Joint work

Adam Jardine, Rutgers University

Central goal: identify abstract computational universals of phonological patterns.

SPE (Chomsky and Halle, 1968) rules describe regular relations (Johnson, 1972; Kaplan and Kay, 1981, 1994), provided they do not reapply to their own structural change.

(1) \( A \rightarrow B / C \_\_ D \)

Such rules can be represented with finite-state transducers (FSTs).

A series of ordered rules can be represented with the composition of FSTs.
A stronger version

- Phonological patterns are *subregular* (Heinz, 2009; Heinz and Lai, 2013; Chandlee and Heinz, 2018, a.o.), or describable with proper subsets of regular.

- Advantages:
  - Better fit to known typology.
  - Enables learning from positive data.
Starting point

- For now, subregular = subsequential
  - Deterministic on the input
  - Closed under composition
  - Learnable from positive data (Oncina et al., 1993)
- Mohri (1997, p. 11): ‘Most phonological and morphological rules correspond to \( p \)-subsequential functions.’
- Supporting evidence found for a range of segmental processes (Heinz and Lai, 2013; Chandlee, 2014; Luo, 2017; Payne, 2017), though see McCollum et al. (2020) for an exception from vowel harmony.
Subsequential FST

(2) Post-nasal obstruent voicing

a. \([-\text{son}] \rightarrow [+\text{voice}] / [+\text{nasal}]\)
   
   b. ampa \rightarrow amba, anta \rightarrow anda, \text{aŋka} \rightarrow \text{anga}
Limitations of FST formalism

- A frequent criticism of this approach is a disconnect with the more familiar intensional representations of phonological generalizations.
- In particular:
  - the use of strings
  - the use of unanalyzed/atomic segments
  - the lack of explanation for why an alternation is occurring
Limitations of FST formalism

![Diagram of a finite-state transducer]

- States: 0, 1
- Transitions: 
  - 0 → 0 with input a,p:t,k
  - 0 → 1 with input n,m,ŋ
  - 1 → 1 with input n,m,ŋ
  - 1 → 0 with input a,p:b,t:d,k:g

- Initial state: 0
- Final state: 1
Optimality Theory

<table>
<thead>
<tr>
<th>/ampa/</th>
<th>*[+nasal][−voice]</th>
<th>IDENT(VOICE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ampa]</td>
<td>*!</td>
<td></td>
</tr>
<tr>
<td>[amba]</td>
<td></td>
<td>*</td>
</tr>
</tbody>
</table>

(Prince and Smolensky, 1993, 2004)
Computational over-generation

▲ OT generates non-regular mappings.

<table>
<thead>
<tr>
<th>aaabb</th>
<th>*ab</th>
<th>MAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>aaabb</td>
<td>*!</td>
<td></td>
</tr>
<tr>
<td>☞ aaa</td>
<td>**</td>
<td></td>
</tr>
<tr>
<td>bb</td>
<td>***!</td>
<td></td>
</tr>
</tbody>
</table>

\[
f(a^n b^m) = \begin{cases} 
  a^n & m < n \\
  b^m & n < m 
\end{cases}
\]

(Gerdemann and Hulden, 2012)
Proposal

- **Boolean Monadic Recursive Schemes (BMRSs)** are a logical characterization of the subsequential functions (Bhaskar et al., 2020).
- They are a restriction on the recursive program schemes of Moschovakis (2019).
- Advantages for phonological analysis:
  - Maintain computational restrictiveness
  - Implement phonological substance
  - Capture commonly-held assumptions about phonological representations and grammars
String models

\[ D = \{1, 2, 3, 4, 5, 6\} \]
\[ s(1) = 2, s(2) = 3, \ldots \]
\[ p(2) = 1, p(3) = 2, \ldots \]
Feature-based representations

\[
\begin{bmatrix}
-\text{cons} \\
+\text{syll} \\
-\text{high} \\
+\text{low} \\
-\text{back} \\
\cdot
\end{bmatrix}
\quad
\begin{bmatrix}
+\text{cons} \\
+\text{son} \\
+\text{nasal} \\
+\text{labial} \\
\cdot
\end{bmatrix}
\quad
\begin{bmatrix}
+\text{cons} \\
-\text{son} \\
-\text{voice} \\
+\text{labial} \\
\cdot
\end{bmatrix}
\quad
\begin{bmatrix}
-\text{cons} \\
+\text{syll} \\
-\text{high} \\
+\text{low} \\
-\text{back} \\
\cdot
\end{bmatrix}
\]
Graph transduction

Input:

Output:
Output predicates

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

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

\[
\begin{align*}
\text{a} & \quad \text{m} \quad \text{p} \quad \text{a} \\
+\text{cons} & \quad +\text{son} & \quad +\text{cons} & \quad -\text{son} \\
+\text{nasal} & \quad +\text{labial} & \quad -\text{voice} & \quad +\text{labial} \\
\end{align*}
\]

\[
\begin{align*}
+\text{voice} \\
\vdots
\end{align*}
\]

\[\text{voice}_o(x) = \text{if } \text{nasal}(p(x)) \text{ then } \top \text{ else } \text{voice}(x)\]

\[\top = \text{True}, \ \bot = \text{False}\]
Output predicates

\[
\begin{bmatrix}
+\text{cons} \\ 
+\text{son} \\ 
+\text{nasal} \\ 
+\text{labial}
\end{bmatrix}
\] \quad \begin{bmatrix}
+\text{cons} \\ 
-\text{son} \\ 
-\text{voice} \\ 
+\text{labial}
\end{bmatrix}
\]

\[
\begin{bmatrix}
+\text{voice} \\ 
\vdots
\end{bmatrix}
\]

\[
\text{voice}_o(x) = \begin{cases} 
\top & \text{if } \text{nasal}(p(x)) \\
\text{voice}(x) & \text{else}
\end{cases}
\]

\[
F_o(x) = F(x), \text{ for all other features } F
\]
BMRS = subsequential (Bhaskar et al., 2020)

All functions:

- Output the type boolean
- Are of arity one
- Include either predecessor or successor terms, but not both
  - $BMRS^p = \text{left subsequential}$
  - $BMRS^s = \text{right subsequential}$
Constraints as substructures

\[ \text{voice}_o(x) = \begin{cases} \top & \text{if } \text{nasal}(p(x)) \text{ then } \top \text{ else } \text{voice}(x) \end{cases} \]

- Recall OT analysis: \([+\text{nasal}][-\text{voice}] >> \text{IDENT(VOICE)}\)
Computational restrictiveness

- Hierarchy of structure evaluation enacts constraint ranking.
- But, this evaluation is necessarily *local* in nature, avoiding the computational over-generation that stems from *global* evaluation (Frank and Satta, 1998; Gerdemann and Hulden, 2012; Lamont, 2018).
Demonstrations

- The inherent logic of how BMRS are structured and evaluated captures certain assumptions about how phonological grammars operate.
  1. Multiple repairs targeting the same marked structure.
  2. Elsewhere condition effects.
Licensing and blocking structures

\[ F_o(x) = \text{if } X \text{ then } \top \text{ else } F(x) \]

- X is a **licensing structure** for F

\[ F_o(x) = \text{if } X \text{ then } \bot \text{ else } F(x) \]

- X is a **blocking structure** for F
Demonstration 1: the typology of *NČ

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

  ▶ Various repair strategies are accounted for in OT with permutations of faithfulness constraints and *NČ.

(4)  Fusion: ex. from Indonesian (Austronesian)
     /mën+pilih/ → [məmilih], ‘to choose/vote’

Demonstration 1: the typology of *NČ

(3) \[ *\text{NČ} \]
No nasal/voiceless obstruent sequences (Pater, 2004)

▶ Various repair strategies are accounted for in OT with permutations of faithfulness constraints and *NČ.

(4) Deletion: ex. from Venda (Niger-Congo; South Africa)
\[ /n+pala/ \rightarrow [pala], 'scratching' \]

Demonstration 1: the typology of *NČ

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

- Various repair strategies are accounted for in OT with permutations of faithfulness constraints and *NČ.

(4) Denasalization: ex. from Mandar (Austronesian; Indonesia)
/man+tunu/ → [mat\text{\texttt{t}}\text{\texttt{t}}unu], ‘to burn’

Demonstration 1: the typology of *NČ

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

▶ Various repair strategies are accounted for in OT with permutations of faithfulness constraints and *NČ.

(4) Voicing: ex. from Puyo Pungo Quechua (Quechuan; Peru)
/kam+pə/ → [kamba], ‘yours’

Demonstration 1: the typology of *NČ

Fusion:
\( \text{Max, Ident(nasal), Ident(voice)} \gg *\text{NČ} \gg \text{Linearity} \)

Deletion:
\( \text{Linearity, Ident(nasal), Ident(voice)} \gg *\text{NČ} \gg \text{Max} \)

Denasalization:
\( \text{Linearity, Max, Ident(voice)} \gg *\text{NČ} \gg \text{Ident(nasal)} \)

Voicing:
\( \text{Linearity, Max, Ident(nasal)} \gg *\text{NČ} \gg \text{Ident(voice)} \)
Demonstration 1: the typology of \( ^*N\delta \)

The structure \( N\delta \) will either license or block features like nasal and voice.

\[
N\delta(x) = \begin{cases} 
\text{if } \text{nasal}(x) \text{ then } \zeta(s(x)) \text{ else } \bot 
\end{cases}
\]

\[
N\delta(x) = \begin{cases} 
\text{if } \zeta(x) \text{ then } \text{nasal}(p(x)) \text{ else } \bot 
\end{cases}
\]
Demonstration 1: the typology of *NČ

- **Voicing:** \( mp \rightarrow mb \)

\[
\text{voice}_o(x) = \text{if } \text{NČ}(x) \text{ then } \top \text{ else } \text{voice}(x)
\]

- **NČ **licenses **voice**
Demonstration 1: the typology of *NC

- Denasalization: nt $\rightarrow$ tt

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

- **NC** blocks nasal
Demonstration 1: the typology of *NC

- **Fusion**: np → m

\[
\text{out}(x) = \begin{cases} 
\bot & \text{if } \text{NC}(x) \text{ then} \\
\top & \text{else }
\end{cases}
\]

\[
\text{labial}_o(x) = \begin{cases} 
\top & \text{if } \text{NasalLabial}(x) \text{ then} \\
\text{labial}(x) & \text{else }
\end{cases}
\]

- **NC** blocks out
Demonstration 1: the typology of *NČ

<table>
<thead>
<tr>
<th></th>
<th>licensing</th>
<th>blocking</th>
<th>licensing</th>
<th>blocking</th>
</tr>
</thead>
<tbody>
<tr>
<td>out</td>
<td>deletion</td>
<td></td>
<td></td>
<td>fusion</td>
</tr>
<tr>
<td>[nasal]₀</td>
<td>denasalization</td>
<td>nasalization</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[voice]₀</td>
<td>devoicing(^1)</td>
<td>voicing</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\(^1\) Attested?
The gaps in the table can be explained if we rule out BMRS that aren’t meaningful.

Definition
A formula of the form if X then B else Y is meaningful if neither of the following holds:
1. B is $\top$ and X implies Y
2. B is $\bot$ and X implies $\neg Y$

e.g., nasal$_o(x) = \text{if } N\text{C}(x) \text{ then } \top \text{ else nasal}(x)$
Demonstration 2: Elsewhere condition effects

- As originally proposed in SPE (Chomsky and Halle, 1968), phonological rules applied conjunctively in sequence.
- The Elsewhere Condition was proposed to allow for *disjunctive* application (Kiparsky, 1973).
- When the context of one rule properly includes that of another, the more specific rule blocks the more general one.
Demonstration 2: Elsewhere condition effects

(5) Diola Fogny (Niger-Congo; Senegal)

a. /let+ku+ɛaw/ → [lekuɛaw], ‘they won’t go’
b. /ku+ɓɔn+ɓɔn/ → [kuɓɔmbɔn], ‘they sent’

▶ Generally, the first of two consecutive consonants is deleted.
▶ When the first consonant is a nasal, it escapes deletion by assimilating in place.
▶ Disjunctive ordering: the more general consonant deletion rule only applies when nasal assimilation does not.

(Sapir 1965, Kiparsky 1973, Baković 2013)
Demonstration 2: Elsewhere condition effects

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

\[ \text{CC}(x) = \begin{cases} \text{if consonant}(x) \text{ then consonant}(s(x)) \text{ else } \bot \end{cases} \]
Demonstration 2: Elsewhere condition effects

\[
\text{out}_o(x) = \begin{cases} 
\text{T} & \text{if NC}(x) \text{ then } \top \\
\text{if CC}(x) \text{ then } \bot \\
\text{else } & \text{else } \\
\text{T} & \text{else}
\end{cases}
\]

Because \( \text{NC}(x) \) implies \( \text{CC}(x) \), if \( \text{CC}(x) \) appeared higher in the hierarchy than \( \text{NC}(x) \), the latter would never be evaluated (and would therefore be extraneous).
Theorem

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 ever take effect.

(Chandlee and Jardine 2021)
Strict Substructure Ordering

**Theorem**

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 ever take effect.

Unlike the Elsewhere Condition\(^1\), the SSOT follows directly from the logic of how BMRSs are evaluated and does not need to be stipulated.

\(^1\)But like Pāṇini’s Theorem on Constraint-ranking in OT (Prince and Smolensky, 2004; Baković, 2013).
Current/Future Work

- Combinations of BMRS (i.e., function composition) for process interaction (Oakden, 2021)
- Further restrictions on BMRS for additional subregular function classes
- Non-linear phonological representations
Subregular functions

Chandlee (2014); Chandlee et al. (2015); also Tier-based Strictly Local (TSL) (Burness and McMullin, 2019; Hao and Andersson, 2019; Hao and Bowers, 2019) and Strictly Piecewise (SP) (Burness and McMullin, 2020).
Conjectures

- Input strictly local (ISL) functions = no recursion
- Output strictly local (OSL) functions = recursion only in combination with successor (ROSL) or predecessor (LOSL)
Non-linear phonological representations

(Goldsmith, 1976; Chandlee and Jardine, 2021)
Non-linear phonological representations

(Strother-Garcia, 2019)
Conclusion

- BMRS compute phonological processes locally, maintaining the desirable computational restrictiveness of subregularity in a way that is more intuitive from a phonological perspective compared to finite-state formalizations.

  ✓ Feature-based representations
  ✓ Non-linear representations
  ✓ Feature licensing/blocking via an evaluation hierarchy
References


References II


Acknowledgements

Thanks to Siddharth Bhaskar (James Madison University) for introducing us to recursive schemes, and to Steven Lindell (Haverford College), Jeffrey Heinz (Stony Brook University), and Eric Baković (UC San Diego) for valuable discussion and feedback.

And thank you to the organizers of MOL and MALIN 2021!