Unordered rules that only apply to the input are not more complex than ordered rules

Scott Nelson
scott.nelson@stonybrook.edu

Bromberger and Halle's (1989) Principle

Phonological rules are ordered with respect to one another. A phonological rule R does not apply necessarily to the underlying representation; rather, R applies to the derived representation that results from the application of each applicable rule preceding R in the order of the rules.

Rule 1: Intervocalic Tapping

{t,d} → ɾ / V́_V

plotting	[ˈplaɾɪ̃ŋ]	*plodding*	[ˈplaɾɪ̃ŋ]
wetting	[ˈwɛɾɪ̃ŋ]	*wedding*	[ˈwɛɾɪ̃ŋ]
butting	[ˈbʌɾɪ̃ŋ]	*budding*	[ˈbʌɾɪ̃ŋ]

Rule 2: Canadian Raising

ˈaɪ → ˈʌɪ / _t

rice	[ɹʌɪs]	*rise*	[ɹaɪz]
tripe	[tʃɹʌɪp]	*tribe*	[tʃɹaɪb]
life	[lʌɪf]	*live*	[laɪv]

Ordering Effects

UR	/ˈɹaɪtɪ̃ŋ/	/ˈɹaɪdɪ̃ŋ/
Rule 1	ˈɹaɪɾɪ̃ŋ	ˈɹaɪɾɪ̃ŋ
Rule 2	–	–
SR	[ˈɹaɪɾɪ̃ŋ]	[ˈɹaɪɾɪ̃ŋ]

UR	/ˈɹaɪtɪ̃ŋ/	/ˈɹaɪdɪ̃ŋ/
Rule 2	ˈɹʌɪtɪ̃ŋ	–
Rule 1	ˈɹʌɪɾɪ̃ŋ	ˈɹaɪɾɪ̃ŋ
SR	[ˈɹʌɪɾɪ̃ŋ]	[ˈɹaɪɾɪ̃ŋ]

- Rule 1 Before Rule 2: Bleeding
- Rule 2 Before Rule 1: Counterbleeding

"Complex" raising rule to account for bleeding map without rule ordering

ˈaɪ → ˈʌɪ / _t *but not* _/tV

- "This rule is more complex than [Rule 2] since it includes an exception stated in the "but not" clause. The inclusion of this clause is motivated solely by the theoretical decision to drop [our] Principle" (Bromberger and Halle, 1989).
- Applying both rules to the input in the counterbleeding order requires no change to the rule descriptions (Joshi and Kiparsky, 1979).

MAIN RESULT

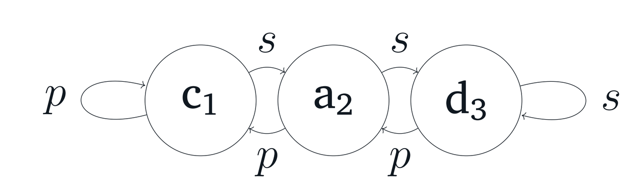
Rules that only apply to the input are not necessarily more complex than ordered rules. This is based on measuring the computational complexity of the function that implements the mapping.

Function Complexity

- Chandlee (2014) shows that an overwhelmingly majority of phonological processes have the property of being *input strictly local* (ISL) and Lambert (2022) confirms that ISL functions belong to one of the most computationally simple function classes.
- **Any phonological process that can be given a logical interpretation using quantifier-free first-order logic (QF) has the property of being ISL (Chandlee and Lindell, 2021).**

Model Theoretic Phonological Structure

- A model theoretic structure is a tuple $\mathcal{M} = \langle \mathcal{D}; \mathcal{R}; \mathcal{F} \rangle$
- $\mathcal{D}$ is a set of domain elements
- $\mathcal{R}$ is a set of predicates/relations
- $\mathcal{F}$ is a set of functions.
- A specific collection of domain elements, relations, and functions is called a model signature.



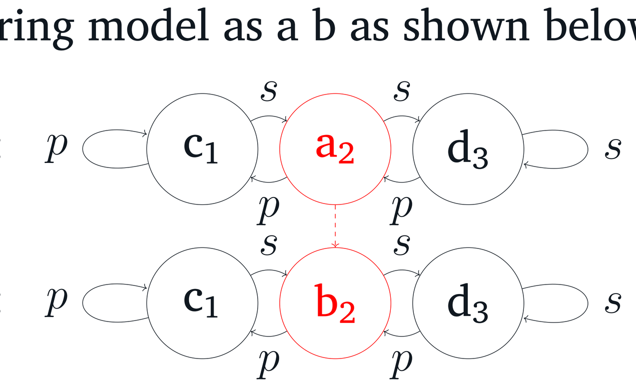
$\langle \mathcal{D} = \{1,2,3\},\ \mathcal{R} = \{\mathtt{a} = \{2\}, \mathtt{b} = \{\}, \mathtt{c} = \{1\}, \mathtt{d} = \{3\}\},\ \mathcal{F} = \{s(1) = 2, s(2) = s(3) = 3,\ p(1) = p(2) = 1, p(3) = 2\} \rangle$

- One model signature than can be used to define phonological strings is $\langle \mathcal{D}, \mathcal{R} = \{\sigma | \sigma \in \Sigma\}, \mathcal{F} = \{s(x), p(x)\} \rangle$. Σ is the alphabet and contains the labels for domain elements of a given string. $s(x)$ and $p(x)$ are the successor and predecessor functions and provide an ordering of the domain elements.
- Specific strings are modeled by explicitly providing a domain and defining the relations and functions that hold between elements of the domain.

Model Theoretic Phonological Transformations

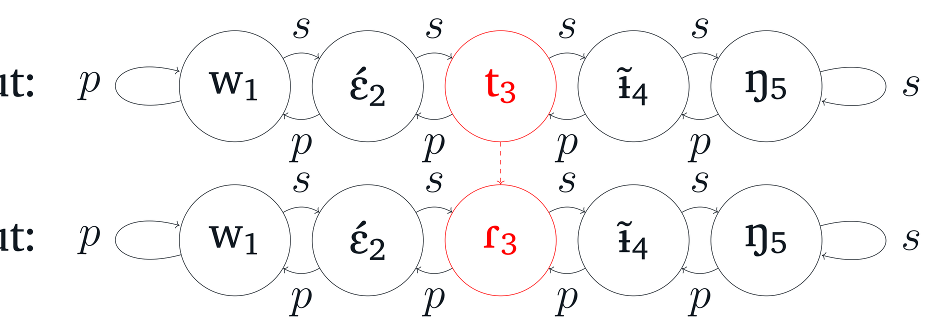
a → b / c_d ⊇ $\phi_{\mathtt{b}}(x) \overset{\text{def}}{=} [\mathtt{a}(x) \wedge \mathtt{c}(p(x)) \wedge \mathtt{d}(s(x))] \vee \mathtt{b}(x)$

- Phonological rules can be turned into logical statements directly by defining output properties in terms of input properties.
- "Domain element x is interpreted as a b on the output if it was an a, preceded by a c, and followed by a d in the input or if it was already a b in the input."
- Since domain element 2 in the string model for *cad* satisfies this formula, it is interpreted in the output string model as a b as shown below.



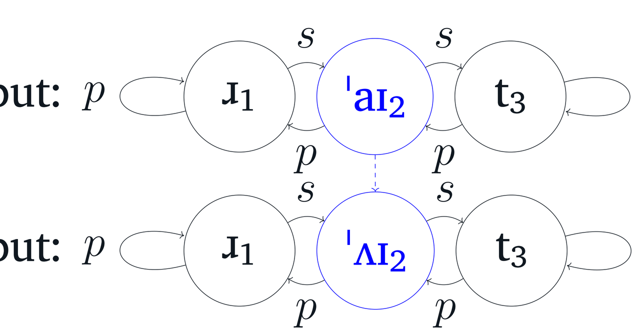
Model Theoretic Intervocalic Tapping

- $\mathrm{T}_1(x) \overset{\text{def}}{=} \acute{\mathrm{V}}(p(x)) \wedge \mathrm{V}(s(x))$
- $\phi_{\mathrm{ɾ}}(x) \overset{\text{def}}{=} ((\mathtt{t}(x) \vee \mathtt{d}(x)) \wedge \mathrm{T}_1(x)) \vee \mathrm{ɾ}(x)$
- $\phi_{\mathtt{d}}(x) \overset{\text{def}}{=} \mathtt{d}(x) \wedge \neg \mathrm{T}_1(x)$
- $\phi_{\mathtt{t}}(x) \overset{\text{def}}{=} \mathtt{t}(x) \wedge \neg \mathrm{T}_1(x)$



Model Theoretic Canadian Raising

- $\mathrm{T}_2(x) \overset{\text{def}}{=} \mathtt{t}(s(x))$
- $\phi_{\mathrm{ʌɪ}}(x) \overset{\text{def}}{=} (\mathrm{ˈaɪ}(x) \wedge \mathrm{T}_2(x)) \vee \mathrm{ˈʌɪ}(x)$
- $\phi_{\mathrm{aɪ}}(x) \overset{\text{def}}{=} \mathrm{ˈaɪ}(x) \wedge \neg \mathrm{T}_2(x)$

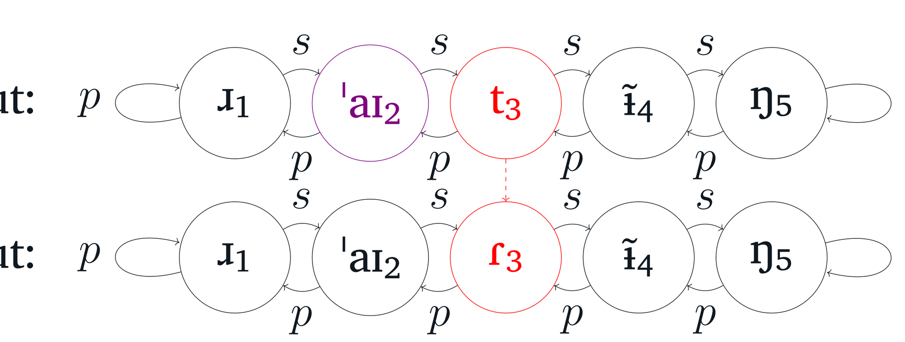


Composition and a "Complex Rule"

- Both processes are ISL since their logical descriptions are quantifier free.
- Their composition is also ISL (Chandlee and Lindell, 2021), which means the bleeding map is ISL and there is no increase in complexity.
- We can interpret the composed logical statement for a bleeding order. It looks exactly like Bromberger and Halle's (1989) "complex" rule.

$\phi_{\mathrm{ʌɪ}}(x) \overset{\text{def}}{=} (\mathrm{ˈaɪ}(x) \wedge \mathtt{t}(s(x)) \wedge \neg(\mathrm{V}(s(s(x))))) \vee \mathrm{ˈʌɪ}(x)$

- "Raise a stressed /aɪ/ when its successor is a /t/ but not if its successor's successor is an unstressed vowel." In other words, raise an /aɪ/ when it's followed by a /t/ unless that /t/ is followed by an unstressed vowel.



- Domain element 2 fails to raise. Domain element 3 taps. **This is based entirely on input properties and matches the bleeding order map.**

Conclusion

- Many factors can increase the computational complexity phonological rules such as whether or not it applies iteratively or over a long distance.
- Adding a local exceptional environment to a phonological rule increases its description, but does not necessarily change the computational complexity of the process.