Modeling phonological processes with recursive program schemes

Chris Oakden, Adam Jardine, and Jane Chandlee

NECPhon
November 16, 2019

Overview

- **Recursive program schemes (RSs)** study structure and complexity of algorithms (Moschovakis, 2019)
- We present **boolean monadic RS (BMRS)** phonological grammars that
  - define a *hierarchy* of local licensing and blocking structures;
  - directly capture *do X unless Y*-type behavior;
  - intensionally express phonologically significant generalizations;
  - are connected to results on computational complexity and learnability (Heinz, 2018);
  - capture both input and output-based mappings, including opacity

Overview

- BMRS provide a glimpse into
  - The *combined map* as a function (available to OT, not to SPE)
  - *Individual* functions which interact (available to SPE, not to OT)
- BMRS offer a framework for describing **operations** (like composition) over individual functions
  - More intuitive than finite-state and logical formalisms

BMRSs: Definition

- An input string is a set of elements $\{1, 2, ..., n\}$
  - ordered by **predecessor function** p, **successor function** s
  - having some **(input) boolean functions** $P(x)$

	$\#$	σ	$\acute{\sigma}$	σ	σ	σ	$\#$
	1	2	3	4	5	6	7
$p(x)$		1	2	3	4	5	6
$s(x)$	2	3	4	5	6	7	
$\#(x)$	$\top$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\top$
$\sigma(x)$	$\bot$	$\top$	$\top$	$\top$	$\top$	$\top$	$\bot$
$\acute{\square}(x)$	$\bot$	$\bot$	$\top$	$\bot$	$\bot$	$\bot$	$\bot$

BMRSs: Definition

- Output string defined by **output boolean functions** $O(x)$

$$\begin{aligned} \#_o(x) &= ? \\ \sigma_o(x) &= ? \\ \acute{\square}_o(x) &= ? \end{aligned}$$

#	σ	σ́	σ	σ	σ	#
1	2	3	4	5	6	7
			↓			
?	?	?	?	?	?	?
1′	2′	3′	4′	5′	6′	7′

(This follows Courcelle 1994; Engelfriet and Hoogeboom 2001)

BMRSs: Definition

- Output string defined by **output boolean functions** $O(x)$

$$
\begin{aligned}
\#_o(x) &= \#(x) \\
\sigma_o(x) &= \sigma(x) \\
\acute{\square}_o(x) &= \acute{\square}(x)
\end{aligned}
$$

#	σ	$\acute{\sigma}$	σ	σ	σ	#
1	2	3	4	5	6	7

$\downarrow$

#	σ	$\acute{\sigma}$	σ	σ	σ	#
$1'$	$2'$	$3'$	$4'$	$5'$	$6'$	$7'$

(This follows Courcelle 1994; Engelfriet and Hoogeboom 2001)

BMRSs: Definition

Logical syntax

- **terms**

$$T \to x \mid p(T) \mid s(T)$$

$$x,\ p(x),\ s(s(x)),\ p(p(p(x))),\ \ldots$$

- **boolean expressions**

$$E \to \top \mid \bot \mid P(T) \mid \texttt{if}\ E\ \texttt{then}\ E\ \texttt{else}\ E$$

BMRSs: Definition

$$T \rightarrow x \mid p(T) \mid s(T)$$
$$E \rightarrow \top \mid \bot \mid P(T) \mid \texttt{if } E \texttt{ then } E \texttt{ else } E$$

$$\text{final}(x) = \texttt{if } \#(s(x)) \texttt{ then } \top \texttt{ else } \bot$$

BMRSs: Definition

$$T \to x \mid p(T) \mid s(T)$$
$$E \to \top \mid \bot \mid P(T) \mid \texttt{if } E \texttt{ then } E \texttt{ else } E$$

$$\text{final}(x) = \texttt{if } \#(s(x)) \texttt{ then } \top \texttt{ else } \bot$$

	#	σ	σ́	σ	σ	σ	#
	1	2	3	4	5	6	7
$\#(x)$	⊤	⊥	⊥	⊥	⊥	⊥	⊤
$\sigma(x)$	⊥	⊤	⊤	⊤	⊤	⊤	⊥
$\acute{\square}(x)$	⊥	⊥	⊤	⊥	⊥	⊥	⊥

BMRSs: Definition

$$T \to x \mid p(T) \mid s(T)$$
$$E \to \top \mid \bot \mid P(T) \mid \texttt{if } E \texttt{ then } E \texttt{ else } E$$

$$\text{final}(x) = \texttt{if } \#(s(x)) \texttt{ then } \top \texttt{ else } \bot$$

	$\#$ 1	σ 2	$\acute{\sigma}$ 3	σ 4	σ 5	σ 6	$\#$ 7
$\#(x)$	$\top$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\top$
$\sigma(x)$	$\bot$	$\top$	$\top$	$\top$	$\top$	$\top$	$\bot$
$\acute{\square}(x)$	$\bot$	$\bot$	$\top$	$\bot$	$\bot$	$\bot$	$\bot$
$\#(s(x))$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$

BMRSs: Definition

$$T \to x \mid p(T) \mid s(T)$$
$$E \to \top \mid \bot \mid P(T) \mid \texttt{if } E \texttt{ then } E \texttt{ else } E$$

$$\text{final}(x) = \texttt{if } \#(s(x)) \texttt{ then } \top \texttt{ else } \bot$$

	$\#$ 1	σ 2	$\acute{\sigma}$ 3	σ 4	σ 5	σ 6	$\#$ 7
$\#(x)$	$\top$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\top$
$\sigma(x)$	$\bot$	$\top$	$\top$	$\top$	$\top$	$\top$	$\bot$
$\acute{\square}(x)$	$\bot$	$\bot$	$\top$	$\bot$	$\bot$	$\bot$	$\bot$
$\#(s(x))$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$
$\text{final}(x)$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$

BMRSs: Definition

- We can define the output boolean functions with a **BMRS system of equations**

$$
\begin{array}{rcl}
O_1(x) & = & E_1 \\
O_2(x) & = & E_2 \\
& \ldots & \\
O_n(x) & = & E_n
\end{array}
$$

BMRSs: Definition

$$
\begin{aligned}
\#_{\mathrm{o}}(x) &= \#(x)\\
\sigma_{\mathrm{o}}(x) &= \sigma(x)\\
\acute{\square}_{\mathrm{o}}(x) &= \texttt{if } \mathrm{final}(x) \;\; \texttt{then } \bot \texttt{ else}\\
& \;\; \texttt{if } \acute{\square}_{\mathrm{o}}(p(x)) \texttt{ then } \top \texttt{ else}\\
& \;\; \acute{\square}(x)
\end{aligned}
$$

BMRSs: Definition

$$
\begin{array}{lcl}
\#_{\mathrm{o}}(x) & = & \#(x) \\
\sigma_{\mathrm{o}}(x) & = & \sigma(x) \\
\acute{\square}_{\mathrm{o}}(x) & = & \texttt{if } \mathrm{final}(x) \texttt{ then } \bot \texttt{ else} \\
 & & \texttt{if } \acute{\square}_{\mathrm{o}}(p(x)) \texttt{ then } \top \texttt{ else} \\
 & & \acute{\square}(x)
\end{array}
$$

	$\#$	σ	$\acute{\sigma}$	σ	σ	σ	$\#$
	1	2	3	4	5	6	7
$\acute{\square}_{\mathrm{o}}(x)$							

BMRSs: Definition

$$
\begin{array}{lcl}
\#_{o}(x) & = & \#(x) \\
\sigma_{o}(x) & = & \sigma(x) \\
\acute{\square}_{o}(x) & = & \texttt{if } \mathrm{final}(x) \texttt{ then } \bot \texttt{ else} \\
 & & \texttt{if } \acute{\square}_{o}(p(x)) \texttt{ then } \top \texttt{ else} \\
 & & \acute{\square}(x)
\end{array}
$$

	$\#$	σ	$\acute{\sigma}$	σ	σ	σ	$\#$
	1	2	3	4	5	6	7
$\acute{\square}_{o}(x)$	$\bot$						

BMRSs: Definition

$$
\begin{array}{rcl}
\#_{\text{o}}(x) & = & \#(x) \\
\sigma_{\text{o}}(x) & = & \sigma(x) \\
\acute{\square}_{\text{o}}(x) & = & \texttt{if } \mathrm{final}(x) \texttt{ then } \bot \texttt{ else} \\
& & \texttt{if } \acute{\square}_{\text{o}}(p(x)) \texttt{ then } \top \texttt{ else} \\
& & \acute{\square}(x)
\end{array}
$$

	$\#$	σ	$\acute{\sigma}$	σ	σ	σ	$\#$
	1	2	3	4	5	6	7
$\acute{\square}_{\text{o}}(x)$	$\bot$	$\bot$					

BMRSs: Definition

$$
\begin{array}{rcl}
\#_o(x) & = & \#(x) \\
\sigma_o(x) & = & \sigma(x) \\
\acute{\square}_o(x) & = & \texttt{if } \mathrm{final}(x) \texttt{ then } \bot \texttt{ else} \\
 & & \texttt{if } \acute{\square}_o(p(x)) \texttt{ then } \top \texttt{ else} \\
 & & \acute{\square}(x)
\end{array}
$$

	#	σ	$\acute{\sigma}$	σ	σ	σ	#
	1	2	3	4	5	6	7
$\acute{\square}_o(x)$	$\bot$	$\bot$	$\top$				

BMRSs: Definition

$$\begin{array}{rcl} \#_o(x) & = & \#(x) \\ \sigma_o(x) & = & \sigma(x) \\ \acute{\Box}_o(x) & = & \texttt{if } \mathrm{final}(x) \texttt{ then } \bot \texttt{ else} \\ & & \texttt{if } \acute{\Box}_o(p(x)) \texttt{ then } \top \texttt{ else} \\ & & \acute{\Box}(x) \end{array}$$

	$\#$	σ	$\acute{\sigma}$	σ	σ	σ	$\#$
	1	2	3	4	5	6	7
$\acute{\Box}_o(x)$	$\bot$	$\bot$	$\top$	$\top$			

BMRSs: Definition

$$
\begin{aligned}
\#_{\mathrm{o}}(x) &= \#(x) \\
\sigma_{\mathrm{o}}(x) &= \sigma(x) \\
\acute{\square}_{\mathrm{o}}(x) &= \texttt{if } \mathrm{final}(x) \texttt{ then } \bot \texttt{ else} \\
&\quad\ \texttt{if } \acute{\square}_{\mathrm{o}}(p(x)) \texttt{ then } \top \texttt{ else} \\
&\quad\ \acute{\square}(x)
\end{aligned}
$$

	$\#$	σ	$\acute{\sigma}$	σ	σ	σ	$\#$
	1	2	3	4	5	6	7
$\acute{\square}_{\mathrm{o}}(x)$	$\bot$	$\bot$	$\top$	$\top$	$\top$		

BMRSs: Definition

$$
\begin{array}{rcl}
\#_o(x) & = & \#(x) \\
\sigma_o(x) & = & \sigma(x) \\
\acute{\Box}_o(x) & = & \texttt{if } \mathrm{final}(x) \ \texttt{ then } \bot \texttt{ else} \\
 & & \texttt{if } \acute{\Box}_o(p(x)) \texttt{ then } \top \texttt{ else} \\
 & & \acute{\Box}(x)
\end{array}
$$

	$\#$	σ	$\acute{\sigma}$	σ	σ	σ	$\#$
	1	2	3	4	5	6	7
$\acute{\Box}_o(x)$	$\bot$	$\bot$	$\top$	$\top$	$\top$	$\bot$	

BMRSs: Definition

$$
\begin{array}{rcl}
\#_o(x) & = & \#(x) \\
\sigma_o(x) & = & \sigma(x) \\
\acute{\square}_o(x) & = & \texttt{if } \mathrm{final}(x) \texttt{ then } \bot \texttt{ else} \\
& & \texttt{if } \acute{\square}_o(p(x)) \texttt{ then } \top \texttt{ else} \\
& & \acute{\square}(x)
\end{array}
$$

	$\#$	σ	$\acute{\sigma}$	σ	σ	σ	$\#$
	1	2	3	4	5	6	7
$\acute{\square}_o(x)$	$\bot$	$\bot$	$\top$	$\top$	$\top$	$\bot$	$\bot$

BMRSs: Definition

$$
\begin{array}{rcl}
\#_{\mathrm{o}}(x) & = & \#(x) \\
\sigma_{\mathrm{o}}(x) & = & \sigma(x) \\
\acute{\square}_{\mathrm{o}}(x) & = & \texttt{if } \mathrm{final}(x) \texttt{ then } \bot \texttt{ else} \\
 & & \texttt{if } \acute{\square}_{\mathrm{o}}(p(x)) \texttt{ then } \top \texttt{ else} \\
 & & \acute{\square}(x)
\end{array}
$$

	$\#$	σ	$\acute{\sigma}$	σ	σ	σ	$\#$
	1	2	3	4	5	6	7
$\#_{\mathrm{o}}(x)$	$\top$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\top$
$\sigma_{\mathrm{o}}(x)$	$\bot$	$\top$	$\top$	$\top$	$\top$	$\top$	$\bot$
$\acute{\square}_{\mathrm{o}}(x)$	$\bot$	$\bot$	$\top$	$\top$	$\top$	$\bot$	$\bot$
	$1'$	$2'$	$3'$	$4'$	$5'$	$6'$	$7'$
	$\#$	σ	$\acute{\sigma}$	$\acute{\sigma}$	$\acute{\sigma}$	σ	$\#$

BMRSs: Definition

- BMRS systems of equations always have a *least-fixed point solution* (Moschovakis, 2019)
- If restricted to recursing on only $p(x)$ or $s(x)$ (but not both), BMRSs describe *subsequential functions* (Bhaskar et al., ms)
- The syntax expresses a **hierarchy** of **blocking structures** and **licensing structures**

$$\begin{aligned}\acute{\square}_{o}(x) \;=\;& \texttt{if}\ \mathrm{final}(x)\ \texttt{then}\ \bot\ \texttt{else}\\ & \texttt{if}\ \acute{\square}_{o}(p(x))\ \texttt{then}\ \top\ \texttt{else}\\ & \acute{\square}(x)\end{aligned}$$

BMRSs: Input/Output-based mappings

- Input-based: output boolean functions defined **without recursion**
  - Compute output by reference to input structure only
  - **ISL** class of functions

BMRSs: Input/Output-based mappings

- Input-based: output boolean functions defined **without recursion**
  - Compute output by reference to input structure only
  - **ISL** class of functions
- Tianjin tone sandhi ‘RR’ rule (Chen, 1986; Chandlee, 2019)
  - Inventory: H(igh), R(ising), L(ow), F(alling)
  - RR $\rightarrow$ HR (simultaneous, ISL); RRR $\rightarrow$ HHR

BMRSs: Input/Output-based mappings

- Input-based: output boolean functions defined **without recursion**
  - Compute output by reference to input structure only
  - **ISL** class of functions
- Tianjin tone sandhi 'RR' rule (Chen, 1986; Chandlee, 2019)
  - Inventory: H(igh), R(ising), L(ow), F(alling)
  - RR $\rightarrow$ HR (simultaneous, ISL); RRR $\rightarrow$ HHR

$$
\begin{aligned}
H_{\mathrm{o}}(x) &= \texttt{if } \underline{R}R(x) \texttt{ then } \top \texttt{ else } H(x) \\
R_{\mathrm{o}}(x) &= \texttt{if } \underline{R}R(x) \texttt{ then } \bot \texttt{ else } R(x) \\
L_{\mathrm{o}}(x) &= L(x) \\
F_{\mathrm{o}}(x) &= F(x)
\end{aligned}
$$

BMRSs: Input/Output-based mappings

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

	$\#$	R	R	R	R	$\#$
	1	2	3	4	5	6
$H_o(x)$	$\bot$	$\top$	$\top$	$\top$	$\bot$	$\bot$
$R_o(x)$	$\bot$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$
	$1'$	$2'$	$3'$	$4'$	$5'$	$6'$
	$\#$	H	H	H	R	$\#$

BMRSs: Input/Output-based mappings

- Output-based: output boolean functions **require recursion**
  - Refer to **current** input, otherwise to output structure only
  - **OSL** class of functions

BMRSs: Input/Output-based mappings

- Output-based: output boolean functions **require recursion**
  - Refer to **current** input, otherwise to output structure only
  - **OSL** class of functions
- Tianjin tone sandhi 'LL' rule (Chen, 1986; Chandlee, 2019)
  - LL $\rightarrow$ RL (iterative, ROSL)
  - LLL $\rightarrow$ LRL, LLLL $\rightarrow$ RLRL

BMRSs: Input/Output-based mappings

- Output-based: output boolean functions **require recursion**
  - Refer to **current** input, otherwise to output structure only
  - **OSL** class of functions
- Tianjin tone sandhi 'LL' rule (Chen, 1986; Chandlee, 2019)
  - LL $\rightarrow$ RL (iterative, ROSL)
  - LLL $\rightarrow$ LRL, LLLL $\rightarrow$ RLRL

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

BMRSs: Input/Output-based mappings

$$\begin{array}{rcl} R_o(x) & = & \texttt{if } \underline{L}L_oR_o(x) \texttt{ then } \top \texttt{ else} \\ & & \texttt{if } \underline{L}L_o(x) \texttt{ then } \top \texttt{ else} \\ & & R(x) \\ L_o(x) & = & \texttt{if } R_o(x) \texttt{ then } \bot \texttt{ else } L(x) \end{array}$$

	#	L	L	L	L	#
	1	2	3	4	5	6
$R_o(x)$	$\bot$	$\top$	$\bot$	$\top$	$\bot$	$\bot$
$L_o(x)$	$\bot$	$\bot$	$\top$	$\bot$	$\top$	$\bot$
	$1'$	$2'$	$3'$	$4'$	$5'$	$6'$
	#	R	L	R	L	#

BMRSs: Function Composition

- BMRS offers intuitive framework for **function composition**
- Given two BMRS systems of equations a and b, $b \circ a$ is defined:
  - In system b, all non-recursively-defined boolean function names refer to *corresponding* definitions in system a

BMRSs: Function Composition

- BMRS offers intuitive framework for **function composition**
- Given two BMRS systems of equations a and b, $b \circ a$ is defined:
  - In system b, all non-recursively-defined boolean function names refer to *corresponding* definitions in system a
- Applications in phonological process **interactions**
  - Tianjin LL (LL $\rightarrow$ RL) rule **feeds** RR (RR $\rightarrow$ HR) rule
  - R<u>LL</u> $\rightarrow$ <u>R**R**</u>L $\rightarrow$ **H**RL
  - 'Combined map' (Chandlee, 2019)
  - **Compose** two BMRS systems
    - System a: LL rule
    - System b: RR rule
  - Can do both easily in BMRS formalism

BMRSs: Function Composition

BMRSs: Function Composition

$$
\begin{array}{rcl}
 & a & \\
R_a(x) & = & \text{if } \underline{L}L_aR_a(x) \text{ then } \top \text{ else} \\
 & & \text{if } \underline{L}L_a(x) \text{ then } \top \text{ else} \\
 & & R(x) \\
L_a(x) & = & \text{if } R_a(x) \text{ then } \bot \text{ else } L(x) \\
H_a(x) & = & H(x) \\
F_a(x) & = & F(x)
\end{array}
$$

BMRSs: Function Composition

$$\begin{array}{rcl} & a & \\ R_a(x) & = & \text{if } \underline{L}L_aR_a(x) \text{ then } \top \text{ else} \\ & & \text{if } \underline{L}L_a(x) \text{ then } \top \text{ else} \\ & & R(x) \\ L_a(x) & = & \text{if } R_a(x) \text{ then } \perp \text{ else } L(x) \\ H_a(x) & = & H(x) \\ F_a(x) & = & F(x) \end{array}$$

$$\begin{array}{rcl} & b & \\ H_b(x) & = & \text{if } \underline{R}R(x)\text{, then } \top \text{ else H(x)} \\ R_b(x) & = & \text{if } \underline{R}R(x)\text{, then } \perp \text{ else R(x)} \\ L_b(x) & = & L(x) \\ F_b(x) & = & F(x) \end{array}$$

BMRSs: Function Composition

$$
\begin{array}{rcl}
 & a & \\
R_a(x) & = & \text{if } \underline{L}L_aR_a(x) \text{ then } \top \text{ else} \\
 & & \text{if } \underline{L}L_a(x) \text{ then } \top \text{ else} \\
 & & R(x) \\
L_a(x) & = & \text{if } R_a(x) \text{ then } \bot \text{ else } L(x) \\
H_a(x) & = & H(x) \\
F_a(x) & = & F(x)
\end{array}
\qquad
\begin{array}{rcl}
 & b & \\
H_b(x) & = & \text{if } \underline{R}R(x)\text{, then } \top \text{ else H(x)} \\
R_b(x) & = & \text{if } \underline{R}R(x)\text{, then } \bot \text{ else R(x)} \\
L_b(x) & = & L(x) \\
F_b(x) & = & F(x)
\end{array}
$$

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

BMRSs: Function Composition

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

	#	R	L	L	#
$R_b(x)$		$\bot$	$\top$	$\bot$	
$R_a(x)$		$\top$	$\top$	$\bot$	
$H_b(x)$		$\top$	$\bot$	$\bot$	
	#	H	R	L	#

Discussion

- BMRS provide a glimpse into
  - The *combined map* as a function (available to OT, not to SPE)
  - *Individual* functions which interact (available to SPE, not to OT)
  - *Operations* over individual functions (less intuitive in finite-state and logical formalisms)

Discussion

- BMRS provide a glimpse into
  - The *combined map* as a function (available to OT, not to SPE)
  - *Individual* functions which interact (available to SPE, not to OT)
  - *Operations* over individual functions (less intuitive in finite-state and logical formalisms)
    - Further refine definitions
    - Identify *new* operations beside composition

Discussion

- BMRS provide a glimpse into
  - The *combined map* as a function (available to OT, not to SPE)
  - *Individual* functions which interact (available to SPE, not to OT)
  - *Operations* over individual functions (less intuitive in finite-state and logical formalisms)
    - Further refine definitions
    - Identify *new* operations beside composition
- The linchpin: if-then-else syntax
  - Capture *do X unless Y*-type behavior (as in OT)
  - Input- *and* output-orientedness

Discussion

- BMRS provide a glimpse into
  - The *combined map* as a function (available to OT, not to SPE)
  - *Individual* functions which interact (available to SPE, not to OT)
  - *Operations* over individual functions (less intuitive in finite-state and logical formalisms)
    - Further refine definitions
    - Identify *new* operations beside composition
- The linchpin: if-then-else syntax
  - Capture *do X unless Y*-type behavior (as in OT)
  - Input- *and* output-orientedness
    - Hierarchy of licensing and blocking structures
    - Elsewhere condition

Conclusion

- Express phonologically significant generalizations with BMRS
- Equivalent to subsequential class of functions
- Unique syntax defines hierarchy of local licensing and blocking structures
- Capture input and output-based mappings
- Intuitive framework for examining phonological process interaction

Thank You

We also thank Siddharth Bhaskar (University of Copenhagen) and the members of the Rutgers mathematical linguistics reading group.

References

Bhaskar, S., Chandlee, J., Jardine, A., and Oakden, C. (ms). Boolean monadic recursive schemes as a logical characterization of the subsequential functions. Ms. submitted to *LATA 2020*.

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.

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

Chen, M. Y. (2004). Changting Hakka tone sandhi: Analytical challenges. *Language and Linguistics*, 5(4):799–820.

Courcelle, B. (1994). Monadic second-order definable graph transductions: a survey. *Theoretical Computer Science*, 126:53–75.

Engelfriet, J. and Hoogeboom, H. J. (2001). MSO definable string transductions and two-way finite-state transducers. *ACM Transations on Computational Logic*, 2:216–254.

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.

Kaplan, R. M. (1987). Three seductions of computational psycholinguistics. In Whitelock, P., Wood, M. M., Somers, H. L., Johnson, R., and Bennett, P., editors, *Linguistic Theory and Computer Applications*, pages 149–188. Academic Press.

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.

Moschovakis, Y. N. (2019). *Abstract recursion and intrinsic complexity*, volume 48 of *Lecture Notes in Logic*. Cambridge University Press.

Oakden, C. and Chandlee, J. (2019). A computational analysis of tone sandhi ordering paradoxes. Poster presented at NELS 50, MIT.

Appendix 1: Other Operations

Appendix 1: Other Operations

- Identify other operations besides composition
- Interactions for which composition (rule-ordering) yields incorrect mapping

Appendix 1: Other Operations

- Identify other operations besides composition
- Interactions for which composition (rule-ordering) yields incorrect mapping
- Changting sandhi ordering paradoxes (Chen, 2004)
  - Inventory: H(igh), M(id), L(ow), R(ising), F(alling)
  - MR rule (MR $\rightarrow$ LR), RM rule (RM $\rightarrow$ HM)
  - Mutual counterbleeding: /MRM/ $\rightarrow$ LHM, /RMR/ $\rightarrow$ [HLR]

Appendix 1: Other Operations

- Identify other operations besides composition
- Interactions for which composition (rule-ordering) yields incorrect mapping
- Changting sandhi ordering paradoxes (Chen, 2004)
  - Inventory: H(igh), M(id), L(ow), R(ising), F(alling)
  - MR rule (MR $\rightarrow$ LR), RM rule (RM $\rightarrow$ HM)
  - Mutual counterbleeding: /MRM/ $\rightarrow$ LHM, /RMR/ $\rightarrow$ [HLR]

MR<RM		**RM<MR**	
$\underline{\text{MR}}\text{M}$	$\text{R}\underline{\text{MR}}$	$\text{M}\underline{\text{RM}}$	$\underline{\text{RM}}\text{R}$
|	|	|	|
$\text{L}\underline{\text{RM}}$	RLR	MHM	$\text{H}\underline{\text{MR}}$
|	|	|	|
LHM	*RLR	*MHM	HLR

Appendix 1: Other Operations

- Interaction is **ISL** (Oakden and Chandlee, 2019)

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

Appendix 1: Other Operations

- Interaction is **ISL** (Oakden and Chandlee, 2019)

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

- **Not** a result of *composing* two systems MR and RM
- Composition recreates ordering paradox

Appendix 1: Other Operations

- New operation '$\ominus$'
- Given two BMRS systems of equations a and b, $b \ominus a$ is defined:
  - Identity-map definitions in b are replaced with corresponding *non-identity* definitions in a
  - Otherwise, leave the definition the same

  - Is $\ominus$ just *priority union*? (Kaplan, 1987; Karttunen, 1998)

Appendix 1: Other Operations

- New operation '$\ominus$'
- Given two BMRS systems of equations a and b, $b \ominus a$ is defined:
  - Identity-map definitions in b are replaced with corresponding *non-identity* definitions in a
  - Otherwise, leave the definition the same
- Changting mutual counterbleeding
  - System a: RM rule
  - System b: MR rule

  - Is $\ominus$ just *priority union*? (Kaplan, 1987; Karttunen, 1998)

Appendix 1: Other Operations

- New operation '$\ominus$'
- Given two BMRS systems of equations a and b, $b \ominus a$ is defined:
  - Identity-map definitions in b are replaced with corresponding *non-identity* definitions in a
  - Otherwise, leave the definition the same
- Changting mutual counterbleeding
  - System a: RM rule
  - System b: MR rule
- Corresponds to **simultaneous application**
  - Is $\ominus$ just *priority union*? (Kaplan, 1987; Karttunen, 1998)

Appendix 1: Other Operations

Appendix 1: Other Operations

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

Appendix 1: Other Operations

$$
\begin{array}{rclrcl}
 & a & & & b & \\
L_a(x) & = & L(x) & L_b(x) & = & \text{if } \underline{M}R(x) \text{ then } \top \text{ else } L(x) \\
M_a(x) & = & M(x) & M_b(x) & = & \text{if } \underline{M}R(x) \text{ then } \bot \text{ else } M(x) \\
H_a(x) & = & \text{if } \underline{R}M(x) \text{ then } \top \text{ else } H(x) & H_b(x) & = & H(x) \\
R_a(x) & = & \text{if } \underline{R}M(x) \text{ then } \bot \text{ else } R(x) & R_b(x) & = & R(x) \\
F_a(x) & = & F(x) & F_b(x) & = & F(x)
\end{array}
$$

Appendix 1: Other Operations

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

$$\begin{array}{rcl} & b & \\ L_b(x) & = & \text{if } \underline{M}R(x) \text{ then } \top \text{ else } L(x) \\ M_b(x) & = & \text{if } \underline{M}R(x) \text{ then } \bot \text{ else } M(x) \\ H_b(x) & = & H(x) \\ R_b(x) & = & R(x) \\ F_b(x) & = & F(x) \end{array}$$

$$\begin{array}{rcl} & b \ominus a & \\ L_b(x) & = & \texttt{if } \underline{M}R(x) \texttt{ then } \top \texttt{ else } L(x) \\ M_b(x) & = & \texttt{if } \underline{M}R(x) \texttt{ then } \bot \texttt{ else } M(x) \\ H_b(x) & = & H_a(x) \\ R_b(x) & = & R_a(x) \\ F_b(x) & = & F(x) \end{array}$$