Formalizing Iterativity and the Computation of Rule Application Modes

Jane Chandlee

Haverford College, Haverford, PA, USA

20 May 2021
Tel Aviv University
Rule application modes

\[ [-\text{cons}] \rightarrow [+\text{nas}] / [+\text{nas}] \]

Language A  Language B

\[ \text{NVVV} \rightarrow \text{ÑVVV} \quad \text{NVVV} \rightarrow \text{ÑVVV} \]
Rule application modes

\[ [-\text{cons}] \rightarrow [+\text{nas}] / [+\text{nas}] \]

Language A | Language B
---|---
\( \text{NVVV} \leftrightarrow \text{N\text{\~}}\text{VVV} \) | \( \text{NVVV} \leftrightarrow \text{N\text{\~}}\text{VVV} \)

iterative | simultaneous
Rule application modes

\[-\text{cons}] \rightarrow [\text{+nas}] \, / \, [\text{+nas}] \_\_

Language A \quad \text{Language B}

\text{N}\_\text{VVV} \leftrightarrow \text{\~N}\_\text{VVVV} \quad \text{N}\_\text{VVV} \leftrightarrow \text{\~N}\_\text{VVVV}

iterative \quad \text{simultaneous}

One rule, different application modes.
Rule application modes

<table>
<thead>
<tr>
<th>Language A</th>
<th>Language B</th>
</tr>
</thead>
<tbody>
<tr>
<td>([-\text{cons}] \rightarrow [+\text{nas}] / [+\text{nas}])</td>
<td>([-\text{cons}] \rightarrow [+\text{nas}] / [+\text{nas}, +\text{cons}])</td>
</tr>
<tr>
<td>(\text{NVVV} \rightarrow \text{ÑVVV})</td>
<td>(\text{NVVV} \rightarrow \text{ÑVVV})</td>
</tr>
<tr>
<td>iterative</td>
<td>iterative</td>
</tr>
</tbody>
</table>

Different rules, one application mode.

(See Chomsky and Halle, 1968; Johnson, 1972; Howard, 1972, among others)
Optimality Theory (Prince and Smolensky, 1993, 2004) is inherently iterative.

<table>
<thead>
<tr>
<th>/NVVV/</th>
<th>[+nas][−nas]</th>
<th>IDENT-NAS</th>
</tr>
</thead>
<tbody>
<tr>
<td>[NVVV]</td>
<td>*!</td>
<td></td>
</tr>
<tr>
<td>[N˘VVV]</td>
<td>*!</td>
<td>*</td>
</tr>
<tr>
<td>[N˘V˘VV]</td>
<td>*!</td>
<td>**</td>
</tr>
<tr>
<td>[N˘V˘V˘V]</td>
<td>*!</td>
<td>***</td>
</tr>
</tbody>
</table>
Optimization

‘iterative’
*+[+nas][−nas] ⟹ IDENT-NAS

‘simultaneous/noniterative’
*+[+nas,+cons][−nas] ⟹ IDENT-NAS ⟹ *+[+nas][−nas]
Optimality Theory

- Emergent Noniterativity Hypothesis (Kaplan, 2008): No formal entity in phonological grammars may require noniterativity.
  - Prediction: ‘There is no phenomenon that must be analyzed with a self-feeding rule that is not permitted to apply to its own output’ (pg. 4).
In terms of computation

- A computational approach isn’t about rules or constraints, but their *extensions*:
  - $\{(NV, N\tilde{V}), (VN, VN), (NVV, N\tilde{VV}), \ldots\}$ (‘noniterative’)
  - $\{(NV, N\tilde{V}), (VN, VN), (NVV, N\tilde{VV}), \ldots\}$ (‘iterative’)
- An extension or *map* either has a given computational property, or it doesn’t.
  - Can’t change the rule or add constraints.
Properties of extensions

- Relevant computational properties for this talk: input strict locality (ISL) and output strict locality (OSL).
- ISL and OSL functions are part of a subregular hierarchy of functions (see Heinz, 2018).
FST characterizations

\[ a \rightarrow b / b \]

**ISL**

\[ \lambda \]

**OSL**

\[ \lambda \]

\[ a : a \]

\[ b : b, a : b \]
Rule application

\[ a \rightarrow b / b ___ \]

\[
\begin{align*}
\text{ISL} \approx \text{noniterative} & \quad \{(baaa, bbba) \ ... \ } \\
\text{OSL} \approx \text{iterative} & \quad \{(baaa, bbbb) \ ... \ }
\end{align*}
\]
Do we need both?

- For many (most?) rules, ISL and OSL are extensionally-equivalent.
  - Final obstruent devoicing: \{(bad, bat), (hund, hunt), \ldots \}
- Are there any rules that are *necessarily* ISL?
Nasal spreading/assimilation

\[ /NVVV/ \leftrightarrow [N\tilde{N}\tilde{N}\tilde{N}] \quad \text{iterative} \]

\[ /NVVV/ \leftrightarrow [N\tilde{N}VV] \quad \text{noniterative} \]
Nasal spreading/assimilation

/NVVV/ → [ÑVṼ] OSL

/NVVV/ → [ÑVVV] ISL
Nasal spreading/assimilation

/NVVV/ → [Nĩĩĩĩ]  OSL

/NVVV/ → [NĩVV]  ISL (and OSL?)
Noniterative OSL map

OSL (iterative)

OSL (noniterative)
Goal: define (phonologically-intuitive) conditions for when OSL will enforce iteration.
Let $\text{Triggers}$ be the set of segments that trigger a process, and let $\text{Outputs}$ be the set of segments that result from that process.

(1) A phonological map is

a. $\text{output-nondistinct}$ if $\text{Outputs} \cap \text{Triggers} \neq \emptyset$.

b. $\text{output-distinct}$ if $\text{Outputs} \cap \text{Triggers} = \emptyset$. 
Triggers and Outputs

/\NVVV\/ \mapsto \{N\tilde{V}\tilde{V}\tilde{V}\}\tilde{V}/ 

\text{Triggers} = \{N, \tilde{V}\} \quad \text{Triggers} = \{N\}

\text{Outputs} = \{\tilde{V}\} \quad \text{Outputs} = \{\tilde{V}\}

\text{output-nondistinct} \quad \text{output-distinct}
## ISL and OSL

<table>
<thead>
<tr>
<th></th>
<th>output-nondistinct</th>
<th>output-distinct</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>OSL</strong></td>
<td>iterative</td>
<td>noniterative</td>
</tr>
</tbody>
</table>
## ISL and OSL

<table>
<thead>
<tr>
<th></th>
<th>output-nondistinct</th>
<th>output-distinct</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSL</td>
<td>iterative</td>
<td>noniterative</td>
</tr>
<tr>
<td>ISL</td>
<td>noniterative</td>
<td>noniterative</td>
</tr>
</tbody>
</table>
## ISL and OSL

<table>
<thead>
<tr>
<th></th>
<th>output-nondistinct</th>
<th>output-distinct</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSL</td>
<td>iterative</td>
<td>noniterative</td>
</tr>
<tr>
<td>ISL</td>
<td>noniterative</td>
<td>noniterative</td>
</tr>
</tbody>
</table>

![Venn diagram showing the overlap between ISL and OSL with their respective properties](image-url)
What are we looking for?

- ‘True noniterativity’ (i.e., necessarily ISL) would be an output-nondistinct noniterative map.
- Vowel harmony is a great place to look.
A note on local versus long-distance

- If we assume vowel harmony rules include $C_0$, they are trivially neither ISL nor OSL.

  \[(2) \quad V_{-\alpha} \rightarrow +\alpha / V_{+\alpha} C_0 \quad \]

- Will instead use strings of vowels, which can be interpreted in two ways:
  1. The map applies on a vowel tier.
  2. The number of consonants is bounded by syllable structure and can be ignored (for presentation purposes).
In Bengali, \{ɛ, ɔ\} → \{e, o\} in nouns when they precede a high vowel (Mahanta, 2007).

kɔṭʰa ‘spoken words’ kɔṭʰito ‘uttered’ kɔṭʰoniyo ‘speakable’
kɔlpo ‘resembling’ kɔlpito ‘invented’ kɔlponiyo ‘imaginable’

Noniterative: /ɔɔi/ → [ɔoi]
**Bengali ATR harmony**

- **TRIGGERS** = \{i, u\} (only high vowels can trigger harmony)
- **OUTPUTS** = \{e, o\}
- ✓ output-distinct
- Both ISL and OSL.
Crimean Tatar labial harmony


(3)  Central Crimean Tatar (N\text{MZR-Poss.3S})

a. /tuz-lw\text{\text{y}-w}/ [tuz-luy-w] ‘salt’

b. /kyz-lig-i/ [kyz-lyg-i] ‘autumn’

c. /toz-lw\text{\text{y}-w}/ [toz-luy-w] ‘dust’

d. /køz-lig-i/ [køz-lyg-i] ‘eye’
Crimean Tatar labial harmony

- **Triggers** = \{y, \emptyset, u, o\}
- **Outputs** = \{y, \emptyset, u, o\}
Crimean Tatar labial harmony

- **TRIGGERS** = \{y^i, \phi^i, u^i, o^i\} (only initial vowels can trigger)
- **OUTPUTS** = \{y, \phi, u, o\}
- ✓ output-distinct
- Both ISL and OSL.
Crimean Tatar labial harmony

\[ V_{-rd} \rightarrow [+rd] / \# V_{+rd} \]

\[^*\# [+rd, −rd] \text{(McCollum and Kavitskaya, 2018)}\]
Kazakh labial harmony

[+rd] vowel triggers harmony on following vowel (McCollum and Kavitskaya, 2018):

a. /mojәn-dә/ [mojәn-dә] ‘neck-Acc’
b. /tүr-mәsә-nәŋ/ [tүr-mәsә-nәŋ] ‘live-NMZR-POSS.3-GEN’
c. /kino-m-әz-dәŋ/ [kino-m-әz-dәŋ] ‘movie-POSS.1-PL-GEN’
Whether a ‘potential’ trigger gets to actually be a trigger is an empirical question.

As long as something distinguishes underlying triggers from derived ones, OSL will not enforce iteration.

Computational support for ‘emergent noniterativity’, but we still need ISL too!
Two additional reasons we need ISL

1. Rules with two-sided contexts are also necessarily ISL.
**Delaying output**

\[ a \rightarrow b / a \_\_ a \]

**ISL**

\[ \lambda \quad a : a \quad a : \lambda \quad a : b \quad \# : a \]

**OSL**

\[ \lambda \quad a : a \quad a : \lambda \]
Two additional reasons we need ISL

1. Rules with two-sided contexts are necessarily ISL (except epenthesis).
2. ISL provides a straightforward solution to:
   - Opacity (Chandlee et al., 2018)
   - Ordering/ranking paradoxes (Oakden and Chandlee, 2020; Oakden, 2021)
   - Nonderived environment blocking (Chandlee, to appear)
Dissimilation

(4) \( L \rightarrow R \ / \ L \_ \)

- **Triggers** = \{L\}
- **Outputs** = \{R\}
- ✔️ output-distinct

Predicts both ISL and OSL will produce a noniterative map.
What is a noniterative dissimilation map?

(5) /LLLL/ $\mapsto$ [LRLL]?

Simultaneous application gives:

(6) /LLLL/ $\mapsto$ [LRRR]
Dissimilation

- Tianjin tone sandhi (Chen, 1986; Hung, 1987; Tan, 1987; Zhang, 1987; Chen, 2000; Lin, 2008; Wee, 2010)

(7) a. $R \rightarrow H / \_ \_ R$
b. $RRR \rightarrow HHR$
Dissimilation

ISL

OSL

LLLL $\Rightarrow$ LRRR

LLLL $\Rightarrow$ LRLR
Dissimilation: future work

- Modify definitions to predict iterativity in OSL dissimilation maps.
- Iterative dissimilation is actually self-bleeding.
- Highlights the problem of using rule-based terms like iterativity/noniterativity to describe extensions.
Conclusions

- Computational (‘theory independent’) support for the idea that phonology is iterative and output-oriented, just not exclusively.
- We can’t dispense with ISL, but we have a better understanding of when we need it and why.
  - Noniterativity does not necessitate ISL.
  - Connection to rule application modes overgeneralizes.
- The original goal of restricting the theory is maintained as a restriction on computational complexity via subregular functions.
Acknowledgements

Thank you!

(And thanks to an earlier audience at Johns Hopkins University and to Adam McCollum for helpful discussion.)
Chandlee, J. Nonderived environment blocking is input strictly local. To appear in *Evolutionary Linguistic Theory*.


