definitions in this chapter include the 'inner' function in each composite definition, as well).

(5.8) a. $\mathcal{T}_2 \otimes \mathcal{T}_1$
$$
\begin{aligned}
a_2(x) &= a_1(x) \\
b_2(x) &= \texttt{if } \underline{b}_1 a_1(x) \texttt{ then } \bot \texttt{ else } b_1(x) \\
c_2(x) &= \texttt{if } \underline{b}_1 a_1(x) \texttt{ then } \top \texttt{ else } c_1(x)
\end{aligned}
$$

b. $\mathcal{T}_1$
$$
\begin{aligned}
a_1(x) &= \texttt{if } \underline{a}a(x) \texttt{ then } \bot \texttt{ else } a(x) \\
b_1(x) &= \texttt{if } \underline{a}a(x) \texttt{ then } \top \texttt{ else } b(x) \\
c_1(x) &= c(x)
\end{aligned}
$$

The modified system in (5.8a) comprises three output boolean functions which determine whether output string positions are marked a, b, or c. Definition $a_2(x)$ contains a single term $a_1(x)$ (the result of indexing non-recursively-defined $a(x)$); this means that, for any string position under consideration, its truth value is identical to the value computed for $a_1(x)$. In other words, a sequence of two consecutive input as will cause this function to be false, otherwise it will be true if that position is an a in the input (otherwise false, as in the definition of $a_1(x)$).

Functions $b_2(x)$ and $c_2(x)$ contain the modified licensing/blocking structure pair as the first term of their if-then-else statements. In order to satisfy (i.e. return a true $\top$ value) for the term $\underline{b}_1 a_1(x)$, the current string position must evaluate to true for $b_1(x)$ from system $\mathcal{T}_1$ *and* its immediate successor must evaluate to true for $a_1(x)$ from system $\mathcal{T}_1$. In the unmodified system $\mathcal{T}_2$, the crucial structure was a *ba* sequence in the input, whereas the structure described here is a *ba* sequence in the *output* of $\mathcal{T}_1$. This structure licenses an output b and blocks an output c, meaning that it will cause function $b_2(x)$ to be true and output a b. Likewise, it will cause $c_2(x)$ to be false and prevent an output c on that position. If the structure is not satisfied, $b_2(x)$ and $c_2(x)$ receive their truth values from $b_1(x)$ and $c_1(x)$, respectively, from $\mathcal{T}_1$, as indicated in the third term of both definitions' if-then-else statements.

The modified system $\mathcal{T}_2 \otimes \mathcal{T}_1$ represents the composition of systems $\mathcal{T}_2$ and $\mathcal{T}_1$, and so accepts input-output mappings where $\mathcal{T}_2$ applies to the output of $\mathcal{T}_1$. As an example, $\mathcal{T}_1(aaaa)$—$\mathcal{T}_1$ applied to the input string *aaaa*—produces [bbba], as in (5.5). Plugging that output directly into $\mathcal{T}_2$, i.e. $\mathcal{T}_2(bbba)$, would predict [bbca], intuitively because the *ba* substring in positions 3 and 4 satisfy the structural description of the rewrite rule b → c / __ a *and no others*, as in (5.7). The composite system $\mathcal{T}_2 \otimes \mathcal{T}_1$ in (5.8) makes the same prediction, illustrated by the evaluation in (5.9):

(5.9)

Input:	a	a	a	a
	1	2	3	4
$a_2(x)$	$\perp$	$\perp$	$\perp$	$\top$
$b_2(x)$	$\top$	$\top$	$\perp$	$\perp$
$c_2(x)$	$\perp$	$\perp$	$\top$	$\perp$
$a_1(x)$	$\perp$	$\perp$	$\perp$	$\top$
$b_1(x)$	$\top$	$\top$	$\top$	$\perp$
$c_1(x)$	$\perp$	$\perp$	$\perp$	$\perp$
Output:	b	b	c	a

As in (5.5), the first three positions of input string /aaaa/ return true values for $b_1(x)$. This has two effects on the evaluation of positions 1 and 2 against $b_2(x)$; first, neither position satisfies the licensing/blocking structure represented by the term $\underline{b}_1 a_1(x)$. Second, when evaluation proceeds to the final term of $b_2(x)$—$b_1(x)$—those truth values are passed directly from $\mathcal{T}_1$, such that 1 and 2 output as b. Position 3 does satisfy $\underline{b}_1 a_1(x)$ (as the evaluation table shows), licensing an output c. Finally, position 4 is output as a given its input specification; that is, it is true for $a_2(x)$ by virtue of being true for $b_1(x)$.

5.2.2 $\otimes$ as rule ordering

The output of composite system $\mathcal{T}_2 \otimes \mathcal{T}_1$ is identical to ordering the aa rule before the ba rule in a rule-based framework, denoted $aa < ba$. The derivation of input string /aaaa/ in (5.10) illustrates, where modifications to a string via some rule are indicated in bold:

(5.10)

		/aaaa/
aa rule:	a → b / __ a	[**bbb**a]
ba rule:	b → c / __ a	[bb**c**a]
Output:		[bbca]

Given this ordering, the aa rule applies to the input string /aaaa/, producing an intermediate representation [bbba]. Then, the ba rule applies to that intermediate representation to yield [bbca]. As no other rules are ordered after ba, the output is [bbca]. This is the same output predicted for the composite system $\mathcal{T}_2 \otimes \mathcal{T}_1$ applied to /aaaa/, with the only difference being that the latter is a direct mapping from input to output.[1]

A rule-based approach describes a stepwise derivation with distinct intermediate forms after each rule applies, while the BMRS composition framework describes a single mapping from input to output. However, the modifications that $\otimes$ makes via indexation in output boolean function definitions serve, in a sense, to reconstruct these intermediate representations. In $\mathcal{T}_2 \otimes \mathcal{T}_1$, the outer function ($\mathcal{T}_2$) is interpreted in terms of the changes made to the input string *by the inner function* ($\mathcal{T}_1$), and not the input string itself. This can be seen, for example, in the modified definition of $c_2(x)$. The structure $\underline{b}_1 a_1(x)$—a ba sequence as determined by transduction $\mathcal{T}_1$—licenses an output c, and this is crucially not an input ba sequence. Other function definitions in $\mathcal{T}_2'$, having also been indexed with definitions from $\mathcal{T}_1$, reconstruct intermediate representations in the same way. This applies not only to the input string /aaaa/ in (5.9) and (5.10), but any string from the alphabet $\Sigma = \{a, b, c\}$.

The derivation in (5.10) also shows that the aa and ba rules interact: *aa feeds ba* given ordering $aa < ba$. That is to say, earlier application of aa provides additional inputs to ba. The locus of this feeding relationship is string position 3 in the input /aaaa/. After aa applies to the input string, it yields an intermediate representation [bbba]. A substring [ba] satisfies the structural

[1] Here, the 'covert' output function definitions from $\mathcal{T}_1$ act as a sort of intermediate representation.

description of the *ba* rule, allowing it to apply. Given that composition of BMRS systems using $\otimes$ is equivalent to rule ordering, this framework can be employed to model process interactions capturable by rule ordering. The remainder of the chapter provides further illustration of this equivalence, using examples from three case studies of tone sandhi interactions in Chinese dialects.

Before proceeding to the case studies, however, I first sketch a broad typology of compositions of strictly-local (SL) functions and examine their properties.

5.2.3 Compositions of strictly-local functions

Since BMRS systems of equations describe subsequential functions (Bhaskar et al., 2020), they also describe the *strictly-local* subclass of subsequential functions (Chandlee, 2014). An earlier chapter summarized further divisions of the strictly-local class of functions based on what information is used to compute the output: input strictly-local (ISL), left output strictly-local (LOSL) and right output strictly-local (ROSL). This section develops an initial typology of $\otimes$ composition with respect to strictly-local functions, and examines the properties of such composite systems. This includes the composition of two ISL functions, composition of two OSL functions, composition of an ISL function with an OSL function, and composition of an OSL function with an ISL function (keeping in mind that order of composition is important). This typology will come into play in the three case studies examined in later sections, as each member of the typology is evinced by an attested interactions. Importantly, the basic facts about SL functions will clarify the nature of tone sandhi directionality in interaction contexts, crucially in a way that earlier theories struggle to account for.

5.2.3.1 ISL $\otimes$ ISL

Chandlee and Lindell (to appear) conjecture that 'finite-to-one' ISL functions are closed under composition. Functions of this type are those with a bound on the ratio between input and output string lengths, such that for a function $f : \Sigma^* \rightarrow \Gamma^*$, it is finite-to-one if its inverse function $f^{-1}(y)$ is always finite. The length-preserving transductions investigated in this dissertation are of the finite-to-one type, so it follows that the same closure property extends to $\otimes$ compositions of BMRS systems of equations describing ISL functions.

Systems of equations corresponding to rewrite rules *aa* and *ba* in (5.4) and (5.5) describe ISL functions because output boolean function definitions are defined only in terms of input structure (that is, they contain no recursion). Their composition via the $\otimes$ operator (5.8) also describes an ISL function, given the closure property described above. In spite of the fact that, given $\mathcal{T}_2 \otimes \mathcal{T}_1$

we say that $\mathcal{T}_2$ determines the output using a local bounded window over $\mathcal{T}_1$'s *output*, since $\mathcal{T}_1$ is computed using reference to input structure only—and $\mathcal{T}_2$ is computed entirely in terms of $\mathcal{T}_1$ under $\otimes$ (i.e. all function calls are indexed)—it *is* the case that the composite function is computed with a local bounded window over the input structure.

This also echoes the generalization that *aa* and *ba* rule applications—either in isolation or combined in a derivation—is simultaneous application. That is, the rule(s) applies to every input substring satisfying the structural description in a single 'pass' (only the input string matters); e.g. the *aa* rule applied simultaneously to /aaaa/ produces [bbba] (5.5).

5.2.3.2 OSL $\otimes$ OSL

The generalization changes when rules apply iteratively, i.e. such that the output of a rule's application can feed/bleed further application of the rule; this type of application is described by OSL functions (Chandlee, 2014, 36). Compare simultaneous and iterative application of the *aa* rule over input string /aaaa/ (5.11).

(5.11)

		Simultaneous		Iterative
aa rule	a.	/aaaa/ → [bbba]	b.	/aaaa/ → [baba]

In (5.11a), all instances of an input *a* followed by an input *a* map to *b*. In (5.11b), however, the rule applies iteratively, starting from the right edge and proceeding leftward. Scanning in this way, the third position is output as *b*, as it satisfies the necessary *aa* environment. However, the rule is unable to apply on the second *a*; earlier application has bled the *aa* environment. Output as an *a* via vacuous application, the second *a* provides the crucial rightward context for *aa* to apply to the first string position. The full output is [baba].

The mapping in (5.11b) is described by an ROSL function. That is, the decision of how to map an input to an output is determinable using only a bound window in the output *to the right* of that input. Access to the recent output is precisely what allows for a process to apply 'iteratively' or repeatedly. Example (5.12) provides a BMRS equivalent definition of the iterative rule, and it is denoted $\mathcal{T}_3$:

(5.12)
$$\begin{aligned} a_3(x) &= \texttt{if } \underline{a}a_3(x) \texttt{ then } \bot \texttt{ else } a(x) \\ b_3(x) &= \texttt{if } \underline{a}a_3(x) \texttt{ then } \top \texttt{ else } b(x) \\ c_3(x) &= c(x) \end{aligned}$$

Note that the definition is identical to the system describing simultaneous *aa* application in (5.4) with the exception that the licensing/blocking structure pair is defined recursively. This recursion appears to the right of the current string position (underlined), which intuitively indicates an ROSL function. An ROSL equivalent of the *ba* rule is defined in a similar way. Let this transduction be denoted $\mathcal{T}_4$ in (5.13).

(5.13)
$$a_4(x) = a(x)$$
$$b_4(x) = \texttt{if } \underline{b}a_4(x) \texttt{ then } \bot \texttt{ else } b(x)$$
$$c_4(x) = \texttt{if } \underline{b}a_4(x) \texttt{ then } \top \texttt{ else } c(x)$$

Like (5.6), $\mathcal{T}_4$ contains a licensing/blocking structure pair for the substring *ba*, however this structure is defined recursively: the term $\underline{b}a_4(x)$ evaluates to true when the current input string is b and its immediate successor is an a in the *output*.

Before proceeding, an important clarification is necessary.[2] Recall in (5.11b) that iterative application of the *aa* rule derives [baba] from /aaaa/, but only when the input is scanned from right to left. This is at the heart of the issues of directionality in tone sandhi interactions. Because ROSL BMRS definitions call functions recursively to the right of some input position, it is tempting to assume a similar directional procedure for the computation: the string is scanned from the right to the left in a step-wise fashion, with outputs being generated at each step, etc. It is important to remember that the BMRS systems defined here do not describe *procedures* or algorithms for generation. Instead, they are (in a sense) 'timeless', simply describing the conditions under which inputs map to outputs. Simply put, an input-output mapping either satisfies the definitions or it does not. This distinction is important, and is implicit in the analyses that follow.[3]

Unlike finite-to-one ISL functions, OSL functions are not closed under composition (Chandlee, 2014, 158-9). However, the following sections investigate tone sandhi interactions formalizable as compositions of OSL functions. Despite the lack of closure under composition, these compositions can be restricted such that they remain within the subsequential bound.

Bhaskar et al. (2020) define two variants of BMRS logic and connect them to the subsequential class of functions. BMRS^p defines the set of systems which contain no terms of the form $s(T_1)$ for any term T_1 (that is, which call a *predecessor* function only), and which correspond to the left-subsequential class. BMRS^s, by contrast, defines the set of systems containing no terms of the form $p(T_1)$ for any term T_1 (that is, which call a *successor* function only), and which correspond to the right-subsequential class. LOSL functions are describable in BMRS^p while ROSL functions are describable in BMRS^s. Given Elgot and Mezei (1965)'s result that any rational function is the

[2] I thank Jane Chandlee for helpful discussion about this issue.

[3] The related distinction between simultaneous and iterative rule application—and thus between ISL and OSL functions—is straightforward with the *aa* rule, but less so for the *ba* rule. For example, both $\mathcal{T}_2$ and $\mathcal{T}_4$ accept the mapping /bbba/ $\mapsto$ [bbca] (and in general accept the same mappings $\Sigma^* \rightarrow \Gamma^*$ when $\Sigma = \Gamma = \{a, b, c\}$). Subsequent sections explore this issue in more detail (see especially §4.1 and §5.1).

composition of a right- and left-subsequential function, composing an LOSL function with an ROSL function may result in a fully-regular composite function.

To prevent this, the main body of this dissertation only considers compositions of two length-preserving ROSL *or* LOSL functions, with the understanding that such composite functions are not strictly more expressive than their individual components—that is, they are right- or left-subsequential.[4] An example of the latter is $\mathcal{T}_4 \otimes \mathcal{T}_3$, the composition of two ROSL functions, defined below in (5.14):

(5.14) a. $\mathcal{T}_4 \otimes \mathcal{T}_3$

$$\begin{aligned} a_4(x) &= a_3(x) \\ b_4(x) &= \texttt{if } \underline{b}_3 a_4(x) \texttt{ then } \bot \texttt{ else } b_3(x) \\ c_4(x) &= \texttt{if } \underline{b}_3 a_4(x) \texttt{ then } \top \texttt{ else } c_3(x) \end{aligned}$$

b. $\mathcal{T}_3$

$$\begin{aligned} a_3(x) &= \texttt{if } \underline{a} a_3(x) \texttt{ then } \bot \texttt{ else } a(x) \\ b_3(x) &= \texttt{if } \underline{a} a_3(x) \texttt{ then } \top \texttt{ else } b(x) \\ c_3(x) &= c(x) \end{aligned}$$

All non-recursively-defined boolean functions in definitions $a_4(x)$, $b_4(x)$, and $c_4(x)$ are indexed with corresponding definitions from $\mathcal{T}_3$ as per the definition of the $\otimes$ operator. Portions of the definition containing recursive function calls—crucially in the licensing/blocking structure pair $\underline{b}a_4(x)$—retain their index. Thus the modified term which licenses an output c and blocks an output b is $\underline{b}_3 a_4(x)$. It only returns a true value when the current string position under evaluation is true for $b_3(x)$ *and* its successor is true for $a_4(x)$. Applied to input string /aaaa/, the composite system accepts the mapping /aaaa/ $\mapsto$ [caca], as illustrated in (5.15).

(5.15)

Input:	a	a	a	a
	1	2	3	4
$a_4(x)$	$\bot$	$\top$	$\bot$	$\top$
$b_4(x)$	$\bot$	$\bot$	$\bot$	$\bot$
$c_4(x)$	$\top$	$\bot$	$\top$	$\bot$
$a_3(x)$	$\bot$	$\top$	$\bot$	$\top$
$b_3(x)$	$\top$	$\bot$	$\top$	$\bot$
$c_3(x)$	$\bot$	$\bot$	$\bot$	$\bot$
Output:	c	a	c	a

Composing the two transductions in this order recreates the effect of ordering the *aa* rule before

[4] This does not amount to a claim that compositions of LOSL and ROSL functions are impossible in phonology; patterns modeled by such compositions are attested in segmental phonology Heinz and Lai (2013); McCollum et al. (2020) and in tone (Jardine, 2016). Instead, the purpose of this restriction is to demonstrate that various sandhi interactions of a *purported* high complexity fall well within the Subregular Hypothesis. See Chapter 7 for an analysis of a harmony pattern that relaxes this restriction.

the *ba* rule, and having both rules apply iteratively. As before, this ordering models a feeding relationship between *aa* and *ba*, as earlier application of the former creates inputs for application of the latter.

5.2.3.3 ISL $\otimes$ OSL and OSL $\otimes$ ISL

ISL and OSL functions may also combine via composition. Two examples of ISL/OSL compositions are given in this section; they represent orderings of rules with either simultaneous or iterative application.

First, let $\mathcal{T}_2 \otimes \mathcal{T}_3$ be a system of equations denoting the composition of an ISL function with an OSL function, defined in (5.16):

(5.16) a. $\mathcal{T}_2 \otimes \mathcal{T}_3$

$$\begin{aligned} a_2(x) &= a_3(x) \\ b_2(x) &= \texttt{if } \underline{b}_3 a_3(x) \texttt{ then } \bot \texttt{ else } b_3(x) \\ c_2(x) &= \texttt{if } \underline{b}_3 a_3(x) \texttt{ then } \top \texttt{ else } c_3(x) \end{aligned}$$

b. $\mathcal{T}_3$

$$\begin{aligned} a_3(x) &= \texttt{if } \underline{a} a_3(x) \texttt{ then } \bot \texttt{ else } a(x) \\ b_3(x) &= \texttt{if } \underline{a} a_3(x) \texttt{ then } \top \texttt{ else } b(x) \\ c_3(x) &= c(x) \end{aligned}$$

Like $\mathcal{T}_2 \otimes \mathcal{T}_1$ in example (5.8), the outer function in the composite system above is ISL, and so *all* boolean functions are indexed with correspondents from $\mathcal{T}_3$. This includes the *ba* licensing/blocking structure in $b_2(x)$ and $c_2(x)$. $\mathcal{T}_3$ describes an ROSL function, indicated by recursive function calls to the right of the current string position under evaluation in its definitions. The composite function mirrors a rule-based grammar where iterative application of the *aa* rule precedes simultaneous application of the *ba* rule. Like (5.14), the former feeds the latter, such that /aaaa/ maps to [caca].

Now, let $\mathcal{T}_4 \otimes \mathcal{T}_1$ be a system of equations denoting the composition of an OSL function with an ISL function, defined in (5.17).

(5.17) a. $\mathcal{T}_4 \otimes \mathcal{T}_1$

$$\begin{aligned} a_4(x) &= a_1(x) \\ b_4(x) &= \texttt{if } \underline{b}_1 a_4(x) \texttt{ then } \bot \texttt{ else } b_1(x) \\ c_4(x) &= \texttt{if } \underline{b}_1 a_4(x) \texttt{ then } \top \texttt{ else } c_1(x) \end{aligned}$$

b. $\mathcal{T}_1$

$$\begin{aligned} a_1(x) &= \texttt{if } \underline{a} a(x) \texttt{ then } \bot \texttt{ else } a(x) \\ b_1(x) &= \texttt{if } \underline{a} a(x) \texttt{ then } \top \texttt{ else } b(x) \\ c_1(x) &= c(x) \end{aligned}$$

Recursive calls in the outer function's definitions—e.g. in the term $\underline{b}a_4(x)$—retain their index, and all non-recursively-defined function calls refer to corresponding definitions in $\mathcal{T}_1$. Licensing/blocking structures in the modified system therefore resemble the OSL/OSL composition $\mathcal{T}_4 \otimes \mathcal{T}_3$ (5.14). That is, computing a truth value for the relevant term requires evaluation from both the inner function *and* outer function. In (5.17), this can be seen in the modified term $\underline{b}_1 a_4(x)$. This also means that the function is still defined recursively, and calculates outputs based on a local output window to the right of the current string position. It mirrors a grammar where simultaneous application of the *aa* rule precedes iterative application of the *ba* rule.

5.2.4 Interim Summary

So far, this chapter has defined a composition operator $\otimes$ over BMRS systems of equations and sketched its properties. When BMRS systems model individual 'rules' in a rule-based formalism, composition provides a formal expression of pairwise rule ordering. Additionally, systems describing strictly-local functions (ISL, LOSL, ROSL) can combine under the operator. Generalizations about mode of application (simultaneous vs. iterative) and ISL/OSL functions extend to these composite systems.

The typology presented in the previous section exhausts the logical combinations of *two* ISL and OSL functions, keeping in mind the restriction on R/LOSL composition. Provided that the restriction is maintained, compositions of *more than two* ISL and OSL functions can be defined which describe subsequential functions (see §3.4 for an example).

In the next three sections, I present case studies of tone sandhi interactions in Tianjin, Changting, and Nanjing. Transparent (feeding) and opaque (counterbleeding) interactions expressible by rule ordering are formalized as compositions of BMRS systems, and all four SL compositions outlined in the broad typology are attested. Importantly, the asymmetry between ISL and OSL functions with respect to mode of 'rule' application—extrapolated to *compositions* of SL functions—provides a straightforward account of so-called directionality effects in Chinese tone sandhi.

5.3 Tianjin

Chapter 2 (§3.1) introduced the sandhi paradigm in Tianjin (Li and Liu, 1985), a dialect with four lexical tones: H(igh), L(ow), R(ising), and F(alling). The three relevant sandhi rules are repeated below.

(5.18) a. FF rule: FF $\mapsto$ LF e.g. $jing^F zhong^F \rightarrow jing^L zhong^F$ 'net weight'

b. LL rule: LL $\mapsto$ RL e.g. $fei^L ji^L \rightarrow fei^R ji^L$ 'air plane'

c. RR rule: RR $\mapsto$ HR e.g. $xi^R lian^R \rightarrow xi^H lian^R$ 'wash one's face'

Two adjacent falling tones surfaces with the first tone realized as a low tone (5.18a), two adjacent low tones surfaces with the first tone realized as a rising tone (5.18b), and two adjacent rising tones surfaces with the first tone realized as a high tone (5.18c).

Recall the tritonal sequences in (5.19), the locus of the directionality 'paradox' in Tianjin.

(5.19) a. FFF $\mapsto$ FLF e.g. $suo^F liao^F bu^F \rightarrow suo^F liao^L bu^F$ 'plastic cloth'

b. LLL $\mapsto$ LRL e.g. $tuo^L la^L ji^L \rightarrow tuo^L la^R ji^L$ 'tractor'

c. RRR $\mapsto$ HHR e.g. $li^R fa^R suo^R \rightarrow li^H fa^H suo^L$ 'barber shop'

Finally, recall the two feeding relationships.

(5.20) a. LL rule feeds RR rule: R$\underline{\text{LL}}$ $\rightarrow$ $\underline{\text{R}\mathbf{R}}$L $\rightarrow$ **H**RL

b. FF rule feeds LL rule: L$\underline{\text{FF}}$ $\rightarrow$ $\underline{\text{L}\mathbf{L}}$F $\rightarrow$ **R**LF

Following Chandlee (2019), this section formalizes transparent feeding in Tianjin sandhi using BMRS. It adopts the same computational classifications; namely, that LL and FF rules describe OSL functions and the RR rule describes an ISL. However, it differs from Chandlee's analysis in that it models the interaction through composition (using the $\otimes$ operator) of BMRS systems. It provides three such composite systems: one representing RR/LL interaction, one representing LL/FF rules interaction, and a full composite system formalizing both interactions. First, however, each individual system is introduced.

5.3.1 Individual rules as systems of equations

The FF, LL, and RR 'rules' in (5.18) are definable as individual BMRS systems. This section shows that corresponding systems make correct predictions about sandhi patterns in isolation. Additionally, their computational classification (ISL vs OSL) is maintained.

Let a BMRS system of equations denoted a model the FF rule (as before, function names are subscripted with 'a' for clarity). It describes an ROSL function. Again, this means that the output is computed by examining the current input and a bounded output window to the right of that input. This is reflected in the output boolean function definitions in (5.21), where structures—crucially those referring to positions to the right of the string position being evaluated—are defined recursively.

$$
\begin{aligned}
(5.21) \quad H_a(x) &= H(x)\\
R_a(x) &= R(x)\\
L_a(x) &= \texttt{if } \underline{F}F_a(x) \texttt{ then } \top \texttt{ else } L(x)\\
F_a(x) &= \texttt{if } \underline{F}F_a(x) \texttt{ then } \bot \texttt{ else } F(x)
\end{aligned}
$$

The locus of iterative FF-rule application is the output boolean function $L_a(x)$. If an input F tone is followed immediately by an output F tone (described by the structure $\underline{F}F_a(x)$), an L tone is licensed in the output. This same configuration blocks an output F in the same position. An example evaluation table in (5.22) illustrates. The system accepts the mapping /FFFF/ $\mapsto$ [LFLF], consistent with the iterative/repeated application of the FF rule attested in the sandhi data, and associated with a 'right-to-left' scan in earlier accounts.

(5.22)

Input:	F 1	F 2	F 3	F 4
$H_a(x)$	$\bot$	$\bot$	$\bot$	$\bot$
$R_a(x)$	$\bot$	$\bot$	$\bot$	$\bot$
$L_a(x)$	$\top$	$\bot$	$\top$	$\bot$
$F_a(x)$	$\bot$	$\top$	$\bot$	$\top$
Output:	L	F	L	F

Note in particular the evaluation of string positions 1 and 3 above. They both satisfy $L_a(x)$ by virtue of being an input F followed immediately by an output F. It is for this same reason that they fail to satisfy the definition of $F_a(x)$, as evidenced by the false $\bot$ value returned at both of those position. Positions 2 and 4 satisfy only the final term of $F_a(x)$ as input-specified F tones. The reader can verify the accuracy of other mappings for this system, including non-application of the rules, e.g. /LFLF/ $\mapsto$ [LFLF].

In a similar manner, let a system of equations b describe the LL rule. Its definition is given in (5.23).

$$
\begin{aligned}
(5.23) \quad H_b(x) &= H(x)\\
R_b(x) &= \texttt{if } \underline{L}L_b(x) \texttt{ then } \top \texttt{ else } R(x)\\
L_b(x) &= \texttt{if } \underline{L}L_b(x) \texttt{ then } \bot \texttt{ else } L(x)\\
F_b(x) &= F(x)
\end{aligned}
$$

Much like L_a in system a, the output boolean function R_b contains a licensing structure with a recursively-defined function to the right of the evaluated (underlined) input position. Thus, this describes an ROSL function and models iterative application of the LL rule. That the system produces the correct mapping /LLLL/ $\mapsto$ [RLRL] is illustrated in the evaluation table in (5.24).

I reiterate here that previous accounts of Tianjin conflate this form of iterative application with right-to-left directionality.

(5.24)

Input:	L	L	L	L
	1	2	3	4
$H_b(x)$	$\perp$	$\perp$	$\perp$	$\perp$
$R_b(x)$	$\top$	$\perp$	$\top$	$\perp$
$L_b(x)$	$\perp$	$\top$	$\perp$	$\top$
$F_b(x)$	$\perp$	$\perp$	$\perp$	$\perp$
Output:	R	L	R	L

The mapping from input string /LLLL/ to output string [RLRL] is evaluated in the following way. Positions 1 and 3 returns a true $\top$ value for the output boolean function $R_b(x)$ by satisfying the licensing structure $\underline{L}L_b$ (an input L tone immediately succeeded by an output L tone). String position 2, an input L tone, evaluates to *false* for $R_b(x)$, and returns a true value for $L_b(x)$ by virtue of being input-specified as L. This is consistent with iterative LL application.

Finally, let c be a BMRS system of equations describing the RR sandhi alternation in Tianjin. This system (5.25) differs from systems a and b in that output boolean function definitions contain *no* recursion; this corresponds to Chandlee (2019)'s observation that an ISL function describes the RR sandhi pattern.

(5.25)
$$
\begin{aligned}
H_c(x) &= \texttt{if } \underline{R}R(x) \texttt{ then } \top \texttt{ else } H(x) \\
R_c(x) &= \texttt{if } \underline{R}R(x) \texttt{ then } \perp \texttt{ else } R(x) \\
L_c(x) &= L(x) \\
F_c(x) &= F(x)
\end{aligned}
$$

A single structure—a sequence of two input R tones—licenses an H tone in the output and blocks an R tone in the output. Output L and F tones map directly from identical inputs. This system thus describes an ISL function because at any point in the string, computing the output can be done solely by referring to a bounded window in the input. Evaluated against the input string /RRRR/ as in (5.26), this system accepts [HHHR], consistent with simultaneous application of RR. Recall that, in earlier accounts, this mapping requires a left-to-right parse.[5]

[5] Chandlee (2019) shows that, in fact, a parse in *either* direction (assuming simultaneous and not iterative application) would produce the same mapping.

(5.26)

Input:	R 1	R 2	R 3	R 4
$H_c(x)$	$\top$	$\top$	$\top$	$\perp$
$R_c(x)$	$\perp$	$\perp$	$\perp$	$\top$
$L_c(x)$	$\perp$	$\perp$	$\perp$	$\perp$
$F_c(x)$	$\perp$	$\perp$	$\perp$	$\perp$
Output:	H	H	H	R

Here, string positions 1, 2, and 3 all return a true value for $H_c(x)$ (and therefore a false value for $R_c(x)$) as an input R tone followed immediately by another input R tone. As the licensing/blocking structure $\underline{R}R(x)$ is defined non-recursively, the output specification of string positions 2, 3, and 4 is irrelevant to computing these positions. A given string position satisfies this structure even if its immediate successor is output as H (as in the case of positions 1 and 2). Thus, the ISL characteristic of this function coincides with 'simultaneous' application of the RR rule in strings of three or more R tones.

5.3.2 LL rule feeds RR rule (ISL $\otimes$ OSL)

Having defined disyllabic Tianjin sandhi patterns as individual systems, I now turn to their interaction. Recall the feeding relationship that obtains between LL and RR rules in (5.20a), repeated below.

(5.27) LL rule feeds RR rule: R<u>LL</u> → <u>R**R**</u>L → **H**RL

This transparent feeding interaction can be modeled using BMRS composition. That is, for any two BMRS systems of equations $\mathcal{T}_1$ and $\mathcal{T}_2$, the composition $\mathcal{T}_2 \otimes \mathcal{T}_1$ is the union of $\mathcal{T}_1$ with a modified outer function $\mathcal{T}_2'$. The modification is as follows: any non-recursively-defined function appearing in $\mathcal{T}_2$'s set of definitions is indexed with the corresponding function definition from system $\mathcal{T}_1$. Indexing in this way, the 'output' of $\mathcal{T}_1$ becomes the 'input' of $\mathcal{T}_2$. The feeding interaction between Tianjin LL and RR patterns—defined as systems b and c above—using this formalism is thus describable with the composite function $c \otimes b$. Composing the systems in this way entails indexing every non-recursively-defined labeling predicate in system c's boolean output function definitions such that it points toward the corresponding output boolean function definition in system b. This is given in (5.28), along with the original system b for reference.[6]

[6] For the rest of this dissertation, combined systems will follow this convention for ease of reference.

(5.28) a. $c \otimes b$

$$\begin{aligned}
H_c(x) &= \texttt{if } \underline{R_b}R_b(x) \texttt{ then } \top \texttt{ else } H_b(x)\\
R_c(x) &= \texttt{if } \underline{R_b}R_b(x) \texttt{ then } \bot \texttt{ else } R_b(x)\\
L_c(x) &= L_b(x)\\
F_c(x) &= F_b(x)
\end{aligned}$$

b. b

$$\begin{aligned}
H_b(x) &= H(x)\\
R_b(x) &= \texttt{if } \underline{L}L_b(x) \texttt{ then } \top \texttt{ else } R(x)\\
L_b(x) &= \texttt{if } \underline{L}L_b(x) \texttt{ then } \top \texttt{ else } L(x)\\
F_b(x) &= F(x)
\end{aligned}$$

Given that c is an ISL function, all labeling predicates within output boolean function definitions are non-recursively-defined. This means that every instance of a licensing/blocking or default structure is indexed with the corresponding definition from system b as above. The composed system corresponds to the composition type shown in §2.3.3; that is, it mirrors a rule-based grammar where a rule applying iteratively precedes a different rule applying simultaneously.

To see how the composite function works, consider the interaction mapping /RLL/ $\mapsto$ [HRL] from (5.20a), correctly predicted by this system, and illustrated in the evaluation table in (5.29).

(5.29)

Input:	R	L	L
	1	2	3
$H_c(x)$	$\top$	$\bot$	$\bot$
$R_c(x)$	$\bot$	$\top$	$\bot$
$L_c(x)$	$\bot$	$\bot$	$\top$
$F_c(x)$	$\bot$	$\bot$	$\bot$
$H_b(x)$	$\bot$	$\bot$	$\bot$
$R_b(x)$	$\top$	$\top$	$\bot$
$L_b(x)$	$\bot$	$\bot$	$\top$
$F_b(x)$	$\bot$	$\bot$	$\bot$
Output:	H	R	L

Mapping from input /RLL/, the composite function $c \otimes b$ accepts only the output [HRL]. When string position 3 is evaluated against the definition $L_c(x)$, it must refer to the corresponding definition $L_b(x)$. Its direct mapping from input L to output L satisfies the final term of $L_b(x)$.

To see whether string position 2 can map to an output R, it must be checked against the blocking structure $\underline{R_b}R_b$, indexed with function names from system b. Doing so requires a truth value for each string position (the current position 2 and its successor 3) with respect to $R_b(x)$ in system b; only if *both* evaluate to true $\top$ does an output R satisfy this definition. String position 2 satisfies the licensing structure $\underline{L}L_b(x)$ in the definition of $R_b(x)$ (i.e. returns a true value), but its successor in position 3 does *not*, and is therefore false. String position 2 therefore moves to the third term $R_b(x)$, which it satisfies—i.e. recalling that b models the LL rule—and is output as R.

Unlike position 2, string position 1 satisfies the entire blocking structure $\underline{R}_b R_b$, and returns a false $\perp$ value for $R_c(x)$. Importantly, this structure is not defined in terms of input string positions only, as is $\underline{R}R(x)$ from system c; it calls $R_b(x)$ from system b, which references output structure. Thus, although string position 2 is input-specified as L, it satisfies the definition of $R_b(x)$—intuitively as a result of the LL rule applying. The same structure $\underline{R}_b R_b$, which blocks output R, *licenses* output H tones via the definition of $H_c(x)$. String position 1 is an input R, and string position 2 is an R tone 'fed' by application of the LL rule. The result is a true value returned for $H_c(x)$ for position 1, and an output H. The mapping /RLL/ $\mapsto$ [HRL] accepted by this composite system is consistent with the Tianjin data.

This analysis is not dependent on the arbitrary choice to parse the string from right to left, as is the case of earlier rule- and optimization-based accounts. [7] Instead, it is simply the composition of an ROSL function (corresponding to iterative application) and an ISL function (corresponding to simultaneous application), following Chandlee (2019)'s classifications. Importantly, the difference in mode of rule application is explained as the difference in subclasses of strictly-local functions, and is not constrained by a 'general principle or principles' for determining the direction of string parse.

5.3.3 FF rule feeds LL rule (OSL $\otimes$ OSL)

In addition to feeding the RR rule, the LL rule is also fed by the FF rule. Recall this interaction from (5.20b), repeated in (5.30).

(5.30) FF rule feeds LL rule: L$\underline{\text{F}}$$\underline{\text{F}}$ $\rightarrow$ $\underline{\text{L}}$$\underline{\textbf{L}}$F $\rightarrow$ **R**LF

Much like the composite system defined in the previous section, this feeding interaction is definable as the composition of system b with system a ($b \otimes a$), where b is the BMRS system of equations describing the LL rule and system a the FF rule. Every non-recursively-defined labeling predicate in system b's output boolean function definitions is indexed with the corresponding definition from system a, as in (5.31), where the original system a is also given for reference.

[7] Recall that Zhang (1987)'s analysis claims right-to-left application as default, as it happens away from the determinant/trigger. This is consistent with Howard (1972)'s theory whereby directionality is determined by the rule's form (i.e. applies toward the target). These types of analyses are therefore not entirely arbitrary; what is arbitrary is that the RR rule should violate this principle. A potential criticism of the current analysis is that the choice between representing a process as ISL or OSL is equally arbitrary. However, this distinction is itself a general principle of the theory; processes can either be sensitive to input- or output-local information. Additionally, analyses in terms of strictly-local functions off a more complete account of sandhi interactions than what is available to SPE and OT (see especially the next chapter).

(5.31) a. $b \otimes a$

$$\begin{aligned}
H_b(x) &= H_a(x)\\
R_b(x) &= \texttt{if } \underline{L_a}L_b(x) \texttt{ then } \top \texttt{ else } R_a(x)\\
L_b(x) &= \texttt{if } \underline{L_a}L_b(x) \texttt{ then } \bot \texttt{ else } L_a(x)\\
F_b(x) &= F_a(x)
\end{aligned}$$

b. a

$$\begin{aligned}
H_a(x) &= H(x)\\
R_a(x) &= R(x)\\
L_a(x) &= \texttt{if } \underline{F}F_a(x) \texttt{ then } \top \texttt{ else } L(x)\\
F_a(x) &= \texttt{if } \underline{F}F_a(x) \texttt{ then } \bot \texttt{ else } F(x)
\end{aligned}$$

Unlike $c \otimes b$ in the previous section, not all labeling predicates in $b \otimes a$'s output boolean function definitions are indexed with calls to corresponding functions in system a. Again, this is because b describes an ROSL function, and thus contains recursive function calls.

This composite system captures the FF/LL feeding relationship in Tianjin. The evaluation table in (5.32) illustrates how $b \otimes a$ accepts the interaction mapping /LFF/ $\mapsto$ [RLF].

(5.32)

Input:	L	F	F
	1	2	3
$H_b(x)$	$\bot$	$\bot$	$\bot$
$R_b(x)$	$\top$	$\bot$	$\bot$
$L_b(x)$	$\bot$	$\top$	$\bot$
$F_b(x)$	$\bot$	$\bot$	$\top$
$H_a(x)$	$\bot$	$\bot$	$\bot$
$R_a(x)$	$\bot$	$\bot$	$\bot$
$L_a(x)$	$\top$	$\top$	$\bot$
$F_a(x)$	$\bot$	$\bot$	$\top$
Output:	R	L	F

Note the evaluation of string position 1; here the 'feeding' effect of the composition becomes clear. Satisfying the licensing structure $\underline{L_a}L_b(x)$ (in $R_b(x)$) will map an input L to an output R (mirroring the intuition behind the LL rule). String position 1 does just that. It satisfies the first part of the licensing structure—$L_a(x)$—by being input-specified as L. Its successor (position 2) returns a true value for $L_b(x)$ crucially by satisfying $\underline{F}F_a(x)$ from the definition $L_a(x)$: that is, by triggering the FF rule. In intuitive terms, then, the L furnished by the FF rule (that is, the output of a) feeds the LL rule, the locus of system b.

This is represented explicitly in the structure $\underline{L_a}L_b(x)$; an L tone as determined by system a, and an L tone as determined by system b. In addition to describing the interaction, the composed system also preserves the individual generalizations; LL sequences trigger LL sandhi in system b

and RR sequences trigger RR sandhi in system a. This recoverability of rules is an advantage of the BMRS formalism—see Chapter 7 for more details.

Again, the mapping is describable as the composition of two ROSL functions, and is not dependent on a procedure that generates outputs step by step in a right-to-left parse.

5.3.4 The fully-composed map (ISL ⊗ OSL ⊗ OSL)

Combining the results of the preceding two sections, the map of both interactions—FF feeds LL which feeds RR—can be described by the composition $c \otimes b \otimes a$. Composing three BMRS systems of equations follows the same procedure as that of two systems; definitions in system c are indexed with system b functions, and those definitions are indexed with system a functions. Thus, as shown above, the 'output' of system a (the FF rule) can 'feed' system b (the LL rule), which in turn can 'feed' system c (the RR rule). The composite system is shown below in (5.33) in three parts: the composition $c \otimes b$, the composition $b \otimes a$, and finally the original system a for reference.

(5.33)

a. $c \otimes b$

$$\begin{aligned}
H_c(x) &= \texttt{if } \underline{R_b}R_b(x) \texttt{ then } \top \texttt{ else } H_b(x)\\
R_c(x) &= \texttt{if } \underline{R_b}R_b(x) \texttt{ then } \bot \texttt{ else } R_b(x)\\
L_c(x) &= L_b(x)\\
F_c(x) &= F_b(x)
\end{aligned}$$

b. $b \otimes a$

$$\begin{aligned}
H_b(x) &= H_a(x)\\
R_b(x) &= \texttt{if } \underline{L_a}L_b \texttt{ then } \top \texttt{ else } R_a(x)\\
L_b(x) &= \texttt{if } \underline{L_a}L_b \texttt{ then } \bot \texttt{ else } L_a(x)\\
F_b(x) &= F_a(x)
\end{aligned}$$

c. a

$$\begin{aligned}
H_a(x) &= H(x)\\
R_a(x) &= R(x)\\
L_a(x) &= \texttt{if } \underline{F}F_a(x) \texttt{ then } \top \texttt{ else } L(x)\\
F_a(x) &= \texttt{if } \underline{F}F_a(x) \texttt{ then } \bot \texttt{ else } F(x)
\end{aligned}$$

This composite system correctly predicts interaction mappings /RLL/ ↦ [HRL] and /LFF/ ↦ [RLF] as did $c \otimes b$ and $b \otimes a$ respectively, as well as non-interaction mappings (as those in §3.1) and even mappings between input and output Tianjin tones which do not undergo sandhi, such as /HHH/ ↦ [HHH]. These are given in the evaluation table below in (5.34).

(5.34)

Input:	R	L	L	L	F	F	H	H	H
	1	2	3	1	2	3	1	2	3
$H_c(x)$	$\top$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\top$	$\top$	$\top$
$R_c(x)$	$\bot$	$\top$	$\bot$	$\top$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$
$L_c(x)$	$\bot$	$\bot$	$\top$	$\bot$	$\top$	$\bot$	$\bot$	$\bot$	$\bot$
$F_c(x)$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$	$\bot$	$\bot$
$H_b(x)$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\top$	$\top$	$\top$
$R_b(x)$	$\top$	$\top$	$\bot$	$\top$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$
$L_b(x)$	$\bot$	$\bot$	$\top$	$\bot$	$\top$	$\bot$	$\bot$	$\bot$	$\bot$
$F_b(x)$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$	$\bot$	$\bot$
$H_a(x)$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\top$	$\top$	$\top$
$R_a(x)$	$\top$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$
$L_a(x)$	$\bot$	$\top$	$\top$	$\top$	$\top$	$\bot$	$\bot$	$\bot$	$\bot$
$F_a(x)$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$	$\bot$	$\bot$
Output:	H	R	L	R	L	F	H	H	H

Evaluation of the first two input strings proceeds in a similar manner as described in §3.2 and §3.3; importantly, the addition of system c's contributions to the computation of the first mapping and the addition of system a's contributions to the computation of the second mapping do not alter the nature of the mappings which satisfy the composite system. That is to say, the mapping described by the composition includes /RLL/ $\mapsto$ [HRL] *and* /LFF/ $\mapsto$ [RLF]. It also includes non-sandhi mappings such as /HHH/ $\mapsto$ [HHH]. Here, each string position returns a true value for $H_a(x)$ (by virtue of being specified as an H tone in the input) and a false value for the other output boolean functions in system a. This means that it also evaluates to true for system b's definition $H_b(x)$, and false for all other functions in that system. This in turn guarantees that each position returns a true value for $H_c(x)$ when it computes the default term $H_b(x)$. Since positions 1, 2, and 3 evaluate to true for $H_c(x)$, we may say that the entire composite system accepts the mapping /HHH/ $\mapsto$ [HHH]. This mapping from input to output tones is consistent with attested Tianjin data.

5.4 Changting

Chapter 2 (§3.2) presented data from the Hakka dialect Changting (Luo, 1982; Chen, 2000, 2004; Chen et al., 2004), a five-tone dialect with H(igh), L(ow), M(id), R(ising), and F(alling) tones. Four sandhi alternations relevant to this chapter are repeated in (5.35).

(5.35) a. LF rule: LF $\mapsto$ MF e.g. $dai^{L}biao^{F} \rightarrow dai^{M}biao^{F}$ 'represent'

b. RM rule: RM $\mapsto$ HM e.g. $han^{R}leng^{M} \rightarrow han^{H}leng^{M}$ 'cold'

c. MR rule: MR $\mapsto$ LR e.g. $hua^{M}qian^{R} \rightarrow hua^{L}qian^{R}$ 'spend money'

d. ML rule: ML $\mapsto$ LL e.g. $gan^{M}yuan^{L} \rightarrow gan^{L}yuan^{L}$ 'willing'

Recall the trisyllabic data (5.36) which illustrate the following feeding relationships: the LF rule (LF $\rightarrow$ MF) feeds the RM rule (RM $\rightarrow$ HM), and the MR rule (MR $\rightarrow$ LR) feeds the ML rule (ML $\rightarrow$ LL).

(5.36) a. LF rule feeds RM rule: R$\underline{\text{LF}}$ $\rightarrow$ $\underline{\text{R}\mathbf{M}}$F $\rightarrow$ **H**MF

b. MR feeds ML rule: M$\underline{\text{MR}}$ $\rightarrow$ $\underline{\text{M}\mathbf{L}}$R $\rightarrow$ **L**LR

The current section focuses on the transparent feeding interactions in (5.36), demonstrating how they can be modeled as the composition of BMRS systems of equations using the $\otimes$ operator (and thus as subsequential functions). In particular, the feeding relationship between LF and RM rules in (5.36a) is the composition of two ISL functions, and the feeding relationship between MR and ML rules in (5.36b) is the composition of an OSL function with an ISL function. Problematic cases noted by Chen (2004) and taken up by Oakden and Chandlee (2020) are introduced in the discussion section of this chapter, and treated in more detail in the next chapter. As before, the following sections define individual rules as separate BMRS systems of equations first, and then explore their interactions as composition.

5.4.1 Individual rules as systems of equations

The four Changting sandhi rules in (5.35) are definable as individual BMRS systems. This section defines systems corresponding to the Changting disyllabic sandhi patterns, and illustrates how they recreate the effect of those processes applying in isolation. Issues facing the designation of a particular rule as representing a properly-ISL or properly-OSL function are also discussed. Where relevant, a particular designation is adopted.

First, let a BMRS systems of equations denoted a model the LF rule, given in (5.37).

(5.37)
$$\begin{aligned}
H_a(x) &= H(x)\\
R_a(x) &= R(x)\\
F_a(x) &= F(x)\\
M_a(x) &= \texttt{if } \underline{L}F(x) \texttt{ then } \top \texttt{ else } M(x)\\
L_a(x) &= \texttt{if } \underline{L}F(x) \texttt{ then } \bot \texttt{ else } L(x)
\end{aligned}$$

The relevant structure here is an input L tone followed by an input F tone; this structure licenses output M and blocks output L. Given that all structures are input-oriented, the function described by a is ISL. It accepts the disyllabic sandhi mapping /LF/ $\mapsto$ [MF], as shown in the evaluation table in (5.38).

(5.38)

Input:	L	F
	1	2
$H_a(x)$	$\bot$	$\bot$
$R_a(x)$	$\bot$	$\bot$
$F_a(x)$	$\bot$	$\top$
$M_a(x)$	$\top$	$\bot$
$L_a(x)$	$\bot$	$\bot$
Output:	M	F

As in the Tianjin examples, the leftmost string position satisfies the relevant licensing structure and models sandhi in isolation. Unlike Tianjin, however, the distinction between the ISL characterization in (5.37) and an equally-plausible OSL characterization is less straightforward. This is because few—in fact, only one—of the Changting disyllabic sandhi rules are of the type $xx \to yx$, often described as OCP effects (Chen, 2000). In Tianjin, the computational classification of the function becomes clear upon examining the rules in trisyllabic contexts, as ISL functions correspond to 'simultaneous' application and OSL functions to iterative application. No such test exists for the Changting rules under consideration because they are triggered by sequences of different tones. Thus there is no way to determine whether the associated function is properly-ISL or properly-OSL; perhaps it occupies the space in which these function classes overlap. For simplicity, the analysis presented here will assume ISL-ness of individual sandhi rules, unless clear evidence from the data suggests otherwise.

The RM rule (RM $\to$ HM) is described by the BMRS system of equations denoted b, and given in (5.39).

(5.39)
$$
\begin{aligned}
H_b(x) &= \texttt{if } \underline{R}M(x) \texttt{ then } \top \texttt{ else } H(x)\\
R_b(x) &= \texttt{if } \underline{R}M(x) \texttt{ then } \bot \texttt{ else } R(x)\\
F_b(x) &= F(x)\\
M_b(x) &= M(x)\\
L_b(x) &= L(x)
\end{aligned}
$$

Computation of an input string /RM/ using this definition follows a similar procedure as in (5.38). The first position evaluates to true for the term $\underline{R}M(x)$ (an input R followed immediately by an input M). This reflects the generalization that an output H tone is licensed in this position and an output R is blocked.

The same generalization holds of a system c describing the MR rule (MR $\rightarrow$ LR), and given in (5.40).

$$
\begin{aligned}
(5.40) \quad H_c(x) &= H(x) \\
R_c(x) &= R(x) \\
F_c(x) &= F(x) \\
M_c(x) &= \texttt{if } \underline{M}R(x) \texttt{ then } \bot \texttt{ else } M(x) \\
L_c(x) &= \texttt{if } \underline{M}R(x) \texttt{ then } \top \texttt{ else } L(x)
\end{aligned}
$$

Applied to an input string /MR/, the function outputs the attested surface string [LR]. The evaluation table in (5.41) shows that the mapping /MR/ $\mapsto$ [LR] satisfies system c.

(5.41)

Input:	M	R
	1	2
$H_c(x)$	$\top$	$\bot$
$R_c(x)$	$\bot$	$\top$
$F_c(x)$	$\bot$	$\bot$
$M_c(x)$	$\bot$	$\bot$
$L_c(x)$	$\bot$	$\bot$
Output:	L	R

The input string satisfies the structure $\underline{M}R(x)$, thus licensing an output L tone on position 1 and blocking output M on the same position. As before, the string position representing the non-alternating tone (input R in this case) maps directly to an output R by virtue of returning a true value for $R_c(x)$. The function described by system c is assumed to be ISL; the discussion section presents evidence in support of an ISL characterization of both RM *and* MR rules as a result of their interaction. This issue is explored in more detail in Chapter 6.

Unlike LF, RM, and MR rules, the trisyllabic sandhi data in Changting motivate a non-ISL characterization of the ML rule (ML $\rightarrow$ LL). Luo (1982) reports the mapping /MML/ $\mapsto$ [LLL] as in (5.42):

(5.42) a. seŋ^M fa^M joŋ^L $\rightarrow$ seŋ^L fa^L joŋ^L ‘starting fresh’

b. soŋ^M pa^M lo^L $\rightarrow$ soŋ^L pa^L lo^L ‘twins’

These forms illustrate iterative application, suggesting that this rule is describable by an OSL function. Intuitively, access to output structure is necessary to generate the mappings above. If computation of the rule's output were to only make reference to a local bounded window in the input, for example, the L tone created by application of the rule on the second and third tones would not trigger another iteration. In this case, the predicted output would be the unattested MML. An ROSL function, by contrast, calculates the output using a local bounded window in the output structure, thus giving it access to the output L tone. Thinking in terms of rules (and not functions), ML *feeds itself* in a sense, deriving the attested output [LLL]. The derivations in (5.43) schematize the difference between ISL and OSL computations of an /MML/ input string. Underlined portions represent the window of computation.

(5.43) a. ISL function:

$$\begin{array}{ccc} \mathrm{MM\underline{L}} & \mathrm{M\underline{ML}} & \mathrm{\underline{MM}L} \\ \downarrow & \downarrow & \downarrow \\ \mathrm{L} & \mathrm{LL} & \mathrm{MLL} \end{array}$$

b. OSL function:

$$\begin{array}{cccc} \mathrm{MM\underline{L}} & \mathrm{M\underline{M}L} & \mathrm{\underline{M}ML} & \mathrm{MML} \\ \downarrow & \downarrow & \downarrow & \downarrow \\ \mathrm{L} & \mathrm{\underline{L}} & \mathrm{\underline{L}L} & \mathrm{LLL} \end{array}$$

The crucial structure for the OSL functional characterization, then, is an *input* M followed immediately by an *output* L (recall the FF and LL rules from Tianjin). This is reflected in the system of equations d in (5.44).

(5.44)

$$\begin{array}{rcl} H_d(x) & = & H(x) \\ R_d(x) & = & R(x) \\ F_d(x) & = & F(x) \\ M_d(x) & = & \texttt{if}\ \underline{M}L_d(x)\ \texttt{then}\ \bot\ \texttt{else}\ M(x) \\ L_d(x) & = & \texttt{if}\ \underline{M}L_d(x)\ \texttt{then}\ \top\ \texttt{else}\ L(x) \end{array}$$

The OSL-ness of the system is reflected in the licensing/blocking structure pair $\underline{M}L_d(x)$. The current string position (underlined) refers to the input, and other local string information refers to the output (a recursive call of output boolean function $L_d(x)$). System d makes accurate predictions about trisyllabic sandhi forms in (5.42). Example (5.45) illustrates its evaluation of input string /MML/ mapping to output [LLL].

(5.45)

Input:	M	M	L
	1	2	3
$H_d(x)$	$\bot$	$\bot$	$\bot$
$R_d(x)$	$\bot$	$\bot$	$\bot$
$F_d(x)$	$\bot$	$\bot$	$\bot$
$M_d(x)$	$\bot$	$\bot$	$\bot$
$L_d(x)$	$\top$	$\top$	$\top$
Output:	L	L	L

Computation of the output string [LLL] with this system is identical to the procedure described in (5.42b). Position 3 maps to L by virtue of input-specification as L (satisfying the final term of $L_d(x)$). Given this, position 2 satisfies $\underline{M}L_d(x)$, licensing an output L. In precisely the same manner, the output structure contributed by position 2 permits the licensing of position 1 as an output L tone.

Individual systems a, b, c, and d model the LF, RM, MR, and ML disyllabic sandhi rules in Changting, respectively. These systems describe both ISL and OSL functions. Attested feeding interactions between these rules is formalized in a BMRS framework via the composition operator $\otimes$.

5.4.2 LF feeds RM (ISL $\otimes$ ISL)

The following two sections model feeding interactions in Changting as the composition of individual BMRS systems. Combination of two systems in a particular order via $\otimes$ recreates the effect of a feeding order over rules. Recall the feeding interaction between LF and RM rules in (5.36a), repeated below in (5.46):

(5.46) LF rule feeds RM rule: R<u>LF</u> $\rightarrow$ <u>R**M**</u>F $\rightarrow$ **H**MF

In terms of SPE rules, the crucial ordering LF < RM is necessary to derive the mapping /RLF/ $\mapsto$ [HMF]. The LF rule applies first, creating the intermediate representation [RMF]. This string satisfies the structural description of the RM rule. It applies to the intermediate form to yield the surface form [HMF]. Earlier application of the LF rule is crucial as it imparts the requisite environment (an M tone) for the RM rule to apply. A functional characterization of the feeding interaction describes a similar procedure: an LF function applies first, and its output becomes the input of an RM function. This relationship is formalized in a BMRS framework using function composition. With systems a and b describing systems LF and RM respectively, the composite function $b \otimes a$ models the same rule ordering LF < RM and thus transparent feeding. Example (5.47) below gives this definition, in addition to individual system a.

(5.47) a. $b \otimes a$

$$\begin{aligned} H_b(x) &= \texttt{if } \underline{R}_a M_a(x) \texttt{ then } \top \texttt{ else } H_a(x) \\ R_b(x) &= \texttt{if } \underline{R}_a M_a(x) \texttt{ then } \bot \texttt{ else } R_a(x) \\ F_b(x) &= F_a(x) \\ M_b(x) &= M_a(x) \\ L_b(x) &= L_a(x) \end{aligned}$$

b. a

$$\begin{aligned} H_a(x) &= H(x) \\ R_a(x) &= R(x) \\ F_a(x) &= F(x) \\ M_a(x) &= \texttt{if } \underline{L}F(x) \texttt{ then } \top \texttt{ else } M(x) \\ L_a(x) &= \texttt{if } \underline{L}F(x) \texttt{ then } \bot \texttt{ else } L(x) \end{aligned}$$

Applying the $\otimes$ operator proceeds as before. All non-recursively-defined labeling predicates in the definitions of system b (the outer function) are indexed with corresponding output boolean function definitions from system a (the inner function). And since a describes an ISL function, every labeling predicate contains such an index. This composite system is also ISL, since all the indexed function names also point to function names that are themselves non-recursively-defined. In other words, the function described by $b \otimes a$ computes outputs using bound reference to input structure only.

Given the order of composition $b \otimes a$, the resulting system is predicted to model the feeding relationship between LF and RM rules. It does exactly that, as shown by the following evaluation of /RLF/ $\mapsto$ [HMF] against the system.

(5.48)

Input:	R	L	F
	1	2	3
$H_b(x)$	$\top$	$\bot$	$\bot$
$R_b(x)$	$\bot$	$\bot$	$\bot$
$F_b(x)$	$\bot$	$\bot$	$\top$
$M_b(x)$	$\bot$	$\top$	$\bot$
$L_b(x)$	$\bot$	$\bot$	$\bot$
$H_a(x)$	$\bot$	$\bot$	$\bot$
$R_a(x)$	$\top$	$\bot$	$\bot$
$F_a(x)$	$\bot$	$\bot$	$\top$
$M_a(x)$	$\bot$	$\top$	$\bot$
$L_a(x)$	$\bot$	$\bot$	$\bot$
Output:	H	M	F

String position 3 returns a true value for $F_a(x)$ (as an input-specified F) which passes directly to $F_b(x)$. Along with $F_b(x)$, $M_b(x)$ and $L_b(x)$ also inherit their truth values directly from $M_a(x)$ and $L_a(x)$, respectively. This means that when string position 2 satisfies the licensing structure $\underline{L}F(x)$—

an *input* LF sequence—in the definition $M_a(x)$, it also evaluates to true for $M_b(x)$ and is output as M.

Importantly, the licensing/blocking structure pair for the modified system b entails evaluation from system a over two string positions (recalling that b describes an ISL function). Returning a true value for $\underline{R}_a M_a(x)$ requires the current string position to be true for $R_a(x)$ and for its immediate successor to be true for $M_a(x)$. String position 1 returns true $\top$ for $R_a(x)$—an input R—and, as shown in the preceding paragraph, position 2 returns a true value for $M_a(x)$. When the input R is evaluated against $H_b(x)$, it returns a true value based entirely on information from the output of system a. Part of that output is the M tone contributed by application of the LF rule, and which crucially allows positions 1 and 2 to satisfy the licensing structure $\underline{R}_a M_a(x)$. This interplay between the functions described by a and b, and mediated through the syntactic operator $\otimes$, is identical to transparent feeding in a rule-ordering paradigm. Despite the fact that a, b, and $b \otimes a$ are all ISL, the feeding relationship is represented in much the same way as it is for the composition of OSL Tianjin feeding interactions discussed in §3.3.

5.4.3 MR feeds ML (OSL $\otimes$ ISL)

The typology developed in §2.3 predicts as possible interactions—formalizable by rule ordering—compositions of two ISL functions (the previous subsection), compositions of two OSL functions (Tianjin FF/LL feeding in §3.3), and the composition of an OSL function with an ISL function (Tianjin LL/RR feeding in §3.2). The feeding relationship between MR and ML rules in Changting thus exhausts the possible combinations from the typology: here, an ISL function composes with an OSL function.

Recall the feeding relationship between these two rules (5.36b), repeated below in (5.49).

(5.49) MR feeds ML rule: M$\underline{\text{MR}}$ $\rightarrow$ $\underline{\text{M}\textbf{L}}$R $\rightarrow$ **L**LR

To derive the surface form [LLR] from underlying /MMR/, the MR rule must be ordered before ML. As in the previous section, application of the latter rule hinges on the appearance of the sequence ML in the intermediate form MLR (furnished by earlier application of MR).

This transparent feeding interaction is formalized as the composition of system d with system c ($d \otimes c$), recalling that d is a BMRS system of equations describing the ML rule and system c the BMRS system of equations describing the MR rule. The output boolean function definitions of system d are modified such that they are indexed with definitions from c wherever a non-recursive labeling predicate appears. This is given below in (5.50) along with the original system c for reference.

(5.50) a. $d \otimes c$

$$
\begin{aligned}
H_d(x) &= H_c(x) \\
R_d(x) &= R_c(x) \\
F_d(x) &= F_c(x) \\
M_d(x) &= \texttt{if } \underline{M}_c L_d(x) \texttt{ then } \bot \texttt{ else } M_c(x) \\
L_d(x) &= \texttt{if } \underline{M}_c L_d(x) \texttt{ then } \top \texttt{ else } L_c(x)
\end{aligned}
$$

b. c

$$
\begin{aligned}
H_c(x) &= H(x) \\
R_c(x) &= R(x) \\
F_c(x) &= F(x) \\
M_c(x) &= \texttt{if } \underline{M}R(x) \texttt{ then } \bot \texttt{ else } M(x) \\
L_c(x) &= \texttt{if } \underline{M}R(x) \texttt{ then } \top \texttt{ else } L(x)
\end{aligned}
$$

Worth noting is the fact that definitions $M_d(x)$ and $L_d(x)$ now contain a modified licensing/blocking structure pair: $\underline{M}_c L_d(x)$. Computing a truth value of this term requires evaluation from system c (on a current string position) and from system d (on the following string position, recalling that d describes an ROSL function). Other labeling predicates directly index system c.

As defined, the composite system accepts the attested interaction mapping /MMR/ $\mapsto$ [LLR] in the Changting data. An evaluation of this mapping is summarized below in (5.51).

(5.51)

Input:	M	M	R
	1	2	3
$H_d(x)$	$\bot$	$\bot$	$\bot$
$R_d(x)$	$\bot$	$\bot$	$\top$
$F_d(x)$	$\bot$	$\bot$	$\bot$
$M_d(x)$	$\bot$	$\bot$	$\bot$
$L_d(x)$	$\top$	$\top$	$\bot$
$H_c(x)$	$\bot$	$\bot$	$\bot$
$R_c(x)$	$\bot$	$\bot$	$\top$
$F_c(x)$	$\bot$	$\bot$	$\bot$
$M_c(x)$	$\top$	$\bot$	$\bot$
$L_c(x)$	$\bot$	$\top$	$\bot$
Output:	L	L	R

Position 3, input-specified as R, maps directly to R in the output (i.e. it evaluates to true for $R_d(x)$ via $R_c(x)$). The second string position satisfies the relevant structure for the MR rule, returning a true value for $L_c(x)$ as is expected for the rule's application in isolation. Crucially, it returns a false $\bot$ value for $M_c(x)$; thus when computing the truth value for $M_d(x)$ and $L_d(x)$, it returns false for the modified licensing/blocking structure $\underline{M}_c L_d(x)$. It then proceeds to the third term in the if-then-else statement. The truth value for $L_c(x)$ is 'fed' to $L_d(x)$, outputting string position 2 as L via this default term.

Position 1 evaluates to true for $M_c(x)$, also via input M-specification. In conjunction with position 2's value for $L_d(x)$, the input M in position 1 conforms to the structure $\underline{M}_c L_d(x)$—blocking output M and licensing output L. The 'feeding' apparent in this evaluation stems from the fact $L_d(x)$'s truth value comes directly from $L_c(x)$. In other words, system c contributes an 'output' L tone to the evaluation of system d. This is reflected in the composite function $d \otimes c$, and the function accepts mappings consistent with attested Changting interactions.

5.5 Nanjing

Chapter 2 (§3.1) introduced di- and tri-syllabic sandhi data from the Jianghuai Mandarin dialect Nanjing (Liu and Li, 1995). Recall that this dialect has five lexical tones: H(igh), L(ow), R(ising), F(alling) and a C(hecked) tone. Relevant sandhi alternations are repeated in (5.52).

(5.52) a. 'LF rule': LF $\mapsto$ RF e.g. $lao^L shi^F \rightarrow lao^R shi^F$ 'teacher'

b. 'FF rule': FF $\mapsto$ HF e.g. $bing^F xiang^F \rightarrow bing^H xiang^F$ 'refrigerator'

c. 'RC rule': RC $\mapsto$ LC e.g. $tong^R xue^C \rightarrow tong^L xue^C$ 'classmate'

d. 'CC rule': CC $\mapsto$ C′C e.g. $qi^C shi^C \rightarrow qi^{C'} shi^C$ 'seventy'

A recent instrumental study by Ma and Li (2014) provides experimental evidence for sandhi interactions in trisyllabic forms and explores the issue of directionality in sandhi application—echoing the left-to-right and right-to-left scanning dichotomy that led to the Tianjin 'paradox'. Among other findings, their results indicate that the trisyllabic paradigm includes both counterbleeding and counterfeeding interactions. The counterbleeding mappings relevant to this chapter are introduced in (5.53).

(5.53) a. /LFF/ $\mapsto$ [RHF]

b. /RCC/ $\mapsto$ [LC′C]

Mappings (5.53a) and (5.53b) show a counterbleeding (on environment) interaction between FF/LF rules and RC/CC rules, respectively.

This section presents a BMRS analysis of counterbleeding interactions in Nanjing. Its primary purpose is to demonstrate that function composition (via the $\otimes$ operator) offers a formalization of rule ordering beyond transparent feeding to include opaque interactions such as counterbleeding. This lends further support to the argument that the set of interactions expressible by pairwise rule ordering is precisely the set of interactions modeled by function composition. Individual rules are defined as BMRS systems of equations describing ISL or OSL functions, which can be composed to model counterbleeding relationships.

Analysis in terms of compositions of strictly-local functions presents a less stipulative account of the data than the directionality-based analyses pursued by Ma and Li (2014) and others (recall discussion in Chapter 2 §3.4.2). The apparent paradox in directionality—much like Chandlee (2019)'s account of Tianjin and the analysis of Changting in (Oakden and Chandlee, 2020)—vanishes when examined through the lens of ISL and OSL functions and composite functions thereof. Nanjing therefore offers additional support to SL-functional analyses of tone sandhi interactions more generally.

5.5.1 Individual rules as systems of equations

Counterbleeding interactions in Nanjing can be formalized as compositions of individual BMRS systems of equations. Those systems—modeling the individual rules in (5.52)—are first defined, and are shown to make correct predictions about disyllabic sandhi. Data from Ma and Li (2014)'s findings support describing rules as either ISL or OSL functions. Note that systems are defined over an output alphabet of surface sandhi tones in the Nanjing dialect $\Sigma = \{H, L, R, F, C, C'\}$. Crucially, this includes the *output sandhi tone* C′.

First, let a system a describe the LF rule (5.52a), as defined in (5.54). The relevant structure here is an input L tone followed by an input F tone. As before, this definition assumes an ISL function as a default.

(5.54)
$$\begin{aligned}
H_a(x) &= H(x)\\
L_a(x) &= \texttt{if } \underline{L}F(x) \texttt{ then } \bot \texttt{ else } L(x)\\
R_a(x) &= \texttt{if } \underline{L}F(x) \texttt{ then } \top \texttt{ else } R(x)\\
F_a(x) &= F(x)\\
C_a(x) &= C(x)\\
C'_a(x) &= \bot
\end{aligned}$$

Applied to a disyllabic input string /LF/, the system accepts the attested disyllabic sandhi mapping /LF/ $\mapsto$ [RF] as in (5.55); string position 1 satisfies the structure $\underline{L}F(x)$, licensing an output R and blocking output L on that position. String position 2 maps directly to an output F.

(5.55)

Input:	L	F
	1	2
$H_a(x)$	$\perp$	$\perp$
$L_a(x)$	$\perp$	$\perp$
$R_a(x)$	$\top$	$\perp$
$F_a(x)$	$\perp$	$\top$
$C_a(x)$	$\perp$	$\perp$
$C'_a(x)$	$\perp$	$\perp$
Output:	R	F

Note also that, for this system, the output boolean function for the checked sandhi variant is set to false.

Let b, defined in (5.56), denote a system of equations modeling the FF rule (5.52b). A single structure—two consecutive, input F tones—licenses an output H tone and blocks an output F tone, and thus the system describes an ISL function.

(5.56)
$$
\begin{aligned}
H_b(x) &= \texttt{if } \underline{F}F(x) \texttt{ then } \top \texttt{ else } H(x) \\
L_b(x) &= L(x) \\
R_b(x) &= R(x) \\
F_b(x) &= \texttt{if } \underline{F}F(x) \texttt{ then } \perp \texttt{ else } F(x) \\
C_b(x) &= C(x) \\
C'_b(x) &= \perp
\end{aligned}
$$

Unlike the LF rule, the FF rule is of the type $xx \mapsto xy$, and thus whether it is properly-ISL or properly-OSL can be determined from trisyllabic contexts, as was the case for Tianjin sandhi. Ma and Li (2014) report the mapping /FFF/ $\mapsto$ [HHF] for this rule—i.e. simultaneous application. Following Chandlee (2019), this supports an ISL designation. As defined, system b accepts the mapping /FFF/ $\mapsto$ [HHF], as illustrated in the evaluation table in (5.57).

(5.57)

Input:	F	F	F
	1	2	3
$H_b(x)$	$\top$	$\top$	$\perp$
$L_b(x)$	$\perp$	$\perp$	$\perp$
$R_b(x)$	$\perp$	$\perp$	$\perp$
$F_b(x)$	$\perp$	$\perp$	$\top$
$C_b(x)$	$\perp$	$\perp$	$\perp$
$C'_b(x)$	$\perp$	$\perp$	$\perp$
Output:	H	H	F

Crucially, both string positions 1 and 2 return a true value for the term $\underline{F}F(x)$, licensing output H and blocking output F on these positions. System b contains no recursive function calls, and thus computing the output proceeds solely with reference to *input* structure—that is, it describes an ISL function. This characteristic permits the same 'simultaneous' application of the FF rule in Nanjing as was observed for the ISL RR rule in Tianjin (recall the evaluation of the relevant system in (5.26)).

The RC rule in (5.52c) can be described by a system of equations denoted c. It is defined in (5.58) below, and receives the same default ISL-ness assumption as the LF rule.

(5.58)
$$\begin{aligned}
H_c(x) &= H(x)\\
L_c(x) &= \texttt{if } \underline{R}C(x) \texttt{ then } \top \texttt{ else } L(x)\\
R_c(x) &= \texttt{if } \underline{R}C(x) \texttt{ then } \bot \texttt{ else } R(x)\\
F_c(x) &= F(x)\\
C_c(x) &= C(x)\\
C'_c(x) &= \bot
\end{aligned}$$

A single structure—a sequence of a rising and a checked tone in the input—licenses an output L and blocks an output R. Applied to a disyllabic sequence /RC/, then, the first string position satisfies this structure, predicting the attested [LC] surface structure.

Like the FF rule, the CC rule's designation as an ISL or OSL function can be determined by its application over sequences of three (or more) tones. Recall Ma and Li (2014)'s claim from the previous section of the right-to-left application of this rule, evidenced by the attested mapping /CCC/ $\mapsto$ [CC′C] and crucially not *[C′C′C] (the outcome of a left-to-right parse). In terms of ISL and OSL functions, this is reminiscent of the the 'iterative' application of LL and FF rules in Tianjin—e.g. /LLL/ $\mapsto$ [LRL] and /FFF/ $\mapsto$ [FLF]. The CC rule (5.52d) can therefore be defined as a system of equations—denoted d as in (5.59)—describing an ROSL function, consistent with the characterization of Tianjin presented in §3 of this chapter.

(5.59)
$$\begin{aligned}
H_d(x) &= H(x)\\
L_d(x) &= L(x)\\
R_d(x) &= R(x)\\
F_d(x) &= F(x)\\
C_d(x) &= \texttt{if } \underline{C}C_d(x) \texttt{ then } \bot \texttt{ else } C(x)\\
C'_d(x) &= \texttt{if } \underline{C}C_d(x) \texttt{ then } \top \texttt{ else } \bot
\end{aligned}$$

The function described by d computes outputs using the current input string position and a bounded local window in the output and to the right of that string position. System d accepts the attested /CCC/ $\mapsto$ [CC′C] but would not accept the unattested (simultaneous application) /CCC/ $\mapsto$ *[C′C′C] describable by an ISL function. An abbreviated evaluation table in (5.60) illustrates.

(5.60)

Input:	C	C	C
	1	2	3
$C_d(x)$	$\top$	$\bot$	$\top$
$C'_d(x)$	$\bot$	$\top$	$\bot$
Output:	C	C′	C

Only string position 2 satisfies the licensing structure $\underline{C}C_d(x)$ in the definition of $C'_d(x)$; an input C tone whose immediate successor is an output C. Positions 1 and 3 return a false value for this term, and thus map directly to C via the default term $C(x)$ in the definition $C_d(x)$.

5.5.2 LF counterbleeds FF (ISL $\otimes$ ISL)

Composition of individual BMRS systems (using the $\otimes$ operator) models counterbleeding orders in the same manner as feeding orders. This is because order of composition recreates the effect of a crucial rule ordering: the output of one function (the earlier rule) serves as the input of another function (the later rule).

In a trisyllabic environment, the Nanjing LF rule counterbleeds the FF rule. This is apparent in the mapping /LFF/ $\mapsto$ [RHF], and shown in the derivation in (5.61).

(5.61)

$\underline{\text{LF}}$F		L$\underline{\text{FF}}$	
|		|	
R$\underline{\text{FF}}$	by LF rule	LHF	by FF rule
|		|	
RHF	by FF rule	*LHF	LF rule n/a

Deriving the correct surface form requires the order LF < FF. A BMRS compositional account recreates this effect with the composite system $b \otimes a$, recalling that a system b describes the Nanjing FF rule and a the LF rule, respectively. In (5.62) below, the composite system is given along with the original system a.

(5.62) a. $b \otimes a$

$$
\begin{aligned}
H_b(x) &= \texttt{if } \underline{F}_a F_a(x) \texttt{ then } \top \texttt{ else } H_a(x)\\
L_b(x) &= L_a(x)\\
R_b(x) &= R_a(x)\\
F_b(x) &= \texttt{if } \underline{F}_a F_a(x) \texttt{ then } \bot \texttt{ else } F_a(x)\\
C_b(x) &= C_a(x)\\
C'_b(x) &= \bot
\end{aligned}
$$

b. a

$$
\begin{aligned}
H_a(x) &= H(x)\\
L_a(x) &= \texttt{if } \underline{L}F(x) \texttt{ then } \bot \texttt{ else } L(x)\\
R_a(x) &= \texttt{if } \underline{L}F(x) \texttt{ then } \top \texttt{ else } R(x)\\
F_a(x) &= F(x)\\
C_a(x) &= C(x)\\
C'_a(x) &= \bot
\end{aligned}
$$

System $b \otimes a$ describes the composition of two ISL functions. The resulting system shares many similarities to the composition of two ISL functions modeling the LF/RM interaction in Tianjin (5.47). All labeling predicates in b are indexed with corresponding definitions from a given that b describes an ISL function (i.e. no recursive function calls). The composite function is also ISL.

The composition order $b \otimes a$ models the counterbleeding order LF < FF. Crucially, it accepts the mapping /LFF/ $\mapsto$ [RHF]. This is shown in the abbreviated evaluation table below in (5.63).

(5.63)

Input:	L	F	F
	1	2	3
$H_b(x)$	$\bot$	$\top$	$\bot$
$L_b(x)$	$\bot$	$\bot$	$\bot$
$R_b(x)$	$\top$	$\bot$	$\bot$
$F_b(x)$	$\bot$	$\bot$	$\top$
$H_a(x)$	$\bot$	$\bot$	$\bot$
$L_a(x)$	$\bot$	$\bot$	$\bot$
$R_a(x)$	$\top$	$\bot$	$\bot$
$F_a(x)$	$\bot$	$\top$	$\top$
Output:	R	H	F

Thinking of in terms of rule 'application', the order of composition $b \otimes a$ guarantees that the FF rule (system b) applies to the output of the LF rule (system a). In the string /LFF/, early 'application' of the LF rule does not modify the structural description of the FF rule: the substring FF. In the evaluation table above, string positions 2 and 3 both return true values for the function $F_a(x)$ in system a. That is to say, modification of the input string imposed by system a does not extend to those positions. That output feeds system b directly via indices on its labeling predicates. String position 2 returns a true value for the term $\underline{F}_a F_a(x)$ in the definition of $H_b(x)$, licensing an output H (i.e. the application of the FF rule). That structure is dependent on any modification made by system a (i.e. application of the LF rule), of which there is none for positions 2 and 3. The counterbleeding interaction in Nanjing is thus captured in a BMRS compositional framework.

Both systems describe ISL functions. The interaction is describable as a composition of ISL functions (each of which corresponds to simultaneous application), provided that the correct order of composition is maintained. This diverges somewhat from Ma and Li (2014)'s directionality-based approach. Specifically, they claim that a left-to-right parse derives /LFF/ ↦ [RHF], while a right-to-left parse derives an unattested output. Example (5.64) outlines the purported predictions of a 'right-to-left' parse.

(5.64)

Input	Right-to-left	Left-to-right
/LFF/	$\mathrm{L}\underline{\mathrm{FF}} \to {}^{*}\mathrm{L}\mathbf{H}\mathrm{F}$	$\underline{\mathrm{LF}}\mathrm{F} \to \mathbf{R}\underline{\mathrm{FF}} \to \mathrm{R}\mathbf{H}\mathrm{F}$

Differences in parsing 'direction' according to Ma and Li (2014) actually correspond to different rule orderings (and thus composition orders) The right-to-left parse follows the bleeding order FF<LF, while the left-to-right parse follows the (attested) counterbleeding order LF<FF. The analysis presented here makes the same distinction between attested and unattested outputs, but does so via an order of composition over ISL functions. Much like the analysis of Tianjin, this avoids arbitrary stipulation of a parsing direction.

5.5.3 RC counterbleeds CC (OSL $\otimes$ ISL)

The attested counterbleeding interaction between RC and CC rules is also amendable to an analysis using composition of BMRS systems with the $\otimes$ operator. Recall the relevant interaction mapping /RCC/ ↦ [LC′C] in (5.53), with corresponding derivations below in (5.65):

(5.65)

$\underline{\mathrm{RC}}\mathrm{C}$		$\mathrm{R}\underline{\mathrm{CC}}$	
|		|	
$\mathrm{L}\underline{\mathrm{CC}}$	by RC rule	RC′C	by CC rule
|		|	
LC′C	by CC rule	*RC′C	RC rule n/a

The order RC < CC is apparent from the derivations above. In a BMRS framework, individual systems d and c—describing CC and RC rules respectively—compose to model this interaction. Example (5.66) defines the composed system $d \otimes c$, with system c repeated.

(5.66) a. $d \otimes c$

$$\begin{aligned}
H_d(x) &= H_c(x)\\
L_d(x) &= L_c(x)\\
R_d(x) &= R_c(x)\\
F_d(x) &= F_c(x)\\
C_d(x) &= \texttt{if } \underline{C}_c C_d(x) \texttt{ then } \bot \texttt{ else } C_c(x)\\
C'_d(x) &= \texttt{if } \underline{C}_c C_d(x) \texttt{ then } \top \texttt{ else } \bot
\end{aligned}$$

b. c

$$\begin{aligned}
H_c(x) &= H(x)\\
L_c(x) &= \texttt{if } \underline{R}C(x) \texttt{ then } \top \texttt{ else } L(x)\\
R_c(x) &= \texttt{if } \underline{R}C(x) \texttt{ then } \bot \texttt{ else } R(x)\\
F_c(x) &= F(x)\\
C_c(x) &= C(x)\\
C'_c(x) &= \bot
\end{aligned}$$

Note that d describes an ROSL function as per the licensing/blocking structure pair in definitions $C_d(x)$ and $C'_d(x)$. Composing with system c results in the following modified structure: $\underline{C}_c C_d(x)$. Much like the composition of OSL and ISL functions introduced for Changting (see §4.2), this term is calculated by evaluating both the output of system c (on the current string position) and the output of system d (on the current string position's successor).

As (5.67) shows, this system accepts the relevant mapping /RCC/ ↦ [LC′C].

(5.67)

Input:	R	C	C
	1	2	3
$C'_d(x)$	$\bot$	$\top$	$\bot$
$C_d(x)$	$\bot$	$\bot$	$\top$
$L_d(x)$	$\top$	$\bot$	$\bot$
$R_d(x)$	$\bot$	$\bot$	$\bot$
$L_c(x)$	$\top$	$\bot$	$\bot$
$R_c(x)$	$\bot$	$\bot$	$\bot$
$C_c(x)$	$\bot$	$\top$	$\top$
Output:	L	C′	C

Evaluating string position 3 against $C_d(x)$ returns a false $\bot$ value for the modified licensing/blocking structure, then proceeds to the final term (indexed $C_c(x)$), and is thus output as C via input specification as such. The second string position does return a true value for $\underline{C}_c C_d(x)$ and, as a result, evaluates to true for $C'_d(x)$.

Note also that position 2 returns a true value for $C_c(x)$ in system c. Thus string position 1 evaluates to true for $L_c(x)$ in the 'RC rule' system c—by satisfying $\underline{R}C(x)$—and passes that truth value directly to $L_d(x)$, such that string position 1 is output as L. Here the order of composition is paramount; the opposite order $c \otimes d$ would index $L_c(x)$'s definition with definitions from system d, resulting in the licensing structure $\underline{R}_d C_d(x)$. This would effectively block (i.e. bleed) application of the RC rule. In other words, reversing the order of composition generates the same effect as reversing rule ordering.

As before, the SL-functional analysis offers a desirable alternative to one that invokes directionality. By Ma and Li (2014)'s account, the RC/CC counterbleeding interaction offers support of left-to-right application, summarized in (5.68):

(5.68)

Input	Right-to-left	Left-to-right
/RCC/	R$\underline{\text{CC}}$ → *R**C**′C	$\underline{\text{RC}}$R → **L**$\underline{\text{CC}}$ → L**C**′C

Such an analysis, similar to those posited for Tianjin, is then tasked with determining why the CC rule should apply left-to-right above but *right-to-left* in isolation. In the SL-functional analysis, on the other hand, the CC rule is describable as an ROSL function, and its interaction with the RC rule as a particular order of composition with a relevant function. It requires no stipulation about directionality, yet derives the same attested surface structures (that is, in isolation and in interaction contexts). As was also the case with the LF/FF interaction, what these competing analyses share is not claims about directionality, but instead the assumption that the application of RC—and in particular the availability of its conditioning environment—is not disturbed by application of CC. This can only be derived through an order of composition (i.e. rule order). What appears to be directionality is merely an epiphenomenon of this more basic fact (not the other way around, as Chen (2004) argues for Changting). Since Nanjing exhibits both so-called 'rightward' and 'leftward' sandhi via its interactions, a directionality account will be as equally as stipulative as previous accounts for similar effects in Tianjin and Changting. A SL-functional analysis, on the other hand, provides a unified account of purported directionality effects using the basic mechanisms of the theory. Nanjing tone sandhi interactions are describable as compositions of ISL and ROSL functions that correspond to simultaneous and iterative application, respectively.

5.6 Discussion

This chapter has defined a composition operator $\otimes$ over BMRS systems of equations. Applied to systems describing individual rules, this operator models phonological process interactions formalizable by pairwise rule ordering in an SPE framework. This includes both transparent interactions such as feeding, as well as opaque interactions such as counterbleeding. The discussion presented here offers additional support to this claim by showing that reversing composition order yields the same effect as reversing rule order. Additionally, it demonstrates the failure of composition to capture sandhi interactions presenting ordering paradoxes; given the equivalence of composition and rule ordering, a compositional BMRS analysis of such cases merely recapitulates the generalizations from rule ordering. This motivates the expansion of the set of operations over BMRS, which is taken up in the following chapter.

5.6.1 Order of composition and rule ordering

Composition of BMRS systems of equations (via the $\otimes$ operator) models the set of transparent and opaque tone sandhi interactions formalized by pairwise rule ordering. This section reinforces the equivalence of rule ordering and composition of BMRS systems by showing that reversing the order of composition makes the same predictions as reversing the order of rewrite rules. Given some composite system of two BMRS systems combined with $\otimes$, inverting the composition order changes a feeding interaction into a counterfeeding interaction, and a counterbleeding interaction into a bleeding interaction. That is to say, it produces the same effect as reversing rule ordering. Two example cases below—LL/FF feeding in Tianjin and LF/FF counterbleeding in Nanjing—offer an illustration.

First, recall the feeding interaction between LL and RR rules in Tianjin (5.20a), repeated in (5.69):

(5.69) LL rule feeds RR rule: $\mathrm{R}\underline{\mathrm{LL}} \rightarrow \underline{\mathrm{R}\mathbf{R}}\mathrm{L} \rightarrow \mathbf{H}\mathrm{RL}$

This interaction hinges on the LL rule being ordered before the RR rule; that is, the opposite order produces a counterfeeding effect, as illustrated in (5.70).

(5.70)

a. **LL < RR**		b. **RR < LL**	
/RLL/		/RLL/	
|		|	
R**R**L	LL rule	RLL	RR rule
|		|	
[**H**RL]	RR rule	*[R**R**L]	LL rule

In (5.70b), the RR rule first applies vacuously to the input string /RLL/, after which the LL rule applies to yield the unattested *[RRL]. The ordering LL < RR is thus crucial in deriving the feeding relationship.

The same generalization can be made in terms of the order of composition over BMRS systems of equations. In §3.2, the composite system $c \otimes b$ in (5.28) defined a function that describes the interaction mapping /RLL/ $\mapsto$ [HRL]. Reversing the order of composition to $b \otimes c$ produces a system of equations describing a different function. This is given in (5.71), with the original system c for reference.

(5.71) a. $b \otimes c$

$$
\begin{aligned}
H_b(x) &= H_c(x)\\
R_b(x) &= \texttt{if } \underline{L_c}L_b(x) \texttt{ then } \top \texttt{ else } R_c(x)\\
L_b(x) &= \texttt{if } \underline{L_c}L_b(x) \texttt{ then } \bot \texttt{ else } L_c(x)\\
F_b(x) &= F_c(x)
\end{aligned}
$$

b. c

$$
\begin{aligned}
H_c(x) &= \texttt{if } \underline{R}R(x) \texttt{ then } \top \texttt{ else } H(x)\\
R_c(x) &= \texttt{if } \underline{R}R(x) \texttt{ then } \bot \texttt{ else } R(x)\\
L_c(x) &= L(x)\\
F_c(x) &= F(x)
\end{aligned}
$$

The inverted order of composition represented by $b \otimes c$ mirrors the effect of ordering the LL rule after the RR rule, and thus accepts the unattested counterfeeding mapping /RLL/ $\mapsto$ *[RRL] in (5.70b). The table in (5.72) illustrates the evaluation of input string /RLL/ against the composite system $b \otimes c$.

(5.72)

Input:	R	L	L
	1	2	3
$H_b(x)$	$\bot$	$\bot$	$\bot$
$R_b(x)$	$\top$	$\top$	$\bot$
$L_b(x)$	$\bot$	$\bot$	$\top$
$F_b(x)$	$\bot$	$\bot$	$\bot$
$H_c(x)$	$\bot$	$\bot$	$\bot$
$R_c(x)$	$\top$	$\bot$	$\bot$
$L_c(x)$	$\bot$	$\top$	$\top$
$F_c(x)$	$\bot$	$\bot$	$\bot$
Output:	R	R	L

Evaluation of string positions 2 and 3 does not differ between $c \otimes b$ and $b \otimes c$. They do differ, however, on the evaluation of string position 1. To see how, recall that, in the former, an output H is licensed by the structure $\underline{R_b}R_b$, that is, the current string position and its immediate successor both return a true value for $R_b(x)$ in system b. This means that an R tone output by 'application' of the LL rule (the locus of system b) can contribute to the licensing structure in the definition of $H_c(x)$. In the latter system $b \otimes c$, however, an output H tone—$H_b(x)$, which gets its truth value directly from $H_c(x)$—is licensed by the structure $\underline{R}R$: two adjacent input R tones. Crucially, the output structure contributed or 'fed' by system b onto string position 2 is irrelevant here, as this licensing structure refers to input string positions only. The input sequence 'RL' does not conform to the structure which licenses H, thereby returning a false value for $H_c(x)$ and ultimately $H_b(x)$. Position 1 does evaluate to true for $R_b(x)$ via the default term in the definition of $R_c(x)$, and is output as R, thus producing the unattested *[RRL]. This is identical to the output derived from the rule ordering RR $<$ LL, that is, the counterfeeding order. Reversing the order of composition in a system describing a feeding interaction yields a system which describes a counterfeeding interaction.

The same generalization applies to opaque interactions, as well. Recall the mapping exhibiting an opaque counterbleeding interaction between LF and FF rules in Nanjing (5.53a), repeated in (5.73):

(5.73) LF rule counterbleeds FF rule: $\underline{\text{LF}}\text{F} \rightarrow \textbf{R}\underline{\text{FF}} \rightarrow \text{R}\textbf{H}\text{F}$

The counterbleeding interaction hinges on the LF rule being ordered before the FF rule. Example (5.74)—reiterating the generalization in (5.61)—shows that the opposite order produces a bleeding effect.

(5.74)

a. LF < FF	/LFF/		**b. FF < LF**	/LFF/	
	|			|	
	RFF	LF rule		L**H**F	FF rule
	|			|	
	[R**H**F]	FF rule		*[LHF]	LF rule

In (5.74b), the FF rule applies to the input string FF, creating the intermediate representation [LHF]. This string no longer satisfies the structural description of LF; its environment has been bled by earlier application of the FF rule. LF applies vacuously, yielding the unattested *[LHF]. The ordering LF $<$ FF is therefore crucial in deriving the counterbleeding relationship.

A BMRS characterization expresses this fact via order of composition over individual systems of equations. In §5.2, the composite system $b \otimes a$ (5.62) defined a function describing the counterbleeding interaction, and which accepts the attested mapping /LFF/ $\mapsto$ [RHF]. Reversing the order of of composition to $a \otimes b$ produces a system of equations describing a different function, and importantly one which recreates a bleeding order on FF and LF rules. The composite system $a \otimes b$ is defined below in (5.75), again with the original system b for reference.

(5.75) a. $a \otimes b$

$$\begin{array}{lcl} H_a(x) & = & H_b(x) \\ L_a(x) & = & \texttt{if } \underline{L}_b F_b(x) \texttt{ then } \bot \texttt{ else } L_b(x) \\ R_a(x) & = & \texttt{if } \underline{L}_b F_b(x) \texttt{ then } \top \texttt{ else } R_b(x) \\ F_a(x) & = & F_b(x) \\ C_a(x) & = & C_b(x) \\ C'_a(x) & = & \bot \end{array}$$

b. b

$$\begin{array}{lcl} H_b(x) & = & \texttt{if } \underline{F}F(x) \texttt{ then } \top \texttt{ else } H(x) \\ L_b(x) & = & L(x) \\ R_b(x) & = & R(x) \\ F_b(x) & = & \texttt{if } \underline{F}F(x) \texttt{ then } \bot \texttt{ else } F(x) \\ C_b(x) & = & C(x) \\ C'_b(x) & = & \bot \end{array}$$

As with the Tianjin feeding interaction, the inverted order of composition represented by $a \otimes b$ mirrors the effect of ordering the Nanjing LF rule after the FF rule, and thus the prediction is that it will accept the bleeding mapping /LFF/ $\mapsto$ *[LHF] in (5.74b). The evaluation of system $a \otimes b$ against input string /LFF/, as in (5.76), shows that this prediction is borne out.

(5.76)

Input:	L	F	F
	1	2	3
$H_a(x)$	$\bot$	$\top$	$\bot$
$L_a(x)$	$\top$	$\bot$	$\bot$
$R_a(x)$	$\bot$	$\bot$	$\bot$
$F_a(x)$	$\bot$	$\bot$	$\top$
$H_b(x)$	$\bot$	$\top$	$\bot$
$L_b(x)$	$\top$	$\bot$	$\bot$
$R_b(x)$	$\bot$	$\bot$	$\bot$
$F_b(x)$	$\bot$	$\bot$	$\top$
Output:	L	H	F

Under this order of composition, the modified licensing/blocking structure pair for system a (which describes LF rule application) is the term $\underline{L}_b F_b(x)$. This differs from the term in the original definition—$\underline{L}F(x)$—in that its locus of evaluation is no longer the input string, but rather the output of system b (or, the input string in light of modifications made to it by system b). Evaluating string position 1 clarifies the effect of this modification. While it does satisfy the original licensing structure $\underline{L}F(x)$ (an LF input sequence) which would predict an output R, it does not evaluate to true for the modified $\underline{L}_b F_b(x)$. This is because its immediate successor, string position 2, returns a false $\bot$ value for $F_b(x)$ and a true $\top$ value for $H_b(x)$—in other words, the FF rule has applied to positions 2 and 3. Since system a's evaluation of the string is dependent on system b's modifications to the input—as evidenced by $\underline{L}_b F_b(x)$—system b can 'block' the application of the rule described by system a, crucially by depriving it of the environment it needs in order to apply. The end result is the same effect observed for the reversed rule ordering FF < LF: over input /LFF/, the FF rule *bleeds* the LF rule's environment.

The examples above offer additional support to the claim that composition of individual BMRS systems mirrors rule ordering. Reversing the order of composition produces the same effects as reversing rule ordering. This generalization holds for both transparent and opaque interactions which may be modeled by pairwise rule ordering.

5.6.2 SL analysis clarifies issues of directionality

This chapter has also shown that a BMRS composition analysis of tone sandhi interactions, and in particular insights regarding so-called 'directionality' effects, preserves the valuable insights from earlier decomposed SL analyses. They provide a less stipulative alternative to explanations in terms of 'rightward' or 'leftward' scans of input tonal strings. Analyses presented in this chapter preserve earlier observations that ISL functions correspond to simultaneous application, and OSL correspond to iterative application. These generalizations scale up to compositions of ISL and OSL functions, with the understanding that interactions can be described using subsequential functions. These generalizations are not stipulations about directionality but rather define the basic properties of SL function subclasses. Therefore, the BMRS composition analyses put forth here—like recent SL analyses using other formalisms (Chandlee, 2019; Oakden, 2019a; Oakden and Chandlee, 2020)—clarify the nature of 'directionality' effects in tone sandhi interactions using basic principles of the theory.

This is not apparent in earlier rule-based and optimization-based frameworks. For example, a sizeable body of work in both formalisms has been devoted to offering a unified account of the Tianjin paradigm. In rule-based accounts, directionality of rule application is either an *ad hoc* stipulation (Zhang, 1987; Tan, 1987) or is achieved in conjunction with phonotactic constraints against certain sequences of tones (Hung, 1987). OT accounts have attempted to explain directionality in a number of ways. Chen (2000) offers an account in terms of the interplay between a 'default' parsing direction (left-to-right) and other markedness pressures (like the OCP). Lin (2008) appeals to prosodic correspondence, while Wee (2010) uses tree structures to represent derivational histories, introducing the notion of inter-tier correspondence. However, these analyses suffer from the same arbitrariness as SPE accounts. For one, Chen (2000, 110)'s default left-to-right application, governed by a constraint TEMPORAL SEQUENCE and which mirrors "planning and execution of speech", is of the same stipulative nature as rule-based accounts. The nature of the OCP constraints proposed for the analysis casts doubt on whether Tianjin sandhi patterns are in fact OCP effects. Chen (2000, 123), for example, defines three separate OCP constraints:

(5.77) a. OCP- no adjacent identical tones (except HH)

b. OCP′- no *FL (=H<u>L.L</u>) sequences

c. OCP″- no adjacent partially identical tones (*<u>L.L</u>H, *<u>H.H</u>L, *H<u>L.L</u>H, etc.)

These constraints again appear as an attempt to coerce a general principle into fitting an empirical generalization. An exception (HH) is coded directly into the general OCP constraint, and it is unclear what motivates separating the ill-formed sub-melody 'H<u>L.L</u>' from the others into its own constraint OCP′, other than to subsume the observed sandhi patterns under the general category of OCP.

Attempts to explain directionality effects in an OT framework also give rise to questionable extensions of the theory, underlying the inherent evasiveness of an account using basic principles. Chen (2000, 2004) and others present OT accounts of Tianjin and Changting in which candidates are derivational histories, not surface representations. This approach is unorthodox (even by Chen (2004)'s own admission) in OT analyses, and essentially abandons a central tenet of classic OT—parallel evaluation—to fit a serial outlook into a non-serial framework.

Ma and Li (2014)'s preliminary analysis of Nanjing tone sandhi demonstrates that the question of directionality is not isolated to well-known cases like Tianjin and Changting. Their conclusion, that the Nanjing paradigm is a 'hybrid' system of left-to-right and right-to-left sandhi application, would present a similar problem to rule-based and optimization frameworks. As with the classic examples, no clear explanation using basic principles is available, and so resorting to stipulations and questionable extensions to the theory becomes necessary. A BMRS composition analysis using SL functions, on the other hand, provides a straightforward account of the Nanjing data and reflects the basic properties of ISL and OSL functions (and their composition).

5.6.3 Composition recapitulates ordering paradoxes

Composition in a BMRS framework makes the same generalizations about interactions as rule ordering in an SPE framework. This extends to interactions which *cannot* be described via rule ordering, for example ordering paradoxes. Such cases pose the same challenge to a compositional BMRS analysis. Given the equivalence between composition and rule ordering developed in this chapter, the expectation is that BMRS composition analyses of ordering paradoxes merely recapitulate the facts from a rule-based account. This section shows that this is the case for an ordering paradox apparent in the Changting data.

Chen et al. (2004) report an interaction between the RM (RM → HM as in (5.35b)) and MR (MR → LR as in (5.35c)) rules in Changting. Two relevant mappings of trisyllabic forms are given in (5.78); like the feeding interactions described in §4, morphological structure is irrelevant to rule application (Chen, 2004, p. 803).

(5.78)

Input	Output	[x x] x	x [x x]
MRM	LHM	*[hua.qian] duo*	*xin [bei.jing]*
		"spending a lot of money"	"new Beijing"
RMR	HLR	*[yi.jin] you*	*chu [feng.tou]*
		"one catty of oil"	"to show off"

The trisyllabic form /MRM/ surfaces as [LHM]; this lends support to ordering the MR rule before the RM rule. In the derivation below in (5.79), only the rule ordering MR < RM derives the correct surface form.

(5.79)

<u>MR</u>M		M<u>RM</u>	
|		|	
L<u>RM</u>	by MR rule	MHM	by RM rule
|		|	
LHM	by RM rule	*MHM	MR rule n/a

This interaction is one of counterbleeding; earlier application of the RM rule would bleed the application of the MR rule by destroying part of its environment (crucially the R). However, the other trisyllabic mapping /RMR/ ↦ [HLR] requires the opposite ordering. This is given in (5.80).[8]

(5.80)

<u>RM</u>R		R<u>MR</u>	
|		|	
H<u>MR</u>	by RM rule	RLR	by MR rule
|		|	
HLR	by MR rule	*RLR	RM rule n/a

Here, the opposite ordering (RM < MR) is necessary to derive the attested output [HLR]. As before, it is a case of counterbleeding; application of the MR rule destroys the necessary environment

[8] As Chen (2004) argues, this ordering would also feed a different 'RL rule' (RL → RF) to produce the unattested *[RFR]. I abstract away from this detail.

(crucially the M tone) for the RM rule. Thus each rule counterbleeds the other, with the consequence being that no pairwise order of these rules can derive the attested forms. The RM and MR rules present an ordering paradox.

Given the fundamental connection between ordering of rules and composition of systems of equations, it follows that no order of composition generates a system which accepts both /MRM/ $\mapsto$ [LHM] and /RMR/ $\mapsto$ [HLR]. Composition $b \otimes c$ (recall that system b describes the RM rule and c the MR rule) defined in (5.81) recreates the order MR $<$ RM.

(5.81) a. $b \otimes c$

$$
\begin{aligned}
H_b(x) &= \texttt{if } \underline{R_c}M_c(x) \texttt{ then } \top \texttt{ else } H_c(x)\\
R_b(x) &= \texttt{if } \underline{R_c}M_c(x) \texttt{ then } \bot \texttt{ else } R_c(x)\\
F_b(x) &= F_c(x)\\
M_b(x) &= M_c(x)\\
L_b(x) &= L_c(x)
\end{aligned}
$$

b. c

$$
\begin{aligned}
H_c(x) &= H(x)\\
R_c(x) &= R(x)\\
F_c(x) &= F(x)\\
M_c(x) &= \texttt{if } \underline{M}R(x) \texttt{ then } \bot \texttt{ else } M(x)\\
L_c(x) &= \texttt{if } \underline{M}R(x) \texttt{ then } \top \texttt{ else } L(x)
\end{aligned}
$$

The composite system $b \otimes c$ and the order MR $<$ RM make the same predictions about the counterbleeding paradox; that is, they can derive /MRM/ $\mapsto$ [LHM], but fail to map /RMR/ to the attested [HLR], and instead map it to *[RLR] as in (5.80). Predictions of this system against the relevant trisyllabic strings are given in (5.82). Note that, as in the previous section, the same bleeding interaction is apparent in the accepted (unattested) /RMR/ $\mapsto$ *[RLR]. This is the result of RM application via system b being dependent on modifications to the input string inherited from MR application via system c, and evidenced in the modified licensing/blocking structure $\underline{R_c}M_c(x)$.

(5.82)

Input:	M	R	M	R	M	R
	1	2	3	1	2	3
$H_b(x)$	$\perp$	$\top$	$\perp$	$\perp$	$\perp$	$\perp$
$R_b(x)$	$\perp$	$\perp$	$\perp$	$\top$	$\perp$	$\top$
$F_b(x)$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$
$M_b(x)$	$\perp$	$\perp$	$\top$	$\perp$	$\perp$	$\perp$
$L_b(x)$	$\top$	$\perp$	$\perp$	$\perp$	$\top$	$\perp$
$H_c(x)$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$
$R_c(x)$	$\perp$	$\top$	$\perp$	$\top$	$\perp$	$\top$
$F_c(x)$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$	$\perp$
$M_c(x)$	$\perp$	$\perp$	$\top$	$\perp$	$\perp$	$\perp$
$L_c(x)$	$\top$	$\perp$	$\perp$	$\perp$	$\top$	$\perp$
Output:	L	H	M	*R	L	R

Reversing the order of composition to $c \otimes b$ recreates the opposite order RM < MR. This composite system is defined in below in (5.83).

(5.83) a. $c \otimes b$

$$
\begin{aligned}
H_c(x) &= H_b(x)\\
R_c(x) &= R_b(x)\\
F_c(x) &= F_b(x)\\
M_c(x) &= \texttt{if } \underline{M}_b R_b(x) \texttt{ then } \perp \texttt{ else } M_b(x)\\
L_c(x) &= \texttt{if } \underline{M}_b R_b(x) \texttt{ then } \top \texttt{ else } L_b(x)
\end{aligned}
$$

b. b

$$
\begin{aligned}
H_b(x) &= \texttt{if } \underline{R}M(x) \texttt{ then } \top \texttt{ else } H(x)\\
R_b(x) &= \texttt{if } \underline{R}M(x) \texttt{ then } \perp \texttt{ else } R(x)\\
F_b(x) &= F(x)\\
M_b(x) &= M(x)\\
L_b(x) &= L(x)
\end{aligned}
$$

System $c \otimes b$ and the order RM < MR also make the same predictions about the counterbleeding paradox in Changting; they can derive /RMR/ $\mapsto$ [HLR] unlike $b \otimes c$ and MR < RM, but fail to map /MRM/ to the attested [LHM], and instead map it to the *[MHM] as in (5.79). Example (5.84) demonstrates the reversed composite system's predictions against the relevant trisyllabic strings: the predictions are the same as the ordering RM < MR. As before, note the bleeding effect on the mapping /MRM/ $\mapsto$ *[MHM], the direct result of the dependency relationship between systems c and b imposed by this ordering.

(5.84)

Input:	M	R	M	R	M	R
	1	2	3	1	2	3
$H_c(x)$	$\bot$	$\top$	$\bot$	$\top$	$\bot$	$\bot$
$R_c(x)$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\top$
$F_c(x)$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$
$M_c(x)$	$\top$	$\bot$	$\top$	$\bot$	$\bot$	$\bot$
$L_c(x)$	$\bot$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$
$H_b(x)$	$\bot$	$\top$	$\bot$	$\top$	$\bot$	$\bot$
$R_b(x)$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\top$
$F_b(x)$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$
$M_b(x)$	$\top$	$\bot$	$\top$	$\bot$	$\top$	$\bot$
$L_b(x)$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$
Output:	*M	H	M	H	L	R

No order over individual rules can derive both counterbleeding mappings. This means that no order of composition can accept both counterbleeding mappings, either. In other words, composition of BMRS individual systems fails to capture attested tone sandhi process interactions.

However, Oakden and Chandlee (2020) show that a single ISL function describes the mutual counterbleeding interaction. An equivalent BMRS system of equations is given below in (5.85).

(5.85)
$$\begin{aligned}
H'(x) &= \texttt{if } \underline{R}M(x) \texttt{ then } \top \texttt{ else } H(x)\\
R'(x) &= \texttt{if } \underline{R}M(x) \texttt{ then } \bot \texttt{ else } R(x)\\
F'(x) &= F(x)\\
M'(x) &= \texttt{if } \underline{L}F(x) \texttt{ then } \top \texttt{ else } M(x)\\
L'(x) &= \texttt{if } \underline{L}F(x) \texttt{ then } \bot \texttt{ else } L(x)
\end{aligned}$$

This system accepts both /MRM/ $\mapsto$ [LHM] *and* /RMR/ $\mapsto$ [HLR], consistent with the Changting data, as illustrated in the evaluation table in (5.86).

(5.86)

Input:	M	R	M	R	M	R
	1	2	3	1	2	3
$H'(x)$	$\bot$	$\top$	$\bot$	$\top$	$\bot$	$\bot$
$R'(x)$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\top$
$F'(x)$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$
$M'(x)$	$\bot$	$\bot$	$\top$	$\bot$	$\bot$	$\bot$
$L'(x)$	$\top$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$
Output:	L	H	M	H	L	R

A single 'combined map' function describes mutual counterbleeding in Changting, meaning that the complexity of the interaction itself is ISL. But it cannot be the case that this function is the composition of two individual systems b and c, and this follows from the fact that no order over individual rules derives both counterbleeding mappings. Therefore, what is the relationship between the individual systems and the combined map function in such cases? The next chapter addresses this question, and proposes an additional operation over BMRS systems which makes accurate predictions about a number of attested tone sandhi ordering paradoxes, while maintaining computational locality.

Before such a proposal, it is worth reiterating that analyses in terms of SL functions (and implemented using BMRS) offer a straightforward account of paradoxical interactions not available to traditional rule-based accounts. Changting counterbleeding, for example, is ISL, as the system in (5.85) illustrates. It is properly subsequential, which means it is also describable as a regular relation. But as Johnson (1972) and Kaplan and Kay (1994) point out, so are SPE grammars of rewrite rules. What this suggests, then, is something of a blind spot in the formalism: a properly regular subset in which phonological interactions are attested, but for which SPE cannot offer an account. The computational formalism, on the other hand, provides an explicit characterization of this pattern as an ISL function.

6 Interactions as Parallel Satisfaction with $\ominus$

6.1 Introduction

Composition of BMRS systems formalizes transparent and opaque tone sandhi interactions by recreating the effect of serial rule ordering. As the end of Chapter 5 demonstrates, however, this approach fails to account for ordering paradoxes such as mutual counterbleeding in Changting (Chen, 2004). No serial order of the MR (M → L / __ R) and RM (R → H / __ M) rules can derive both /MRM/ ↦ [LHM] and /RMR/ ↦ [HLR] mappings; they require opposite orderings of these rules. Example (6.1) summarizes the paradox with derivations using both orders.

(6.1) a.

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

b.

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

Because no order of rewrite rules can derive this pattern, it follows that no order of composition will capture the interaction when RM and MR rules are defined as separate BMRS systems. §6.3 of the previous chapter demonstrates this in detail.

Oakden and Chandlee (2020) show that the mutual counterbleeding interaction in Changting is ISL. They do so by defining a QF logical transduction that accepts /MRM/ ↦ [LHM] and /RMR/ ↦ [HLR]. An equivalent BMRS system is given in (6.2), defined over the output alphabet $\{H, R, F, M, L\}$.

(6.2)
$$\begin{aligned} H'(x) &= \texttt{if } \underline{R}M(x) \texttt{ then } \top \texttt{ else } H(x) \\ R'(x) &= \texttt{if } \underline{R}M(x) \texttt{ then } \bot \texttt{ else } F(x) \\ F'(x) &= F(x) \\ M'(x) &= \texttt{if } \underline{M}R(x) \texttt{ then } \bot \texttt{ else } M(x) \\ L'(x) &= \texttt{if } \underline{M}R(x) \texttt{ then } \top \texttt{ else } L(x) \end{aligned}$$

The solution they present hinges on the *non*-interaction of RM and MR rules. While inevitable in a serial derivation (given intermediate representations), the logical characterization shows that reference to the same input string allows both rules to apply in parallel without interference. This type of 'simultaneous' application derives precisely the outputs observed in Changting.

Given this insight, we may ask: how do individual RM and MR rules in Changting 'combine' to produce the combined-map function described in (6.2)? More specifically, what operation over BMRS systems guarantees that individual ISL functions describing each rule compute outputs using bounded reference to the same input structure? Composition merely recapitulates serial ordering paradoxes, and so is a nonstarter.

This chapter defines a new operation over BMRS systems of equations. It relates systems such as those describing Changting RM and MR rules to combined map functions in (6.2). The operator is termed 'parallel satisfaction' (PS, denoted $\ominus$); combining two or more systems with PS—termed a 'PS-join'—describes a function that satisfies both systems, crucially with reference to the same input and output strings. That is, neither system has access to the modifications to the input string rendered by the other system. It is in this way that it differs from BMRS composition with the $\otimes$ operator.

Taken together, composition and parallel satisfaction bear some resemblance to the serial vs. simultaneous application dichotomy in a rule-based formalism (see, for example, early discussion in Chomsky and Halle, 1968a; Postal, 1968; McCawley, 1968; Harms, 1968). While the ISL-definable interaction in (6.2) conforms to those intuitions, $\ominus$ does not model 'simultaneous application' of rules generally. Instead, it describes an operation that collapses input/output reference across separate systems into a single input and output string. Reference to the output string in particular forces a departure from a rule-based conception; rules in SPE do not describe surface (i.e. output) structure as part of their conditioning environments.[1] In addition, PS formalizes some interactions—such as mutual bleeding in §3.3—not derivable in a simultaneous application paradigm (see Pullum, 1972, for a relevant demonstration in Spanish).

Conceptually, the PS operator also brings to mind two-level phonology in the spirit of Koskenniemi (1983). A PS-joined function can be thought of as a set of elements (the individual BMRS systems), each containing information about some facet of the mapping, and which refer to the same lexical (input) and surface (output) character pairs. That is, the elements operate in tandem. As Karttunen (1993) puts it: "a surface form is a realization of a lexical form just in case all transducers [BMRS systems in our formalism] accepts the pair." This approach differs from the two-level formalism, however, in that $\ominus$ is not intersection, as in Koskenniemi's model.

PS-joins, regardless of their relation or non-relation to previous approaches, can be leveraged to analyze a variety of tone sandhi interactions, and therefore are a meaningful addition to the set of operations over BMRS systems. Specifically, they account for paradoxical sandhi interactions

[1] Rules with iterative application do, in a way, describe output structure as part of the conditioning environment.

for which a compositional analysis fails. They also model interactions which *can* be derived by rule ordering (and thus composition). In the sections that follow, both types of analyses will be presented.

This chapter is organized as follows. §2 defines the PS operator ($\ominus$) and summarizes its formal properties, namely that it models parallel evaluation of more than one BMRS-definable transduction over a single input and output string. Reference to one input and one output—that is, no access to intermediate representations—provides a solution to ordering/ranking paradoxes in Changting; §3 builds on Oakden and Chandlee (2020)'s results, demonstrating the applicability of a PS-join analysis to mutual counterbleeding and mutual bleeding interactions in Changting. §4 revisits interactions in Nanjing sandhi to show that some opaque interactions derivable by rule ordering—counterfeeding and counterbleeding on environment—also have a PS-join solution. A third case study of Xiamen tone circles is presented in §5. Like Changting, the Xiamen paradigm evades straightforward characterization in rule-based (ordering paradox) and optimization-based (ranking paradox) formalisms, but receives a straightforward account in terms of parallel satisfaction over BMRS systems of equations. §6 concludes the chapter with a discussion of the implications of this operator.

6.2 Definition and Formal Properties

In this section, I define a syntactic operator $\ominus$, termed 'parallel satisfaction' (PS), over BMRS systems of equations. Applied to BMRS systems that describe individual processes, the PS operator recreates the effect of both processes applying in tandem. That is, they operate over the same input (and output) representation. This contrasts with BMRS-composition $\otimes$, where one function operates over an input in light of the other function's modification to that input (i.e. its output). Combining two systems via the $\ominus$ operator is referred to as a PS-join.

Because two systems evaluate the same input and output string, both PS-join orders—$A \ominus B$ and $B \ominus A$ for two systems A and B—typically describe extensionally-equivalent functions. This property obtains for the tone sandhi interactions explored in this dissertation. Order only becomes relevant when a *conflict* arises; the notion of conflict is defined in this section, and is explored in more detail in §6.

As in the previous chapter, this section also sketches a typology of PS-joins of strictly-local functions, and examines their features. Preservation of a single input and output string is crucial to formalizing tone sandhi interactions with $\ominus$, and applies to joins of systems describing ISL and OSL functions, and combinations thereof.

6.2.1 A syntactic operator $\ominus$

What follows is a definition of a syntactic operator $\ominus$ over BMRS systems of equations. As specified in specified in chapter 4, this dissertation focuses on PS-joins of length-preserving transductions, crucially defined over a single copy set.

Definition 4 *For BMRS transductions $\mathcal{T}_1, \mathcal{T}_2$ from strings in Σ^* to strings in Δ^*, let $\mathcal{T}_1 \ominus \mathcal{T}_2$ be a syntactic operation on BMRS transductions as follows. Let*

$$\mathcal{T}_1 = \{\ \mathtt{f}_1(x_1) = T_1,\ \ \ldots\ ,\ \mathtt{f}_n(x_n) = T_n\}$$

be a system of equations over a signature of strings in Σ^*: a set of recursively-defined, boolean-type function symbols $\mathtt{F}$ *where each* $\mathtt{f}_i$ *corresponds to some δ_i in the output alphabet Δ, where $n = |\Delta|$. Similarly, let*

$$\mathcal{T}_2 = \{\ \mathtt{g}_1(x_1) = T_1,\ \ \ldots\ ,\ \mathtt{g}_n(x_n) = T_n\}$$

be a system of equations also over a signature of strings in Σ^*: a set of recursively-defined, boolean-type function symbols $\mathtt{G}$ *where each* $\mathtt{g}_j$ *corresponds to some δ_j in the output alphabet Δ, where $n = |\Delta|$ as before.*

Let $\mathcal{T}_1'$ then be identical to $\mathcal{T}_1$ with the exception that the final term in T_i in the definition for each boolean-type function symbol $\mathtt{f}_i$ *in $\mathcal{T}_1$ is replaced with the* entire *definition T_i for corresponding function* $\mathtt{g}_i$ *in $\mathcal{T}_2$. Then, let $\mathcal{T}_1 \ominus \mathcal{T}_2 := \mathcal{T}_1' \cup \mathcal{T}_2$.*

Parallel satisfaction $\ominus$ differs from composition $\otimes$ in two crucial ways. First, whereas composition combines transductions with (potentially) different input alphabets—but guaranteeing that $\mathcal{T}_1$'s output alphabet and $\mathcal{T}_2$'s input alphabet are the same—parallel satisfaction is only definable over transductions with identical input and output alphabets. Again, this scenario is relevant to tone sandhi interactions, where individual processes target the same set of lexical tones, and thus forms the focus of this dissertation. A more general definition encompassing combination of transductions with different alphabets is left for future work.

Second, whereas composition replaces boolean terms wherever they appear in a given function definition, parallel satisfaction replaces a term in a fixed position: that is, in final position. The definition below gives a precise indication of this position for all possible boolean terms.

Definition 5 *For a function definition $\mathtt{f}_i(x) = T_i$, the final term in T_i is as follows:*

$$\begin{array}{ll} \top & \text{if } T_i = \top \\ \bot & \text{if } T_i = \bot \\ \sigma(T) & \text{if } T_i = \sigma(T) \\ \mathtt{f}(T) & \text{if } T_i = \mathtt{f}(T) \\ T_3 & \text{if } T_i = \mathtt{if}\ \ T_1\ \ \mathtt{then}\ \ T_2\ \ \mathtt{else}\ \ T_3 \end{array}$$

Since most of the function definitions introduced in this dissertation comprise either if-then-else statements or default-mappings from the input, the majority of replacements will be T_3 from $\mathtt{if}\ \ T_1\ \ \mathtt{then}\ \ T_2\ \ \mathtt{else}\ \ T_3$ (the former) or $\sigma(T)$ (the latter).

Crucially, parallel satisfaction joins BMRS systems of equations such that they evaluate the same input and output string, irrespective of potential modifications made by the other system. To illustrate, consider two systems $\mathcal{T}_1$ and $\mathcal{T}_2$ defined over the same alphabet $\Sigma = \Delta = \{a, b, c, d\}$. $\mathcal{T}_1$ (6.3) describes an ISL function that maps 1) every input a to c when it appears before an input b (and to a otherwise), 2) every input b to b, 3) every input c to c, and 4) every input d to d. That is, $a \to c$ / __ b in rule form.

(6.3)
$$\begin{array}{lcl} a_1(x) & = & \mathtt{if}\ \ \underline{a}b(x)\ \ \mathtt{then}\ \ \bot\ \ \mathtt{else}\ \ a(x) \\ b_1(x) & = & b(x) \\ c_1(x) & = & \mathtt{if}\ \ \underline{a}b(x)\ \ \mathtt{then}\ \ \top\ \ \mathtt{else}\ \ c(x) \\ d_1(x) & = & d(x) \end{array}$$

An input-oriented (non-recursively-defined) structure $\underline{a}b(x)$ licenses output c and blocks output a. The map /aba/ $\mapsto$ [cba] in (6.4) satisfies $\mathcal{T}_1$, as string positions 1, 2, and 3 return true $\top$ values for $c_1(x)$, $b_1(x)$, and $a_1(x)$, respectively.

(6.4)

Input:	a	b	a
	1	2	3
$a_1(x)$	$\bot$	$\bot$	$\top$
$b_1(x)$	$\bot$	$\top$	$\bot$
$c_1(x)$	$\top$	$\bot$	$\bot$
$d_1(x)$	$\bot$	$\bot$	$\bot$
Output:	c	b	a

$\mathcal{T}_2$ (defined in (6.5)) describes a different ISL function; it maps 1) input a to a, 2) every input b to d when it appears before an input a (and to b otherwise), 3) input c to c, and 4) input d to d. That is, $b \to d$ / __ a in rule form.

(6.5)
$$
\begin{aligned}
a_2(x) &= a(x)\\
b_2(x) &= \texttt{if } \underline{b}a(x) \texttt{ then } \bot \texttt{ else } b(x)\\
c_2(x) &= c(x)\\
d_2(x) &= \texttt{if } \underline{b}a(x) \texttt{ then } \top \texttt{ else } d(x)
\end{aligned}
$$

Like $\mathcal{T}_1$, output boolean function definitions in $\mathcal{T}_2$ contain a licensing/blocking structure pair: input-oriented $\underline{b}a(x)$ licenses output d but blocks output b. The ISL function represented by $\mathcal{T}_2$ maps input /aba/ to a different output [ada], mirroring the application of $b \to d$ / __ a. $\mathcal{T}_2$ accepts this mapping (and only this mapping for /aba/), as illustrated in (6.6).

(6.6)

Input:	a	b	a
	1	2	3
$a_2(x)$	$\top$	$\bot$	$\top$
$b_2(x)$	$\bot$	$\bot$	$\bot$
$c_2(x)$	$\bot$	$\bot$	$\bot$
$d_2(x)$	$\bot$	$\top$	$\bot$
Output:	a	d	a

A third ISL function could be defined. Given input string /aba/, it maps to neither [cba] (6.4) nor [ada] (6.6), but instead to [cda]. In a rule-based conception, this may be thought of as a counterbleeding order over rules $a \to c$ / __ b and $b \to d$ / __ a such that the former is ordered before the latter.

(6.7)

Input:	/aba/
$a \to c$ / __ b	**c**ba
$b \to d$ / __ a	c**d**a
Output:	[cda]

Alternatively, the two rules may be thought of as applying *simultaneously* to the input string /aba/. In a BMRS framework, $\mathcal{T}_1$ and $\mathcal{T}_2$ can join via $\circleddash$ to model this ISL function. A PS-joined system $\mathcal{T}_1 \circleddash \mathcal{T}_2$ is one in which the final term of each output boolean function definition in $\mathcal{T}_1$ is replaced with the corresponding definition from $\mathcal{T}_2$. Since both are defined over the same output alphabet $\Delta = \{a, b, c, d\}$, output boolean functions responsible for labeling output positions are identical. In the following definition (6.8), substituted final terms are shown in bold for clarity, and the unmodified $\mathcal{T}_2$ is given for reference.

(6.8) a. $\mathcal{T}_1 \ominus \mathcal{T}_2$

$$\begin{aligned} a_1(x) &= \texttt{if } \underline{a}b(x) \texttt{ then } \bot \texttt{ else } \mathbf{a_2(x)} \\ b_1(x) &= \mathbf{b_2}(x) \\ c_1(x) &= \texttt{if } \underline{a}b(x) \texttt{ then } \top \texttt{ else } \mathbf{c_2(x)} \\ d_1(x) &= \mathbf{d_2}(x) \end{aligned}$$

b. $\mathcal{T}_2$

$$\begin{aligned} a_2(x) &= a(x) \\ b_2(x) &= \texttt{if } \underline{b}a(x) \texttt{ then } \bot \texttt{ else } b(x) \\ c_2(x) &= c(x) \\ d_2(x) &= \texttt{if } \underline{b}a(x) \texttt{ then } \top \texttt{ else } d(x) \end{aligned}$$

When the system in (6.8) evaluates input-output mappings, truth values for terms $\underline{a}b(x)$ and $\underline{b}a(x)$—the licensing/blocking structures from the original systems—are assigned with reference to a single input string. Any input sequence *ab* licenses output *c*, and any input sequence *ba* licenses output *d*. Whether those two *b*s are the same input string position is irrelevant, because the terms are evaluated in parallel. $\mathcal{T}_1 \ominus \mathcal{T}_2$ thus describes the ISL function that maps /aba/ to [cda]; this system accepts the mapping as in (6.9):

(6.9)

Input:	a	b	a
	1	2	3
$a_1(x)$	$\bot$	$\bot$	$\top$
$b_1(x)$	$\bot$	$\bot$	$\bot$
$c_1(x)$	$\top$	$\bot$	$\bot$
$d_1(x)$	$\bot$	$\top$	$\bot$
$a_2(x)$	$\top$	$\bot$	$\top$
$b_2(x)$	$\bot$	$\bot$	$\bot$
$c_2(x)$	$\bot$	$\bot$	$\bot$
$d_2(x)$	$\bot$	$\top$	$\bot$
Output:	c	d	a

Given the property of evaluation over a single input string, the PS-joined system also accepts /bab/ $\mapsto$ [dcb] (6.10). Here, input substrings *ab* (licenses *c*) and *ba* (licenses *d*) converge on a single *a* position.

(6.10)

Input:	b	a	b
	1	2	3
$a_1(x)$	$\bot$	$\bot$	$\bot$
$b_1(x)$	$\bot$	$\bot$	$\top$
$c_1(x)$	$\bot$	$\top$	$\bot$
$d_1(x)$	$\top$	$\bot$	$\bot$
$a_2(x)$	$\bot$	$\top$	$\bot$
$b_2(x)$	$\bot$	$\bot$	$\top$
$c_2(x)$	$\bot$	$\bot$	$\bot$
$d_2(x)$	$\top$	$\bot$	$\bot$
Output:	d	c	b

In a rule-based framework, this outcome is predicted by a simultaneous application but crucially *not* with serial ordering. To derive [dca] from /bab/ requires the opposite rule ordering as that which gives /aba/ $\rightarrow$ [cda]. The toy example above constitutes an ordering paradox similar to the Changting interaction introduced at the end of the previous chapter. It is explored in detail in §3.

6.2.2 Order and structure-conflict

When two BMRS systems of equations are composed, the order in which they combine via $\otimes$ makes a difference. Often, systems with opposite composition orders—for instance $\mathcal{T}_1 \otimes \mathcal{T}_2$ and $\mathcal{T}_2 \otimes \mathcal{T}_1$ for two BMRS systems $\mathcal{T}_1$ and $\mathcal{T}_2$—describe different functions entirely.

These generalizations do not apply in the same manner to PS-join orders. Intuitively, this is because processes that apply simultaneously to an input are not ordered with respect to one another. Combination using $\ominus$ has an inherent 'order' such that one system occupies the position of 'outer' function and the other occupies the 'inner' position, as with composition. However, for the PS-joins defined in this chapter (i.e. those that model tone sandhi interactions), either 'order' describes the same function.

To illustrate with a concrete example, the PS-joined system $\mathcal{T}_1 \ominus \mathcal{T}_2$ in (6.8) models the simultaneous application of two rules. Reversing the order, $\mathcal{T}_2 \ominus \mathcal{T}_1$ in (6.11), describes an extensionally-equivalent ISL function. It thus models simultaneous application of rules in the same way.

(6.11) a. $\mathcal{T}_2 \ominus \mathcal{T}_1$

$$\begin{aligned} a_2(x) &= \mathbf{a_1(x)} \\ b_2(x) &= \texttt{if } \underline{b}a(x) \texttt{ then } \bot \texttt{ else } \mathbf{b_1(x)} \\ c_2(x) &= \mathbf{c_1(x)} \\ d_2(x) &= \texttt{if } \underline{b}a(x) \texttt{ then } \top \texttt{ else } \mathbf{d_1(x)} \end{aligned}$$

b. $\mathcal{T}_1$

$$\begin{aligned} a_1(x) &= \texttt{if } \underline{a}b(x) \texttt{ then } \bot \texttt{ else } a(x) \\ b_1(x) &= b(x) \\ c_1(x) &= \texttt{if } \underline{a}b(x) \texttt{ then } \top \texttt{ else } c(x) \\ d_1(x) &= d(x) \end{aligned}$$

Two input-oriented structures license specific output symbols and block others, as before: the *ab* substring licenses output *c* while blocking output *a*, the *ba* substring licenses output *d* while blocking output *b*. Additionally, the function scans a single input string, including those for which *a*s and *b*s overlap. This PS-joined system accepts the same mappings as $\mathcal{T}_1 \ominus \mathcal{T}_2$. An evaluation table (6.12) illustrates with /aba/ ↦ [cda].

(6.12)

Input:	a	b	a
	1	2	3
$a_2(x)$	$\bot$	$\bot$	$\top$
$b_2(x)$	$\bot$	$\bot$	$\bot$
$c_2(x)$	$\top$	$\bot$	$\bot$
$d_2(x)$	$\bot$	$\top$	$\bot$
$a_1(x)$	$\bot$	$\bot$	$\top$
$b_1(x)$	$\bot$	$\top$	$\bot$
$c_1(x)$	$\top$	$\bot$	$\bot$
$d_1(x)$	$\bot$	$\bot$	$\bot$
Output:	c	d	a

Thus in this case order is irrelevant. In the analysis of tone sandhi interactions to follow (§3-§5), both PS-join orders are given, and their acceptance of crucial interaction mappings is demonstrated.

The reason why order is irrelevant hinges on the nature of licensing/blocking structure pairs. In particular, the order in which systems join via $\ominus$ is inconsequential when the licensing/blocking structures from each system do not generate a *conflict*. Conflict can be defined as follows.

Definition 6 *For any two terms of the form* `if` T_1 `then` $\top/\bot$ `else` T_3*, a conflict obtains when the following conditions are met:*

a. The first term is of the form `if` T_1 `then` $\top$ `else` T_3

b. The second term is of the form `if` T_1 `then` $\bot$ `else` T_3

c. T_1 refers to the same structure in both terms

d. Both terms appear in the same output boolean function definition

Conditions *a*-*c* describe the notion of a licensing/blocking structure pair. In a PS-join, when both converge on a single output boolean function definition, a conflict arises. Using $\mathcal{T}_1$ and $\mathcal{T}_2$ as an example, terms (i.e. T_1) $\underline{a}b(x)$ and $\underline{b}a(x)$ cannot be in conflict because they do not refer to the same structure. Thus, either PS-join order will describe the same ISL function.

As mentioned above, this chapter analyzes sandhi interactions with non-conflicting PS-joins, meaning that order does not matter for those cases (but see §6.3 a discussion of a case where order does matter).

6.2.3 PS with strictly-local functions

Parallel satisfaction over BMRS systems formalizes the application of multiple processes, but with reference to a single input string and a single output string. This section presents a basic typology of PS-joins, with a focus on ISL and OSL functions.

The extended example introduced in previous sections (§2.1 and §2.2) joins two ISL functions using the $\ominus$ operator. Recall $\mathcal{T}_1 \ominus \mathcal{T}_2$ in (6.8), repeated below in (6.13). $\mathcal{T}_1$ and $\mathcal{T}_2$ describe ISL functions.

(6.13) a. $\mathcal{T}_1 \ominus \mathcal{T}_2$

$$\begin{array}{lcl} a_1(x) & = & \texttt{if } \underline{a}b(x) \texttt{ then } \bot \texttt{ else } \mathbf{a_2}(\mathbf{x}) \\ b_1(x) & = & \mathbf{b_2}(x) \\ c_1(x) & = & \texttt{if } \underline{a}b(x) \texttt{ then } \top \texttt{ else } \mathbf{c_2}(\mathbf{x}) \\ d_1(x) & = & \mathbf{d_2}(x) \end{array}$$

b. $\mathcal{T}_2$

$$\begin{array}{lcl} a_2(x) & = & a(x) \\ b_2(x) & = & \texttt{if } \underline{b}a(x) \texttt{ then } \bot \texttt{ else } b(x) \\ c_2(x) & = & c(x) \\ d_2(x) & = & \texttt{if } \underline{b}a(x) \texttt{ then } \top \texttt{ else } d(x) \end{array}$$

As separate systems, $\mathcal{T}_1$ and $\mathcal{T}_2$ contain an input-oriented licensing/blocking structure pair. The result of combining with parallel satisfaction is that each structure $\underline{a}b(x)$ and $\underline{b}a(x)$ is evaluated with respect to the same input string. 'Default' terms—$b(x)$ and $d(x)$ in $\mathcal{T}_1$; $a(x)$ and $c(x)$ in $\mathcal{T}_2$—also refer to a single input.

Joins of ISL functions preserve a single output string vacuously, as neither function refers to the output. When ISL and OSL functions combine with parallel satisfaction, the output-orientedness of a single system is co-opted by the joined system. This is implicit in Definition 4 given that two systems share an output alphabet—a recursively defined function in either system refers to the same output string in the PS-join. Recursive function calls in joined systems are thus labeled with the joined system. To illustrate, consider a BMRS system $\mathcal{T}_3$ (6.14). It is similar to $\mathcal{T}_2$ in that it maps b to d, but differs in that it does so only when b is followed by an *output* a.

(6.14)
$$\begin{aligned} a_3(x) &= a(x) \\ b_3(x) &= \texttt{if } \underline{b}a_3(x) \texttt{ then } \bot \texttt{ else } b(x) \\ c_3(x) &= c(x) \\ d_3(x) &= \texttt{if } \underline{b}a_3(x) \texttt{ then } \top \texttt{ else } d(x) \end{aligned}$$

The function represented by $\mathcal{T}_3$ computes outputs with reference to the current input position, and a bounded window in the output structure, specifically one position to the right, shown by the term $\underline{b}a_3(x)$. As such, it describes an ROSL function. Combined with $\mathcal{T}_1$, the PS-join $\mathcal{T}_1 \ominus \mathcal{T}_3$ (6.15) accepts mappings which satisfy both in parallel, crucially with reference to a single input and a single output string. This is denoted by '${}_{1\ominus 3}$' in recursive function calls.

(6.15) a. $\mathcal{T}_1 \ominus \mathcal{T}_3$
$$\begin{aligned} a_1(x) &= \texttt{if } \underline{a}b(x) \texttt{ then } \bot \texttt{ else } \mathbf{a_3(x)} \\ b_1(x) &= \mathbf{b_3}(x) \\ c_1(x) &= \texttt{if } \underline{a}b(x) \texttt{ then } \top \texttt{ else } \mathbf{c_3(x)} \\ d_1(x) &= \mathbf{d_3}(x) \end{aligned}$$

b. $\mathcal{T}_3$
$$\begin{aligned} a_3(x) &= a(x) \\ b_3(x) &= \texttt{if } \underline{b}a_{1\ominus 3}(x) \texttt{ then } \bot \texttt{ else } b(x) \\ c_3(x) &= c(x) \\ d_3(x) &= \texttt{if } \underline{b}a_{1\ominus 3}(x) \texttt{ then } \top \texttt{ else } d(x) \end{aligned}$$

Example (6.16) collapses both into an equivalent single system for clarity.

(6.16)
$$\begin{aligned} a_{1\ominus 3}(x) &= \texttt{if } \underline{a}b(x) \texttt{ then } \bot \texttt{ else } a(x) \\ b_{1\ominus 3}(x) &= \texttt{if } \underline{b}a_{1\ominus 3}(x) \texttt{ then } \bot \texttt{ else } b(x) \\ c_{1\ominus 3}(x) &= \texttt{if } \underline{a}b(x) \texttt{ then } \top \texttt{ else } c(x) \\ d_{1\ominus 3}(x) &= \texttt{if } \underline{b}a_{1\ominus 3}(x) \texttt{ then } \top \texttt{ else } d(x) \end{aligned}$$

Thus the function maps input a to c when followed by an input b, and maps b to d when followed by an *output* a. Given an input string /baba/, the output is [bcda]. The table in (6.17) illustrates evaluation by the PS-join system $\mathcal{T}_1 \ominus \mathcal{T}_3$.

(6.17)

Input:	b	a	b	a
	1	2	3	4
$a_{1\ominus 3}(x)$	$\bot$	$\bot$	$\bot$	$\top$
$b_{1\ominus 3}(x)$	$\top$	$\bot$	$\bot$	$\bot$
$c_{1\ominus 3}(x)$	$\bot$	$\top$	$\bot$	$\bot$
$d_{1\ominus 3}(x)$	$\bot$	$\bot$	$\top$	$\bot$
Output:	b	c	d	a

The difference between positions 1 and 3 is crucial. First, position 3 satisfies the output-oriented structure $\underline{b}a_{1\ominus 3}(x)$; it is input-specified as b followed immediately by an output a (which itself maps directly from input a). Position 1, despite having an a as its input successor, does not evaluate to true for $\underline{b}a_{1\ominus 3}(x)$. This is because position 2 maps to c by virtue of satisfying the relevant licensing structure ($\underline{a}b(x)$ in the definition $c_{1\ominus 3}(x)$). The PS-join $\mathcal{T}_1 \ominus \mathcal{T}_3$ therefore integrates the ISL properties of $\mathcal{T}_1$ and the OSL properties of $\mathcal{T}_3$. It models the satisfaction of both systems in tandem, crucially with reference to single input and output strings. Note that this mapping /baba/ $\mapsto$ [bcba] is different from what $\mathcal{T}_3(x)$ would accept in isolation—/baba/ $\mapsto$ [**d**a**d**a]. Additionally, it differs from $\mathcal{T}_1 \ominus \mathcal{T}_2$'s mapping—/baba/ $\mapsto$ [**d**c**d**a].

Two OSL functions join via $\ominus$ in a similar fashion. That is, the PS-join 'collapses' reference to separate output strings (in recursive function calls) into a single output string. To illustrate, consider a system $\mathcal{T}_4$ in (6.18), the ROSL equivalent of $\mathcal{T}_1$.

(6.18)
$$\begin{aligned} a_4(x) &= \texttt{if } \underline{a}b_4(x) \texttt{ then } \bot \texttt{ else } a(x) \\ b_4(x) &= b(x) \\ c_4(x) &= \texttt{if } \underline{a}b_4(x) \texttt{ then } \top \texttt{ else } c(x) \\ d_4(x) &= d(x) \end{aligned}$$

Instead of the input-oriented structure $\underline{a}b(x)$ licensing an output c, it is the output-oriented structure represented by $\underline{a}b_1(x)$ (input a with output b as its successor) which causes a to map to c.

$\mathcal{T}_3 \ominus \mathcal{T}_4$ (6.19) is a PS-join of two ROSL functions. This means that definitions with recursive function calls across both systems refer to a single output structure. As before, this is denoted with '$_{3\ominus 4}$' in the definitions below.

(6.19) a. $\mathcal{T}_3 \ominus \mathcal{T}_4$

$$\begin{aligned} a_3(x) &= \mathbf{a_4(x)} \\ b_3(x) &= \texttt{if } \underline{b}a_{3\ominus 4}(x) \texttt{ then } \bot \texttt{ else } \mathbf{b_4(x)} \\ c_3(x) &= \mathbf{c_4(x)} \\ d_3(x) &= \texttt{if } \underline{b}a_{3\ominus 4}(x) \texttt{ then } \top \texttt{ else } \mathbf{d_4(x)} \end{aligned}$$

b. $\mathcal{T}_4$

$$\begin{aligned} a_4(x) &= \texttt{if } \underline{a}b_{3\ominus 4}(x) \texttt{ then } \bot \texttt{ else } a(x) \\ b_4(x) &= b(x) \\ c_4(x) &= \texttt{if } \underline{a}b_{3\ominus 4}(x) \texttt{ then } \top \texttt{ else } c(x) \\ d_4(x) &= d(x) \end{aligned}$$

As an equivalent single system (6.20):

(6.20)

$$\begin{aligned} a_{3\ominus 4}(x) &= \texttt{if } \underline{a}b_{3\ominus 4}(x) \texttt{ then } \bot \texttt{ else } a(x) \\ b_{3\ominus 4}(x) &= \texttt{if } \underline{b}a_{3\ominus 4}(x) \texttt{ then } \bot \texttt{ else } b(x) \\ c_{3\ominus 4}(x) &= \texttt{if } \underline{a}b_{3\ominus 4}(x) \texttt{ then } \top \texttt{ else } c(x) \\ d_{3\ominus 4}(x) &= \texttt{if } \underline{b}a_{3\ominus 4}(x) \texttt{ then } \top \texttt{ else } d(x) \end{aligned}$$

$\mathcal{T}_3 \ominus \mathcal{T}_4$ describes an ROSL function; it computes outputs with reference to the current input string position and a bounded output window to the right of that string position. The output window is a single output string referenced in the PS-join. This detail is crucial to the modeling of tone sandhi interactions presented in §3 of this chapter. §6.1 explores the issue in more detail.

As with composition, the analyses presented in this dissertation limit their focus to PS-joins of ROSL functions. Future work may investigate the combination of ROSL and LOSL functions with the parallel satisfaction operator.

6.2.4 Interim summary

So far, this chapter has defined a parallel satisfaction operator $\ominus$ over BMRS systems of equations and provided a sketch of its basic properties. When two individual systems form a PS-join, the resulting system describes a function whose mappings from input to output satisfy both systems in tandem, crucially with reference to a single input and output string. This bears a resemblance to both simultaneous application of rules and two-level phonological approaches as no intermediate representations factor into computation. Additionally, systems describing strictly-local functions can combine under this operator. Preservation of single input and single output strings applies to PS-joins of two ISL functions, two OSL functions, and combinations of ISL and OSL functions.

Due to the absence of intermediate forms in the PS-join framework, such analysis provides a straightforward account of ordering paradoxes in tone sandhi, such as the Changting interaction sketched at the end of the previous chapter. The following three sections present case studies of sandhi interactions in Changting, Nanjing, and Xiamen. Transparent and opaque patterns—some ordering paradoxes and others not—are formalized as PS-joins of individual BMRS systems.

6.3 Changting

As mentioned in the previous chapter, ordering paradoxes are inherent to the Changting sandhi paradigm. No order of rules—and thus no order of composition in a BMRS framework—can derive output strings from underlying forms for certain three-syllable combinations. Recall that Changting has five lexical tones: low (L), mid (M), high (H), rising (R), and falling (F). Out of the 25 possible two-tone combinations, Chen (2004) reports 15 combinations which undergo tone sandhi. The sandhi patterns which are of concern for this chapter are repeated in (6.21).

(6.21) a. 'RM rule': R $\rightarrow$ H / __ M

b. 'MR rule': M $\rightarrow$ L / __ R

c. 'LM rule': L $\rightarrow$ M / __ M

d. 'ML rule': M $\rightarrow$ L / __ L

These basic patterns extend to sequences of three or more tones, with overlapping targets and triggers giving rise to interactions. This section introduces two interactions in Changting which, in a rule-based framework, present ordering paradoxes: mutual counterbleeding between rules in (6.21a-b) and mutual bleeding between rules in (6.21c-d).

Changting 'RM' and 'MR' rules interact act such that each rule counterbleeds the other. Recall the relevant trisyllabic data from the previous chapter (5.78), repeated in ((6.22); Chen, 2004, p.803).

(6.22)

Input	Output	[x x] x	x [x x]
MRM	LHM	*[hua.qian] duo*	*xin [bei.jing]*
		"spending a lot of money"	"new Beijing"
RMR	HLR	*[yi.jin] you*	*chu [feng.tou]*
		"one catty of oil"	"to show off"

In order to derive both outputs from their corresponding inputs, conflicting rule orderings are necessary. For example, the trisyllabic form /MRM/ surfaces as [LHM]; this requires the ordering MR $<$ RM, as illustrated by the following parallel derivations (6.23).

(6.23)	$\underline{\mathrm{MR}}\mathrm{M}$		$\mathrm{M}\underline{\mathrm{RM}}$	
	$\mathrm{L}\underline{\mathrm{RM}}$	by MR rule	MHM	by RM rule
	LHM	by RM rule	*MHM	MR rule n/a

RM counterbleeds MR; earlier application of the RM rule would *bleed* the application of the MR rule by destroying part of its environment (crucially the R). In a different three-tone sequence, however—/RMR/ → [HLR]—the opposite ordering is necessary. This is given in (6.24).[2]

(6.24)	$\underline{\mathrm{RM}}\mathrm{R}$		$\mathrm{R}\underline{\mathrm{MR}}$	
	$\mathrm{H}\underline{\mathrm{MR}}$	by RM rule	RLR	by MR rule
	HLR	by MR rule	*RLR	RM rule n/a

Here, RM < MR is necessary to derive the attested output [HLR]. As before, it is a case of counterbleeding. That is, application of the MR rule destroys the necessary environment (crucially the M tone) for the RM rule. Thus each rule counterbleeds the other, with the consequence being that a crucial ordering of rules *cannot* derive the attested forms. The RM and MR rules present an ordering paradox.

Another ordering paradox is apparent, this time between the LM (6.21c) and ML rules (6.21d). Example (6.25) gives the relevant data from (Chen, 2004, p. 804-5).

(6.25)

Input	Output	[x x] x	x [x x]
MLM	MMM	*[gan.yuan] jiao*	*wo [shi.zhai]*
		"willing to teach"	"I am a vegetarian"
LML	LLL	*[ren.zhen] du*	*shi [xi.yao]*
		"seriously study"	"take western medication"

Forms *ganyuan jiao* 'willing to teach' and *wo shizhai* 'I am a vegetarian' motivate the ordering LM < ML. Only this order derives [MMM] from input /MLM/, as in (6.26).

(6.26)	$\mathrm{M}\underline{\mathrm{LM}}$		$\underline{\mathrm{ML}}\mathrm{M}$	
	MMM	by LM rule	$\mathrm{L}\underline{\mathrm{LM}}$	by ML rule
	MMM	ML rule n/a	*LMM	by LM rule

[2]As Chen (2004) argues, this ordering would also feed a different 'RL rule' (RL → RF) to produce the unattested *[RFR]. I abstract away from this detail for now.

These derivations show that earlier application of LM bleeds the ML rule. Unfortunately, in the mapping of these forms' tonal palindromes *renzhen du* 'seriously study' and *shi xiyao* 'take western medication'—/LML/ $\mapsto$ [LLL] as in (6.27)—the opposite ordering and resultant bleeding are necessary.

(6.27)

$\mathrm{L\underline{ML}}$		$\mathrm{\underline{LM}L}$	
|		|	
LLL	by ML rule	$\mathrm{M\underline{ML}}$	by LM rule
|		|	
LLL	LM rule n/a	*MLL	by ML rule

So again, in order to account for attested forms, both rules must bleed one another. This is a challenge for traditional rule-based accounts as no single order can derive both outputs from their respective input forms.

As discussed in Chapter 2, these interactions also lead to ranking paradoxes in an OT account. According to Chen (2004)'s analysis whereby candidates are derivations, ranking Temp (apply rules from left to right) above Econ (minimize derivational steps) is motivated as it correctly chooses the derivation MRM → LRM → LHM over MRM → MHM, as in (6.28) (repeated from (2.27)).

(6.28)

	/MRM/	Temp	Econ
☞	$\mathrm{\underline{MR}M}$ - $\mathrm{L\underline{RM}}$ - LHM		**
	$\mathrm{M\underline{RM}}$ - $\mathrm{\underline{MH}M}$	*!	*

But then the opposite ranking—Econ<<Temp in (6.29) (repeated from (2.28))—is necessary to select the derivation MLM → MMM over a different candidate containing an unattested surface form MLM → LLM → LMM.

(6.29)

	/MLM/	Econ	Temp
	$\mathrm{\underline{ML}M}$ - $\mathrm{L\underline{LM}}$ - LMM	**!	
☞	$\mathrm{M\underline{LM}}$ - $\mathrm{\underline{MM}M}$	*	*

Despite the challenges Changting poses to rule-based and optimization-based theories of phonology, these interactions are strictly-local when defined as single combined map functions. In their computational analysis of Changting, Oakden and Chandlee (2020) use logical transduction to show that the mutual counterbleeding interaction (6.22) describes an ISL function, and that the mutual bleeding interaction (6.25) describes an ROSL function. The crucial assumption—unavailable to rule-based accounts—is that both 'rules' in an interaction refer to the same input and output strings; that is, when intermediate representations are not considered. This section builds on that earlier work by providing a BMRS analysis of Changting tone sandhi. In particular, it demonstrates that mutual counterbleeding and mutual bleeding interactions in Changting can be characterized as the PS-join of systems of equations which describe the respective rules in isolation. These analyses are further distinguished from compositional analyses in that the ordering of systems is irrelevant, given the non-conflicting nature of licensing/blocking structure pairs (but see §5 for more discussion). Systems which describe individual rules are given first, followed by BMRS analyses of two interactions.

6.3.1 Individual rules as systems of equations

Four disyllabic Changting sandhi 'rules' (6.21) are relevant to counterbleeding and bleeding interactions to be explored here. Each is modeled with a separate BMRS system of equations, defined over the alphabet $\Sigma = \Gamma = \{H, R, F, M, L\}$. First, let a BMRS system of equations denoted a model the RM rule (6.21a). Its definition is given in (6.30), and it describes an ISL function.[3]

(6.30)
$$\begin{aligned} H_a(x) &= \texttt{if } \underline{R}M(x) \texttt{ then } \top \texttt{ else } H(x) \\ R_a(x) &= \texttt{if } \underline{R}M(x) \texttt{ then } \bot \texttt{ else } R(x) \\ F_a(x) &= F(x) \\ M_a(x) &= M(x) \\ L_a(x) &= L(x) \end{aligned}$$

A system b, defined in (6.31), models the MR rule in a similar manner, i.e. by means of a single input-specified licensing/blocking structure pair.

(6.31)
$$\begin{aligned} H_b(x) &= H(x) \\ R_b(x) &= R(x) \\ F_b(x) &= F(x) \\ M_b(x) &= \texttt{if } \underline{M}R(x) \texttt{ then } \bot \texttt{ else } M(x) \\ L_b(x) &= \texttt{if } \underline{M}R(x) \texttt{ then } \top \texttt{ else } L(x) \end{aligned}$$

BMRS systems corresponding to LM and ML rules (6.21c-d) are defined using recursion. This reflects Oakden and Chandlee (2020)'s observation that both rules are describable by OSL functions (as is their interaction). A system denoted c and which models the LM rule is given in (6.32).

(6.32)
$$\begin{aligned} H_c(x) &= H(x) \\ R_c(x) &= R(x) \\ F_c(x) &= F(x) \\ M_c(x) &= \texttt{if } \underline{L}M_c(x) \texttt{ then } \top \texttt{ else } M(x) \\ L_c(x) &= \texttt{if } \underline{L}M_c(x) \texttt{ then } \bot \texttt{ else } L(x) \end{aligned}$$

[3]Note that this is equivalent to system (5.39) defined for the same sandhi pattern in the previous chapter. The same holds for all other systems defined in this chapter.

The crucial structure in this system is an input L (the current string position) followed immediately by an *output* M which licenses an output L and blocks output M. Reference to output structure is also essential to a system of equations modeling the ML rule, denoted d and defined in (6.33).

(6.33)
$$\begin{aligned}
H_d(x) &= H(x)\\
R_d(x) &= R(x)\\
F_d(x) &= F(x)\\
M_d(x) &= \texttt{if } \underline{M}L_d(x) \texttt{ then } \bot \texttt{ else } M(x)\\
L_d(x) &= \texttt{if } \underline{M}L_d(x) \texttt{ then } \top \texttt{ else } L(x)
\end{aligned}$$

As the following sections will show, the $\ominus$ operator combines these systems to formalize so-called paradoxical interactions in Changting. Unlike composition, which provides access to intermediate representations (and therefore to the ordering paradoxes apparent in serial derivations), the procedure for joining by parallel satisfaction described in §2 limits the information available to the function to a single input and a single output string. Additionally, given the lack of conflicts between licensing/blocking structure pairs across definitions, the order in which the systems are joined is inconsequential (but see §5 for more discussion).

6.3.2 Mutual counterbleeding with RM/MR rules (ISL $\ominus$ ISL)

Mutual counterbleeding between RM and MR rules in (6.23) and (6.24) presents an ordering paradox; no ordering over individual rules can derive attested trisyllabic surface forms. As the previous chapter demonstrates, this means a compositional BMRS account is doomed to fail as well. Knowing that the interaction *is* describable by a single ISL function, and having defined individual systems for each rule, there must exist another operation by which they may combine to yield a function extensionally-equivalent to the combined-map. One viable operation is $\ominus$.

Consider the PS-joined system $a \ominus b$ defined below in (6.34) (recalling that systems a and b describe RM and MR rules, respectively), with the original system b for reference. In the joined system, replaced terms are given in bold.

(6.34) a. $a \ominus b$

$$\begin{aligned}
H_a(x) &= \texttt{if } \underline{R}M(x) \texttt{ then } \top \texttt{ else } \mathbf{H_b(x)}\\
R_a(x) &= \texttt{if } \underline{R}M(x) \texttt{ then } \bot \texttt{ else } \mathbf{R_b(x)}\\
F_a(x) &= \mathbf{F_b(x)}\\
M_a(x) &= \mathbf{M_b(x)}\\
L_a(x) &= \mathbf{L_b(x)}
\end{aligned}$$

b. b

$$\begin{aligned}
H_b(x) &= H(x)\\
R_b(x) &= R(x)\\
F_b(x) &= F(x)\\
M_b(x) &= \texttt{if } \underline{M}R(x) \texttt{ then } \bot \texttt{ else } M(x)\\
L_b(x) &= \texttt{if } \underline{M}R(x) \texttt{ then } \top \texttt{ else } L(x)
\end{aligned}$$

The *final* term of each output boolean function definition in a is replaced with the right side of the corresponding definition (also a term) in system b. In definitions $H_a(x)$ and $R_b(x)$, this is the final term of an if-then-else statement; in the other definitions, it is the licensing predicate from the input (default case of identity mapping).

Applying the $\ominus$ operator, the resulting system models Changting mutual counterbleeding. It crucially accepts *both* /MRM/ $\mapsto$ [LHM] and /RMR/ $\mapsto$ [HLR] interaction mappings, as illustrated in the evaluation table in (6.35).

(6.35)

Input:	M	R	M	R	M	R
	1	2	3	1	2	3
$H_a(x)$	$\bot$	$\top$	$\bot$	$\top$	$\bot$	$\bot$
$R_a(x)$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\top$
$F_a(x)$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$
$M_a(x)$	$\bot$	$\bot$	$\top$	$\bot$	$\bot$	$\bot$
$L_a(x)$	$\top$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$
$H_b(x)$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$
$R_b(x)$	$\bot$	$\top$	$\bot$	$\top$	$\bot$	$\top$
$F_b(x)$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$
$M_b(x)$	$\bot$	$\bot$	$\top$	$\bot$	$\bot$	$\bot$
$L_b(x)$	$\top$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$
Output:	L	H	M	H	L	R

In the first mapping /MRM/ $\mapsto$ [HLM], string position 1 (input M) returns a true value for $L_b(x)$ by satisfying the licensing structure $\underline{M}R$. This evaluation is passed directly to $L_a(x)$—originally defined as an identity mapping—and string position 1 surfaces as an L tone. The second string position is output as H via $H_a(x)$ (in spite of returning a *false* value for $H_b(x)$). String position 3 evaluates to true for $M_a(x)$ and $M_b(x)$ by virtue of being input-specified as such. Crucially, the licensing structures which permit these mappings are computed with reference to *input structure* only, and when a and b combine in this way, they refer to a single input string. This means, in intuitive terms, that RM and MR rules can 'apply' *simultaneously* to the trisyllabic string /MRM/; it is not the case that the application of one rule *destroys* the environment of the other.[4] Such an interaction is not possible under a composite system $a \otimes b$ or $b \otimes a$, as evaluation of non-recursively-defined licensing and blocking structures in these definitions would necessarily refer to the potential 'output' of one rule or another. The same generalization applies to the second mapping /RMR/ $\mapsto$ HLR; the input-orientedness of terms $\underline{R}M(x)$ and $\underline{M}R(x)$—preserved through application of $\ominus$—allows both rules to apply without one bleeding the environment of the other.

Additionally, the terms $\underline{R}M(x)$ and $\underline{M}R(x)$ that define the systems' licensing/blocking structure pairs are non-overlapping. Therefore switching the order does not alter the predictions, and in fact describes an (almost) identical function. In (6.36), the inverse ordering $b \ominus a$ is defined.

(6.36) a. $b \ominus a$

$$\begin{aligned}
H_b(x) &= \mathbf{H_a(x)}\\
R_b(x) &= \mathbf{R_a(x)}\\
F_b(x) &= \mathbf{F_a(x)}\\
M_b(x) &= \texttt{if } \underline{M}R(x) \texttt{ then } \bot \texttt{ else } \mathbf{M_a(x)}\\
L_b(x) &= \texttt{if } \underline{M}R(x) \texttt{ then } \top \texttt{ else } \mathbf{L_a(x)}
\end{aligned}$$

b. b

$$\begin{aligned}
H_a(x) &= \texttt{if } \underline{R}M(x) \texttt{ then } \top \texttt{ else } H(x)\\
R_a(x) &= \texttt{if } \underline{R}M(x) \texttt{ then } \bot \texttt{ else } R(x)\\
F_a(x) &= F(x)\\
M_a(x) &= M(x)\\
L_a(x) &= L(x)
\end{aligned}$$

Because licensing/blocking structure pairs in a and b appear in different output boolean function definitions with respect to one another ($H_a(x)$ and $R_a(x)$ in the case of a, and $M_b(x)$ and $L_b(x)$ in b), neither order of PS-join disturbs the system-internal hierarchy of these structures. What results is identical to the combined-map system of the interaction introduced at the end of the previous chapter. In other words, $a \ominus b$ and $b \ominus a$ describe the function which maps /MRM/ to [LHM] and /RMR/ to [HLR].

[4] This sense of 'simultaneous'—in which a *set* of rules applies to an underlying representation in a single step—is distinct from the 'simultaneous' used earlier whereby a *single* rule applies to an input string by scanning for targets, then applying all changes to those targets in a single step.

6.3.3 Mutual bleeding with ML/LM rules (OSL $\ominus$ OSL)

Mutual bleeding between LM and ML rules in (6.25) and (6.27) constitutes an ordering paradox in the same way as the mutual counterbleeding interaction. No total order over the two rules may derive both /MLM/ $\mapsto$ [MMM] and /LML/ $\mapsto$ [LLL]. This guarantees that no composition of systems c and d—describing LM and ML rules, respectively (6.32-6.33)—will yield a function describing the interaction. However, we know that the interaction itself is ROSL (Oakden and Chandlee, 2020), suggesting that individual systems might combine via *some* operation to constitute the full map. Indeed, applying the parallel satisfaction $\ominus$ operator to c and d describes a function which accepts both interaction mappings of three-tone sequences described above. According to the definition described above, the system $c \ominus d$ would be defined as in (6.37), with the original system d for reference. Replaced terms are given in bold for clarity, and recursive definitions are subscripted '$_{c\ominus d}$' to reflect collapsing multiple output strings into a single output.

(6.37) a. $c \ominus d$

$$\begin{aligned}
H_{c\ominus d}(x) &= \mathbf{H_d(x)} \\
R_{c\ominus d}(x) &= \mathbf{R_d(x)} \\
F_{c\ominus d}(x) &= \mathbf{F_d(x)} \\
M_{c\ominus d}(x) &= \texttt{if } \underline{L}M_{c\ominus d}(x) \texttt{ then } \top \texttt{ else } \mathbf{M_d(x)} \\
L_{c\ominus d}(x) &= \texttt{if } \underline{L}M_{c\ominus d}(x) \texttt{ then } \bot \texttt{ else } \mathbf{L_d(x)}
\end{aligned}$$

b. d

$$\begin{aligned}
H_d(x) &= H(x) \\
R_d(x) &= R(x) \\
F_d(x) &= F(x) \\
M_d(x) &= \texttt{if } \underline{M}L_{c\ominus d}(x) \texttt{ then } \bot \texttt{ else } M(x) \\
L_d(x) &= \texttt{if } \underline{M}L_{c\ominus d}(x) \texttt{ then } \top \texttt{ else } L(x)
\end{aligned}$$

In the same way that PS-join of ISL functions references the *same input string*, joining OSL functions (defined in a BMRS framework using recursive function calls) references the *same output string* in its computation. This is line with the single OSL function defining the interaction (see more in the discussion). It accepts *only* the outputs [MMM] and [LLL] from respective input strings /MLM/ and /LML/ as illustrated below in (6.38).

(6.38)

Input:	M	L	M	L	M	L
	1	2	3	1	2	3
$H_{c\ominus d}(x)$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$
$R_{c\ominus d}(x)$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$
$F_{c\ominus d}(x)$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$
$M_{c\ominus d}(x)$	$\top$	$\top$	$\top$	$\bot$	$\bot$	$\bot$
$L_{c\ominus d}(x)$	$\bot$	$\bot$	$\bot$	$\top$	$\top$	$\top$
Output:	M	M	M	L	L	L

Individual systems c and d and the PS-joined function describe an ROSL function; thus computation of the output string begins at the right edge and proceeds leftward. String position 3 in both mappings, as a final element, satisfies *none* of the licensing/blocking structures, and maps directly based on its input specification. The second string position in /MLM/ conforms to $\underline{L}M_{c\ominus d}(x)$—an input L followed immediately by an *output* M. This structure licenses an output M. String position 2 in /LML/ returns a true value for $L_{c\ominus d}(x)$ by satisfying the relevant licensing structure: an input M followed by an output L.

Computation of the first string position of each mapping is crucial. Importantly, string position 1 in the first mapping does *not* satisfy $\underline{M}L_{c\ominus d}(x)$ (which licenses output L and blocks output M); though it is an input M followed by an *input* L, that L has already been output as M. It evaluates to false for structures in both if-then-else statements, and is output as M as a result of being input-specified as M. Intuitively, this is the effect of the LM rule *bleeding* the ML rule. The same principle applies to the second mapping; here, the effect is one of the ML rule bleeding the LM rule. Thus both mappings accepted by the system are consistent with observed three-syllable forms in Changting.

Note that, in system c, an output M tone is licensed by a structure $\underline{L}M_c(x)$ and in system d, output L is licensed by a structure $\underline{M}L_d(x)$. Despite being defined using the same tones M and L, these structures do not overlap. What this predicts is that, like the mutual counterbleeding interaction above, reversing the order does not alter the nature of the resulting system. Example (6.39) defines a system whereby c and d join via PS, but in the reverse order.

(6.39) a. $d \ominus c$

$$\begin{aligned}
H_{d\ominus c}(x) &= \mathbf{H_c(x)}\\
R_{d\ominus c}(x) &= \mathbf{R_c(x)}\\
F_{d\ominus c}(x) &= \mathbf{F_c(x)}\\
M_{d\ominus c}(x) &= \texttt{if } \underline{M}L_{d\ominus c}(x) \texttt{ then } \bot \texttt{ else } \mathbf{M_c(x)}\\
L_{d\ominus c}(x) &= \texttt{if } \underline{M}L_{d\ominus c}(x) \texttt{ then } \top \texttt{ else } \mathbf{L_c(x)}
\end{aligned}$$

b. c

$$\begin{aligned}
H_c(x) &= H(x)\\
R_c(x) &= R(x)\\
F_c(x) &= F(x)\\
M_c(x) &= \texttt{if } \underline{L}M_{d\ominus c}(x) \texttt{ then } \top \texttt{ else } M(x)\\
L_c(x) &= \texttt{if } \underline{L}M_{d\ominus c}(x) \texttt{ then } \bot \texttt{ else } L(x)
\end{aligned}$$

Both systems define rules which manipulate M and L tones; i.e. their licensing/blocking structure pairs are defined over the same functions governing output M and L. Reversing the order of the PS join effectively creates two hierarchies of these structures, schematized below.

(6.40)

$c \ominus d$

$$\begin{aligned}
M(x) &= \texttt{if } \underline{L}M_{c\ominus d}(x) \texttt{ then } \top \texttt{ else}\\
&\quad\ \texttt{if } \underline{M}L_{c\ominus d}(x) \texttt{ then } \bot \texttt{ else } M(x)\\
L(x) &= \texttt{if } \underline{L}M_{c\ominus d}(x) \texttt{ then } \bot \texttt{ else}\\
&\quad\ \texttt{if } \underline{M}L_{c\ominus d}(x) \texttt{ then } \top \texttt{ else } L(x)
\end{aligned}$$

$d \ominus c$

$$\begin{aligned}
M(x) &= \texttt{if } \underline{M}L_{d\ominus c}(x) \texttt{ then } \bot \texttt{ else}\\
&\quad\ \texttt{if } \underline{L}M_{d\ominus c}(x) \texttt{ then } \top \texttt{ else } M(x)\\
L(x) &= \texttt{if } \underline{M}L_{d\ominus c}(x) \texttt{ then } \bot \texttt{ else}\\
&\quad\ \texttt{if } \underline{L}M_{d\ominus c}(x) \texttt{ then } \bot \texttt{ else } L(x)
\end{aligned}$$

Despite this, $c \ominus d$ and $d \ominus c$ make the same correct predictions about relevant interaction mappings in Changting, given that their respective licensing and blocking structures do not conflict with one another. Thus, joining two OSL functions via parallel satisfaction $\ominus$ (in either order) models the Changting mutual bleeding interaction, crucially by enforcing reference to a single output string.

6.4 Nanjing

This section builds on the analysis presented in the previous chapter. Recall the rules in (2.39), repeated in (6.41) with the addition of an 'HC' rule in (6.41e).

(6.41) a. 'LF rule': LF $\mapsto$ RF e.g. $lao^{L}shi^{F} \rightarrow lao^{R}shi^{F}$ 'teacher'

b. 'FF rule': FF $\mapsto$ HF e.g. $bing^{F}xiang^{F} \rightarrow bing^{H}xiang^{F}$ 'refrigerator'

c. 'RC rule': RC $\mapsto$ LC e.g. $tong^{R}xue^{C} \rightarrow tong^{L}xue^{C}$ 'classmate'

d. 'CC rule': CC $\mapsto$ C′C e.g. $qi^{C}shi^{C} \rightarrow qi^{C'}shi^{C}$ 'seventy'

e. 'HC rule': HC $\mapsto$ FC e.g. $shu^{H}xue^{C} \rightarrow shu^{F}xue^{C}$ 'mathematics'

In addition to the counterbleeding interactions presented in Chapter 4 (repeated in (6.42a-b)), Ma and Li (2014) report a counterfeeding interaction between HC and FF rules (6.42c).

(6.42) a. /LFF/ $\mapsto$ [RHF]

b. /RCC/ $\mapsto$ [LC′C]

c. /FHC/ $\mapsto$ [FFC]

The mapping in (6.42c) is an example of counterfeeding on environment. Derivations in (6.43) motivate the ordering FF < HC, as only this ordering of rules produces the attested mapping /FHC/ $\mapsto$ [FFC].

(6.43)

$$\begin{array}{ll}
\text{FHC} & \\
\mid & \\
\text{F}\underline{\text{HC}} & \text{FF rule n/a} \\
\mid & \\
\text{FFC} & \text{by HC rule}
\end{array}
\qquad\qquad
\begin{array}{ll}
\text{F}\underline{\text{HC}} & \\
\mid & \\
\underline{\text{FF}}\text{C} & \text{by HC rule} \\
\mid & \\
\text{*HFC} & \text{by FF rule}
\end{array}$$

In the output, it appears that the FF rule has *underapplied* (i.e. it is not surface-true); this is the result of the ordering which, if reversed, would have the HC rule *feeding* the FF rule to produce the unattested *[HFC].

While both counterbleeding and counterfeeding interactions in Nanjing can be derived via serial rule ordering—and thus do not present an ordering paradox—their interaction may also be formalized using the $\ominus$ operator. The following sections present a BMRS parallel satisfaction analysis of counterfeeding (6.42b) and counterbleeding (6.42c) interactions. Joining individual BMRS systems via PS indicates that counterfeeding and counterbleeding on environment are amenable to two-level accounts, i.e. those which assume a single input and single output. Additionally, and unlike composition, the order is inconsequential. For each interaction, both orders of PS join are given to demonstrate that they describe extensionally-equivalent functions. The availability of both parallel satisfaction and composition analyses of the data is explored in further detail in the discussion section. Before proceeding to the main analysis, the next section specifies the set of individual BMRS systems to be joined with $\ominus$.

6.4.1 Individual rules as systems of equations

Two pairs of disyllabic sandhi rules comprise the interactions in (6.42b) and (6.42c): a FF/HC rule pair (6.41b,e), and a RC/CC rule pair (6.41c,d). The previous chapter defined BMRS systems to model FF, RC, and CC rules in isolation, motivating ISL and OSL functional characterizations for these systems based on observed data. These systems are repeated below, and the reader is referred to the previous chapter (§5.1) for relevant discussion.

A system b (6.44) describes an ISL function, and models the FF sandhi rule. That is, a sequence of two input F tones blocks an output F on the first tone, and licenses an output H tone.

(6.44)
$$\begin{aligned}
H_b(x) &= \texttt{if } \underline{F}F(x) \texttt{ then } \top \texttt{ else } H(x)\\
L_b(x) &= L(x)\\
R_b(x) &= R(x)\\
F_b(x) &= \texttt{if } \underline{F}F(x) \texttt{ then } \bot \texttt{ else } F(x)\\
C_b(x) &= C(x)\\
C'_b(x) &= \bot
\end{aligned}$$

A system c (6.45) also describes an ISL function, and models the RC sandhi rule. The definition of output boolean functions below prevents an input R from mapping to the output when followed by a checked tone ($R_c(x)$). Instead, the function maps this tone to L ($L_c(x)$). The remaining output boolean functions map other input tones directly to the output.

(6.45)
$$\begin{aligned}
H_c(x) &= H(x)\\
L_c(x) &= \texttt{if } \underline{R}C(x) \texttt{ then } \top \texttt{ else } L(x)\\
R_c(x) &= \texttt{if } \underline{R}C(x) \texttt{ then } \bot \texttt{ else } R(x)\\
F_c(x) &= F(x)\\
C_c(x) &= C(x)\\
C'_c(x) &= \bot
\end{aligned}$$

As in the previous chapter, system d (6.46) describes the CC rule.

(6.46)
$$\begin{aligned}
H_d(x) &= H(x)\\
L_d(x) &= L(x)\\
R_d(x) &= R(x)\\
F_d(x) &= F(x)\\
C_d(x) &= \texttt{if } \underline{C}C_d(x) \texttt{ then } \bot \texttt{ else } C(x)\\
C'_d(x) &= \texttt{if } \underline{C}C_d(x) \texttt{ then } \top \texttt{ else } \bot
\end{aligned}$$

Recall that d describes an ROSL function. That is, it computes outputs using the current string position's input and a bounded local window in the output, crucially to the right of the string position under evaluation, evinced by recursive definitions of $C_d(x)$ and $C'_d(x)$. Support for this characterization comes from the mapping /CCC/ $\mapsto$ [CC′C] attested by Ma and Li (2014), and accepted by system d ((5.60), repeated in (6.47)):

(6.47)

Input:	C	C	C
	1	2	3
$C_d(x)$	$\top$	$\perp$	$\top$
$C'_d(x)$	$\perp$	$\top$	$\perp$
Output:	C	C′	C

Finally, let e denote a BMRS system of equations simulating the HC rule in (6.41e). It describes an ISL function—which is assumed in the absence of evidence to the contrary—and is given below in (6.48).

(6.48)
$$
\begin{aligned}
H_e(x) &= \texttt{if } \underline{H}C(x) \texttt{ then } \perp \texttt{ else } H(x)\\
L_e(x) &= L(x)\\
R_e(x) &= R(x)\\
F_e(x) &= \texttt{if } \underline{H}C(x) \texttt{ then } \top \texttt{ else } F(x)\\
C_e(x) &= C(x)\\
C'_e(x) &= \perp
\end{aligned}
$$

Here, the relevant licensing/blocking structure is an input sequence of a rising tone followed by a checked tone. This sequences licenses an output falling tone for the former, and thus recreates the effect of the HC rule in isolation. An abbreviated evaluation table (6.49) illustrates.

(6.49)

Input:	H	C
	1	2
$H_e(x)$	$\perp$	$\perp$
$F_e(x)$	$\top$	$\perp$
$C_e(x)$	$\perp$	$\top$
Output:	F	C

System d accepts only the surface string [FC] from input /HC/ and nothing else.

6.4.2 HC counterfeeds FF (ISL $\ominus$ ISL)

The following two sections model counterfeeding and counterbleeding sandhi interactions in Nanjing. They are formalized as the combination of individual BMRS systems via $\ominus$. For these interactions, the order of PS-join is inconsequential. That is, given two systems a and b defined over the same alphabets and which model distinct disyllabic rules in isolation, $a \ominus b$ and $b \ominus a$ describe extensionally-equivalent functions.

Recall the counterfeeding order on HC and FF rules necessary to derive the mapping /FHC/ $\mapsto$ [FFC] in (6.42c). This is motivated by the derivations in (6.43), repeated in (6.50).

(6.50)

FHC		F<u>HC</u>	
|		|	
F<u>HC</u>	FF rule n/a	<u>FF</u>C	by HC rule
|		|	
FFC	by HC rule	*HFC	by FF rule

Serial ordering of rules—specifically the order HC < FF—derives the correct output. This means that the interaction is amenable to a compositional BMRS analysis whereby systems e and a are composed in a certain order. However, combining the same systems with the PS operator also recreates the counterfeeding effect. Let one such system be denoted $e \ominus a$, defined in (6.51). As before, modified elements are given in bold.

(6.51) a. $e \ominus a$

$$
\begin{array}{rcl}
H_e(x) & = & \texttt{if } \underline{H}C(x) \texttt{ then } \bot \texttt{ else } \mathbf{H_a(x)} \\
L_e(x) & = & \mathbf{L_a(x)} \\
R_e(x) & = & \mathbf{R_a(x)} \\
F_e(x) & = & \texttt{if } \underline{H}C(x) \texttt{ then } \top \texttt{ else } \mathbf{F_a(x)} \\
C_e(x) & = & \mathbf{C_a(x)} \\
C'_e(x) & = & \mathbf{C'_a(x)}
\end{array}
$$

b. a

$$
\begin{array}{rcl}
H_a(x) & = & \texttt{if } \underline{F}F(x) \texttt{ then } \top \texttt{ else } H(x) \\
L_a(x) & = & L(x) \\
R_a(x) & = & R(x) \\
F_a(x) & = & \texttt{if } \underline{F}F(x) \texttt{ then } \bot \texttt{ else } F(x) \\
C_a(x) & = & C(x) \\
C'_a(x) & = & \bot
\end{array}
$$

Following the definition in §2, the $\ominus$ operator generates a system that is identical to e (the outer function), with the exception that the final term in each boolean function definition is replaced with the corresponding boolean function definition from a (the inner function). That $e \ominus a$ recreates the effect of the counterfeeding order in (6.43)—albeit not the ordering itself—is demonstrated in the following abbreviated evaluation table (6.52); the system accepts interaction mapping /FHC/ $\mapsto$ [FFC].

(6.52)

Input:	F	H	C
	1	2	3
$H_e(x)$	$\perp$	$\perp$	$\perp$
$F_e(x)$	$\top$	$\top$	$\perp$
$C_e(x)$	$\perp$	$\perp$	$\top$
$H_a(x)$	$\perp$	$\top$	$\perp$
$F_a(x)$	$\top$	$\perp$	$\perp$
$C_a(x)$	$\perp$	$\perp$	$\top$
Output:	F	F	C

Both licensing/blocking structure pairs, represented by terms $\underline{H}C(x)$ and $\underline{F}F(x)$, refer to input structure only. Unlike composition, they are not modified such that they refer to the output of some other function. This predicts 'application' of the HC rule, as an HC sequence is present in the input. For the same reason, it also predicts that the FF rule will not apply; there is no such sequence present in the input string. Thus, the first string position F returns a true value for $F_e(x)$ (via $F_a(x)$) and maps to F in the output.

Neither licensing/blocking structure overlaps with the other resulting in potential conflict. This means that reversing the order generates an extensionally-equivalent function. To see how, consider a system $a \ominus e$, the mirror image of $e \ominus a$. It is defined in (6.53).

(6.53) a. $a \ominus e$

$$\begin{aligned} H_a(x) &= \texttt{if } \underline{F}F(x) \texttt{ then } \top \texttt{ else } \mathbf{H_e(x)} \\ L_a(x) &= \mathbf{L_e(x)} \\ R_a(x) &= \mathbf{R_e(x)} \\ F_a(x) &= \texttt{if } \underline{F}F(x) \texttt{ then } \perp \texttt{ else } \mathbf{F_e(x)} \\ C_a(x) &= \mathbf{C_e(x)} \\ C'_a(x) &= \mathbf{C'_e(x)} \end{aligned}$$

b. e

$$\begin{aligned} H_e(x) &= \texttt{if } \underline{H}C(x) \texttt{ then } \perp \texttt{ else } H(x) \\ L_e(x) &= L(x) \\ R_e(x) &= R(x) \\ F_e(x) &= \texttt{if } \underline{H}C(x) \texttt{ then } \top \texttt{ else } F(x) \\ C_e(x) &= C(x) \\ C'_e(x) &= \perp \end{aligned}$$

This has the effect of switching the orders of hierarchies on definitions $H_a(x)/H_e(x)$ and $F_a(x)/F_e(x)$. Compare the respective definitions of each in (6.54):

(6.54) a.

$$\begin{array}{rcl} H_{a\ominus e}(x) & = & \texttt{if } \underline{F}F(x) \texttt{ then } \top \texttt{ else} \\ & & \texttt{if } \underline{H}C(x) \texttt{ then } \bot \texttt{ else } H(x) \\ F_{a\ominus e}(x) & = & \texttt{if } \underline{F}F(x) \texttt{ then } \bot \texttt{ else} \\ & & \texttt{if } \underline{H}C(x) \texttt{ then } \top \texttt{ else } F(x) \end{array}$$

b.

$$\begin{array}{rcl} H_{e\ominus a}(x) & = & \texttt{if } \underline{H}C(x) \texttt{ then } \bot \texttt{ else} \\ & & \texttt{if } \underline{F}F(x) \texttt{ then } \top \texttt{ else } H(x) \\ F_{e\ominus a}(x) & = & \texttt{if } \underline{H}C(x) \texttt{ then } \top \texttt{ else} \\ & & \texttt{if } \underline{F}F(x) \texttt{ then } \bot \texttt{ else } F(x) \end{array}$$

However, their reference to the same input is preserved. Thus $a \ominus e$ also describes a function which maps /FHC/ to [FFC], illustrated by the evaluation in (6.55).

(6.55)

Input:	F	H	C
	1	2	3
$H_a(x)$	$\bot$	$\bot$	$\bot$
$F_a(x)$	$\top$	$\top$	$\bot$
$C_a(x)$	$\bot$	$\bot$	$\top$
$H_e(x)$	$\bot$	$\bot$	$\bot$
$F_e(x)$	$\top$	$\top$	$\bot$
$C_e(x)$	$\bot$	$\bot$	$\top$
Output:	F	F	C

Note that the same order $a \ominus e$ in a compositional account would merely reproduce the serial ordering FF < HC. This composite system would accept the unattested *[HFC] from input /FFC/ as in (6.50). A parallel satisfaction analysis preserves the licensing/blocking structure pairs such that they refer to the same input string. Given that these pairs do not conflict, both orders—$e \ominus a$ and $a \ominus e$—formalize the counterfeeding interaction of HC and FF rules, accepting the mapping /FHC/ $\mapsto$ [FFC] observed in the data.

6.4.3 RC counterbleeds CC (OSL $\ominus$ ISL)

Combining BMRS systems with PS also models counterbleeding on environment. In Nanjing, the RC and CC disyllabic sandhi rules interact such that the former counterbleeds the latter. Evidence for this interaction comes from the trisyllabic output [LC′C] from input /RCC/; parallel derivations for this form are given in (6.56).

(6.56)

$$\begin{array}{ll}
\underline{\mathrm{RC}}\mathrm{C} & \\
| & \\
\mathrm{L}\underline{\mathrm{CC}} & \text{by RC rule} \\
| & \\
\mathrm{LC'C} & \text{by CC rule}
\end{array}
\qquad\qquad
\begin{array}{ll}
\mathrm{R}\underline{\mathrm{CC}} & \\
| & \\
\mathrm{RC'C} & \text{by CC rule} \\
| & \\
^*\mathrm{RC'C} & \text{RC rule n/a}
\end{array}$$

In the previous chapter, individual systems c and d were composed to formalize the counterbleeding interaction between RC and CC rules in Nanjing. Importantly, a particular order of composition over those systems ($d \otimes c$) expressed the counterbleeding order RC $<$ CC in a BMRS framework. This section uses the same systems c and d to model RC/CC counterbleeding, but instead makes use of the $\ominus$ operator. As the respective systems' licensing/blocking structures do not conflict, the order is irrelevant; that is, either order describes the function which maps /RCC/ to [LC′C]. Reference to a single input and a single output—recalling that system d describes an ROSL function—is sufficient to model this interaction.

First, consider the system in (6.57) that results from joining systems d and c via the $\ominus$ operator. This order $d \ominus c$ mirrors the order of the composite system $d \otimes c$ in the previous chapter. In the definition below, modifications to the outer function are given in bold, and the inner function is repeated for reference.

(6.57) a. $d \ominus c$

$$\begin{aligned}
H_{d\ominus c}(x) &= \mathbf{H_c(x)} \\
L_{d\ominus c}(x) &= \mathbf{L_c(x)} \\
R_{d\ominus c}(x) &= \mathbf{R_c(x)} \\
F_{d\ominus c}(x) &= \mathbf{F_c(x)} \\
C_{d\ominus c}(x) &= \texttt{if } \underline{C}C_{d\ominus c}(x) \texttt{ then } \bot \texttt{ else } \mathbf{C_c(x)} \\
C'_{d\ominus c}(x) &= \texttt{if } \underline{C}C_{d\ominus c}(x) \texttt{ then } \top \texttt{ else } \mathbf{C'_c(x)}
\end{aligned}$$

b. c

$$\begin{aligned}
H_c(x) &= H(x) \\
L_c(x) &= \texttt{if } \underline{R}C(x) \texttt{ then } \top \texttt{ else } L(x) \\
R_c(x) &= \texttt{if } \underline{R}C(x) \texttt{ then } \bot \texttt{ else } R(x) \\
F_c(x) &= F(x) \\
C_c(x) &= C(x) \\
C'_c(x) &= \bot
\end{aligned}$$

Final terms in each of d's boolean function definitions are replaced with the corresponding definition from c. Additionally, recursive function calls refer to the modified (PS-joined) system, denoted with the subscript '${}_{d\ominus c}$'. This has the effect of applying both functions simultaneously such that they compute outputs using the same input (and output) strings—without the intermediate representations that arise from serial ordering. The abbreviated table in (6.58) illustrates evaluation of input string /RCC/ against $d \ominus c$. It accepts the attested surface string [LC′C].

(6.58)

Input:	R	C	C
	1	2	3
$C'_{d\ominus c}(x)$	$\perp$	$\top$	$\perp$
$C_{d\ominus c}(x)$	$\perp$	$\perp$	$\top$
$L_{d\ominus c}(x)$	$\top$	$\perp$	$\perp$
$R_{d\ominus c}(x)$	$\perp$	$\perp$	$\perp$
$L_c(x)$	$\top$	$\perp$	$\perp$
$R_c(x)$	$\perp$	$\perp$	$\perp$
$C_c(x)$	$\perp$	$\top$	$\top$
Output:	L	C′	C

Applying the parallel satisfaction operator in this order yields a function similar in nature to the corresponding composite system $d \otimes c$. In particular, the input-orientedness of c's licensing/blocking structure pair ($\underline{R}C(x)$) is preserved in both cases. This is essential because it allows output L to be licensed on position 1 in spite of the fact that its successor is C′ in the *output*. In the case of composition, this is due to order: c is the inner function, and so remains unaffected. While switching the order of composition would disrupt the input-orientedness of c's $\underline{R}C(x)$ term (and thus predict the unattested *[RC′C] for input /RCC/), the same effect does not obtain with PS. The opposite order of $d \ominus c$—$c \ominus d$—is defined in (6.59). Note that the inner function d is modified such that its recursive function calls refer to the joined system (i.e. a single output string).

(6.59) a. $c \ominus d$

$$
\begin{aligned}
H_{c\ominus d}(x) &= \mathbf{H_d(x)} \\
L_{c\ominus d}(x) &= \texttt{if } \underline{R}C(x) \texttt{ then } \top \texttt{ else } \mathbf{L_d(x)} \\
R_{c\ominus d}(x) &= \texttt{if } \underline{R}C(x) \texttt{ then } \perp \texttt{ else } \mathbf{R_d(x)} \\
F_{c\ominus d}(x) &= \mathbf{F_d(x)} \\
C_{c\ominus d}(x) &= \mathbf{C_d(x)} \\
C'_{c\ominus d}(x) &= \mathbf{C'_d(x)}
\end{aligned}
$$

b. d

$$
\begin{aligned}
H_d(x) &= H(x) \\
L_d(x) &= L(x) \\
R_d(x) &= R(x) \\
F_d(x) &= F(x) \\
C_d(x) &= \texttt{if } \underline{C}C_{c\ominus d}(x) \texttt{ then } \perp \texttt{ else } C(x) \\
C'_d(x) &= \texttt{if } \underline{C}C_{c\ominus d}(x) \texttt{ then } \top \texttt{ else } \perp
\end{aligned}
$$

System c is the outer function, but the input-orientedness of term $\underline{R}C(x)$ is preserved nonetheless. Like $d \ominus c$ (and unlike composite $c \otimes d$), it describes the function which maps /RCC/ to [LC′C].

(6.60)

Input:	R	C	C
	1	2	3
$C'_{c\ominus d}(x)$	$\bot$	$\top$	$\bot$
$C_{c\ominus d}(x)$	$\bot$	$\bot$	$\top$
$L_{c\ominus d}(x)$	$\top$	$\bot$	$\bot$
$R_{c\ominus d}(x)$	$\bot$	$\bot$	$\bot$
$L_d(x)$	$\bot$	$\bot$	$\bot$
$R_d(x)$	$\top$	$\bot$	$\bot$
$C_d(x)$	$\bot$	$\bot$	$\top$
$C'_d(x)$	$\bot$	$\top$	$\bot$
Output:	L	C′	C

6.5 Xiamen

Another oft-cited sandhi ordering paradox comes from southern Min, and in particular Xiamen (Dong, 1960; Chen, 1987, 2000). Chen (1987) gives the following data from this five-tone system comprising R(ising), H(igh), M(id), L(ow), and F(alling) tones:[5]

(6.61)	a.	*p'ang*	H	"fragrant"
		p'ang tsui	MF	"perfume" (lit. fragrant + water)
	b.	*we*	R	"shoes"
		we tua	ML	"shoe laces"
	c.	*pĩ*	M	"sick"
		pĩ lang	LR	"patient" (lit. sick + person)
	d.	*ts'u*	L	"house"
		ts'u ting	FF	"roof" (lit. house + top)
	e.	*hai*	F	"ocean"
		hai kĩ	HR	"ocean front"

The Xiamen data present a *tone circle* sandhi pattern which Chen (2000, 42) describes as "a musical-chair pattern produced by the replacement of tone A by tone B, which is in turn replaced by tone C, and so forth." This is shown diagrammatically below.

[5] Chen gives Chao letter equivalents 24, 44, 22, 21, and 53 respectively. In addition, this analysis follows the conventional assumption that citation forms are underlying.

(6.62)

$$\begin{tikzcd} & R \arrow[d] & \\ & M \arrow[dr] & \\ H \arrow[ur] & & L \arrow[dl] \\ & F \arrow[ul] & \end{tikzcd}$$

Intuitively, a citation tone is realized as the 'next' tone in the circle when it appears in non-final position. Another way to state these generalizations is in the form of rewrite rules. A set of five rules in (6.63) describe the individual sandhi transformations in (6.61), where 'T' denotes any lexical tone.

(6.63)
a. R → M / __ T
b. H → M / __ T
c. M → L / __ T
d. L → F / __ T
e. F → H / __ T

Given the circular nature of the mappings in (6.62), Xiamen sandhi outputs cannot be produced by ordering individual rewrite rules. Thus like mutual counterfeeding and bleeding patterns in Changting, these data present an ordering paradox. Consider the ordering $e < d < c < b < a$ in (6.64) as an example; this produces some correct surface forms but not others.

(6.64)

Input	RT	MT	LT	FT	HT
e	–	–	–	HT	–
d	–	–	FT	–	–
c	–	LT	–	–	–
b	–	–	–	MT	MT
a	MT	–	–	–	–
Output	MT	LT	FT	*MT	MT

Indeed, any permutation over the rules will result in an unwanted feeding relationship between two rules; above, rule e feeds rule b over input /FT/ producing *[MT] when [HT] is attested. Xiamen tone sandhi can therefore be described as a type of circular counterfeeding. Rule ordering, and therefore a rule-based analysis, is ill-equipped to account for this phenomenon.

Constraint-based analyses have had limited success at explaining Xiamen sandhi, as well. An early admission of failure is due to Moreton (2004), who demonstrates that Xiamen and other circular chain shifts are non-computable functions in classic OT, in part due to their non-idempotence.

Numerous subsequent attempts have incorporated various extensions to OT, sometimes defining new families of constraints, with some being more desirable than others. This includes anti-merger constraints militating against syncretism in a tonal paradigm (Hsieh, 2005), contrast preservation (Barrie, 2006), so-called 'linear faithfulness' constraints defined over candidates which are mappings from sets of citation tones to sets of sandhi tones (Thomas, 2008), and comparative markedness (Hsiao, 2015), among others. However, the fact that the basic principles of the theory fail to derive the pattern—like serial ordering in an SPE framework—remains.

This has led many to question whether tone circles are psychologically-real for speakers or are present in synchronic grammars. Chen (2000, 42) notes that the Xiamen tone circle and others like it "often strike the analyst as bizarre because they seem to relate or map one tone to another in an essentially arbitrary and whimsical manner." In a footnote, he echoes earlier dismissals by Anderson (1987) and Ballard (1988), who assert that such patterns are irrelevant to questions of tonal phonology (at least a feature-based theory of tone), and go as far to say that the synchronic rules comprising them are "neither learnable, nor productive, in fact 'not a part of the speakers' grammars, but historical artifacts.'" This section will challenge that claim, showing that the Xiamen tone circle interaction is easily describable by a PS $\ominus$ operator over ISL functions, which have a demonstrably strong connection to phonological transformations, both tonal and segmental.

An important first observation is that the Xiamen tone circle, from the computational perspective, is remarkably simple: it describes an ISL-2 function. Example (6.65) gives a 'combined map' BMRS system for the function, defined over $\Sigma = \Gamma = \{M, L, H, F, R\}$. As above, 'T' is a shorthand for any lexical tone.

(6.65)
$$
\begin{array}{rcl}
M'(x) & = & \texttt{if } \underline{R}T(x) \texttt{ then } \top \texttt{ else} \\
 & & \texttt{if } \underline{H}T(x) \texttt{ then } \top \texttt{ else} \\
 & & \texttt{if } \underline{M}T(x) \texttt{ then } \bot \texttt{ else } M(x) \\
L'(x) & = & \texttt{if } \underline{M}T(x) \texttt{ then } \top \texttt{ else} \\
 & & \texttt{if } \underline{L}T(x) \texttt{ then } \bot \texttt{ else } L(x) \\
H'(x) & = & \texttt{if } \underline{F}T(x) \texttt{ then } \top \texttt{ else} \\
 & & \texttt{if } \underline{H}T(x) \texttt{ then } \bot \texttt{ else } H(x) \\
F'(x) & = & \texttt{if } \underline{L}T(x) \texttt{ then } \top \texttt{ else} \\
 & & \texttt{if } \underline{F}T(x) \texttt{ then } \bot \texttt{ else } F(x) \\
R'(x) & = & \texttt{if } \underline{R}T(x) \texttt{ then } \bot \texttt{ else } R(x)
\end{array}
$$

An evaluation table below shows that each mapping in the tone circle paradigm (6.61) is accepted by this combined map system, and nothing else.

(6.66)

Input:	R	T	M	T	L	T	F	T	H	T
	1	2	1	2	1	2	1	2	1	2
$M'(x)$	$\top$		$\bot$		$\bot$		$\bot$		$\top$	
$L'(x)$	$\bot$		$\top$		$\bot$		$\bot$		$\bot$	
$H'(x)$	$\bot$		$\bot$		$\bot$		$\top$		$\bot$	
$F'(x)$	$\bot$		$\bot$		$\top$		$\bot$		$\bot$	
$R'(x)$	$\bot$		$\bot$		$\bot$		$\bot$		$\bot$	
Output:	M	T	L	T	F	T	H	T	M	T

Despite its 'whimsical' nature, the ISL-ness of this interaction is uncontroversial. But relating the function in (6.65) to the individual rules in (6.63) is not straightforward. Because the tone circle presents an ordering paradox, this also means that any order of composition over equivalent BMRS systems of equations will recapitulate some serial ordering—it cannot capture the tone circle. Individual systems describing (6.63a-e) can, however, combine via the $\ominus$ operator. The resulting system is extensionally-equivalent to (6.65) and so accepts mappings consistent with the attested tone circle data. Demonstrating this is the main purpose of this section. Before doing so, the following section presents individual BMRS systems of equations describing the rules in (6.63).

6.5.1 Individual rules as systems of equations

Five separate BMRS systems define mappings between non-final citation and sandhi tones. This section defines them, where systems are denoted with letters $a - e$ following the ordering in (6.63) and defined over the alphabet $\Sigma = \Gamma = \{M, L, H, F, R\}$. In the definitions below, 'T' is a shorthand for any lexical tone, and omitted output boolean function definitions are understood to be direct mappings from the input. Importantly, each system describes an ISL function.

The rule in (6.63a) is formalized as a BMRS system a. It contains the licensing/blocking structure 'RT', a non-final R input tone followed by any other (input) tone.

(6.67)
$$M_a(x) = \texttt{if } \underline{R}T(x) \texttt{ then } \top \texttt{ else } M(x)$$
$$R_a(x) = \texttt{if } \underline{R}T(x) \texttt{ then } \bot \texttt{ else } R(x)$$

System b describes the rule in (6.63b) whereby any non-final input H maps to M.

(6.68)
$$M_b(x) = \texttt{if } \underline{H}T(x) \texttt{ then } \top \texttt{ else } M(x)$$
$$H_b(x) = \texttt{if } \underline{H}T(x) \texttt{ then } \bot \texttt{ else } H(x)$$

Similarly, the rule M → L / __ T in (6.63c) can be described by a BMRS system c, defined below.

$$(6.69)\quad \begin{aligned} L_c(x) &= \texttt{if}\ \underline{M}T(x)\ \texttt{then}\ \top\ \texttt{else}\ L(x) \\ M_c(x) &= \texttt{if}\ \underline{M}T(x)\ \texttt{then}\ \bot\ \texttt{else}\ M(x) \end{aligned}$$

Let d denote a BMRS system of equations describing the rule in (6.63d); this maps any non-final input L to F.

$$(6.70)\quad \begin{aligned} F_d(x) &= \texttt{if}\ \underline{L}T(x)\ \texttt{then}\ \top\ \texttt{else}\ F(x) \\ L_d(x) &= \texttt{if}\ \underline{L}T(x)\ \texttt{then}\ \bot\ \texttt{else}\ L(x) \end{aligned}$$

Finally, the rule in (6.63e) is formalized as a BMRS system e; the input sequence 'FT' (a non-final falling tone) licenses H in the output.

$$(6.71)\quad \begin{aligned} H_e(x) &= \texttt{if}\ \underline{F}T(x)\ \texttt{then}\ \top\ \texttt{else}\ H(x) \\ F_e(x) &= \texttt{if}\ \underline{F}T(x)\ \texttt{then}\ \bot\ \texttt{else}\ F(x) \end{aligned}$$

Systems $a - e$ thus comprise the individual "links" in the Xiamen tone circle pattern. Crucially these links do not join via composition. Instead, the PS operator $\ominus$ is applied to individual systems to model the circular chain shift in Xiamen. In doing so, the resulting system preserves the generalization implicit in the combined map (6.65); each licensing/blocking structure is computed over the same input string, not the intermediate representation generated by imparting a serial order on rules. The order in which these systems are combined—much like interactions in Changting—is inconsequential, owing to the fact that licensing/blocking structure pairs do not overlap.

6.5.2 Tone circle

The majority of function definitions in each system $a-e$ is a direct mapping from input symbols—i.e. is of the form $T'(x) = T(x)$. Applying $\ominus$ to these systems has a similar effect as interleaving definitions of the form $\texttt{if}\ T_1\ \texttt{then}\ T_1\ \texttt{else}\ T_3$. Depending on the order, a different hierarchy of licensing/blocking structures is created. For clarity, this section assembles a total order $e \ominus d \ominus c \ominus b \ominus a$ (the same order over rules as in (6.64)), but does so piece by piece. Example (6.72) joins systems e and d.

$$
\begin{array}{llllcl}
(6.72) & \text{a.} & e \ominus d & M_e(x) & = & \mathbf{M_d(x)} \\
 & & & L_e(x) & = & \mathbf{L_d(x)} \\
 & & & H_e(x) & = & \texttt{if } \underline{F}T(x) \texttt{ then } \top \texttt{ else } \mathbf{H_d(x)} \\
 & & & F_e(x) & = & \texttt{if } \underline{F}T(x) \texttt{ then } \bot \texttt{ else } \mathbf{F_d(x)} \\
 & & & R_e(x) & = & \mathbf{R_d(x)} \\
 & \text{b.} & d & M_d(x) & = & M(x) \\
 & & & L_d(x) & = & \texttt{if } \underline{L}T(x) \texttt{ then } \bot \texttt{ else } L(x) \\
 & & & H_d(x) & = & H(x) \\
 & & & F_d(x) & = & \texttt{if } \underline{L}T(x) \texttt{ then } \top \texttt{ else } F(x) \\
 & & & R_d(x) & = & R(x)
\end{array}
$$

Note that e and d's $F(x)$ function definitions both contain if-then-else statements; applying $\ominus$ in this way generates an ordered hierarchy with the former ranking above the latter. This ordering carries no consequences in terms of evaluation, since $\underline{F}T(x)$ and $\underline{L}T(x)$ do not conflict.

The next step is to combine $e \ominus d$ with c; this is given in (6.73).

$$
\begin{array}{llllcl}
(6.73) & \text{a.} & (e \ominus d) \ominus c & M_{e\ominus d}(x) & = & \mathbf{M_c(x)} \\
 & & & L_{e\ominus d}(x) & = & \texttt{if } \underline{L}T(x) \texttt{ then } \bot \texttt{ else } \mathbf{L_c(x)} \\
 & & & H_{e\ominus d}(x) & = & \texttt{if } \underline{F}T(x) \texttt{ then } \top \texttt{ else } \mathbf{H_c(x)} \\
 & & & F_{e\ominus d}(x) & = & \texttt{if } \underline{F}T(x) \texttt{ then } \bot \texttt{ else} \\
 & & & & & \texttt{if } \underline{L}T(x) \texttt{ then } \top \texttt{ else } \mathbf{F_c(x)} \\
 & & & R_{e\ominus d}(x) & = & \mathbf{R_c(x)} \\
 & \text{b.} & c & M_c(x) & = & \texttt{if } \underline{M}T(x) \texttt{ then } \bot \texttt{ else } M(x) \\
 & & & L_c(x) & = & \texttt{if } \underline{M}T(x) \texttt{ then } \top \texttt{ else } L(x) \\
 & & & H_c(x) & = & H(x) \\
 & & & F_c(x) & = & F(x) \\
 & & & R_c(x) & = & R(x)
\end{array}
$$

Another hierarchy is created with this application, this time within the output function definition of $L(x)$; the blocking structure $\underline{L}T(x)$ (originating in d) ranks above the licensing structure $\underline{M}T(x)$ from c. As before, no conflict arises.

System $(e \ominus d) \ominus c$ then combines with b to form $((e \ominus d) \ominus c) \ominus b$, abbreviated '$(e ... \ominus b$' in (6.74).

(6.74) a. $(e ... \ominus b$

$$\begin{aligned}
M_{(e\ominus d)\ominus c}(x) &= \texttt{if } \underline{M}T(x) \texttt{ then } \bot \texttt{ else } \mathbf{M_b(x)} \\
L_{(e\ominus d)\ominus c}(x) &= \texttt{if } \underline{L}T(x) \texttt{ then } \bot \texttt{ else} \\
&\quad\ \texttt{if } \underline{M}T(x) \texttt{ then } \top \texttt{ else } \mathbf{L_b(x)} \\
H_{(e\ominus d)\ominus c}(x) &= \texttt{if } \underline{F}T(x) \texttt{ then } \top \texttt{ else } \mathbf{H_b(x)} \\
F_{(e\ominus d)\ominus c}(x) &= \texttt{if } \underline{F}T(x) \texttt{ then } \bot \texttt{ else} \\
&\quad\ \texttt{if } \underline{L}T(x) \texttt{ then } \top \texttt{ else } \mathbf{F_b(x)} \\
R_{(e\ominus d)\ominus c}(x) &= \mathbf{R_b(x)}
\end{aligned}$$

b. b

$$\begin{aligned}
M_b(x) &= \texttt{if } \underline{H}T(x) \texttt{ then } \top \texttt{ else } M(x) \\
L_b(x) &= L(x) \\
H_b(x) &= \texttt{if } \underline{H}T(x) \texttt{ then } \bot \texttt{ else } H(x) \\
F_b(x) &= F(x) \\
R_b(x) &= R(x)
\end{aligned}$$

Two new hierarchies emerge. One is in the definition of output function $M(x)$, combining the blocking structure $\underline{M}T(x)$ from c and licensing structure $\underline{H}T(x)$ from b. The other is in $H(x)$; $\underline{H}T(x)$ from b—here a blocking structure—ranks below licensing structure $\underline{F}T(x)$, originally from system e.

Finally, the PS operator is applied to $((e \ominus d) \ominus c) \ominus b$ and the remaining system a to generate the full system $(((e \ominus d) \ominus c) \ominus b) \ominus a$. This is given in (6.75).

(6.75) a. $(e ... \ominus a$

$$\begin{aligned}
M_{(e...\ominus b}(x) &= \texttt{if } \underline{M}T(x) \texttt{ then } \bot \texttt{ else} \\
&\quad\ \texttt{if } \underline{H}T(x) \texttt{ then } \top \texttt{ else } \mathbf{M_a(x)} \\
L_{(e...\ominus b}(x) &= \texttt{if } \underline{L}T(x) \texttt{ then } \bot \texttt{ else} \\
&\quad\ \texttt{if } \underline{M}T(x) \texttt{ then } \top \texttt{ else } \mathbf{L_a(x)} \\
H_{(e...\ominus b}(x) &= \texttt{if } \underline{F}T(x) \texttt{ then } \top \texttt{ else} \\
&\quad\ \texttt{if } \underline{H}T(x) \texttt{ then } \bot \texttt{ else } \mathbf{H_a(x)} \\
F_{(e...\ominus b}(x) &= \texttt{if } \underline{F}T(x) \texttt{ then } \bot \texttt{ else} \\
&\quad\ \texttt{if } \underline{L}T(x) \texttt{ then } \top \texttt{ else } \mathbf{F_a(x)} \\
R_{(e...\ominus b}(x) &= \mathbf{R_a(x)}
\end{aligned}$$

b. a

$$\begin{aligned}
M_a(x) &= \texttt{if } \underline{R}T(x) \texttt{ then } \top \texttt{ else } M(x) \\
L_a(x) &= L(x) \\
H_a(x) &= H(x) \\
F_a(x) &= F(x) \\
R_a(x) &= \texttt{if } \underline{R}T(x) \texttt{ then } \bot \texttt{ else } R(x)
\end{aligned}$$

$\underline{R}T(x)$ is added to the hierarchy in $M(x)$ (as a licensing structure), and is interleaved (as a blocking structure) into the as yet direct-mapping definition of $R(x)$. Example (6.76) gives the full definition using simplified notation.

(6.76)
$$\begin{array}{lcl}
M'(x) & = & \texttt{if } \underline{M}T(x) \texttt{ then } \bot \texttt{ else} \\
 & & \texttt{if } \underline{H}T(x) \texttt{ then } \top \texttt{ else} \\
 & & \texttt{if } \underline{R}T(x) \texttt{ then } \top \texttt{ else } M(x) \\
L'(x) & = & \texttt{if } \underline{L}T(x) \texttt{ then } \bot \texttt{ else} \\
 & & \texttt{if } \underline{M}T(x) \texttt{ then } \top \texttt{ else } L(x) \\
H'(x) & = & \texttt{if } \underline{F}T(x) \texttt{ then } \top \texttt{ else} \\
 & & \texttt{if } \underline{H}T(x) \texttt{ then } \bot \texttt{ else } H(x) \\
F'(x) & = & \texttt{if } \underline{F}T(x) \texttt{ then } \bot \texttt{ else} \\
 & & \texttt{if } \underline{L}T(x) \texttt{ then } \top \texttt{ else } F(x) \\
R'(x) & = & \texttt{if } \underline{R}T(x) \texttt{ then } \bot \texttt{ else } R(x)
\end{array}$$

Combining a series of ISL systems via $\ominus$ yields a single ISL function. In this system, all licensing/blocking structure pairs are evaluated over the same input string. This is key, because it guarantees that no single 'rule' will feed another rule, and so successfully models the circular counterfeeding interaction between those rules. Relatedly, the ISL function in (6.76) is not intensionally-equivalent to the function described in (6.65), but the two are extensionally-equivalent. It accepts the circular chain shift maps conceptualized in (6.62), as shown in the evaluation table below.

(6.77)

Input:	R	T	M	T	L	T	F	T	H	T
	1	2	1	2	1	2	1	2	1	2
$M'(x)$	$\top$		$\bot$		$\bot$		$\bot$		$\top$	
$L'(x)$	$\bot$		$\top$		$\bot$		$\bot$		$\bot$	
$H'(x)$	$\bot$		$\bot$		$\bot$		$\top$		$\bot$	
$F'(x)$	$\bot$		$\bot$		$\top$		$\bot$		$\bot$	
$R'(x)$	$\bot$		$\bot$		$\bot$		$\bot$		$\bot$	
Output:	M	T	L	T	F	T	H	T	M	T

Output boolean function definitions across both systems comprise a single set of if-then-else statements which can be thought of as contributions from the separate systems $a - e$. What differs is the relative ordering of those statements within a hierarchy. For example, $L'(x)$ in the joined system (6.76) first evaluates the blocking structure $\underline{L}T(x)$, then the licensing structure $\underline{M}T(x)$; the corresponding function in the combined map system (6.76) evaluates the opposite order.

Yet both are equivalent. This is explained by the fact that no two licensing/blocking structure pairs in $a - e$ are in conflict; they describe distinct input structures. Thus switching the order of the

PS operator over these systems—thereby rearranging the hierarchies apparent within single function definitions—has no impact on the set of mappings accepted by the resulting system, i.e. the function it describes. The reverse PS-join of (6.76), $a \ominus b \ominus c \ominus d \ominus e$, is given below in (6.78) using simplified notation.

(6.78)
$$\begin{aligned}
M'(x) &= \texttt{if } \underline{R}T(x) \texttt{ then } \top \texttt{ else} \\
&\quad\; \texttt{if } \underline{H}T(x) \texttt{ then } \top \texttt{ else} \\
&\quad\; \texttt{if } \underline{M}T(x) \texttt{ then } \bot \texttt{ else } M(x) \\
L'(x) &= \texttt{if } \underline{M}T(x) \texttt{ then } \top \texttt{ else} \\
&\quad\; \texttt{if } \underline{L}T(x) \texttt{ then } \bot \texttt{ else } L(x) \\
H'(x) &= \texttt{if } \underline{H}T(x) \texttt{ then } \bot \texttt{ else} \\
&\quad\; \texttt{if } \underline{F}T(x) \texttt{ then } \top \texttt{ else } H(x) \\
F'(x) &= \texttt{if } \underline{L}T(x) \texttt{ then } \top \texttt{ else} \\
&\quad\; \texttt{if } \underline{F}T(x) \texttt{ then } \bot \texttt{ else } F(x) \\
R'(x) &= \texttt{if } \underline{R}T(x) \texttt{ then } \bot \texttt{ else } R(x)
\end{aligned}$$

Note that the definition-internal hierarchies expressed in the system above are the mirror image of those in (6.76). Due to the lack of conflict in the expressed structures, however, the systems in (6.78), (6.76), and (6.65) all describe the same function. This is the function that maps Xiamen citation tones to corresponding sandhi tones in a circular fashion.

6.6 Discussion

A variety of opaque and transparent sandhi interactions can be modeled by the simultaneous application of two or more rules. To formalize this effect in a BMRS framework, this chapter has defined a parallel satisfaction operator $\ominus$ over BMRS systems of equations. Applied to individual systems (i.e. those which describe single rules), the operator models counterbleeding, bleeding, and counterfeeding relationships which are not derivable via conventional rule ordering, and thus cannot be described as the composition of systems using $\otimes$. The discussion presented here summarizes these generalizations and explores their ramifications.

First, this section reiterates the effect of preserving single input and output strings via $\ominus$, showing that failure to preserve the latter recreates ordering paradoxes in Changting. It also relates this effect to the absence of intermediate representations, which Oakden and Chandlee (2020) argue is the key to the Changting paradigm. This insight unifies seemingly disconnected interactions—mutual counterbleeding/bleeding in Changting, classic counterfeeding/counterbleeding on environment in

Nanjing, and the Min tone circle—which have posed different challenges for rule- and constraint-based theories of phonology.

The same quality of $\ominus$ also limits its ability to formalize certain interactions, namely transparent feeding and bleeding which depend on the information present in intermediate forms. This is illustrated with a feeding interaction analyzed in the previous chapter. Finally, this chapter presents a demonstration that the $\ominus$ operator is not commutative (i.e. order does matter), despite the fact that order does not affect many of the joined systems modeling tone sandhi interactions.

6.6.1 One input, one output

This section reiterates a key component of the $\ominus$ operator: collapsing the input/output structures referred to by separate BMRS systems into a single input and output string.

6.6.1.1 Enforcing a single output

When two systems join by $\ominus$, ensuring a single output requires that recursion in either system refer to the PS-joined system itself. This guarantees that it is evaluated using one output string and not two. Limiting computation to a single output string is crucial, for example, in the mutual bleeding interaction in Changting (§3.3). To see why, consider a system $c \ominus' d$ (defined in (6.79)); it combines two systems in the usual way (i.e. by replacing final terms), but also preserves recursion from the separate systems.

(6.79) a. $c \ominus' d$

$$\begin{aligned} H_c(x) &= \mathbf{H_d(x)} \\ R_c(x) &= \mathbf{R_d(x)} \\ F_c(x) &= \mathbf{F_d(x)} \\ M_c(x) &= \texttt{if } \underline{L}M_c(x) \texttt{ then } \top \texttt{ else } \mathbf{M_d(x)} \\ L_c(x) &= \texttt{if } \underline{L}M_c(x) \texttt{ then } \bot \texttt{ else } \mathbf{L_d(x)} \end{aligned}$$

b. d

$$\begin{aligned} H_d(x) &= H(x) \\ R_d(x) &= R(x) \\ F_d(x) &= F(x) \\ M_d(x) &= \texttt{if } \underline{M}L_d(x) \texttt{ then } \bot \texttt{ else } M(x) \\ L_d(x) &= \texttt{if } \underline{M}L_d(x) \texttt{ then } \top \texttt{ else } L(x) \end{aligned}$$

Combining two systems in this way generates two separate *outputs* over which output boolean function definitions across both systems are evaluated. This gives rise to the intermediate representations that characterize the ordering paradox over rules, an undesirable result. When $c \ominus' d$ evaluates the relevant three-tone sequences /MLM/ and /LML/ from Changting, it accepts the same unattested mappings as would the composition $c \otimes d$. That is, it recapitulates the rule ordering ML < LM: /LML/ correctly maps to [LLL], but /MLM/ maps to the unattested *[LMM]. An evaluation table in (6.80) illustrates the relevant mappings accepted by $c \ominus' d$.

(6.80)

Input:	M	L	M	L	M	L
	1	2	3	1	2	3
$H_c(x)$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$
$R_c(x)$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$
$F_c(x)$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$
$M_c(x)$	$\bot$	$\top$	$\top$	$\bot$	$\bot$	$\bot$
$L_c(x)$	$\top$	$\bot$	$\bot$	$\top$	$\top$	$\top$
$H_d(x)$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$
$R_d(x)$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$
$F_d(x)$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$
$M_d(x)$	$\bot$	$\bot$	$\top$	$\bot$	$\bot$	$\bot$
$L_d(x)$	$\top$	$\top$	$\bot$	$\top$	$\top$	$\top$
Output:	*L	M	M	L	L	L

The crux of the issue lies in the evaluation of input /MLM/'s second string position. It returns a true value for both $M_c(x)$ (by conforming to the licensing structure $\underline{L}M_c(x)$) *and* $L_d(x)$ (maps directly from input L as a default). Since the joined system comprises recursive function calls from two systems c and d, computation refers to two distinct output structures while moving through the string. When string position 1 is evaluated against definitions $M_c(x)$ and $L_c(x)$, it skips the first two terms in the if-then-else statements—since the current position is not input-specified as L. This moves evaluation directly to relevant definitions from system d. There, it checks for licensing/blocking structures defined in terms *of the output generated by system d*.

Since string position 2 is true for $L_d(x)$, position 1 returns a true value for $L_d(x)$, modeling the application of the ML rule, and unaware of the 'surface' output generated by system c. This also means that $L_c(x)$ evaluates to true. String position 1 surfaces as L, thus producing the unattested *[LMM]. In intuitive terms, the availability of output string positions produced by d (the application of the ML rule) results in a *failed* blocking of ML.

To prod the issue further, we may reverse the order of this operation over systems c and d and observe its effects. Consider $d \ominus' c$, defined in (6.81).

(6.81) a. $d \ominus' c$

$$\begin{aligned}
H_d(x) &= \mathbf{H_c}(\mathbf{x})\\
R_d(x) &= \mathbf{R_c}(\mathbf{x})\\
F_d(x) &= \mathbf{F_c}(\mathbf{x})\\
M_d(x) &= \texttt{if } \underline{M}L_d(x) \texttt{ then } \bot \texttt{ else } \mathbf{M_c}(\mathbf{x})\\
L_d(x) &= \texttt{if } \underline{M}L_d(x) \texttt{ then } \top \texttt{ else } \mathbf{L_c}(\mathbf{x})
\end{aligned}$$

b. c

$$\begin{aligned}
H_c(x) &= H(x)\\
R_c(x) &= R(x)\\
F_c(x) &= F(x)\\
M_c(x) &= \texttt{if } \underline{L}M_c(x) \texttt{ then } \top \texttt{ else } M(x)\\
L_c(x) &= \texttt{if } \underline{L}M_c(x) \texttt{ then } \bot \texttt{ else } L(x)
\end{aligned}$$

As before, computation proceeds with reference to two distinct output strings. What differs is that now it is the output of d that constitutes the 'surface' output of the function described by the system. Given the situation described above, a reasonable prediction is that the reversed PS-join $d \ominus' c$ would mirror the predictions of rule ordering LM < ML: /MLM/ correctly maps to [MMM], but /LML/ maps to the unattested *[MLL]. This prediction is confirmed, as illustrated by the evaluation table in (6.82).

(6.82)

Input:	M	L	M	L	M	L
	1	2	3	1	2	3
$H_d(x)$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$
$R_d(x)$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$
$F_d(x)$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$
$M_d(x)$	$\top$	$\top$	$\top$	$\top$	$\bot$	$\bot$
$L_d(x)$	$\bot$	$\bot$	$\bot$	$\bot$	$\top$	$\top$
$H_c(x)$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$
$R_c(x)$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$
$F_c(x)$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$
$M_c(x)$	$\top$	$\top$	$\top$	$\top$	$\top$	$\bot$
$L_c(x)$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\top$
Output:	M	M	M	*M	L	L

A discrepancy in each system's output for string position 2 and the resulting evaluation is the locus of the anomaly, as before. Reproducing the effects of rule ordering LM < ML is also apparent; the ML rule is 'blocked' as evinced by /MLM/ $\mapsto$ [MMM], but the failure to block LM produces the aberrant /LML/ $\mapsto$ *[MLL].

Interactions between OSL functions illustrate the motivation for enforcing a single output when $\ominus$ is applied. Failing to do so generates the intermediate representations which give rise to ordering paradoxes.

6.6.1.2 No intermediate representations

Joining systems of equations via $\ominus$ yields functions whose outputs are computed with reference to one input and one output string. This models simultaneous application of individual (i.e. separate) rules; application neither generates nor is sensitive to intermediate representations like those generated in a serial ordering framework.

Opaque and transparent interactions explored in this chapter are thus unified in their amenability to an analysis using PS. What counterfeeding/counterbleeding on environment (in Nanjing), mutual counterfeeding and mutual bleeding (in Changting), and so-called circular counterfeeding (in Xiamen) all have in common is that they can be understood as the sum of their parts—i.e. in terms of the individual 'rules' which comprise interactions—but that those parts refer to the same input/output structure, and no inherent order between the parts is necessary. This seems to clash with intuitions about counterfeeding/counterbleeding on environment, which assume, as part of their definition, a specific ordering of rules. While an ordering analysis is available to these cases, as illustrated in the preceding chapter, this chapter has shown that it is not necessary.

That a PS analysis is equally tenable for these cases is also noteworthy given that they pose different challenges to rule- and constraint-based formalisms. Classic opaque interactions are trivial in a serial framework. Ordering paradoxes such as those posed by transparent and opaque interactions Changting, on the other hand, are intractable. And while Changting has been argued to exhaust the limits of OT because of its directionality effects (Chen et al., 2004), it is instead the non-idempotency of circular chain shifts in Xiamen which have earned it the title of 'non-computable function' (Moreton, 2004).[6] In a BMRS framework, these interactions can be accounted for as the PS-join of individual systems of equations; that is, they share the crucial assumption that 'rules' apply simultaneously with bounded reference to a single input and output string. Combined with the SL perspective on directionality developed in chapter 4, the computational framework advocated here provides a unified solution to these sandhi interactions, some of which are outstanding cases in the literature.

[6]This is different from linear chain shifts, e.g. the classic case of Danish vowel lowering (Hyman, 1975; Lundskær-Nielsen and Holmes, 2011). (Kirchner, 1996) shows that linear chain shifts like Danish are computable with classic OT grammars. (Chandlee et al., 2018) show that, as a single function, Danish vowel lowering is ISL. In terms of the analyses proposed here, the vowel lowering chain shift is amenable to an analysis as a PS-join of three functions. However since there is no ordering paradox in a rule-based account, it can also be modeled using a BMRS compositional analysis assuming a specific order.

6.6.2 Interactions not formalized by $\ominus$

BMRS analyses using the $\ominus$ operator model interactions which do not depend on intermediate representations. This also means that they cannot formalize analyses of feeding and bleeding interactions for which intermediate representations are a crucial component. To illustrate, recall the feeding relationship between LF and RM rules in Changting, repeated in (6.83).

(6.83) LF rule feeds RM rule: R<u>LF</u> → <u>R**M**</u>F → **H**MF

In the previous chapter, this interaction was formalized as the composition of two BMRS systems $b \otimes a$. The outer function b is evaluated in terms of modifications to the input string made by the inner function a. Thus for an input string /RLF/, system a 'feeds' an intermediate M tone (on the second string position) which, in conjunction with the input R tone in the first string position, licenses an output H. Importantly, only this order of composition produces the desired effect.

This effect disappears when the same systems combine with the PS operator, as in (6.84) below.

(6.84) a. $b \ominus a$

$$\begin{aligned}
H_b(x) &= \texttt{if } \underline{R}M(x) \texttt{ then } \top \texttt{ else } \mathbf{H_a(x)} \\
R_b(x) &= \texttt{if } \underline{R}M(x) \texttt{ then } \bot \texttt{ else } \mathbf{R_a(x)} \\
F_b(x) &= \mathbf{F_a(x)} \\
M_b(x) &= \mathbf{M_a(x)} \\
L_b(x) &= \mathbf{L_a(x)}
\end{aligned}$$

b. a

$$\begin{aligned}
H_a(x) &= H(x) \\
R_a(x) &= R(x) \\
F_a(x) &= F(x) \\
M_a(x) &= \texttt{if } \underline{L}F(x) \texttt{ then } \top \texttt{ else } M(x) \\
L_a(x) &= \texttt{if } \underline{L}F(x) \texttt{ then } \bot \texttt{ else } L(x)
\end{aligned}$$

When input /RLF/ is evaluated against $b \ominus a$, licensing structures $\underline{L}F(x)$ ($M_a(x)$) and $\underline{R}M(x)$ ($H_b(x)$) compute truth values by examining the same input string. Because systems b and a evaluate in parallel, neither can feed the other. The evaluation table in (6.85) demonstrates this effect.

(6.85)

Input:	R	L	F
	1	2	3
$H_b(x)$	$\bot$	$\bot$	$\bot$
$R_b(x)$	$\top$	$\bot$	$\bot$
$F_b(x)$	$\bot$	$\bot$	$\top$
$M_b(x)$	$\bot$	$\top$	$\bot$
$L_b(x)$	$\bot$	$\bot$	$\bot$
$H_a(x)$	$\bot$	$\bot$	$\bot$
$R_a(x)$	$\top$	$\bot$	$\bot$
$F_a(x)$	$\bot$	$\bot$	$\top$
$M_a(x)$	$\bot$	$\top$	$\bot$
$L_a(x)$	$\bot$	$\bot$	$\bot$
Output:	*R	M	F

Term $\underline{R}M(x)$ evaluates to false for input /RLF/. That string position 2 returns a true value for $M_a(x)$ is irrelevant; an output H is licensed by an input RM sequence only. Feeding cannot be formalized as a PS-join of systems in this way.

Nor can switching the order of systems b and a. Recall that, throughout the chapter, both orders on systems with non-conflicting licensing/blocking structures generate extensionally-equivalent functions. The same generalization holds for the operator's failure to formalize transparent feeding, illustrated by the mirror image of $b \ominus a$ in (6.86).

(6.86) a. $a \ominus b$

$$\begin{aligned}
H_a(x) &= \mathbf{H_b(x)} \\
R_a(x) &= \mathbf{R_b(x)} \\
F_a(x) &= \mathbf{F_b(x)} \\
M_a(x) &= \texttt{if } \underline{L}F(x) \texttt{ then } \top \texttt{ else } \mathbf{M_b(x)} \\
L_a(x) &= \texttt{if } \underline{L}F(x) \texttt{ then } \bot \texttt{ else } \mathbf{L_b(x)}
\end{aligned}$$

b. b

$$\begin{aligned}
H_b(x) &= \texttt{if } \underline{R}M(x) \texttt{ then } \top \texttt{ else } M(x) \\
R_b(x) &= \texttt{if } \underline{R}M(x) \texttt{ then } \bot \texttt{ else } R(x) \\
F_b(x) &= F(x) \\
M_b(x) &= M(x) \\
L_b(x) &= L(x)
\end{aligned}$$

Applied to /RLF/, $a \ominus b$ accepts the unattested *[RMF] by the same evaluation. Transparent interactions whose analyses rely on intermediate representations—in particular rule application triggered by intermediate representations—cannot be formalized in a BMRS framework using the parallel satisfaction operator.

6.6.3 Does order matter?

Unlike composition with the $\otimes$ operator, the nature of the resulting function does not covary with the order in which individual systems are joined via $\ominus$. That is, given two BMRS systems 1 and 2 whose licensing and blocking structures do not conflict, $1 \ominus 2$ describes the same function as $2 \ominus 1$. This property holds when the crucial requirement of non-conflict is satisfied, and applies to ISL functions, OSL functions, and various combinations of thereof.

Individual systems typically contain a single licensing structure and a single blocking structure, distributed across two separate output boolean function definitions. A common scenario is one in which two systems define structures over non-overlapping function definitions. When they combine through $\ominus$, if-then-else statements containing those licensing/blocking structures replace default direct mapping definitions ($T'(x) = T(x)$). For example, joined systems describing the interaction of Nanjing CC and RC rules (§4.3) are of this type: a system d models the CC rule with a licensing/blocking structures in definitions $C_d(x)$ and $C'_d(x)$ while a system c models the RC rule with licensing/blocking structures in definitions $L_c(x)$ and $R_c(x)$. Both $d \ominus c$ and $c \ominus d$ generate the same system in (6.87), denoted 'u'.

(6.87)
$$
\begin{array}{lcl}
H_u(x) & = & H(x) \\
L_u(x) & = & \texttt{if } \underline{R}C(x) \texttt{ then } \top \texttt{ else } L(x) \\
R_u(x) & = & \texttt{if } \underline{R}C(x) \texttt{ then } \bot \texttt{ else } R(x) \\
F_u(x) & = & F(x) \\
C_u(x) & = & \texttt{if } \underline{C}C_u(x) \texttt{ then } \bot \texttt{ else } C(x) \\
C'_u(x) & = & \texttt{if } \underline{C}C_u(x) \texttt{ then } \top \texttt{ else } \bot
\end{array}
$$

Sometimes, this distribution overlaps, as in the case of Xiamen, creating a hierarchy of licensing and blocking structures within a single definition. Opposite orders of PS-join thus generate opposite hierarchies. For example, compare the respective definitions of Xiamen output M tone $M'(x)$ in a system $e \ominus d \ominus c \ominus b \ominus a$ and in a system which reverses the order $a \ominus b \ominus c \ominus d \ominus e$, in (6.88).

(6.88) a. $e \ominus d \ominus c \ominus b \ominus a$
$$
\begin{array}{lcl}
M'(x) & = & \texttt{if } \underline{M}T(x) \texttt{ then } \bot \texttt{ else} \\
 & & \texttt{if } \underline{H}T(x) \texttt{ then } \top \texttt{ else} \\
 & & \texttt{if } \underline{R}T(x) \texttt{ then } \top \texttt{ else } M(x)
\end{array}
$$

b. $a \ominus b \ominus c \ominus d \ominus e$
$$
\begin{array}{lcl}
M'(x) & = & \texttt{if } \underline{R}T(x) \texttt{ then } \top \texttt{ else} \\
 & & \texttt{if } \underline{H}T(x) \texttt{ then } \top \texttt{ else} \\
 & & \texttt{if } \underline{M}T(x) \texttt{ then } \bot \texttt{ else } M(x)
\end{array}
$$

The hierarchy in (6.88a) is a mirror image of the hierarchy in (6.88b). However, since the structures described by terms $\underline{R}T(x)$, $\underline{H}T(x)$, and $\underline{M}T(x)$ are non-conflicting, $M'(x)$ functions in both systems evaluate input-output mappings in the same way. Therefore, even when combination through $\ominus$ generates hierarchies of licensing/blocking structures that are not apparent in either individual system, this does not affect evaluation provided the structures do not conflict.

More generally, however, the $\ominus$ operation is *not* commutative; there exist cases in which varying order of PS-join does result in a different function. As a simple illustration, consider two functions defined over an alphabet $\Sigma = \{a, b\}$. Function 1 maps any string-final position to b, as in (6.89). It is an ISL-2 function.

(6.89)
$$\begin{aligned} a_1(x) &= \texttt{if } \#(s(x)) \texttt{ then } \bot \texttt{ else } a(x) \\ b_1(x) &= \texttt{if } \#(s(x)) \texttt{ then } \top \texttt{ else } b(x) \end{aligned}$$

In this definition, the structure $\#(s(x))$—immediate successor is a word edge—licenses output b and blocks an output a. Now consider Function 2 (6.90), which maps any string-final position to a. It is also an ISL-2 function.

(6.90)
$$\begin{aligned} a_2(x) &= \texttt{if } \#(s(x)) \texttt{ then } \top \texttt{ else } a(x) \\ b_2(x) &= \texttt{if } \#(s(x)) \texttt{ then } \bot \texttt{ else } b(x) \end{aligned}$$

The same structure has the opposite effect in this system: it blocks an output b and licenses an output a. Across both output boolean function definitions $a_{1/2}(x)$ and $b_{1/2}(x)$, a conflict arises in which the same structure licenses the output in one function but blocks it in another.

When 1 and 2 join via $\ominus$, a conflicting hierarchy is generated. Whether output a or b surfaces in string-final position is entirely dependent on order. Example (6.91) below represents the hierarchy created by the order $1 \ominus 2$.

(6.91)
$$\begin{aligned} a_{1\ominus 2}(x) &= \texttt{if } \#(s(x)) \texttt{ then } \bot \texttt{ else} \\ &\quad\;\; \texttt{if } \#(s(x)) \texttt{ then } \top \texttt{ else } a(x) \\ b_{1\ominus 2}(x) &= \texttt{if } \#(s(x)) \texttt{ then } \top \texttt{ else} \\ &\quad\;\; \texttt{if } \#(s(x)) \texttt{ then } \bot \texttt{ else } b(x) \end{aligned}$$

The outer function 1 determines the string-final output, as hierarchies are evaluated from the first term downward. As a result, $1 \ominus 2$ only accepts mappings from strings of as and bs to b-final strings, illustrated in the evaluation in (6.92), where x denotes either a or b.

(6.92)

Input:	...	x	$\#$
$a_{1\ominus 2}(x)$		$\bot$	
$b_{1\ominus 2}(x)$		$\top$	
Output:	...	b	

Under the reversed order $2 \ominus 1$ (given below in (6.93)), however, the same principle dictates that the outer function 2 determines the string final output.

(6.93)
$$
\begin{aligned}
a_{2\ominus 1}(x) &= \texttt{if } \#(s(x)) \texttt{ then } \top \texttt{ else} \\
&\quad \texttt{if } \#(s(x)) \texttt{ then } \bot \texttt{ else } a(x) \\
b_{2\ominus 1}(x) &= \texttt{if } \#(s(x)) \texttt{ then } \bot \texttt{ else} \\
&\quad \texttt{if } \#(s(x)) \texttt{ then } \top \texttt{ else } b(x)
\end{aligned}
$$

This means that $2 \ominus 1$ only accepts mappings from strings of *a*s and *b*s to *a*-final strings (see the evaluation table in (6.94)), and thus describes a different function from $1 \ominus 2$.

(6.94)

Input:	...	x	$\#$
$a_{2\ominus 1}(x)$		$\bot$	
$b_{2\ominus 1}(x)$		$\top$	
Output:	...	a	

Edge cases such as those above are beyond the scope of this dissertation and are therefore left for future work. Rather, the focus of this chapter is on the relevance of $\ominus$ to formalizing tone sandhi interactions, in particular those which present ordering paradoxes. Modeling simultaneous application of rules over a single input and output string—while still allowing reference to those strings—successfully derives interaction mappings in a number of outstanding cases.

7 Discussion

7.1 Introduction

The preceding chapters develop a formal theory of phonological process interactions. Such a theory formalizes interactions as a set of individual functions (as BMRS systems) that combine using a distinct set of operators. This set includes composition $\otimes$ and parallel satisfaction $\ominus$, and these are sufficient to model a series of sandhi interactions in Chinese tone. In this chapter I pause to interpret these results and discuss their ramifications.

There are three main goals to this chapter: to highlight the benefits of BMRS in modeling phonological process interactions, especially compared to other formalisms (§2); to expand on underdeveloped/overlooked issues introduced in previous chapters (§§3-4); and to identify new avenues of research using the formalism developed within the dissertation (§5).

7.2 Interactions over BMRS vs other formalisms

This chapter begins with a continuation of the discussion in chapter 2 §5.2, that is, the advantages of the BMRS formalism in modeling phonological process interactions. It identifies benefits of BMRS through a comparison of BMRS analyses of interactions (§2.1) and definition of function operations over BMRS (§2.2) with other computational formalisms, namely logical and finite-state frameworks. It is important to note that these formalisms are equivalent in the sense that they describe the same class of functions; BMRS systems of equations (Bhaskar et al., 2020), QFLFP logical transductions (Chandlee and Jardine, 2019b) and deterministic finite-state transducers (Mohri, 1997) all describe the subsequential class. Instead, the focus here is on how one formalism might be more conducive to analysis of phonological process interactions, the main focus of this dissertation.

7.2.1 Modeling interactions: BMRS vs QFLFP

BMRS systems of equations model phonological transformations via a series of statements about how output structure is computed. These statements identify structures in the input (using non-recursive definitions) or output (using recursive definitions) which license certain output elements or block them. Thus the underlying motivation for phonological processes—in the form of marked/ill-formed structures—can be stated clearly. Processes in isolation are definable as single functions, and interactions with other processes can be modeled as combinations of individual functions with operators. This clarifies the contribution of each individual process to the interaction map.

Formalization using QF logical transduction (see Lindell and Chandlee, 2016) do not share the same properties. Properly-QF transductions model input-oriented phonological processes, but not output-oriented ones. While enhancements to QF using least-fixed point logic (QFLFP; see Chandlee and Jardine, 2019b, for more details as well as (Libkin, 2013)) can describe output-oriented processes, these definitions can become unruly when modeling multiple generalizations. Stated briefly, QFLFP transductions define predicates which refer to the output structure through inductive definitions, but still in a restricted local way. These predicates contain base case (necessarily input-oriented) and recursive case (using a free set variable) components to allow for output reference, for example in iterative rule application or non-(input-)local spreading patterns. Unlike the BMRS formalism, the motivation for the processes being described tend to be obscured by these logical statements. And without a means to combine individual generalizations via operations like composition, combined map functions of even simple interactions are difficult to define and interpret. To illustrate, recall the functions modeling the Tianjin FF (FF $\rightarrow$ LF) and LL (LL $\rightarrow$ RL) sandhi rules from chapter 5. Each 'rule' is defined as a separate system, denoted a and b, respectively. I focus on functions marking output elements as L(ow) tones, repeated in (7.1a-b) below.

(7.1) a. $L_a(x) \;=\; \texttt{if}\ \underline{F}F_a(x)\ \texttt{then}\ \top\ \texttt{else}\ L(x)$

b. $L_b(x) \;=\; \texttt{if}\ \underline{L}L_b(x)\ \texttt{then}\ \bot\ \texttt{else}\ L(x)$

The definitions clearly identify structures that license or block output L: a sequence of one input F and one output F in the former, and a sequence of one input L and one output L in the latter. Equivalent QFLFP definitions from Oakden (2019a)'s analysis of Tianjin interactions are given in (7.2a-b).[1]

(7.2)

a. $P_L^a(x) = P_L(x) \vee [\texttt{lfp}(P_F(y) \wedge \texttt{last}_F(s(y)) \wedge \neg\texttt{last}(y)) \vee (A(s(s(y))) \wedge P_F(y) \wedge P_F(s(y)))](x)$

b. $P_L^b(x) = P_L(x) \wedge \neg[\texttt{lfp}(P_L(y) \wedge \texttt{last}_L(s(y)) \wedge \neg\texttt{last}(y)) \vee (A(s(s(y))) \wedge P_L(y) \wedge P_L(s(y)))](x)$

QFLFP definitions above represent the same logical statements as those in (7.1), but are much more opaque regarding the relevant structures that determine the computation of output L.

[1]This comparison is a bit unfair, because the QFLFP analysis assumes a total successor function, requiring more statements than assuming a partial successor function. A partial successor function analysis would not, however, improve the legibility of these definitions to a great extent, as it would merely allow deletion of the $\neg\texttt{last}(y)$ conjuncts in the $\texttt{lfp}$ base cases.

In Tianjin, the FF rule feeds the LL rule. Chapter 5 presented a BMRS analysis of the interaction by composing b with a. Output function $L_{a \otimes b}(x)$ is defined simply by indexing non-recursively-defined functions with their equivalents from system a. This is given in (7.3).

$$(7.3) \quad L_{a\otimes b}(x) \quad = \quad \texttt{if } \underline{L_a} L_b(x) \texttt{ then } \bot \texttt{ else } L(x)$$

A new blocking structure $\underline{L_a}L_b(x)$ indicates the relationship between the two sandhi rules: LL blocks output L when the first L is either an input L (recalling the final term from $L_a(x)$) or a *derived* L from the FF rule. Without a composition operator over QFLFP transductions, the equivalent logical statement can only join the definitions in (7.2) by disjunction.

$$\begin{aligned}(7.4) \quad P'_L(x) \quad = \quad & (P_L(x) \wedge \neg[\texttt{lfp}(P_L(y) \wedge \texttt{last}_L(s(y)) \wedge \neg\texttt{last}(y)) \vee (A(s(s(y))) \\ & \wedge P_L(y) \wedge P_L(s(y)))](x)) \vee ([\texttt{lfp}(P_F(y) \wedge \texttt{last}_F(s(y)) \\ & \wedge\neg\texttt{last}(y)) \vee (A(s(s(y))) \wedge P_F(y) \wedge P_F(s(y)))](x))\end{aligned}$$

Not only is this definition difficult to interpret and manipulate, it also fails to explicate the FF/LL interaction in any meaningful way.[2] BMRS definitions, on the other hand, are transparent with regard to the motivation for processes and in the interplay of multiple generalizations.

7.2.2 Defining operators: BMRS vs FST

As mentioned above, BMRS offers an intuitive framework for defining operations over functions. This dissertation has defined two such operators—composition and parallel satisfaction. These join two separate functions into a single function by manipulating well-formed BMRS terms in different ways. In this section I compare BMRS operators to operations over finite-state machines. This is because operations (such as composition) in the finite-state formalism are well-understood, especially when compared to logical transduction. I conjecture on the extent to which BMRS provides a comparatively straightforward and intuitive schema both for defining operators and understanding their effects, and therefore might be more favorable in formalizing process interactions using operations over functions.

One popular approach to composition of finite state transducers generalizes an intersection operation over finite-state acceptors (Hopcroft and Ullman, 1979; Mohri and Sproat, 1996; Mohri et al., 2000, 2002). Karttunen (1993, 5) gives a semi-informal overview of the operation:

[2] Worse still, it likely makes incorrect predictions. To properly capture this relationship, it would be necessary to embed a copy of the first `lfp` predicate inside the base case of the second one, which in earlier conceptions is ill-formed. However the real point of this demonstration is to contrast the burdensome nature of QFLFP with the uncomplicated nature of BMRS in formalizing interactions.

> The basic idea in the composition algorithm for finite-state transducer is quite simple. Every state in the new machine corresponds to a pair of original states in the two transducers, starting with the two initial states. Every *x:z* arc in the new machine corresponds to an *x:y* arc in the upper transducer and an *y:z* arc in the lower transducer. The matching middle symbol y is eliminated. The destination of the new arc is a state that corresponds to the destinations of the two original arcs. In general a composite transducer is larger than its components. In the worst case, the number of states in the composite machine is the product of the sizes of the original transducers.

Despite this simplicity, there two are potential disadvantages as it relates to formalizing process interactions as composite functions. One less serious consequence is that, as the number of states in the composite machine grows, so does the difficulty in interpreting it, and indeed in actually performing the composition without the assistance of automation. The other concern is that the contribution of each original machine is obscured as the two combine into a single machine. This is in addition to the more general disadvantage of finite-state approaches that Chandlee and Jardine (2020) point out: the machines do not intensionally capture motivations for patterns in a way that gels with traditional phonological analysis.

BMRS composition is not subject to the same pitfalls. First, while the size of the full composite system does grow proportionately to that of the individual systems—recall that $\mathcal{T}_2 \otimes \mathcal{T}_1$ is the *PS-join* of $\mathcal{T}_1$ and modified $\mathcal{T}_2'$—the modified outer function $\mathcal{T}_2'$ does not increase in size. Indexing refers to the inner function which itself has not increased in size either, making manual interpretation less taxing. More importantly, though, application of the composition operator keeps both functions mostly intact. This delineates the contribution of each function in the composite system's computation. Given this accessibility, BMRS composition is perhaps better suited to modeling process interactions using operations over functions.

Comparing the formalisms' relative ease of defining new operators is another dimension by which they can be compared. For example, a FST-equivalent to the parallel satisfaction operator might follow a similar state-pair construction protocol as composition. If so, it is on a par with BMRS composition and PS operators, in that both work by replacing *terms* from the outer function with terms from the inner function. However, the resulting machine would still be subject to the same pitfalls as the composite machine, namely potential difficulties with interpretation and the inability to disambiguate the contribution of each function. A serious attempt at defining PS over FSTs and comparing it with the BMRS equivalent is beyond the scope of this dissertation. Future work may examine these similarities in greater detail. The discussion here is preliminary illustration of the advantages of BMRS in modeling interactions compared to other approaches.

7.3 Representation

This section examines issues relevant to the representational choices motivated in chapter 4, and to a lesser extent the version of the BMRS formalism adopted in this dissertation. First, I consider how modifications to one's representational assumptions—combined with slight adjustments to the formalism—allow for alternative analyses of sandhi interactions. Then I extend the application of BMRS operations to the issue of notational equivalence introduced in chapter 4.

7.3.1 Alternative analyses

Analyses of sandhi interactions presented in previous chapters rely on representational assumptions about the string models over which transductions are defined. For example, input models of tonal strings consist of a universe of string indices. Individual string positions are labeled with at most one tone from a set of relations, and a set of index-type functions impose a linear order over these elements. By modifying certain assumptions about these models and the systems that define transductions, it is possible to consider alternative analyses of the same data. The purpose of this section is not to present fully-formed alternative analyses, but instead to consider how altering representational parameters might allow for such analyses.

Here I sketch an alternative analysis of the Xiamen tone circle. Instead of a PS-join of individual systems whereby each system models a single arc in the circle, I emulate Mortensen (2006)'s account using logical scales. Briefly, logical scales are a representational device that encode relationships (via an ordering relation) over phonological constituents of a particular type, such as lexical tones in Chinese sandhi. First recall the circular pattern, repeated below in (7.5).

(7.5)
$$
\begin{tikzcd}
 & \mathrm{R} \arrow[d] & \\
 & \mathrm{M} \arrow[dr] & \\
\mathrm{H} \arrow[ur] & & \mathrm{L} \arrow[dl] \\
 & \mathrm{F} \arrow[ul] &
\end{tikzcd}
$$

Suppose we wanted to encode the circular pattern *not* as a system of recursive function names for each output tone, but instead as part of the input structure by means of a logical scale. One way to achieve this is to relax the boolean restriction on BMRS such that recursive function names can be of type index. Next, input models under consideration contain two scale functions: one for the circular chain shift ($c(x)$ in 7.6a), and another identity map scale ($d(x)$ in 7.6b).

(7.6) a. $c(x) = \{(R, M), (M, L), (L, F), (F, H), (H, M)\}$

b. $d(x) = \{(R, R), (M, M), (L, L), (F, F), (H, H)\}$

Then, the circular pattern can be represented as a single index function $T'(x)$, as in (7.7). Note that this is somewhat similar to Wang (1967)'s single-rule account.

(7.7) $T'(x) \;=\; \texttt{if nf}(x) \texttt{ then } c(x) \texttt{ else } d(x)$

An output tonal index is selected from the scale function, based on whether it is non-final (indicated by $\texttt{nf}(x)$ above) within some domain. If domain-final, it returns the index from the identity map function $d(x)$. The same function could be extended to any circular chain shift pattern by modifying the scale function in the input signature.

Clearly the analysis needs further refinement. One issue deals with imposing both a linear order and a scalar order over the same set of string positions. If indices are represented directly as tones H, R, L, etc., defining predecessor/successor functions seems untenable for string structures with repeating tones, e.g. /HHRR/. However, the scale functions $c(x)$ and $d(x)$ depend on such representation, and it is unclear how a logical scale as an order over elements might be implemented using boolean terms $\sigma(x)$ for $\sigma \in \Sigma$.

Another alternative analysis relates to the conceptual motivation for adopting syllabic string representations as presented in chapter 4: namely, that decomposing contour tones into sequences of level tones captures edge effects and potential conspiracies not available to syllabic strings. Akinlabi (p.c.), for example, maintains that the Changting data present such a case. Recall the 15 disyllabic sandhi patterns reported by Chen (2004), repeated from chapter 2.

(7.8)

	M	**R**	**F**	**H**	**L**
M		**Lx**			**Lx**
R	Hx				xF
F	Rx	Lx	Mx	Lx; xM	Rx
H	Fx	Fx			Fx
L	**Mx**		**Mx**	**Mx**	

Notice the rows for Mid and Low tone, presented in bold.[3] Mid tone's sandhi form is Low, surfacing only before another Low tone (representing Rising as LH). Likewise, Low tone's sandhi form is Mid, surfacing only before another *non-low* tone: Mid, High, and Falling (HL). One interpretation of these

[3]There is another generalization whereby High becomes Falling (HL) before another non-high tone: M or L. I do not discuss this.

data is that Mid and Low tones assimilate in a stepwise fashion. There are a variety of options for condensing this generalization into a set of two rules (using melodic segments M and L) or even one rule (using alpha notation for a tonal feature [$\pm$ low]). What is important here is that the melodic analysis provides an alternative view which casts these alternations as a more general process of stepwise assimilation. The detectability of such edge effects results in a more elegant analysis—one for which the whole paradigm is captured using fewer rules/constraints. Importantly, this view is unavailable to a theory of sandhi that adopts arbitrary syllabic strings as representations. A syllabic string account, for example, treats L and R as separate entities, and thus has no explanation for why M surfaces as L before Low and Rising tones to the exclusion of all others.

While this may be the case for Changting, the representation of tone adopted in this dissertation presents a better fit for the typology of tone sandhi. Assuming a theory that predicts step-wise assimilation over melodies (to the exclusion of other types of alternations) may provide an eloquent account of Changting disyllabic sandhi, but it will ultimately fail the test of descriptive adequacy for sandhi in Chinese dialects. This is due to the often non-phonologically-grounded and phonetically-arbitrary nature of tone sandhi processes, discussed in chapters 2 and 4. In other words, the theory pursued here may (appear to) miss generalizations about conspiracies and edge effects, but it ultimately provides greater empirical coverage of the data.

7.3.2 Bi-inerpretability with composition

Operations over BMRS systems are not limited to modeling phonological process interactions. This section extends BMRS composition to proving the notational equivalence of representational theories.

Chapter 4 argues for the equivalence of melodic and syllable string representations with the notion of bi-interpretability (Friedman and Visser, 2014; Oakden, 2020). For any representational models A and B, this metric requires that they exhibit two properties. One is that they are intertranslatable, such that there exist interpretations T_A and T_B translating A to B and B to A. The second is the two are contrast-preserving; no contrast is lost as a result of translation. Formally this requires that the composition of the interpretations be isomorphic to the identity map. That is, $T_A \circ T_B$—translating A to B then back to A—is the same as mapping A to itself ($id(A)$). The inverse, $T_B \circ T_A \cong id(B)$ (where $\cong$ denotes isomorphism), must also hold.

Two BMRS transductions defined in chapter 4 translate any syllabic structure from $\Sigma = \{H, R, F, L\}$ into an equivalent melodic string structure from $\Sigma' = \{H, L, \bullet\}$[4], and any melodic structure into an

[4]Recall the crucial assumption about syllable boundaries '$\bullet$' in the melodic representation.

equivalent syllabic string representation. These are repeated as $\mathcal{T}^{sm}$ (7.9) and $\mathcal{T}^{ms}$ (7.10) below. $\mathcal{T}^{sm}$ is defined over copy set of size three, and $\mathcal{T}^{ms}$ over a single copy set.

(7.9) a. $\mathcal{T}^{sm}$

$$
\begin{aligned}
H^1(x) &= \texttt{if}\ F(x)\ \texttt{then}\ \bot\ \texttt{else}\ H(x)\\
H^2(x) &= R(x)\\
H^3(x) &= \bot\\
L^1(x) &= \texttt{if}\ R(x)\ \texttt{then}\ \bot\ \texttt{else}\ L(x)\\
L^2(x) &= F(x)\\
L^3(x) &= \bot\\
\bullet^1(x) &= \bot\\
\bullet^2(x) &= \bot\\
\bullet^3(x) &= \top
\end{aligned}
$$

(7.10) b. $\mathcal{T}^{ms}$

$$
\begin{aligned}
R^1(x) &= \texttt{if}\ L(x)\ \texttt{then}\\
&\quad\ \texttt{if}\ H(s(x))\ \texttt{then}\ \bullet(s(s(x)))\ \texttt{else}\ \bot\\
&\ \ \texttt{else}\ \bot\\
F^1(x) &= \texttt{if}\ H(x)\ \texttt{then}\\
&\quad\ \texttt{if}\ L(s(x))\ \texttt{then}\ \bullet(s(s(x)))\ \texttt{else}\ \bot\\
&\ \ \texttt{else}\ \bot\\
H^1(x) &= \texttt{if}\ H(x)\ \texttt{then}\\
&\quad\ \texttt{if}\ \bullet(p(x))\ \texttt{then}\ \bullet(s(x))\ \texttt{else}\ \bot\\
&\ \ \texttt{else}\ \bot\\
L^1(x) &= \texttt{if}\ L(x)\ \texttt{then}\\
&\quad\ \texttt{if}\ \bullet(p(x))\ \texttt{then}\ \bullet(s(x))\ \texttt{else}\ \bot\\
&\ \ \texttt{else}\ \bot
\end{aligned}
$$

These are sufficient to prove the first requirement of bi-interpretability. However, in demonstrating contrast preservation, chapter 4 does not compose the two systems; the composition operator over BMRS is not introduced until a later chapter. Instead it shows with an example that applying one transduction to a structure, then applying the other transduction to its output yields the same structure as the original input—i.e. is equivalent to $id(A)$ for some string representation and $id(B)$ for some melodic representation.

Having defined a composition operator over BMRS, it is now possible to compose the two systems, moving a step forward in proving bi-interpretability of string and melodic tonal representations. This section presents one half of this proof. Since it deals with non-size-preserving transductions, I follow Lindell and Chandlee (2016) and Jardine and Oakden (2020) in using typed variables for output copies. That is, for a copy set $C = \{1, \ldots, m\}$ and an input x, we have output copies $x^1, x^2, \ldots, x^m$.[5] (7.9) and (7.10) are repeated in (7.11) and (7.12) using typed variables.

(7.11) a. $\mathcal{T}^{sm}$

$$
\begin{aligned}
H(x^1) &= \texttt{if } F(x) \texttt{ then } \bot \texttt{ else } H(x)\\
H(x^2) &= R(x)\\
H(x^3) &= \bot\\
L(x^1) &= \texttt{if } R(x) \texttt{ then } \bot \texttt{ else } L(x)\\
L(x^2) &= F(x)\\
L(x^3) &= \bot\\
\bullet(x^1) &= \bot\\
\bullet(x^2) &= \bot\\
\bullet(x^3) &= \top
\end{aligned}
$$

(7.12) b. $\mathcal{T}^{ms}$

$$
\begin{aligned}
R(x^1) &= \texttt{if } L(x) \texttt{ then}\\
&\quad\ \texttt{if } H(s(x)) \texttt{ then } \bullet(s(s(x))) \texttt{ else } \bot\\
&\quad \texttt{else } \bot\\
F(x^1) &= \texttt{if } H(x) \texttt{ then}\\
&\quad\ \texttt{if } L(s(x)) \texttt{ then } \bullet(s(s(x))) \texttt{ else } \bot\\
&\quad \texttt{else } \bot\\
H(x^1) &= \texttt{if } H(x) \texttt{ then}\\
&\quad\ \texttt{if } \bullet(p(x)) \texttt{ then } \bullet(s(x)) \texttt{ else } \bot\\
&\quad \texttt{else } \bot\\
L(x^1) &= \texttt{if } L(x) \texttt{ then}\\
&\quad\ \texttt{if } \bullet(p(x)) \texttt{ then } \bullet(s(x)) \texttt{ else } \bot\\
&\quad \texttt{else } \bot
\end{aligned}
$$

If the composition $\mathcal{T}^{sm} \otimes \mathcal{T}^{ms}$ is isomorphic to the identity map on string models $id(\mathcal{M}_s)$, this will serve as proof of half of the contrast-preservation requirement. Following (Jardine and Oakden,

[5] Thus the functions defined here are *polymorphic*, as each function represents a set of functions relativized to the variable type.

2020), composition for non-size-preserving transductions proceeds as follows. Given $\mathcal{T}^{sm}$ with a copy set $C = \{1, 2, 3\}$ and $\mathcal{T}^{ms}$ with a copy set $D = \{1\}$, $\mathcal{T}^{sm} \otimes \mathcal{T}^{ms}$ is defined over a copy set $E = C \cup (C \times D)$. $\mathcal{T}^{ms}$ is expanded to $\mathcal{T}'^{ms}$, defined over $C \times D$, where every expression $\delta(p(x))/\delta(s(x))$[6] from $\mathcal{T}^{sm}$ with an untyped term is replaced with $\delta(p_1(x^c))/\delta(s_1(x^c))$, where p_1 and s_1 denote the order-preserving index functions[7] from $\mathcal{T}^{sm}$. $\mathcal{T}^{sm} \otimes \mathcal{T}^{ms}$ is thus equal to $\mathcal{T}^{sm} \cup \mathcal{T}'^{ms}$.

The definition of $\mathcal{T}^{sm} \otimes \mathcal{T}^{ms}$ is given in (7.13); recall $\mathcal{T}^{sm}$ in (7.11) for relevant definitions.

$$
\text{(7.13)} \quad
\begin{array}{rcl}
R(x^{1,1}) & = & \texttt{if } L(x^1) \texttt{ then} \\
 & & \quad \texttt{if } H(s_1(x^1)) \texttt{ then } \bullet(s_1(s_1(x^1))) \texttt{ else } \perp \\
 & & \texttt{else } \perp \\
R(x^{2,1}) & = & \texttt{if } L(x^2) \texttt{ then} \\
 & & \quad \texttt{if } H(s_1(x^2)) \texttt{ then } \bullet(s_1(s_1(x^2))) \texttt{ else } \perp \\
 & & \texttt{else } \perp \\
R(x^{3,1}) & = & \texttt{if } L(x^3) \texttt{ then} \\
 & & \quad \texttt{if } H(s_1(x^3)) \texttt{ then } \bullet(s_1(s_1(x^3))) \texttt{ else } \perp \\
 & & \texttt{else } \perp \\
F(x^{1,1}) & = & \texttt{if } H(x^1) \texttt{ then} \\
 & & \quad \texttt{if } L(s_1(x^1)) \texttt{ then } \bullet(s_1(s_1(x^1))) \texttt{ else } \perp \\
 & & \texttt{else } \perp \\
F(x^{2,1}) & = & \texttt{if } H(x^2) \texttt{ then} \\
 & & \quad \texttt{if } L(s_1(x^2)) \texttt{ then } \bullet(s_1(s_1(x^2))) \texttt{ else } \perp \\
 & & \texttt{else } \perp \\
F(x^{3,1}) & = & \texttt{if } H(x^3) \texttt{ then} \\
 & & \quad \texttt{if } L(s_1(x^3)) \texttt{ then } \bullet(s_1(s_1(x^3))) \texttt{ else } \perp \\
 & & \texttt{else } \perp \\
H(x^{1,1}) & = & \texttt{if } H(x^1) \texttt{ then} \\
 & & \quad \texttt{if } \bullet(p_1(x^1)) \texttt{ then } \bullet(s_1(x^1)) \texttt{ else } \perp \\
 & & \texttt{else } \perp \\
H(x^{2,1}) & = & \texttt{if } H(x^2) \texttt{ then} \\
 & & \quad \texttt{if } \bullet(p_1(x^2)) \texttt{ then } \bullet(s_1(x^2)) \texttt{ else } \perp \\
 & & \texttt{else } \perp \\
H(x^{3,1}) & = & \texttt{if } H(x^3) \texttt{ then} \\
 & & \quad \texttt{if } \bullet(p_1(x^3)) \texttt{ then } \bullet(s_1(x^3)) \texttt{ else } \perp \\
 & & \texttt{else } \perp \\
L(x^{1,1}) & = & \texttt{if } L(x^1) \texttt{ then} \\
 & & \quad \texttt{if } \bullet(p_1(x^1)) \texttt{ then } \bullet(s_1(x^1)) \texttt{ else } \perp \\
 & & \texttt{else } \perp \\
L(x^{2,1}) & = & \texttt{if } L(x^2) \texttt{ then} \\
 & & \quad \texttt{if } \bullet(p_1(x^2)) \texttt{ then } \bullet(s_1(x^2)) \texttt{ else } \perp \\
 & & \texttt{else } \perp \\
L(x^{3,1}) & = & \texttt{if } L(x^3) \texttt{ then} \\
 & & \quad \texttt{if } \bullet(p_1(x^3)) \texttt{ then } \bullet(s_1(x^3)) \texttt{ else } \perp \\
 & & \texttt{else } \perp
\end{array}
$$

This composed system maps any syllabic string representation to an equivalent syllabic string representation, and is isomorphic to the identity map on syllabic string models $id(\mathcal{M}_s)$. To illustrate,

[6] That is, from $\delta \in \Delta = \{H, L, \bullet\}$

[7] Order is preserved on transductions by setting the definition of each $p(x^i)$. See (Jardine and Oakden, 2020) for more details

consider the syllabic string $RHLFF$. Its evaluation against the composite system is provided in (7.14). To save space, some functions for which no string position returns true are omitted.

(7.14)

Input:		R_1	H_2	L_3	F_4	F_5
	x^1	L	H	L	H	H
	x^2	H			L	L
	x^3	$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\bullet$
Output:	$x^{1,1}$	R	H	L	F	F
	$x^{2,1}$					
	$x^{3,1}$					

	1	2	3	4	5
$p_1(x^1)$		1^3	2^3	3^3	4^3
$p_1(x^2)$	1^1			4^1	5^1
$p_1(x^3)$	1^2	2^1	3^1	4^2	5^2
$s_1(x^1)$	1^2	2^3	3^3	4^2	5^2
$s_1(x^2)$	1^3			4^3	5^3
$s_1(x^3)$	2^1	3^1	4^1	5^1	
$H(x^1)$		$\top$		$\top$	$\top$
$H(x^2)$	$\top$				
$L(x^1)$	$\top$		$\top$		
$L(x^2)$				$\top$	$\top$
$\bullet(x^3)$	$\top$	$\top$	$\top$	$\top$	$\top$
$R(x^{1,1})$	$\top$				
$F(x^{1,1})$				$\top$	$\top$
$H(x^{1,1})$		$\top$			
$L(x^{1,1})$			$\top$		

The reader can verify (perhaps painstakingly) that the output of the composite system is $RHLFF$, the same as the identity map on $RHLFF$: $id(RHLFF) = RHLFF$. This extends to the set of syllabic string models $\mathcal{M}_s$, meaning that translating from a syllabic string model to a melodic model does not entail any loss of contrast.

A full proof—crucially the inverse: $\mathcal{T}^{ms} \otimes \mathcal{T}^{sm} \cong id(\mathcal{M}_m)$—is left for future work, but the demonstration above illustrates the application of operations over BMRS to questions of notational equivalence in representation.

7.4 Further properties of the PS operator

This section explores the PS operator $\ominus$ in more detail. It ponders some of its formal properties—closure properties with classes of SL functions in §4.1 and its non-commutativity in §4.2—outlining directions for future work. The remaining sections examine possible connections with phonological and linguistic theory: §4.3 connects the operator with two-level phonology, and §4.4 considers whether PS is similar to a priority union operator.

7.4.1 Closure properties

One question of interest is the closure properties of classes of functions in regard to the PS operator. For example, it is known that the regular relations are closed under composition (Kaplan and Kay, 1994). That is, for any two regular relations f and g, their composition $g \circ f$ is guaranteed to be regular. This result is significant because it shows that interactions modeled by ordering over individual rewrite rules (represented by regular regulations) does not increase the expressivity of the entire grammar beyond the regular threshold. However subregular classes, and in particular strictly-local classes of functions, do not share this quality uniformly. Lindell and Chandlee (2016) conjecture the closure of finite-to-one ISL functions[8] under composition, but Chandlee (2014) shows that OSL functions are not closed under composition. As this dissertation deals primarily with interactions of SL functions, the closure properties of ISL and OSL classes under the PS operator are most relevant. I begin with ISL.

Given two ISL functions f and g, is $f \ominus g$ guaranteed to be ISL? Recall that the class of NR-BMRS-definable transductions correspond to the ISL class, and that $\ominus$ joins two transductions $\mathcal{T}_1$ and $\mathcal{T}_2$ such that the final term in each function definition in $\mathcal{T}_1$ is replaced with the entire right-hand side of the corresponding definition (which itself is also a term) from $\mathcal{T}_2$. Let $\mathcal{T}_1$, $\mathcal{T}_2$ be NR-BMRS-definable[9] transductions, that is, for which no term T_i in list $(f_{1_{\mathcal{T}_1}}(x_{1_{\mathcal{T}_1}})...f_{k_{\mathcal{T}_1}}(x_{k_{\mathcal{T}_1}}))$ or $(f_{1_{\mathcal{T}_2}}(x_{1_{\mathcal{T}_2}})...f_{k_{\mathcal{T}_2}}(x_{k_{\mathcal{T}_2}}))$ contains a term of the type $\mathtt{f}(T)$. Thus $\mathcal{T}_2'$, the result of replacing definitions as described above, is also NR-BMRS, because no terms of the type $\mathtt{f}(T)$ are introduced as a result of the operation. $\mathcal{T}_1 \ominus \mathcal{T}_2$ is also NR-BMRS and therefore ISL because the resulting transduction—$\mathcal{T}_1 \cup \mathcal{T}_2'$—contains no terms of the type $\mathtt{f}(T)$. It is clear that this extends to any number of PS joins involving two or more NR-BMRS-definable transductions and, therefore it holds in the general case. Thus, ISL is closed under PS.

A similar intuition holds for L-OSL and R-OSL classes. For example, let $\mathcal{T}_1$, $\mathcal{T}_2$ be OR-BMRS-definable[10] transductions, that is, for which no term of the form $\sigma(T)$ takes index sort terms of the type $s(T)$ or $p(T)$. Additionally, restrict $\mathcal{T}_1$ and $\mathcal{T}_2$ such that recursive function terms $\mathtt{f}(T)$ in both systems take either terms $p(T)$ or $s(T)$, but not both. Let us denote these OR-BMRS^p and OR-BMRS^s, respectively. This is tantamount to limiting PS-joins to two L-OSL functions or two R-OSL functions. Thus $\mathcal{T}_2'$, the result of replacing definitions as described above, is also OR-BMRS$^{p/s}$, because no terms $\sigma(T)$ that take $p(T)$ or $s(T)$ are introduced as a result of the operation. $\mathcal{T}_1 \ominus \mathcal{T}_2$

[8]Recall from the composition chapter that functions of this type impose a bound on the ratio between input and output string lengths. Therefore a function $f : \Sigma^* \rightarrow \Gamma^*$ is finite-to-one if its inverse function $f^{-1}(y)$ is always finite.

[9]Where NR-BMRS denotes the class of non-recursive BMRS systems. See §4.2 in the formal chapter.

[10]Where OR-BMRS denotes the class of output-restricted BMRS systems. See §4.3 in the formal chapter.

is also OR-BMRS$^{p/s}$ and therefore L/R-OSL because the resulting transduction—$\mathcal{T}_1 \cup \mathcal{T}_2'$—contains no terms of the type $\sigma(T)$ that take index sort terms $p(T)$ or $s(T)$. Extending this to the general case, it would seem that these classes of OSL are closed under PS as well. Barring a solid proof of the equivalence between OR-BMRS and the OSL class, however, L/R-OSL closure under PS is merely speculative, and is therefore left for future work. Closure of the entire OSL class is also set aside for future work.

7.4.2 The non-commutativity of PS

An earlier chapter demonstrates the non-commutativity of the PS operator; given two BMRS-definable transductions $\mathcal{T}_1$ and $\mathcal{T}_2$, it is not guaranteed that $\mathcal{T}_1 \ominus \mathcal{T}_2$ and $\mathcal{T}_2 \ominus \mathcal{T}_1$ describe extensionally-equivalent functions. This section provides additional illustration using an example from natural language, and generalizes the notion of 'conflict' to substructures in a hierarchy of licensing and blocking structures.[11]

Example (7.15) shows a quantitive-sensitive stress paradigm in Lushootseed (Hayes, 1995); this pattern stresses the leftmost heavy ('H') syllable, or, if there are no heavy syllables, the leftmost light ('L') syllable. Stress is indicated with an acute accent.[12]

(7.15) a. LLLH $\mapsto$ LLLḦ́

b. LLLL $\mapsto$ ĹLLL

c. HLHL $\mapsto$ H́LHL

Now consider a function that maps input unstressed heavy and light syllables $\Sigma = \{H, L\}$ to output syllables with the attested stress application $\Gamma = \{H, L, \acute{\Box}\}$, where '$\acute{\Box}$' denotes stress marking on a syllable. As a BMRS system, the recursive function name of interest is $\acute{\Box}(x)$ (H and L inputs will map directly to the output). For clarity, three auxiliary functions are defined. Two (7.16a-b) are defined recursively: $foll\text{-}H(x)$ evaluates true when a position follows an H somewhere in the string, and $prec\text{-}H(x)$ evaluates true when a position precedes an H somewhere in the string. The other $first\text{-}L(x)$ (7.16c) is non-recursive, and returns a true value only for an initial string position input-marked as a light (L) syllable.[13]

[11] This section is inspired by lively discussion with Nate Koser and Adam Jardine.

[12] Koser and Jardine (2020) show that this pattern is not subsequential (specifically, it is weakly deterministic) due to recursive calls of both predecessor and successor functions within the definitions. I abstract away from this issue to illustrate the relevant property of the PS operator.

[13] $first(x)$ is definable assuming a total predecessor function, and is shorthand for: `if` $p(x) = x$ `then` $\top$ `else` $\bot$.

(7.16) a. $\mathit{foll}\text{-}H(x) \;=\; \texttt{if}\;\; H(p(x)) \;\;\texttt{then}\;\; \top \;\;\texttt{else}\;\; \mathit{foll}\text{-}H(p(x))$

c. $\mathit{prec}\text{-}H(x) \;=\; \texttt{if}\;\; H(s(x)) \;\;\texttt{then}\;\; \top \;\;\texttt{else}\;\; \mathit{prec}\text{-}H(s(x))$

c. $\mathit{first}\text{-}L(x) \;=\; \texttt{if}\;\; L(x) \;\;\texttt{then}\;\; \mathit{first}(x) \;\;\texttt{else}\;\; \bot$

The definition for $\acute{\square}(x)$ is in (7.17):

$$
\begin{aligned}
(7.17)\quad \acute{\square}(x) \;=\; & \texttt{if}\;\; \mathit{foll}\text{-}H(x) \;\;\texttt{then}\;\; \bot \;\;\texttt{else} \\
& \texttt{if}\;\; H(x) \;\;\texttt{then}\;\; \top \;\;\texttt{else} \\
& \texttt{if}\;\; \mathit{prec}\text{-}H(x) \;\;\texttt{then}\;\; \bot \;\;\texttt{else} \\
& \texttt{if}\;\; \mathit{first}\text{-}L(x) \;\;\texttt{then}\;\; \top \;\;\texttt{else}\;\; \bot
\end{aligned}
$$

It generalizes the LHOL pattern by stating the following conditions (also notice their order): (line 1) if a syllable follows an H, it is not stressed; otherwise (line 2) if it is input-marked H, it is stressed; otherwise (line 3) if it precedes an H, it is not stressed; otherwise (line 4) if it is the first L in the string it is stressed, otherwise it is not stressed. This system accepts the mappings in (7.15), and thus models the LHOL stress pattern. The table in (7.18) provides evaluations of each representative mapping.

(7.18)

	L	L	L	H	L	L	L	L	H	L	H	L
	1	2	3	4	1	2	3	4	1	2	3	4
$\acute{\square}$(x)	$\bot$	$\bot$	$\bot$	$\top$	$\top$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$	$\bot$	$\bot$
$\mathit{foll}\text{-}H(x)$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\top$	$\top$	$\top$
$H(x)$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$	$\top$	$\bot$
$\mathit{prec}\text{-}H(x)$	$\top$	$\top$	$\top$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\top$	$\top$	$\bot$	$\bot$
$\mathit{first}\text{-}L(x)$	$\top$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$
Output	L	L	L	Ḣ	Ĺ	L	L	L	Ḣ	L	H	L

Suppose we wanted to define two properly-subsequential functions from (7.17) such that lookahead is restricted to a single direction in each function. Example (7.19) presents one option; separation of recursively-defined $\mathit{foll}\text{-}H(x)$ and $\mathit{prec}\text{-}H(x)$ into $\mathcal{T}_1$ and $\mathcal{T}_2$ guarantees the desired restriction on lookahead.

(7.19)

$\mathcal{T}_1$

$$
\begin{aligned}
\acute{\square}(x) \;=\; & \texttt{if}\;\; \mathit{foll}\text{-}H(x) \;\;\texttt{then}\;\; \bot \;\;\texttt{else} \\
& \texttt{if}\;\; H(x) \;\;\texttt{then}\;\; \top \;\;\texttt{else}\;\; \bot
\end{aligned}
$$

$\mathcal{T}_2$

$$
\begin{aligned}
\acute{\square}(x) \;=\; & \texttt{if}\;\; \mathit{prec}\text{-}H(x) \;\;\texttt{then}\;\; \bot \;\;\texttt{else} \\
& \texttt{if}\;\; \mathit{first}\text{-}L(x) \;\;\texttt{then}\;\; \top \;\;\texttt{else}\;\; \bot
\end{aligned}
$$

$\mathcal{T}_1$ and $\mathcal{T}_2$ can be combined via PS. It is clear that $\mathcal{T}_1 \ominus \mathcal{T}_2$ is extensionally-equivalent to the system in (7.17) and thus would accept the same maps; replacing the final term in $\mathcal{T}_1$ with $\mathcal{T}_2$ generates an identical hierarchy of structures. What about $\mathcal{T}_2 \ominus \mathcal{T}_1$, given in (7.20)?

(7.20) a. $\mathcal{T}_2 \ominus \mathcal{T}_1$ $\acute{\square}(x)$ $=$ `if` $prec\text{-}H(x)$ `then` $\bot$ `else`

`if` $first\text{-}L(x)$ `then` $\top$ `else` $\acute{\square}_{\mathcal{T}_1}(\mathbf{x})$

b. $\mathcal{T}_1$ $\acute{\square}(x)$ $=$ `if` $foll\text{-}H(x)$ `then` $\bot$ `else`

`if` $H(x)$ `then` $\top$ `else` $\bot$

In this case, the result is a different function from $\mathcal{T}_1 \ominus \mathcal{T}_2$. This serves as another illustration of the non-communativity of the PS operator. An evaluation table in (7.21) highlights differences between functions described by $\mathcal{T}_2 \ominus \mathcal{T}_1$ and $\mathcal{T}_1 \ominus \mathcal{T}_2$ PS joins.

(7.21)

	L	L	L	H	L	L	L	L	H	L	H	L
	1	2	3	4	1	2	3	4	1	2	3	4
$\acute{\square}$(x)	$\bot$	$\bot$	$\bot$	$\top$	$\top$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$
$prec\text{-}H(x)$	$\top$	$\top$	$\top$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\top$	$\top$	$\bot$	$\bot$
$first\text{-}L(x)$	$\top$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$
$foll\text{-}H(x)$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\top$	$\top$	$\top$
$H(x)$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$	$\top$	$\bot$
Output	L	L	L	Ḣ́	Ĺ	L	L	L	H	L	H	L

Although it accepts the same mappings from inputs /LLLL/ and /LLLH/, this function does not assign stress to any syllable for the input string /HLHL/. Why?

An earlier chapter demonstrates the non-communativity of the PS operator with the existence of structural *conflicts*, that is, PS-joins for which the same structure both licenses and blocks the same output across two systems. The case presented above reveals that a more general principle is applicable and approaches perhaps a better generalization. To illustrate, consider how $\mathcal{T}_1 \ominus \mathcal{T}_2$ and $\mathcal{T}_2 \ominus \mathcal{T}_1$ differ in the mapping from input /HLHL/, summarized in (7.22).

(7.22) $\mathcal{T}_1 \ominus \mathcal{T}_2$ /HLHL/ $\mapsto$ [ḢLHL]

$\mathcal{T}_2 \ominus \mathcal{T}_1$ /HLHL/ $\mapsto$ [HLHL]

The discrepancy is attributable to relationships between licensing/blocking structures in the distinct hierarchies created by each PS-join. In $\mathcal{T}_1 \ominus \mathcal{T}_2$, $H(x)$ (licenses stress) evaluation is *higher* in the hierarchy than $prec\text{-}H(x)$ (blocks stress); the first string position <u>H</u>LHL receives stress despite also evaluating to true for $prec\text{-}H(x)$. Order between these two structures is reversed in $\mathcal{T}_2 \ominus \mathcal{T}_1$. The initial heavy syllable in <u>H</u>LHL does not receive stress, as evaluation of $prec\text{-}H(x)$—here outranking $H(x)$—returns a false value for $\acute{\square}(x)$. Satisfying $prec\text{-}H(x)$ blocks stress on that position, rendering satisfaction of $H(x)$ immaterial.

This is related to what Chandlee and Jardine (2020) term the Strict Substructure Ordering Principle (SSOP), defined in (7.23).

(7.23) For any ranking of structures in a BMRS definition, whenever $\mathrm{STRUCT}_j(x)$ implies $\mathrm{STRUCT}_i(x)$ but the converse is not true (i.e., STRUCT_i is a strict substructure of STRUCT_j, it must be the case that $i < j$ in the order, otherwise j will never be evaluated.

The set of structures $H(x)$ is *not* a strict subset of the structures *prec*-$H(x)$ (and clearly not vice versa), so this example of the non-commutativity of the PS operator cannot be explained as a SSOP effect. They do, however, have a non-empty intersection, as the mapping of input /HLHL/ illustrates, and similar effects (specifically non-evaluation of the lower-ranked structure) are observed in such cases. A more accurate description of *conflict*, then, might be: a set of structures $\mathcal{S}_1$ licenses some output in $\mathcal{T}_1$, another set of structures $\mathcal{S}_2$ blocks the same output in $\mathcal{T}_2$, and $\mathcal{S}_1 \cap \mathcal{S}_2 \neq \varnothing$.[14] Here, PS-join order matters, because the operator creates hierarchies that make different predictions for a subset of maps. Future work can narrow in on a more precise characterization of what types of PS-joins have this property.

7.4.3 PS and two-level phonology

Applying the PS operator to individual systems models interactions whereby multiple processes refer to the same input and output string structures in their computation. No intermediate representations are generated as in composition. This bears a resemblance to two-level rules in the finite-state phonological framework introduced by Koskenniemi (1983). In the parallel conceptualization of phonological transformations, multiple rules range over the same sets of underlying and surface forms without intermediate stages. As Karttunen (1993, 1) puts it, a "parallel rule does not 'apply' in the sense of changing one representation to another one, it is simply true or false for some *pair* of forms" [emphasis mine]. In terms of models over strings, this pair is precisely the single input (for ISL functions) and output (for OSL functions) string structures to which computation is restricted in PS joins.

Both formalisms agree in modeling transformations without intermediate stages, but there are a great many differences between the two. Koskenniemi's model maps lexical strings to surface strings through an unordered set of constraints formalized as transducers, and working in tandem. Order is consequential for PS-joins of individual systems—as the previous section illustrates—despite the fact that order does not seem to matter for a majority of sandhi interactions analyzed in chapter 6.

Another important difference is in how the parallel approach relates to the sequential approach;

[14] This definition of conflict might connect with other approaches to defining interactions, e.g. (Meinhardt et al., 2020), in particular as it relates to predicting when order of combination under some operator matters. Connecting these notions is left for future work.

that is, two-level rules vs. cascades of composed transducers on the one hand, and PS-joins and composed BMRS systems on the other. Karttunen (1993) maintains that sequential and parallel systems in finite-state phonology are formally equivalent. They do not differ in the types of phenomena that they can describe. This dissertation has shown instead that the set of interaction maps definable as composition does not correspond to the set of interaction maps defined from PS-joins. In fact, the insufficiency of composition in modeling ordering paradoxes and circular chain shifts is what necessitated an enrichment of the theory to include an alternative.

This section has examined mostly superficial similarities and differences between these approaches. A serious, comprehensive accounting of the connections between BMRS PS-joins and Koskennieni's two-level rules in a finite-state framework is beyond the scope of the current dissertation, but offers a potentially fruitful avenue for future work.

7.4.4 PS and priority union

Chapter 6 introduces PS as a new operation over BMRS systems of equations. Is this a truly novel operation, or does it correspond to some well-understood general operation over functions/relations? One possible candidate is priority union, which Karttunen (1998) discusses as part of a larger composition operation relevant to OT grammars.[15] Priority union "was originally defined as an operation for unifying two feature structures in a way that eliminates any risk of failure by stipulating that one of the two has priority in a case of conflict" (12). For example, consider two relations $\mathtt{Q}$ and $\mathtt{R}$, shown below.

(7.24)
$$\mathtt{Q} = \begin{cases} a \mapsto x \\ b \mapsto y \end{cases} \qquad \mathtt{R} = \begin{cases} b \mapsto z \\ c \mapsto w \end{cases}$$

There is a conflict between these relations for the first element b;[16] $\mathtt{Q}$ maps it to y while $\mathtt{R}$ maps it to z. The *priority union* of these relations such that $\mathtt{Q}$ is prioritized over $\mathtt{R}$—denoted $\mathtt{Q\,.P.\,R}$—comprises the following mappings.

(7.25)
$$\mathtt{Q\,.P.\,R} = \begin{cases} a \mapsto x \\ b \mapsto y \\ c \mapsto w \end{cases}$$

[15] This section was inspired by a lively conversation with Adam McCollum, Eric Baković, Anna Mai, and Eric Meinhardt.

[16] Karttunen calls this the 'upper element'

That is, it includes all pairs from `Q` and every pair from `R` whose first element is not also in `Q`. When there is a mapping for this element in both relations *and* they conflict (as with b), the priority union of `Q` and `R` only includes the pair from `Q`. Does PS do something similar?

As a point of departure in probing this question, one avenue is to consider whether priority union might work for a case that is amenable to analysis as a PS-join. Let us consider as an example mutual counterbleeding opacity in Changting (that is, the interaction of MR and RM rules; see chapter 6). If we take the priority union of these two mappings, do we get the attested interaction? At first glance it appears that the answer is no. The case Karttunen describes is for relations whose 'upper elements' in a pair are non-overlapping; only cases of overlap (and conflict) evince prioritization. If we think of separate *total* functions describing individual MR and RM rules as relations `A` and `B`—such that the upper element is the input and the lower input is the output—we find total overlap. This is because the two systems of equations considered in the analysis are total functions over the same input alphabet $\Sigma = \{L, M, H, R, F\}$ and thus consider the same set of string inputs ('upper elements' in a pair) which map to outputs. Priority union as defined would obscure the effect of one rule entirely and thus fail to capture the interaction. Consider the predictions of `A .P. B` and `B .P. A` under this conception below, neither of which are consistent with observed outputs—recall that the correct mappings are /MRM/ $\mapsto$ [LHM] and /RMR/ $\mapsto$ [HLR].

$$(7.26) \qquad \texttt{A .P. B} = \begin{cases} MRM \mapsto {}^*MHM \\ RMR \mapsto {}^*HMR \end{cases} \qquad \texttt{B .P. A} = \begin{cases} MRM \mapsto {}^*LRM \\ RMR \mapsto {}^*RLR \end{cases}$$

In other words, prioritizing with total functions seems to mean applying one rule and not the other: `A .P. B` is equivalent to `A` and `B .P. A` is equivalent to `B`.

Now suppose that we relax certain restrictions on total functions. That is, let two relations `A` and `B` represent *partial* functions for the RM and MR rules. In BMRS terms, function names for which no change obtains between input and output—non-mutating mappings—are undefined.[17] Then `A .P. B` might yield a total, combined-map function. However, there is no priority to speak of because there is no overlap. Thus Changting mutual counterbleeding would be a special case of priority(less) union.

As the Changting example suggests, the question of how PS relates to priority union is tied to totality/partiality of functions joined by the operator. Future work can explore this issue in more detail, but the demonstration provided here hints that equivalence between the PS and priority

[17] An equivalent alternative is to say that they are defined over output alphabets consisting only of tonal segments that vary from the input.

union (as an operation over functions) is spurious.

7.5 Beyond sandhi, beyond interactions

Finally, this section goes beyond the empirical scope (sandhi interactions) of this dissertation to examine other applications of BMRS. First, I extends BMRS composition to harmony interactions in Palestinian Arabic, showing how process-specific constraint (PSC) effects fall out automatically from a characteristic of BMRS: preservation of hierarchies under composition. Then, I consider potential applications of BMRS to learning.

7.5.1 Harmony and PSC

Here I analyze a non-sandhi interaction while demonstrating how PSC phenomena can be captured within the BMRS formalism. This section presents a case study of a PSC effect in the RTR harmony system of Palestinian Arabic (Davis, 1995; McCarthy, 1997); rightward RTR harmony can be blocked by high, front segments, but leftward RTR harmony proceeds unimpeded within a word. This is shown in (7.27), where triggers are capitalized and RTR-harmony spans are underlined.[18]

(7.27)
a. *Leftward harmony*: <u>ballaS</u> 'thief'
b. *Rightward harmony*: <u>Tuubak</u> 'your blocks'
c. *Blocking of rightward harmony*: <u>Sa</u>yyad 'hunter'

Davis (1995)'s rule-based analysis achieves the PSC effect by tagging the rightward spread rule with the target condition "RTR/Hi and RTR/Fr" such that segments with these features block rightward spread but not leftward spread. The additional claim is that OT does not predict these effects. McCarthy (1997) responds by showing that OT not only captures PSCs—a direct result of constraint ranking—but it also presents a more restrictive theory of PSCs. The transitive nature of ranking predicts that if some crucial ranking between markedness constraints produces a blocking effect for one process, the same effect will be observed for any other process compelled by a markedness constraint lower in the hierarchy. This principle is termed the *Subset Criterion*, and no such prediction is made by PSCs tagged on individual rules.

Following McCarthy (1997)'s generalization about rankings for PSC interactions in OT grammars, the analysis presented here shows that ordered hierarchies of licensing and blocking structures

[18] Davis briefly discusses dialectical variability in Arabic RTR harmony. In some dialects, spreading is limited to the adjacent vowel, while in others RTR spreads throughout the word, and there are no blockers. The forms presented here align most closely with data from (Herzallah, 1990).

in BMRS systems of equations produce the same effects. These obtain both when the interaction is defined as a single, combined map system of equations and for composition of separate systems (modeling each individual spreading process). Crucially, this is because hierarchical relations within systems are preserved under composition. It also means that the subset criterion is predicted by this basic mechanism in BMRS in much the same way as it is predicted in OT as a result of ranking transitivity.

7.5.1.1 PSC effects in a combined map system

The Arabic data motivate the constraint ranking in (7.28), where RTR-Left/Right triggers leftward/rightward spread and RTR/Hi&Fr is a local conjunction representing the blocking condition.

(7.28) RTR-Left >> RTR/Hi&Fr >> RTR-Right >> Ident-RTR

In general, a PSC interaction as in Arabic RTR harmony obtains when some constraint $\mathbb{C}$ ranks *between* two markedness constraints $\mathbb{M}_i$ and $\mathbb{M}_j$, each of which outrank some faithfulness constraint $\mathbb{F}$. Given $\mathbb{M}_i >> \mathbb{C} >> \mathbb{M}_j >> \mathbb{F}$, the effect of $\mathbb{C}$ is 'specific' to the process triggered by $\mathbb{M}_j >> \mathbb{F}$ and not to the one triggered by $\mathbb{M}_i$.

The PSC effects attested in Palestinian Arabic can be captured as a single combined map BMRS system of equations using the same intuition. Here, instead of a ranking between constraints, the effect is captured by a hierarchy of licensing and blocking structures. (7.29) gives the output boolean function for the feature RTR.

(7.29)
$$
\begin{aligned}
\mathrm{RTR}'(x) \quad = \quad & \texttt{if}\ \ \mathrm{RTR}'(s(x)) && \texttt{then}\ \top\ \texttt{else}\\
& \texttt{if}\ \ \mathrm{RTR/H\textsc{i}\&F\textsc{r}}(x) && \texttt{then}\ \bot\ \texttt{else}\\
& \texttt{if}\ \ \mathrm{RTR}'(p(x)) && \texttt{then}\ \top\ \texttt{else}\\
& \mathrm{RTR}(x)
\end{aligned}
$$

The first (i.e. highest-ranked) licensing structure in the hierarchy permits leftward RTR harmony via the recursive definition $\mathrm{RTR}'(s(x))$ as illustrated in the evaluation table for the form in (7.27a):

(7.30)

	b	a	l	l	a	S
	1	2	3	4	5	6
$\mathrm{RTR}'(s(x))$	$\top$	$\top$	$\top$	$\top$	$\top$	$\bot$
$\mathrm{RTR}(x)$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\top$
	b̲	a̲	l̲	l̲	a̲	S̲

This definition works in conjunction with the default condition RTR(x) in the following way. String position 6 evaluates to true for RTR′(x) by virtue of being specified as RTR in the input. When position 5 is evaluated against the definition, it satisfies the recursively-defined RTR′($s(x)$) and is output with the RTR feature. Further evaluation proceeds in an identical manner; given the licensing structure's hierarchical position, the RTR span extends unhindered to the edge of the word.

Rightward spread obtains via initial satisfaction of RTR(x) followed by iterative evaluation of RTR′($p(x)$). However, since the blocking condition RTR/Hi&Fr(x) comes before RTR′($p(x)$) in the hierarchy, spreading can only proceed provided the current input symbol does *not* return a true value for RTR/Hi&Fr(x). If it does, then spreading is blocked as in (7.27c). See the evaluation below in (7.31).

(7.31)

	S	a	y	y	a	d
	1	2	3	4	5	6
RTR/Hi&Fr(x)	$\bot$	$\bot$	$\top$	$\top$	$\bot$	$\bot$
RTR′($p(x)$)	$\bot$	$\top$	$\top$	$\bot$	$\bot$	$\bot$
RTR(x)	$\top$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$
	$\underline{\text{S}}$	$\underline{\text{a}}$	y	y	a	d

RTR spreads to string position 2 from the trigger 'S', but since string position 3 satisfies the higher-ranked blocking structure (in spite of also evaluating to true for RTR′($p(x)$)), it returns false for RTR′(x), blocking further rightward spread. Importantly, the same hierarchical stratification that blocks rightward spread also permits leftward spread over high, front segments. Consider another form $\underline{\text{xayyaT}}$ 'tailor' which exhibits leftward spread over the high, front blocker 'y'.

(7.32)

	x	a	y	y	a	T
	1	2	3	4	5	6
RTR′($s(x)$)	$\top$	$\top$	$\top$	$\top$	$\top$	$\bot$
RTR/Hi&Fr(x)	$\bot$	$\bot$	$\top$	$\top$	$\bot$	$\bot$
RTR(x)	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\top$
	$\underline{\text{x}}$	$\underline{\text{a}}$	$\underline{\text{y}}$	$\underline{\text{y}}$	$\underline{\text{a}}$	$\underline{\text{T}}$

In spite of returning a 'true' value for RTR/Hi&Fr(x), string positions 3 and 4 surface with an RTR feature by virtue of satisfying the licensing structure RTR′($s(x)$) higher in the hierarchy. This allows the span to spread to the beginning of the word. The hierarchy and the observed PSC blocking thus mirror McCarthy's constraint ranking.

7.5.1.2 PSC effects in a composite system

The observed effects are captured in a combined map BMRS system of equations, but they also persist when each spreading process is defined as a separate system and the two combine through composition. For example, let some system of equations L model unhindered leftward RTR spreading. Its output boolean function $\mathrm{RTR}_L(x)$ is defined in (7.33).

$$\text{(7.33)}\quad \mathrm{RTR}_L(x) \;=\; \texttt{if}\ \mathrm{RTR}_L(s(x))\ \texttt{then}\ \top\ \texttt{else}\ \mathrm{RTR}(x)$$

Similarly, let R denote a BMRS system of equations modeling rightward spreading with the blocking condition. The equivalent output boolean function in this system is $\mathrm{RTR}_R(x)$. Note in (7.34) that the blocking condition supersedes the licensing condition in the hierarchy; this produces the effect of rightward spreading which is blocked by any high, front segment.

$$\text{(7.34)}\quad \mathrm{RTR}_R(x) \;=\; \begin{array}{lll} \texttt{if}\ \mathrm{RTR/H\scriptsize{I}\normalsize{\&}F\scriptsize{R}}(x) & \texttt{then}\ \bot & \texttt{else} \\ \texttt{if}\ \mathrm{RTR}_R(p(x)) & \texttt{then}\ \top & \texttt{else} \\ \mathrm{RTR}(x) & & \end{array}$$

Now let $L \otimes R$ be the composition of these systems as defined in this dissertation. The composite system is given in (7.35)

$$\text{(7.35)}\quad \mathrm{RTR}_L(x) \;=\; \texttt{if}\ \mathrm{RTR}_L(s(x))\ \texttt{then}\ \top\ \texttt{else}\ \mathrm{RTR}_R(x)$$

The reader can confirm that the composite system is extensionally equivalent to the combined map system in (7.29), and computes the same set of mappings, e.g. (7.30), (7.31), and (7.32). Computation of an output string proceeds first through the portion of the hierarchy responsible for leftward spread ($\mathrm{RTR}_L(x)$), then through the established hierarchy for rightward spread ($\mathrm{RTR}_R(x)$). Since composition does not alter existing hierarchical relations within individual systems, we may say that hierarchies are *preserved under composition*.

7.5.1.3 BMRS preserves the Subset Criterion

One direct consequence of hierarchical relations in BMRS systems of equations and their preservation under composition is that McCarthy/Prince's Subset Criterion falls out automatically. McCarthy presents a general schema for this criterion as in (7.36), where $\mathbb{L}$ represents a constraint imposing a specific limitation on $\mathbb{M}_i$ (which, recall, is the markedness constraint compelling some process not influenced by blocker $\mathbb{C}$):

(7.36) $\mathbb{L} >> \mathbb{M}_i >> \mathbb{C} >> \mathbb{M}_j >> \mathbb{F}$

$\mathbb{L}$ necessarily outranks $\mathbb{M}_j$ in addition to $\mathbb{M}_i$ given the transitive nature of strict ordering over constraints. The Subset Criterion is thus derived from a basic property of OT: "if $\mathbb{M}_i >> \mathbb{M}_j >> \mathbb{F}$, then the set of constraints that can, in principle, impinge on $\mathbb{M}_i$ is a subset of the set of constraints that can, in principle, impinge on $\mathbb{M}_j$" (239). In other words, when higher-ranked $\mathbb{M}_i$ is subject to a PSC, lower-ranked $\mathbb{M}_j$ may also be subject to the same PSC.

Hierarchies of licensing and blocking structures and their preservation under composition yields the same effect in the BMRS formalism. A simplified example using McCarthy's notation demonstrates this fact. Let some system F model a process licensed by a structure $\mathbb{M}_i$ but which is subject to some (output-oriented) PSC limitation $\mathbb{L}$. A relevant output boolean function $\mathrm{A}_F(x)$ is defined in (7.37); the PSC is ordered before the licenser in the hierarchy.

(7.37)
$$\begin{aligned} \mathrm{A}_F(x) \quad = \quad & \texttt{if } \mathbb{L}(x) \texttt{ then } \bot \texttt{ else} \\ & \texttt{if } \mathbb{M}_i(x) \texttt{ then } \top \texttt{ else} \\ & \mathrm{A}(x) \end{aligned}$$

Similarly, let system G model a process licensed by a structure $\mathbb{M}_j$ but subject to some (also output-oriented) PSC limitation $\mathbb{C}$. An equivalent output boolean function definition for $\mathrm{A}_G(x)$ is shown in (7.38).

(7.38)
$$\begin{aligned} \mathrm{A}_G(x) \quad = \quad & \texttt{if } \mathbb{C}(x) \texttt{ then } \bot \texttt{ else} \\ & \texttt{if } \mathbb{M}_j(x) \texttt{ then } \top \texttt{ else} \\ & \mathrm{A}(x) \end{aligned}$$

Composition $F \otimes G$ follows by the normal mechanism; in this case the default condition $\mathrm{A}(x)$ in system F—the only non-recursively defined boolean function—is indexed with the corresponding definition from G. Importantly, all hierarchical relations between licensing and blocking structures are preserved. The result is that the outer function's PSC limitation $\mathbb{L}(x)$ is calculated before $\mathbb{M}_i(x)$ given the hierarchy *and necessarily* before $\mathbb{M}_j$ given the hierarchical relation between $\mathbb{L}(x)$ and $\mathrm{A}(x)$ in the original system F. In other words, when $\mathbb{M}_i$ is subject to $\mathbb{L}$, so is $\mathbb{M}_j$. An equivalent system illustrates the full hierarchy.

(7.39)
$$\begin{aligned} \mathrm{A}_F(x) \quad = \quad & \texttt{if } \mathbb{L}(x) \texttt{ then } \bot \texttt{ else} \\ & \texttt{if } \mathbb{M}_i(x) \texttt{ then } \top \texttt{ else} \\ & \texttt{if } \mathbb{C}(x) \texttt{ then } \bot \texttt{ else} \\ & \texttt{if } \mathbb{M}_j(x) \texttt{ then } \top \texttt{ else} \\ & \mathrm{A}(x) \end{aligned}$$

This reflects precisely the total order in (7.36), and produces the same effects. Therefore the nature of hierarchical relations and their preservation under composition in BMRS mirrors the "irreflexive, asymmetric, and transitive" nature of the strict ordering relation over OT constraints, a property not derived by rule-based accounts with PSC tags on individual rules.

7.5.1.4 BMRS avoids pathological PSC effects

By McCarthy's account, the Subset Criterion—driven by the basic mechanism of constraint interaction—results in a more restrictive theory of PSC than is available to the rule-based formalism. Davis (1995) posits a hypothetical harmony system where rightward spread is subject to one condition and leftward spread is subject to a different condition. Such a case is predicted to be impossible in OT because it would require a circular ranking. McCarthy illustrates with a toy example using the de-conjoined RTR/Hi and RTR/Fr as separate conditions on rightward and leftward spread. The required rankings are as in (7.40):

(7.40)		Ranking	Interpretation
	a.	RTR/Hi >> RTR-Right	High segments block rightward harmony.
	b.	RTR-Right >> RTR/Fr	Front segments don't block rightward harmony.
	c.	RTR/Fr >> RTR-Left	Front segments block leftward harmony.
	d.	RTR-Left >> RTR/Hi	High segments don't block leftward harmony.

A total order over constraints with these sub-rankings is impossible; RTR/Hi cannot rank *above* RTR-Right and *below* RTR-Left when RTR-Right >> RTR-Left via transitivity.

BMRS systems of equations make the same predictions about the hypothetical case above and thus align with the restrictions on PSC imposed by the Subset Criterion. To see how, consider two systems of equations R (7.41a) and L (7.41b) modeling rightward and leftward spreading with separate PSC conditions:

(7.41) a.
$$\begin{aligned}\mathrm{RTR}_R(x) \;=\;& \texttt{if}\ \mathrm{RTR/H{\small I}}(x)\ \texttt{then}\ \bot\ \texttt{else}\\ & \texttt{if}\ \mathrm{RTR}_R(p(x))\ \texttt{then}\ \top\ \texttt{else}\\ & \mathrm{RTR}(x)\end{aligned}$$

b.
$$\begin{aligned}\mathrm{RTR}_L(x) \;=\;& \texttt{if}\ \mathrm{RTR/F{\small R}}(x)\ \texttt{then}\ \bot\ \texttt{else}\\ & \texttt{if}\ \mathrm{RTR}_L(s(x))\ \texttt{then}\ \top\ \texttt{else}\\ & \mathrm{RTR}(x)\end{aligned}$$

Preservation of hierarchical relations under composition guarantees that every composite system definable from single systems will meet the Subset Criterion, and so also predicts that no BMRS system of equations can describe the hypothetical grammar in (7.40). Instead, the possible compositions $R \otimes L$ and $L \otimes R$ for the example above maintain the subset/superset relationship between conditions on two spreading patterns.

However, BMRS preserves the Subset Criterion for PSC effects differently from OT. Recall that a hierarchy with subrankings in (7.40) is an impossible total order over constraints (and thus an impossible OT grammar) because it requires contradictory ranking relationships—RTR/Hi >> RTR-Left via transitivity with RTR-Right but also the opposite ranking RTR-Left >> RTR/Hi. In principle, it is possible to define separate BMRS systems of equations with these exact relations intact, and then compose them to form a full grammar. The systems in (7.42) append the definitions in (7.41) to include all subrankings from McCarthy's pathological hierarchy.

(7.42) a.
$$\begin{aligned} \mathrm{RTR}_R(x) \;=\; & \texttt{if}\ \mathrm{RTR/H{\scriptstyle I}}(x)\ \texttt{then}\ \bot\ \texttt{else} \\ & \texttt{if}\ \mathrm{RTR}_R(p(x))\ \texttt{then}\ \top\ \texttt{else} \\ & \texttt{if}\ \mathrm{RTR/F{\scriptstyle R}}(x)\ \texttt{then}\ \bot\ \texttt{else} \\ & \mathrm{RTR}(x) \end{aligned}$$

b.
$$\begin{aligned} \mathrm{RTR}_L(x) \;=\; & \texttt{if}\ \mathrm{RTR/F{\scriptstyle R}}(x)\ \texttt{then}\ \bot\ \texttt{else} \\ & \texttt{if}\ \mathrm{RTR}_L(s(x))\ \texttt{then}\ \top\ \texttt{else} \\ & \texttt{if}\ \mathrm{RTR/H{\scriptstyle I}}(x)\ \texttt{then}\ \bot\ \texttt{else} \\ & \mathrm{RTR}(x) \end{aligned}$$

Composing these systems in either direction yields a well-formed (albeit semi-redundant) system that still observes the Subset Criterion, again because of preservation of hierarchical relations under composition. A system of equations equivalent to $R \otimes L$ is given below in (7.43); high segments block rightward and leftward spreading while front segments only block leftward spreading.

(7.43)
$$\begin{aligned} \mathrm{RTR}'(x) \;=\; & \texttt{if}\ \mathrm{RTR/H{\scriptstyle I}}(x)\ \texttt{then}\ \bot\ \texttt{else} \\ & \texttt{if}\ \mathrm{RTR}'(p(x))\ \texttt{then}\ \top\ \texttt{else} \\ & \texttt{if}\ \mathrm{RTR/F{\scriptstyle R}}(x)\ \texttt{then}\ \bot\ \texttt{else} \\ & \texttt{if}\ \mathrm{RTR/F{\scriptstyle R}}(x)\ \texttt{then}\ \bot\ \texttt{else} \\ & \texttt{if}\ \mathrm{RTR}'(s(x))\ \texttt{then}\ \top\ \texttt{else} \\ & \texttt{if}\ \mathrm{RTR/H{\scriptstyle I}}(x)\ \texttt{then}\ \bot\ \texttt{else} \\ & \mathrm{RTR}(x) \end{aligned}$$

Thus unlike OT, the BMRS formalism can capture Davis' hypothetical systems where individual 'rules' (in this case separate systems of equations) are subject to distinct conditions. Unlike Davis' rule-based conception, though, the composition of those systems necessarily obeys the Subset Condition. This is a direct consequence of two basic components of BMRS systems of equations: hierarchies of licensing and blocking structures, and their preservation under composition.

7.5.2 Learning

This final section briefly considers the application of the theory of process interactions introduced in the dissertation to learning. The main purpose here is to identify avenues for future work, and two main veins are addressed. One concerns how the BMRS formalism (and especially BMRS operations) might be leveraged in developing learning algorithms and evaluating existing ones. A second concerns learning of interaction grammars, and whether the conceptualization of them pursued here is tractable from a phonological learning perspective; that is, when the target of learning is a set of functions that combine via a set of operations.

An important result relevant to learning phonological transformations is due to Oncina et al. (1993), who show that subsequential functions are learnable within the Gold paradigm of identification with the limit using positive data (Gold, 1967). More recent work builds on that result, developing learning algorithms for classes of the subsequential functions (Chandlee and Jardine, 2014; Chandlee et al., 2014; Jardine et al., 2014; Chandlee et al., 2015b, and others). This includes procedures for learning ISL (Chandlee, 2014) and OSL (Chandlee et al., 2015a) functions discussed in previous chapters.[19]

These learning procedures are implemented primarily using finite-state methods. Given the advantages of BMRS discussed in §2 of this chapter and in chapter 2, it remains to be seen whether these properties are applicable to learning algorithms. For example, could the ability of BMRS to capture the underlying motivation for phonological processes (recalling Chandlee and Jardine (2020)'s generaliations) be extended to model learning scenarios in a more naturalistic way? And given the similarities between licensing/blocking structures and OT constraints—as well as hierarchies of such structures and rankings of OT constraints—might BMRS provide a means to fruitfully compare learning models in the subsequential framework with OT-based models, where the ultimate learning goal is a hierarchy of constraints (Tesar and Smolensky, 1998, 2000; Tesar, 2014)? Exploring these issues requires a foundation in BMRS learnability. This has yet to be established, but any such undertaking must begin with basic questions such has how a BMRS representation of a function can be learned. Future work can enhance this understanding in tandem with a more thorough understanding of the BMRS formalism.

Previous learning results demonstrate the learnability of subsequential classes like ISL and OSL.

[19] See also (Burness and McMullin, 2019) for a related learning result for tier-based OSL string-to-string functions.

It is known that single ISL functions can model multiple, potentially-interacting generalizations, including opaque ones (Chandlee and Heinz, 2018; Chandlee et al., 2018). So to a certain extent the learnability of ISL functions already presupposes the learnability of at least ISL interactions. What, then, could the current dissertation contribute to a general theory of learning interactions? One possibility is an alternative to a combined map learning scenario, for which the objective is a single function modeling multiple generalizations. Building on the conception of interactions as single functions and operators, an alternate scenario is one for which the learner has two or more separate generalizations (perhaps from data where the process occurs in isolation), and then learns the interaction as an application of operators over those individual generalizations. Given a set of operations, a learner may consider a number of hypotheses concerning which operations to apply and in what order, with the ultimate goal being a composite/PS-join function that conforms to the data. It is yet to be determined whether this perspective offers any insight into learning that the combined map approach does not. Future work can examine this issue in more detail.

8 Conclusion

This dissertation has developed a computational theory of process interactions in phonology. In this theory, individual phonological processes are input-output mappings (functions) represented as BMRS systems of equations, and interactions are the combination of functions via a set of BMRS-definable operators. The current work has proposed a set of two operators. One is a composition operator $\otimes$ whereby the output of one function serves as input to another function. Composite BMRS systems capture the set of interactions derivable via serial rule ordering: feeding, bleeding, counterfeeding, and counterbleeding. The insufficiency of composition prompted the introduction of an additional operator termed parallel satisfaction $\ominus$. This operator combines BMRS systems by enforcing reference to a single input and output string. Parallel satisfaction captures a subset of interactions formalizable via composition, and crucially describes interactions unavailable to a composition analysis.

Using this theory, the analyses presented here have demonstrated that the computational approach—enhanced with BMRS operators—offers wider empirical coverage of attested phonological interactions than is available to SPE and OT. It did so specifically by focusing on a number of outstanding cases in Chinese tone sandhi. Given the compactness of tonal inventories in Chinese, richness of disyllabic systems, and the tendency for targets and triggers to overlap in longer sequences of tones, Chinese tone sandhi is well-suited to such an investigation. Applied to both well-known and recently-described sandhi interactions, the BMRS analyses were shown to outperform rule- and optimization-based accounts. Building on a recent automata-theoretic study by Chandlee (2019), analysis in terms of strictly-local functions clarifies issues of directionality in both Tianjin and Nanjing. The introduction of the parallel satisfaction operator, likewise, provides a solution to so-called paradoxical interactions in Changting as well as the Xiamen tone circle. While these cases have plagued the literature on Chinese tone for distinct reasons—even been purported to exhaust the analytical tools of current theories—both enjoy a straight-forward account using operators over BMRS systems.

It should also be emphasized that the BMRS formalism, in addition to furnishing computational analyses of sandhi interactions that fare better than previous attempts in SPE and OT, stands out among other computational formalisms in its suitability for pursuing such a theory. Despite the equivalence of BMRS with other logical and finite-state approaches (as discussed in the previous chapter), BMRS provides an intuitive and easily-interpretable means for both performing operations over BMRS-definable functions, as well as defining new operators. It also clarifies the contribution of each function joined by an operator by keeping systems mostly intact. The degree of clarity and interpretability afforded by BMRS is not available to other computational formalisms. This is in addition to its general advantages in phonological analysis, namely that it captures the intensional motivation for phonological processes.

The theory pursued in the preceding chapters hinges on the ability to define operators over BMRS systems. This dissertation has also shown that these operations have applications outside of a theory of interactions, namely to questions of representation. A composition operator over BMRS systems of equations was fruitfully applied to the demonstration of bi-interpretability between representational models using BMRS transductions. As discussed in earlier chapters, proving bi-interpretability is a key component in arguments about the notational equivalence of representational theories, but earlier formalisms lack an established composition procedure for logical transduction. The current work closes this gap by defining a composition operator for BMRS transductions.

While tone sandhi is an ideal empirical focus for developing a theory of interactions, the sandhi patterns explored here do not exhaust the full range of possible interactions in phonology. Instead, I address outstanding cases that highlight the insufficiency of other approaches but still fall well within the predictions of the subregular hypothesis. Future work will seek interactions in other domains to test the adequacy of this theory and its predictions.

References

Alderete, J. D. (1999). *Morphologically governed accent in Optimality Theory.* PhD thesis, UMass Amherst.

Alderete, J. D. (2001). Dominance effects as trans-derivational anti-faithfulness. *Phonology*, 18:201–253.

Anderson, S. R. (1987). Tone features. In Fromkin, V. A., editor, *Tone: a Linguistic Survey*, pages 133–176. Academic Press.

Baković, E. (2007). A revised typology of opaque generalisations. *Phonology*, 24(2):217–259.

Baković, E. (2011). Opacity and ordering. In Goldsmith, J., Riggle, J., and Yu, A., editors, *The Blackwell Handbook of Phonological Theory.* Wiley-Blackwell.

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

Baković, E. and Blumenfeld, L. (2018). Overapplication conversion. In Bennett, R., Angeles, A., Brasoveanu, A., Buckley, D., Kalivoda, N., Kawahara, S., McGuire, G., and Padgett, J., editors, *Hana-bana: A Festschrift for Junko Itô and Armin Mester.* Santa Cruz, CA.

Baković, E. and Blumenfeld, L. (2020). Rule interaction conversion operations. *Loquens*, 6(2):062.

Ballard, W. L. (1988). *The History and Development of Tonal Systems and Tonal Alternations in South China.* Institute for the Study of Languages and Cultures of Asia and Africa.

Bao, Z. (1990). *On the nature of tone.* PhD thesis, Massachusetts Institute of Technology.

Bao, Z. (2004). Accentualism in Chinese. *Language and Linguistics*, 5(4):863–889.

Barrie, M. (2006). Tone circles and contrast preservation. *Linguistic Inquiry*, 37(1):131–141.

Beckman, J. N. (1999). *Positional faithfulness.* Garland, New York.

Bhaskar, S., Chandlee, J., Jardine, A., and Oakden, C. (2020). Boolean monadic recursive schemes as a logical characterization of the subsequential functions. *Ms. submitted to LATA.*

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

Bowern, C., Alpher, B., and Round, E. (2013). Yidiny stress, length and truncation reconsidered. *Poster presented at the 44th North East Linguistic Society.*

Büchi, J. R. (1960). Weak second-order arithmetic and finite automata. *Zeitschrift für Mathematische Logik und Grundlagen der Mathmatik*, 6:66—92.

Burness, P. and McMullin, K. (2019). Efficient learning of output tier-based strictly 2-local functions. In *Proceedings of the 16th Meeting on the Mathematics of Language*, pages 78–90.

Cao, Y. (2007). Shandong Lelinghua de liandu biandiao [Pitch shifting and neutral tones of Shandong Leling dialect]. *Dezhou Xueyuan Xuebao [Journal of Dezhou University]*, 2007:55–60.

Chan, M. K. (1995a). An autosegmental analysis of danyang tone sandhi: Some historical and theoretical issues. *Wuyu yanjiu [Studies of the Wu dialects]*, pages 145–184.

Chan, M. K. M. (1985). Fuzhou phonology: a non-linear analysis of tone and stress.

Chan, M. K. M. (1989). On the status of 'basic' tones. *Acta Linguistica Hafniensia*, 21(2):35–64.

Chan, M. K. M. (1995b). An autosegmental analysis of Danyang tone sandhi: Some historical and theoretical issues. *Wuyu yanjiu [Studies of the Wu dialects]*, pages 145–184.

Chan, M. K. M. and Ren, H. M. (1989). Wuxi tone sandhi. *Acta Linguistica Hafniensia*, 21(2):35–64.

Chandlee, J. (2014). *Strictly local phonological processes.* PhD thesis, University of Delaware.

Chandlee, J. (2019). A computational account of tone sandhi interaction. In Hout, K., Mai, A., McCollum, A., Rose, S., and Zalansky, M., editors, *Proceedings of the 2018 Annual Meeting on Phonology.* LSA.

Chandlee, J., Eyraud, R., and Heinz, J. (2014). Learning strictly local subsequential functions. *Transactions of the Association for Computational Linguistics*, 2:491–503.

Chandlee, J., Eyraud, R., and Heinz, J. (2015a). Output strictly local functions. In Kuhlmann, M., Katazawa, M., and Kobele, G. M., editors, *Proceedings of the 14th Meeting on the Mathe-*

matics of Language (MoL 2015), pages 112–125, Stroudsburg, PA. Association for Computational Linguistics.

Chandlee, J., Eyraud, R., Heinz, J., Jardine, A., and Rawski, J. (2019). Learning with partially ordered representations. *arXiv preprint arXiv:1906.07886*.

Chandlee, J. and Heinz, J. (2012). Bounded copying is subsequential: Implications for metathesis and reduplication. In *Proceedings of the 12th Meeting of the ACL Special Interest Group on Computational Morphology and Phonology (SIGMORPHON)*, pages 42–51.

Chandlee, J. and Heinz, J. (2018). Strict locality and phonological maps. *LI*, 49:23–60.

Chandlee, J., Heinz, J., and Jardine, A. (2018). Input strictly local opaque maps. *Phonology*, 35(2):171–205.

Chandlee, J. and Jardine, A. (2014). Learning phonological mappings by learning Strictly Local functions. In Kingston, J., Moore-Cantwell, C., Pater, J., and Staubs, R., editors, *Proceedings of the 2013 Meeting on Phonology (UMass Amherst)*, Proceedings of the Annual Meetings on Phonology. LSA.

Chandlee, J. and Jardine, A. (2019a). Autosegmental input strictly local functions. *Transactions of the Association for Computational Linguistics*, 7:157–168.

Chandlee, J. and Jardine, A. (2019b). Quantifier-free least fixed point functions for phonology. In *Proceedings of the 16th Meeting on the Mathematics of Language*, pages 50–62.

Chandlee, J. and Jardine, A. (2019c). Quantifier-free monadic least-fixed point transductions and subsequential functions. Unpublished ms., Haverford College and Rutgers University.

Chandlee, J. and Jardine, A. (2020). Recursive schemes for phonological analysis. Unpublished ms, Haverford College and Rutgers University.

Chandlee, J., Jardine, A., and Heinz, J. (2015b). Learning repairs for marked structures. In *Proceedings of the 2015 Annual Meeting on Phonology*. LSA.

Chandlee, J. and Lindell, S. (”to appear”). A logical characterization of strictly local functions. In Heinz, J., editor, *Doing Computational Phonology*. OUP.

Chang, L. M.-c. (1992). *A prosodic account of tone, stress, and tone sandhi in Chinese languages*. PhD thesis, University of Hawaii.

Chao, Y.-R. (1930). A system of tone letters. *Le maître phonétique*, 45:24–27.

Chao, Y.-R. (1968). *A grammar of spoken Chinese*. University of California, Berkeley.

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

Chen, M. Y. (1987). The syntax of Xiamen tone sandhi. *Phonology Yearbook*, 4:109–150.

Chen, M. Y. (1992). Argument vs adjenct: Xiamen tone sandhi revisited. Ms., University of California, San Diego.

Chen, M. Y. (2000). *Tone sandhi: Patterns across Chinese dialects*, volume 92. Cambridge University Press.

Chen, M. Y. (2004). Changting Hakka tone sandhi: Analytical challenges. *Language and Linguistics*, 5(4):799–820.

Chen, M. Y., Yan, X., and Wee, L.-H. (2004). *Hakka tone sandhi: Corpus and analytical challenges*. Project on Linguistic Analysis, University of California.

Chen, M. Y. and Zhang, H. (1997). Lexical and postlexical tone sandhi in Chongming. In Wang, J. and Smith, N., editors, *Studies in Chinese phonology*, pages 13–52. North-Holland.

Chen, S. and Wiltshire, C. (2013). Tone realization in younger versus older speakers of Nanjing dialect. In Jing-Schmidt, Z., editor, *Increased Empiricism: recent advances in Chinese linguistics*, pages 147–170. John Benjamins.

Chen, T.-Y. (2010). Some remarks on contour tone units. *Journal of East Asian Linguistics*, 19(2):103–135.

Chen, Y. (2008). Revisiting the phonetics and phonology of Shanghai tone sandhi. In *Proceedings of the Fourth Conference on Speech Prosody*, pages 253–256.

Cheng, R. L. (1968). Tone sandhi in Taiwanese. *Linguistics*, 6(41):19–42.

Cheng, R. L. (1973). Some notes on tone sandhi in Taiwanese. *Linguistics*, 11(100):5–25.

Chiang, M.-h. (1985). Taichung xian Dongshi keyu yinyun yanjiu [A phonological study of Dongshi Hakka in Taichung County. MA thesis, National Taiwan University.

Chomsky, N. (1956). Three models for the description of language. *IRE Transactions on Information Theory*, pages 113–124.

Chomsky, N. (1959). On certain formal properties of grammars. *Information and Control*, 2:137–167.

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

Chomsky, N. and Halle, M. (1968b). The sound pattern of english.

Clements, G. N. (1983). The hierarchical representation of tone features. *Current Approaches to African linguistics*, 1:145–176.

Clements, G. N. (1985). The geometry of phonological features. *Phonology*, 2:225–252.

Courcelle, B. (1994). Monadic second-order definable graph transductions: a survey. *Theoretical Computer Science*, 126:53–75.

Dai, Q. (1990). Jingpo yu de shengdiao [The tones of the Jingpho language]. In Dai, Q., editor, *Zang-Mian yuzu yuyan yanjiu [Research on Tibeto-Burman languages]*, pages 199–212. Yunnan Minzu Chubanshe.

Danis, N., Blum, E., Iacoponi, L., Mitchley, H., and Jardine, A. (2017). A computational method for evaluating theories of phonological representation. Talk given at Chicago Linguistic Society 53.

Danis, N. and Jardine, A. (2019). Q-Theory representations are logically equivalent to autosegmental representations. In *Proceedings of the Society for Computation in Linguistics*, volume 2.

Davis, S. (1995). Emphasis spread in Arabic and grounded phonology. *Linguistic Inquiry*, 26:465–498.

Dong, T.-h. (1960). Sige Minnan fangyan [Four southern Min dialects]. *The Bulletin of the Institute of History and Philology, Academia Sinica*, 30:729–1042.

Du, T.-c. (1983). *Tone and stress in Taiwanese*. PhD thesis, University of Illinois, Urbana-Champaign.

Duanmu, S. (1990). *A formal study of syllable, tone, stress and domain in Chinese languages*. PhD thesis, Massachusetts Institute of Technology.

Duanmu, S. (1991). Stress and syntax-phonology mismatches: tonal domains in Danyang and Shanghai. In Bates, D., editor, *Proceedings of the Tenth West Coast Conference on Formal Linguistics*, pages 127–137. Stanford Linguistic Association.

Duanmu, S. (1994). Contour tone units. *Linguistic Inquiry*, 25(4):555–608.

Duanmu, S. (1995). Metrical and tonal phonology of compounds in two Chinese dialects. *Language*, pages 225–259.

Elgot, C. C. and Mezei, J. E. (1965). On relations defined by generalized finite automata. *IBM Journal of Research and Development*, 9:47—68.

Enderton, H. B. (2001). *A mathematical introduction to logic*. Elsevier.

Engelfriet, J. and Hoogeboom, H. J. (2001). Mso definable string transductions and two-way finite transducers. *ACM Transactions on Computational Logic*, 2:216–254.

Fei, J. and Sun, L. (1993). *Nanjing fangyan zhi [Survey of the Nanjing dialect]*. Nanjing Chubanshe, Nanjing.

Filiot, E. (2015). Logic-automata connections for transformations. In *Logic and Its Applications (ICLA)*, pages 30–57. Springer.

Friedman, H. M. and Visser, A. (2014). When bi-interpretability implies synonymy. *Logic Group Preprint Series*, 320:1–19.

Gold, E. M. (1967). Language identification in the limit. *Information and control*, 10(5):447–474.

Goldsmith, J. A. (1976). *Autosegmental phonology*, volume 159. Indiana University Linguistics Club Bloomington.

Gu, Q. (2015). Nanjing fangyan dangjiang liangshe rushengyun de gongshi bianyi ji shili yanbian yanjiu [On synchronic variation and diachronic evolution of the entering tone in the dang and jiang classes in the Nanjing dialect]. *Yuyan Kexue*, 14(4):384–396.

Halle, M. (1962). Phonology in generative grammar. *Word*, 18(1-3):54–72.

Harms, R. T. (1968). *Introduction to phonological theory*. Prentice-Hall, Englewood Cliffs.

Hayes, B. (1995). *Metrical stress theory: Principles and case studies*. University of Chicago Press.

Heinz, J. (2018). The computational nature of phonological generalizations. In Hyman, L. and Plank, F., editors, *Phonological Typology*, Phonetics and Phonology, chapter 5, pages 126–195. De

Gruyter Mouton.

Heinz, J. and Idsardi, W. (2011). Sentence and word complexity. *Science*, 333(6040):295–297.

Heinz, J. and Idsardi, W. (2013). What complexity differences reveal about domains in language. *Topics in cognitive science*, 5(1):111–131.

Heinz, J. and Lai, R. (2013). Vowel harmony and subsequentiality. In *Proceedings of the 13th Meeting on the Mathematics of Language (MoL 13)*, pages 52–63.

Herzallah, R. (1990). *Aspects of Palestinian Arabic phonology: a non-linear approach*. PhD thesis, Cornell University.

Hodges, W. (1997). *A shorter model theory*. Cambridge University press.

Hopcroft, J. and Ullman, J. (1979). *Introduction to Automata Theory, Languages, and Computation*. Addison Wesley, Reading, MA.

Hou, J. (1980). Pingyao fangyan de liandu biandiao [tone sandhi in the yinchuan dialect]. *Fangyan*, 1980(1):1–14.

Howard, I. (1972). *A directional theory of rule application in phonology*. PhD thesis, Massachusetts Institute of Technology.

Hsiao, Y. E. (1991). *Syntax, rhythm and tone: a triangular relationship*. PhD thesis, University of California, San Diego.

Hsiao, Y. E. (2015). Tonal chain shifts in Taiwanese: a comparative markedness approach. *Capturing phonological shades within and across languages*, pages 142–165.

Hsieh, F.-f. (2005). Tonal chain-shifts as anti-neutralization-induced tone sandhi. In *U. Penn Working Papers in Linguistics, Volume 11.1*, pages 99–112.

Hsieh, H.-I. (1970). The psychological reality of tone sandhi rules in Taiwanese. In *6th meeting of the Chicago Linguistic Society*, pages 489–503.

Hsu, H. (1994). *Constraint-based phonology and morphology: a survey of languages in China*. PhD thesis, University of California, San Diego.

Hsu, H. (2005). Two constraints on tonal derivation in chinese. *Taiwan Journal of Linguistics*, 3(2):119–147.

Hsu, H.-c. (1992). Domain of tone sandhi in idioms: A tug of war between the foot formation rule and the tone group formation. Paper presented at the Fifth North American Conference on Chinese Linguistics, Ann Arbor, Michigan.

Hsu, H.-c. (1995). Trisyllabic tone sandhi in the Changting Hakka dialect. *Journal of Chinese Linguistics*, 23(1):42–86.

Hsu, H.-c. (2002). On non-recycling of tonal derivation in Chinese dialects. *Tsing Hua Journal of Chinese Studies*, 32:357–387.

Hung, T. T. (1985). Tianjing tone sandhi: a simple analysis. Ms., University of California, San Diego.

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

Hyman, L. (2011). Tone: Is it different? In Goldsmith, J., Riggle, J., and Yu, A., editors, *The Blackwell Handbook of Phonological Theory*, pages 197–238. Wiley-Blackwell.

Hyman, L. M. (1975). *Phonology: Theory and Analysis*. New York: Holt, Rinehart and Winston.

Hyman, L. M. (1986). The representation of multiple tone heights. *The phonological representation of suprasegmentals: studies on African languages offered to John M. Stewart on his 60th birthday*, pages 109–152.

Hyman, L. M. (2007). Universals of tone rules: 30 years later. In Riad, T. and Gussenhoven, C., editors, *Tones and tunes: studies in word and sentence prosody*, pages 1–34. De Gruyter Mouton.

Hyman, L. M. (2014). How autosegmental is phonology? *The Linguistic Review*, 31(2):363–400.

Hyman, L. M. and Schuh, R. G. (1974). Universals of tone rules. *Linguistic Inquiry*, 5:81–115.

Hyman, L. M. and VanBik, K. (2004). Directional rule application and output problems in Hakha Lai tone. *Language and Linguistics*, 5(4):821–861.

Jansche, M. (1998). A two-level take on Tianjin tone. *Proceedings of the Third ESSLLI Student Session, 10th European Summer School on Logic, Language and Information, Saarbrücken, Germany*, pages 162–174.

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

Jardine, A. (2018). Learning local and non-local interactions in tone. Unpublished ms., Rutgers University.

Jardine, A. (2020). Melody learning and long-distance phonotactics in tone. *Natural Language and Linguistic Theory*, 38:1145–1195.

Jardine, A., Chandlee, J., Eyraud, R., and Heinz, J. (2014). Very efficient learning of structured classes of subsequential functions from positive data. In *Proceedings of the 12th International Conference on Grammatical Inference (ICGI 2014)*, JMLR Workshop Proceedings, pages 94–108.

Jardine, A. and Oakden, C. (2020). Some further properties of BMRS transductions. Unpublished ms, Rutgers University.

Johnson, C. D. (1972). *Formal aspects of phonological description*. Mouton.

Kaplan, R. and Kay, M. (1994). Regular models of phonological rule systems. *Computational Linguistics*, 20:331–78.

Karttunen, L. (1993). Finite-state constraints. In Goldsmith, J., editor, *The last phonological rule: reflections on constraints and derivations*, pages 173–194. University of Chicago Press, Chicago.

Karttunen, L. (1998). The proper treatment of optimality in computational phonology: plenary talk. In *Proceedings of the International Workshop on Finite State Methods in Natural Language Processing*, pages 1–12. Association for Computational Linguistics.

Kean, M.-L. (1974). The strict cycle in phonology. *Linguistic Inquiry*, 5(2):179–203.

Kennedy, G. A. (1953). Two tone patterns in Tangsic. *Language*, 29:367–373.

Kenstowicz, M. and Kisseberth, C. (1971). Unmarked bleeding orders. *Studies in the Linguistic Sciences*, 1(1):8–28.

Kiparsky, P. (1971). Historical linguistics. In Dingwall, W. O., editor, *A Survey of Linguistic Science*, pages 576–642. University of Maryland Linguistics Program.

Kiparsky, P. (1976). Abstractness, opacity, and global rules. In Koutsoudas, A., editor, *The Ordering and Application of Rules*. Mouton.

Kiparsky, P. (1982). From cyclic phonology to lexical phonology. In van der Hulst, H. and Smith, N., editors, *The Structure of Phonological Representations, vol 1*, pages 131–175. Dordrecht.

Kirchner, R. (1996). Synchronic chain shifts in Optimality Theory. *Linguistic Inquiry*, 27(2):341–350.

Kisseberth, L. M. and Kenstowicz, M. (1977). *Topics in phonological theory*. New York: Academic Press.

Kong, H. (2008). Hefei fangyan shengdiao de youxuanlun fenxi [an ot analysis of tones in the hefei dialect]. *Hefei Xueyuan Xuebao*, 25(2):30–32.

Koser, N. and Jardine, A. (2020). The computational nature of stress assignment. In Baek, H., Takahashi, C., and Yeung, A. H.-L., editors, *Proceedings of the 2019 Meeting on Phonology*, Proceedings of the Annual Meetings on Phonology. Washington, DC: LSA.

Koser, N., Oakden, C., and Jardine, A. (2019). Tone association and output locality in non-linear structures. In Hout, K., Mai, A., McCollum, A., Rose, S., and Zalansky, M., editors, *Supplemental Proceedings of the 2018 Annual Meeting on Phonology*. LSA.

Koskenniemi, K. (1983). *Two-level morphology: a general computational model for word-form recognition and production*. PhD thesis, University of Helsinki.

Ku, T.-c. (1993). How does 'metrical structures' predict 'tonal domains' in Taiwanese? Unpublished ms, University of Michigan, Ann Arbor.

Lai, W. (2002). Jingpo yu shuang yinjie hecheng ci bian diao de youxuan jieshi [An Optimality Theory analysis of tone sandhi in Jingpho disyllabic compounds]. Ms, Chinese University of Hong Kong.

Lamont, A. (2018). Precedence is pathological: The problem of alphabetical sorting. In *Proceedings of WCCFL 36*.

Li, C. N. and Thompson, S. A. (1989). *Mandarin Chinese: A functional reference grammar*. University of California Press.

Li, P. J.-K. (1985). A secret language in Taiwanese. *Journal of Chinese Linguistics*, 13(1):91–121.

Li, P. J.-k. (1997). Taiwan de mimi yuyan [Secret languages in Taiwan]. pages 76–102.

Li, Q. and Chen, Y. (2012). Trisyllabic tone sandhi in Tianjin Mandarin. In *Tonal Aspects of Languages-Third International Symposium*.

Li, R. (1965). Changtinghua liangyinjie, sanyinjie de liandu biandiao [Disyllabic and trisyllabic tone

sandhi in Changting]. *Xiamen Daxue Xuebao*, 1965(2):57–61.
Li, R. (1984). Wenling fangyan de liandu biandiao [Tone sandhi in the Wenling dialect]. *Fangyan*, 1:1–29.
Li, X. and Liu, S. (1985). Tianjin fangyan de liandu biandiao [Tone sandhi in the Tianjin dialect]. *Zhongguo Yuwen*, 1:76–80.
Li, Y. (2015). Tone sandhi and tonal coarticulation in Fuzhou Min. In *Proceedings of ICPhS 2015*.
Liberman, M. and Pierrehumbert, J. (1984). Intonational invariance under changes in pitch range and length. In Aronoff, M. and Oehrle, R. T., editors, *Language, sound, structure*, pages 157–233. MIT Press, Cambridge, MA.
Libkin, L. (2013). *Elements of finite model theory*. Springer Science & Business Media.
Lin, H.-s. (2008). Variable directional applications in Tianjin tone sandhi. *Journal of East Asian Linguistics*, 17(3):181.
Lin, J.-w. (1994). Lexical government and tone group formation in Xiamen Chinese. *Phonology*, 11(2):237–275.
Lindell, S. and Chandlee, J. (2016). A logical characterization of strictly local functions. Paper presented at the Fourth Workshop on Natural Language and Computer Science, New York, NY.
Liu, D. and Li, R. (1995). *Nanjing Fangyan Cidian [Dictionary of Nanjing Dialect]*. Jiangsu Jiaoyu Chubanshe.
Lü, S. (1980). Danyang fangyan de shengdiao xitong [tone system of the Danyang dialect]. *Fangyan*, 1980(2):85–122.
Łubowicz, A. (2003). *Contrast preservation in phonological mappings*. PhD thesis, UMass Amherst.
Lundskær-Nielsen, T. and Holmes, P. (2011). *Danish: An essential grammar*. Routledge.
Luo, H. (2013). Long-distance consonant agreement and subsequentiality. ms., University of Delaware.
Luo, H. (2017). Long-distance consonant agreement and subsequentiality. *Glossa*, 2(1):1–25.
Luo, M. (1982). Fujian Changting Kejiahua de liandu biandiao [Tone sandhi in the Hakka dialect of Changting, Fujian]. *Yuyan Yanjiu*, 3:188–197.
Ma, Q. (2005). Tianjinhua liandu biandiao zhi mi de youxuanlun jieshi [An OT account of the Tianjin tone sandhi paradox]. *Zhongguo Yuwen*, 2005(6):561–568.
Ma, Q. (2009). Nanjing fangyan liangzizu liandu biandiao youxuanlun fenxi [An OT solution to bisyllabic tone sandhi in Nanjing dialect]. *Yuyan Yanjiu*, 29(1):27–32.
Ma, Q. and Li, X. (2014). Nanjing fangyan sanzizu liandu biandiao de shiyan diaocha yu fenxi [an experimental study of trisyllabic tone sandhi in the nanjing dialect]. *Tongji Daxue Xuebao*, 25(4):100–107.
Magri, G. (2018). Idempotency in optimality theory. *Journal of Linguistics*, 54(1):139–187.
Mamadou, T. and Jardine, A. (2020). Representation and the computation of long distance tone processes. In Baek, H., Takahashi, C., and Yeung, A. H.-L., editors, *Supplemental Proceedings of the 2019 Meeting on Phonology*, Proceedings of the Annual Meetings on Phonology. Washington, DC: LSA.
Mascaró, J. (1976). *Catalan phonology and the phonological cycle*. PhD thesis, Massachusetts Institute of Technology.
Matisoff, J. A. (1974). The tones of Jinghpaw and Lolo-Burmese: common origin vs. independent development. *Acta Linguistica Hafniensia*, 15(2):153–212.
McCarthy, J. J. (1997). Process-specific constraints in Optimality Theory. *Linguistic Inquiry*, 28:231–251.
McCarthy, J. J. (1999). Sympathy and phonological opacity. *Phonology*, 16:331–399.
McCarthy, J. J. (2003). Sympathy, cumulativity, and the Duke-of-York gambit. In Féry, C. and van de Vijver, R., editors, *The Syllable in Optimality Theory*, pages 23–76. Cambridge University Press.
McCarthy, J. J. (2007). *Hidden generalizations: phonological opacity in Optimality Theory*. Equinox Publishing.
McCawley, J. D. (1968). *The phonological component of a grammar of Japanese*. Mouton, The Hague.
McCollum, A., Bakovic, E., Mai, A., and Meinhardt, E. (2017). Conditional blocking in Tutrugbu

requires non-determinism: implications for the subregular hypothesis. Paper presented at NELS 48.

McCollum, A. G., Baković, E., Mai, A., and Meinhardt, E. (2020). Unbounded circumambient patterns in segmental phonology. *Phonology*, 37(2):215–255.

McCollum, A. G. and Essegbey, J. (2018). Unbounded harmony is not always myopic: Evidence from Tutrugbu. In *35th West Coast Conference on Formal Linguistics*, pages 251–258. Cascadilla Proceedings Project.

Mei, T. (1977). Tones and tone sandhi in 16th century Mandarin. *Journal of Chinese Linguistics*, 5:237–260.

Mei, T.-L. (1970). Tones and prosody in Middle Chinese and the origin of the rising tone. *Harvard Journal of Asiatic Studies*, 30:86–110.

Meinhardt, E., Mai, A., Baković, E., and McCollum, A. (2020). On the proper treatment of weak determinism: subsequentiality and simultaneous application in phonological maps. Ms., UC San Diego and Rutgers University.

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

Mohri, M., Pereira, F., and Riley, M. (2000). The design principles of a weighted finite-state transducer library. *Theoretical Computer Science*, 231(1):17–32.

Mohri, M., Pereira, F., and Riley, M. (2002). Weighted finite-state transducers in speech recognition. *Computer Speech & Language*, 16(1):69–88.

Mohri, M. and Sproat, R. (1996). An efficient compiler for weighted rewrite rules. In *34th Meeting of the Association for Computational Linguistics (ACL '96), Proceedings of the Conference, Santa Cruz, California*. ACL.

Moreton, E. (2004). Non-computable functions in Optimality Theory. In McCarthy, J. J., editor, *Optimality theory in phonology: A reader*, pages 141–164. Blackwell Publishing.

Mortensen, D. (2002). Semper infidelis: Theoretical dimensions of tone sandhi chains in Jingpho and A-hmao. Unpublished ms, University of California, Berkeley.

Mortensen, D. (2004). Semper infidelis: Anti-identity in A-hmao and Jingpho tone sandhi chains. In *78th annual meeting of Linguistic Society of America*.

Mortensen, D. (2006). *Logical and substantive scales in phonology*. PhD thesis, University of California Berkeley.

Moschovakis, Y. N. (2019). *Abstract recursion and intrinsic complexity*, volume 48. Cambridge University Press.

Nespor, M. and Vogel, I. (1987). *Prosodic phonology*. Foris, Dordrecht.

Norman, J. (1988). *Chinese*. Cambridge University Press, Cambridge.

Oakden, C. (2012). Tone sandhi in the Nanjing dialect: a phonological analysis. MA thesis, University of Arizona.

Oakden, C. (2017). Checked tone merger in the Nanjing dialect: An acoustic analysis. In *Proceedings of the 29th North American Conference on Chinese Linguistics (NACCL-29)*, volume 1.

Oakden, C. (2019a). The output also matters: QFLFP logic and rule interaction in Tianjin sandhi. Unpublished ms., Rutgers University.

Oakden, C. (2019b). Spreading, copying, and notational equivalence in tonal geometry. Unpublished ms., Rutgers University.

Oakden, C. (2020). Notational equivalence in tonal geometry. *Phonology*, 37(2):1–40.

Oakden, C., Bhaskar, S., and Jardine, A. (2020). Some further properties of BMRS transductions. Unpublished ms., Rutgers University and Københavns Universitet. Available at https://chrisoakden.com/files/Further_properties_BMRS.pdf.

Oakden, C. and Chandlee, J. (2020). A computational analysis of tone sandhi ordering paradoxes. In *Proceedings of the 50th Annual Meeting of the North East Linguistic Society (NELS 50)*, volume 1.

Oakden, C. and Jardine, A. (2020). Process-specific constraints in BMRS. Unpublished ms, Rutgers University.

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

Onn, F. M. (1980). *Aspects of Malay phonology and morphology: A generative approach*. Universiti

Kebangsaan Malaysia.

Padgett, J. (1995). Partial class behavior and nasal place assimilation. In Suzuki, K. and Elzinga, D., editors, *Proceedings of the 1995 Southwestern Workshop on Optimality Theory*.

Pater, J. (2018). Substance matters: a reply to Jardine (2016). *Phonology*, 35(1):151–156.

Payne, A. (2014). Dissimilation as a subsequential process. In *Proceedings of the 44th Meeting of the North East Linguistic Society (NELS)*.

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

Pierrehumbert, J. and Beckman, M. E. (1988). *Japanese tone structure*. MIT Press, Cambridge, MA.

Postal, P. M. (1968). *Aspects of phonological theory*. Harper & Row, New York.

Prince, A. and Smolensky, P. (1993). Optimality Theory: Constraint interaction in generative grammar. *Rutgers University Center for Cognitive Science Technical Report*, 2.

Prince, A. and Smolensky, P. (2004). *Optimality Theory: Constraint Interaction in Generative Grammar*. Blackwell Publishing.

Pulleyblank, D. (1986). *Tone in lexical phonology*, volume 4. Springer Science & Business Media.

Pulleyblank, E. G. (1978). The nature of the Middle Chinese tones and their development to Early Mandarin. *Journal of Chinese Linguistics*, 6(2):173–203.

Pullum, G. K. (1972). Sequential and simultaneous rule application in Spanish phonology. *Lingua*, 38:221–262.

Qian, Z. (1981). *Yantai Fangyan Baogao [A Report on the Dialect of Yantai]*. Jilin: Qilu Shushe.

Qian, Z. and Zhu, G. (1998). *Jinanhua yindang [The sound system of Jinan dialect*. Shanghai Jiaoyu Chubanshe, Shanghai.

Rao, C. (1987). Fujian Changting (Kejia) fangyan de liandu biandiao [Tone sandhi in the Hakka dialect of Changting, Fujian]. *Zhongguo Yuwen*, 1987(3):191–200.

Sagey, E. C. (1986). *The representation of features and relations in non-linear phonology*. PhD thesis, Massachusetts Institute of Technology.

Schuh, R. G. (1978). Tone rules. In Fromkin, V., editor, *Tone: a linguistic survey*, pages 221–256. Academic Press.

Schützenberger, M. P. (1977). Sur une variante des fonctions séquentielles. *Theoretical Computer Science*, 4(1):47–57.

Selkirk, E. and Shen, T. (1990). Prosodic domains in Shanghai Chinese. In Inkelas, S. and Zec, D., editors, *The Phonology-Syntax Connection*, pages 313–338. UCP.

Selkirk, E. O. (1984). *Phonology and syntax*. MIT Press, Cambridge, MA.

Shen, X. S. (1990). Tonal coarticulation in Mandarin. *Journal of Phonetics*, 18(2):281–295.

Shen, X. S. (1992). On tone sandhi and tonal coarticulation. *Acta Linguistica Hafniensia*, 25(1):83–94.

Sherard, M. (1972). *Shanghai phonology*. PhD thesis, Cornell University.

Shi, F. (1990). *Hanyu he Dong-Tai yu de shengdiao geju [Tone paradigms in Chinese and Kam-Tai languages]*. PhD thesis, Nankai University.

Shieber, S. M. (1985). Evidence against the context-freeness of natural language. *Linguistics and Philosophy*, 8:333–343.

Shih, C. (1986). *The Prosodic Domain of Tone Sandhi in Chinese*. PhD thesis, University of California, San Diego.

Shih, C. (1988). Tone and intonation in Mandarin. Ms., Cornell University and AT&T Laboratories.

Shih, C. (1997). Mandarin third tone sandhi and prosodic structure. In Jialing, W. and Smith, N., editors, *Studies in Chinese phonology*, pages 81–123. Mouton de Gruyter, Berlin.

Shih, C. and Sproat, R. (1992). Variations of the Mandarin rising tone. In *Proceedings of the IRCS workshop on prosody in natural speech*, volume 92, pages 193–200.

Shih, S.-h. (2017). Binary and focus in prosodic phrasing: new evidence from Taiwan Mandarin. In *Proceedings of AMP 2016*.

Shih, S.-h. (2018). On the existence of sonority-driven stress in Gujarati. *Phonology*, 35(2):327–364.

Snider, K. L. (1990). Tonal upstep in Krachi: Evidence for a register tier. *Language*, pages 453–474.

Song, Y. (2006). Nanjing fangyan shengdiao shiyan yanjiu [a phonetic experiment on Nanjing tones. MA thesis, Nanjing Normal University.

Strother-Garcia, K. (2018). Imdlawn Tashlhiyt Berber syllabification is quantifier-free. *Proceedings of the Society for Computation in Linguistics (SCiL) 2018*, pages 145–153.

Strother-Garcia, K. (2019). *Using model theory in phonology: a novel characterization of syllable structure and syllabification*. PhD thesis, University of Delaware.

Strother-Garcia, K. and Heinz, J. (2015). Logical foundations of syllable representations. Poster presented at the 2015 Annual Meeting on Phonology.

Strother-Garcia, K., Heinz, J., and Hwangbo, H. J. (2016). Using model theory for grammatical inference: a case study from phonology. In *Proceedings of the 13th International Conference on Grammatical Inference*, pages 66–78.

Sun, H. (2003). Nanjing fangyan shengdiao de ruogan wenti [a discussion on tones in the Nanjing dialect]. *Nanjing Xiaozhuang Xueyuan Xuebao*, 1:34–40.

Sun, X. and Huang, T. (2015). Gradience in contextual tonal realization processes: An instrumental study of Nanjing Chinese. In *Proceedings of ICPhS 2015*.

Tan, F. (1987). Tone sandhi in the Tianjin dialect. *Journal of Chinese Linguistics*, pages 228–246.

Tang, Z. (2019). Rusheng housaiwei shengli biaozheng de jiliang fenxi [a quantitative study on the physiological representation of glottal stops in entering tone]. *Zhongguo Yuwen*, 2019(6):709–722.

Tesar, B. (2014). *Output-Driven Phonology: Theory and Learning*. Cambridge Studies in Linguistics.

Tesar, B. and Smolensky, P. (1998). Learnability in optimality theory. *Linguistic Inquiry*, 29(2):229–268.

Tesar, B. and Smolensky, P. (2000). *Learnability in optimality theory*. MIT Press.

Thomas, G. (2008). An analysis of the Xiamen tone circle. In *Proceedings of the 27th West Coast Conference on Formal Linguistics*, pages 422–430. Cascadilla Proceedings Project, Somerville, MA.

Tsay, J. and Myers, J. (1996). Taiwanese tone sandhi as allomorph selection. In *Annual Meeting of the Berkeley Linguistics Society*, volume 22, pages 395–405.

Wang, H. S. (1995). *Experimental studies in Taiwanese phonology*. Crane Publishing Company.

Wang, P. (1988). Changzhou fangyan de liandu biandiao [Tone sandhi in the Changzhou dialect]. *Fangyan*, 1988(3):177–194.

Wang, W. S. and Li, K.-P. (1967). Tone 3 in Pekingese. *Journal of Speech and Hearing Research*, 10(3):629–636.

Wang, W. S.-Y. (1967). Phonological features of tone. *International Journal of American Linguistics*, 33:93–105.

Wang, X. and Lin, Y.-H. (2017). No more conflicting directionality: metrical conditions on Tianjin Chinese trisyllabic tone sandhi. In *Proceedings of AMP 2016*.

Wee, L.-H. (2010). A percolative account of tianjin tone sandhi. *Language and Linguistics*, 11(1):21–64.

Wen, D. and Shen, M. (1999). *Taiyuanhua yindang [The sound system of Taiyuan dialect*. Shanghai Jiaoyu Chubanshe, Shanghai.

Wilson, C. (2003). Analyzing unbounded spreading with constraints: marks, targets, and derivations. Unpublished ms., UCLA.

Wright, M. S. (1983). *A metrical approach to tone sandhi in Chinese dialects*. PhD thesis, UMass Amherst.

Wu, X. and Liu, D. (1981). Shandong rizhao fangyan zhong de liandu biandiao [tone sandhi in the rizhao dialect of shandong]. *Xiandai Yuwen*, 2010(1):106–107.

Xu, B., Z.Z., T., and Qian, N. (1981). Xinpai shanghai fangyan de liandu biandiao [tone sandhi in the new shanghai dialect]. *Fangyan*, 1981(2):145–155.

Xu, Y. (1994). Production and perception of coarticulated tones. *The Journal of the Acoustical Society of America*, 95(4):2240–2253.

Yan, S. (1981). Gaoan (Laowu Zhoujia) fangyan de yuyin xitong [The phonetic system of Gao'an (Laowu Zhoujia]. *Fangyan*, 1981(2):104–121.

Yang, F.-m. (1963). A preliminary study of the Yaoping hakka dialect as spoken in Hsinchu, Taiwan. In *Proceedings of the International Conference on Oriental Studies. Tokyo*.

Yip, M. (1980). *The tonal phonology of Chinese*. PhD thesis, Massachusetts Institute of Technology.

Yip, M. (1989). Contour tones. *Phonology*, 6:149–174.

Yip, M. (2002). *Tone*. Cambridge University Press.

You, R. (1992). *Hanyu fangyan gaiyao [An introduction to Chinese dialectology]*. Jiaoyu Chubanshe, Shanghai.

Yuan, J. (1960). *Hanyu fangyan gaiyao [An introduction to Chinese dialectology]*. Wenzi Gaige Chubanshe, Beijing.

Yue-Hashimoto, A. O. (1987). Tone sandhi across chinese dialects. *Chinese Language Society of Hong Kong (Ed.), Wang Li memorial volumes, English volume*, pages 445–474.

Zee, E. and Maddieson, I. (1980). Tones and tone sandhi in Shanghai: phonetic evidence and phonological analysis. *Glossa*, 14:45–88.

Zhan, B. (1993). Guangdong sheng Yaoping fangyan jiyin [Transcription of Yaoping dialects in Guangdong province. *Fangyan*, 58:129–141.

Zhang, H. (1985). Zhenjiang fangyan de liandu biandiao [Tone sandhi in the Zhenjiang dialect]. *Fangyan*, 1985(3):191–204.

Zhang, J. (2007). A directional asymmetry in chinese tone sandhi systems. *Journal of East Asian Linguistics*, 16(4):259–302.

Zhang, J. (2014). Tones, tonal phonology, and tone sandhi. In Huang, C.-T., Li, Y.-H., and Simpson, A., editors, *The Handbook of Chinese Linguistics*, chapter 4, pages 443–464. John Wiley and Sons.

Zhang, J., Lai, Y., and Turnbull-Sailor, C. (2006). Wug-testing the tone circle in Taiwanese. In *Proceedings of the 25th West Coast Conference on Formal Linguistics*, pages 453–461. Somerville: MA: Cascadilla Proceedings Project.

Zhang, J., Lai, Y., and Turnbull-Sailor, C. (2009). Opacity, phonetics, and frequency in Taiwanese tone sandhi. In *Current issues in unity and diversity of languages: Collection of papers selected from the 18th International Congress of Linguists*, pages 3019–3038. Linguistic Society of Korea Seoul.

Zhang, J., Lai, Y., and Turnbull-Sailor, C. (2011). Modeling Taiwanese speakers knowledge of tone sandhi in reduplication. *Lingua*, 121(2):181–206.

Zhang, J. and Liu, J. (2011). Tone sandhi and tonal coarticulation in Tianjin Chinese. *Phonetica*, 68(3):161–191.

Zhang, N. (1997). The avoidance of the third tone sandhi in Mandarin Chinese. *Journal of East Asian Linguistics*, 6(4):293–338.

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