Recursive Schemes for Phonological Analysis

Jane Chandlee | Adam Jardine
Haverford College | Rutgers University

SIGMORPHON 2020

10 July 2020
Central goal: identify abstract computational universals of phonological patterns.

- e.g., phonological rules are *regular relations* (Johnson, 1972; Kaplan and Kay, 1981, 1994)
  - i.e., can be computed for any string using a fixed amount of memory
- More recent work has posited further constraints on this memory
  - e.g., phonological patterns are *subregular* (Heinz, 2009; Heinz and Lai, 2013; Chandlee and Heinz, 2018, a.o.)

A frequent criticism of this approach is a disconnect with the intensional representations of generalizations that phonologists use and recognize.
Introduction

- We present a computational formalism that retains computational restrictiveness and better aligns with commonly-held assumptions about phonological representations and grammars.

- Specifically, we introduce *Boolean Monadic Recursive Schemes (BMRSs)* and demonstrate their utility for phonological analysis.

- Advantages:
  1. well-understood complexity bound
  2. means of implementing phonological substance
Graph transduction

- We represent strings as graphs and processes as maps from an input to an output graph.

(1) German /hʊnd/ $\mapsto$ [hʊnt], ‘dog’

Input:

Output:
Graph transduction

- We represent strings as graphs and processes as maps from an input to an output graph.

\[(2) \quad \text{German } /\text{hünd}/ \leftrightarrow [\text{hunt}], \text{‘dog’} \]

Input:

Output:
Output predicates

\[ \text{voice}_o(x) = \ldots \text{ (under what conditions is an output segment specified as +voice?)} \]
Boolean Monadic Recursive Schemes

\[ \text{voice}_o(x) = \text{if } X \text{ then } B \text{ else } Y \]
Output predicates

\[
\text{voice}_o(x) = \begin{cases} 
\top & \text{if sonorant}(x) \\
\bot & \text{if } \#(s(x)) \\
\text{voice}(x) & \text{else}
\end{cases}
\]

- \( \top = \text{True} \)
- \( \bot = \text{False} \)
Output predicates

Input:

\[
\begin{array}{cccccc}
\# & h & v & n & d & \#\\
1 & 2 & 3 & 4 & 5 & 6
\end{array}
\]

Output:

\[
\begin{array}{ccccccc}
1' & 2' & 3' & 4' & 5' & 6'
\end{array}
\]

\[\text{voice}_o(x) = \begin{cases} 
\top & \text{if sonorant}(x) \\
\bot & \text{if } \#(s(x)) \text{ then else} \\
\text{voice}(x) & \end{cases}\]
Output predicates

\[
\begin{align*}
\text{Input:} & \quad \# \quad h \quad u \quad n \quad d \quad \#
\end{align*}
\]
\[
1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6
\]

\[
\begin{align*}
\text{Output:} & \quad u \quad n
\end{align*}
\]
\[
1' \quad 2' \quad 3' \quad 4' \quad 5' \quad 6'
\]

\[
\begin{align*}
\text{voice}_o(x) &= \text{if sonorant}(x) \text{ then } \top \text{ else } \\
&\quad \text{if } \#(s(x)) \text{ then } \bot \text{ else } \\
&\quad \text{voice}(x)
\end{align*}
\]
Output predicates

Input:  \# h u n d \#  
\begin{align*} 
1 & 2 & 3 & 4 & 5 & 6 
\end{align*}

Output: u n t  
\begin{align*} 
1' & 2' & 3' & 4' & 5' & 6' 
\end{align*}

\[
\text{voice}_o(x) = \begin{cases} 
\top & \text{if sonorant}(x) \text{ then } \\
\bot & \text{if } \#(s(x)) \text{ then } \\
\text{voice}(x) & \text{else} 
\end{cases}
\]
Output predicates

Input:  
# h u n d #
1 2 3 4 5 6

Output:  
  h u n t
1' 2' 3' 4' 5' 6'

\[ \text{voice}_o(x) = \begin{cases} \top & \text{if \ sonorant}(x) \text{ then} \\ \bot & \text{if \ #(s(x)) \ then} \\ \text{voice}(x) & \text{else} \end{cases} \]
Advantages

- Similar to OT, BMRSs employ a hierarchy of constraint-like predicates that refer to particular structures in the input/output.
- These structures may alternately license or block particular feature values from surfacing.
  - In previous example, #(s(x)) is a blocking structure for voicing...
  - while sonorant(x) licenses voicing.

\[
\text{voice}_o(x) = \begin{cases} 
\top & \text{if sonorant}(x) \text{ then } \\
\bot & \text{if } \#(s(x)) \text{ then } \\
\text{voice}(x) & \text{else}
\end{cases}
\]
Advantages

- In contrast to OT, the evaluation of a BMRS hierarchy is necessarily *local* in nature.
  - Avoids the computational over-generation of global evaluation (Frank and Satta, 1998; Gerdemann and Hulden, 2012; Lamont, 2018).
- BMRSs are guaranteed to describe a strict subclass of the regular functions (Bhaskar et al., 2020).
Demonstration 1: Elsewhere condition effects

- Stressed vowels in binary feet are long when followed by /iV/ and short elsewhere (Chomsky and Halle, 1968; Myers, 1987; Halle, 1995; Baković, 2006)

(3)  
  a. re(mēdi)al *re(mědi)al  
  b. co(lōni)al *co(lōni)al  
  c. (rādi)al *(rǎdi)al

(4)  
  a. (nātu)ral (cf. nāture)  
  b. di(vīni)ty (cf. divīne)  
  c. (rādi)cal
Demonstration 1: Elsewhere condition effects

\[ \text{HEAD}(x) \]

\[ \text{NON HIGH} \text{CiV}(x) = \begin{cases} \text{if } \text{high}(x) \text{ then } \bot \text{ else } \\
\text{if } C(s(x)) \text{ then } \\
\text{if } i(s(s(x))) \text{ then } V(s(s(s(x)))) \\
\text{else } \bot \end{cases} \]

\[ \text{HEAD} \& \text{NON HIGH} \text{CiV}(x) = \begin{cases} \text{if } \text{HEAD}(x) \text{ then } \text{NON HIGH} \text{CiV}(x) \text{ else } \bot \end{cases} \]
Demonstration 1: Elsewhere condition effects

\[ \text{long}_o(x) = \begin{cases} \top & \text{if HEAD\&\text{\underline{NONHIGH}}CiV(x) then} \\ \bot & \text{else if HEAD}(x) \text{ then} \\ \text{long}(x) & \text{else} \end{cases} \]
Demonstration 1: Elsewhere condition effects

$$\text{long}_o(x) = \begin{cases} \text{if } \text{HEAD} \& \text{NONHIGH} \text{CiV}(x) \text{ then } \top \text{ else } \\ \text{if } \text{HEAD}(x) \text{ then } \bot \text{ else } \\ \text{long}(x) \end{cases}$$

- Note that $\text{HEAD} \& \text{NONHIGH} \text{CiV}(x)$ implies $\text{HEAD}(x)$.
- If $\text{HEAD}(x)$ appeared higher in the hierarchy than $\text{HEAD} \& \text{NONHIGH} \text{CiV}(x)$, the latter would never be evaluated (and would therefore be extraneous).
Strict Substructure Ordering Principle (SSOP)

(5) If structure $j$ is a strict substructure of structure $i$ (i.e., structure $i$ implies structure $j$ but not vice versa), then $i$ must precede $j$ in the hierarchy in order to be evaluated.
Strict Substructure Ordering Principle (SSOP)

(5) If structure $j$ is a strict substructure of structure $i$ (i.e., structure $i$ implies structure $j$ but not vice versa), then $i$ must precede $j$ in the hierarchy in order to be evaluated.

- Captures the disjunctive nature of shortening and lengthening.
- This principle is inherent to the logic of how BMRSs are evaluated and does not need to be stipulated.
Demonstration 2: the typology of *NČ

(6) *NČ
No nasal/voiceless obstruent sequences (Pater, 2004)

- Various repair strategies are accounted for in OT with permutations of faithfulness constraints and *NČ.
- With BMRS, this variation is captured with the placement of blocking and licensing structures.
Demonstration 2: the typology of $*\text{NC}$

\[
\text{NC}(x) = \begin{cases} 
\text{nasal}(x) & \text{then } \text{VOICELESSOBS}(s(x)) \text{ else } \bot \\
\end{cases}
\]

\[
\text{NC}(x) = \begin{cases} 
\text{VOICELESSOBS}(x) & \text{then } \text{nasal}(p(x)) \text{ else } \bot \\
\end{cases}
\]
Demonstration 2: the typology of *NÇ

\[ \text{Obstruent}(x) = \text{if } \text{sonorant}(x) \text{ then } \bot \text{ else } \top \]

\[ \text{VoicelessObs}(x) = \text{if } \text{voice}(x) \text{ then } \bot \text{ else } \text{Obstruent}(x) \]
Demonstration 2: the typology of *NC

(7) Indonesian fusion (Onn, 1980)
/mənpilih/ \rightarrow [məmilih], ‘to choose, vote’

(8) \text{out}(x) = \text{if } \text{NC}(x) \text{ then } \bot \text{ else } \top

\text{labial}_o(x) = \text{if } \text{NasalLabial}(x) \text{ then } \top \text{ else } \text{labial}(x)

\text{NC} \quad \text{blocks out}
Demonstration 2: the typology of *NC

(9) Puyo Pungo Quechua post-nasal voicing (Orr, 1962)
/kampa/ $\mapsto$ [kamba], ‘yours’

(10) $\text{voice}_o(x) = \begin{cases} \top & \text{if } \text{NC}(x) \text{ then } \top \text{ else } \\ \text{voice} (x) \end{cases}$

\[ \text{NC} \text{ licenses voice} \]
Demonstration 2: the typology of *NČ

(11) Nasalization in Konjo (Austronesian; Sulawesi; Friberg and Friberg, 1991)

/aŋpinahąc/ \rightarrow [œmminahąc] ‘to follow’
/aŋkanre/ \rightarrow [œ sınıf] ‘to eat’

(12) nasal\_o(x) = \text{if } NČ(x) \text{ then } \top \text{ else } nasal(x)
voice\_o(x) = \text{if } nasal\_o(x) \text{ then } \top \text{ else } voice(x)

\underline{NČ} \text{ licenses nasal, and nasal licenses voice}
Demonstration 2: the typology of *NC

(13) Mandar gemination (Austronesian; Sulawesi; Mills, 1975)
/mantunu/ ➔ [mattunu], ‘to burn’

(14) \( \text{nasal}_o(x) = \text{if } \text{NC}(x) \text{ then } \bot \text{ else } \text{nasal}(x) \)

\( \text{NC} \) blocks nasal
Demonstration 2: the typology of *NC

<table>
<thead>
<tr>
<th>NÇ blocks out</th>
<th>fusion</th>
</tr>
</thead>
<tbody>
<tr>
<td>NÇ licenses [voice]</td>
<td>post-nasal voicing</td>
</tr>
<tr>
<td>NÇ licenses [nasal]</td>
<td>nasalization</td>
</tr>
<tr>
<td>NÇ blocks [nasal]</td>
<td>gemination</td>
</tr>
</tbody>
</table>
## Demonstration 2: the typology of *N\_C\_\_\_C

<table>
<thead>
<tr>
<th>Licensing</th>
<th>Blocking</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>out</strong></td>
<td><strong>Swahili</strong></td>
</tr>
<tr>
<td>[nasal] \textsubscript{o}</td>
<td><strong>Mandar</strong></td>
</tr>
<tr>
<td>[voice] \textsubscript{o}</td>
<td>attested?</td>
</tr>
</tbody>
</table>
Meaningful structures

(15) Meaningful Structures in BMRSs

$\text{if } X \text{ then } B \text{ else } Y$

$X$ is meaningful iff **neither** of the following holds.

$B$ is $\top$ and $X$ implies $Y$,

or $B$ is $\bot$ and $X$ implies $\neg Y$. 
(16) Meaningful Structures in BMRSs

A formula of the form

\[
\text{if } X \text{ then } B \text{ else } Y
\]

is meaningful iff **neither** of the following holds.

- \( B \) is \( \top \) and \( X \) implies \( Y \),
- or \( B \) is \( \bot \) and \( X \) implies \( \neg Y \).

E.g., \( \text{nasal}_o(x) = \text{if } \text{NC}_o(x) \text{ then } \top \text{ else nasal}(x) \)

- By definition (16), this formula is *meaningless*. 
Open questions

- What kinds of operations are needed to combine multiple BMRSs into a complete phonological grammar?
  - Composition?
  - Priority union?
  - Something else?
Open questions

How can recursion be used to capture long-distance processes?

(17) Kikongo (Ao, 1991; Odden, 1994)
/ku-kin-ila/ [ku-kin-ina] ‘to dance for’
/ku-dumuk-ila/ [ku-dumuk-ina] ‘to jump for’
/ku-dumuk-is-ila/ [ku-dumuk-is-ina] ‘to make jump for’

(18) follows-nas(x) = if #(x) then ⊥ else
if nas(p(x)) then T else
follows-nas(p(x))
Open questions

- Consequences of using non-linear representations?
  - Recall $\text{HEAD}(x)$
Open questions

- Extension of prior finite-state-based learnability results for subregular functions.
  - How can we learn a BMRS template from positive data?
Conclusion

- BMRS capture phonological generalizations in a local way, maintaining the desirable computational restrictiveness of subregularity in a way that is more intuitive from a phonological perspective.
- Unlike prior finite-state characterizations of subregular functions, the restrictiveness of BMRSs doesn’t depend on determinism.
  - Enables the use of feature-based representations.
  - Captures relationship among multiple generalizations more transparently.


References II


References III


