Formalizing (Non)iterativity and the Computation of Rule Application Modes

Jane Chandlee
Haverford College

April 28, 2022
Iterative and noniterative

- The terms *iterative* vs. *noniterative* have their origin in rule-based phonology, but even in that context what they mean depends on the theory.
- Outside of rule-based phonology, they serve as descriptive labels for certain phonological maps, but it is unclear to what extent they reflect actual properties of those maps.
- Viewing phonological maps in terms of their computational properties leads to a more formal and precise understanding of what it means for a map to be iterative, or noniterative, or both, or neither...
Simultaneous application

‘To apply a rule, the entire string is first scanned for segments that satisfy the environmental constraints of the rule. After all such segments have been identified in the string, the changes required by the rule are applied simultaneously’ (Chomsky and Halle, 1968).

- Assumption of noniterativity = rules cannot reapply to their own outputs.

(1) \([-\text{cons}] \rightarrow [+\text{nasal}] / [+\text{nasal}]\)
   a. \(\text{NVVV} \rightarrow \text{N}\tilde{V}VV\)
Simulating iteration

(2) \([-\text{cons}] \rightarrow [+\text{nasal}] / [+\text{nasal}] ([-\text{cons}])^* \) __

a. \([-\text{cons}] \rightarrow [+\text{nasal}] / [+\text{nasal}] __

b. \([-\text{cons}] \rightarrow [+\text{nasal}] / [+\text{nasal}] [−\text{cons}] __

c. \([-\text{cons}] \rightarrow [+\text{nasal}] / [+\text{nasal}] [−\text{cons}][−\text{cons}] __

d. NVVV \rightarrow N\tilde{V}\tilde{V}\tilde{V}\tilde{V}
Directional rules

- Criticisms of parenthesis-star motivated proposals in which rules can apply to their own outputs (Johnson, 1972; Howard, 1972; Lightner, 1972; Anderson, 1974; Kenstowicz and Kisseberth, 1977).

- In some theories, direction is stipulated:
  \[ [-\text{cons}] \rightarrow [+\text{nasal}] / [+\text{nasal}] \]

  left-to-right  right-to-left

  \[ \text{NVVV} \rightarrow \text{N\~NV\~V} \quad \text{NVVV} \rightarrow \text{N\~VVV} \]
Directional rules

• In other theories, direction is determined by the rule’s form:

\begin{align*}
(3) & \quad \text{a. } [-\text{cons}] \rightarrow [+\text{nasal}] / [+\text{nasal}] \quad \text{(left-to-right)} \\
& \quad \text{b. } [-\text{cons}] \rightarrow [+\text{nasal}] / \_ [+] [\text{nasal}] \quad \text{(right-to-left)}
\end{align*}

• Noniteration remains available as an ad hoc solution when needed.

• The availability of both options is more official in theories with an iterativity parameter (e.g., Archangeli and Pulleyblank, 1994).
Enter OT (Prince and Smolensky, 1993)

- Optimality Theory is inherently iterative, in that iteration is optimal.

<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>[NVVV]</td>
<td>*!</td>
<td>*</td>
</tr>
<tr>
<td>[ÑVVV]</td>
<td>*!</td>
<td>**</td>
</tr>
<tr>
<td>[ÑVVV]</td>
<td></td>
<td>***</td>
</tr>
</tbody>
</table>
Feature not a bug

- Emergent Noniterativity Hypothesis (Kaplan, 2008): No formal entity in phonological grammars may require noniterativity.
- Noniterativity is always epiphenomenal and can be explained by other means.
Emergent noniterativity

(4) Lango (Nilo-Saharan; Noonan, 1992)
   a. /bɔŋo-ní/ → [bɔŋo-ní], ‘your (sg.) dress’
   b. /cɔŋo-ní/ → [cɔŋo-ní], ‘your (sg.) beer’

• Positional licensing: suffix [ATR] needs to be linked to the root (Kaplan, 2008).
‘True noniterativity’

- But, some cases of noniterativity cannot be reanalyzed as emergent in this way (Ampofo and Rasin 2021, McCollum under review).
Defining (non)iterativity

- Extension: NVVV $\rightarrow$ ÑVVV
- Intensional description:
  
  $[-\text{cons}] \rightarrow [+\text{nasal}] / [+\text{nasal}]$ ___ (simultaneous application/−iterative)

  $[-\text{cons}] \rightarrow [+\text{nasal}] / [+\text{nasal}]$ ___ (right-to-left iterative)

  $[-\text{cons}] \rightarrow [+\text{nasal}] / [+\text{nasal}, +\text{cons}]$ ___ (left-to-right, right-to-left, −iterative/simultaneous)
• FLT models of phonological maps reflect more invariant properties compared to their intensional descriptions.

• Finite-state models typically implement iteration vs. noniteration by matching rule contexts on the output vs. input string, respectively (Kaplan and Kay, 1994; Hulden, 2009; Gorman and Sproat, 2021).
Matching context on input
Matching context on output
Input/output and +/- iterative

- It’s tempting to then define iterative as ‘output-based’ and noniterative as ‘input-based’, in terms of the computations they require.
- But, the sets of maps that can be generated with these computation types are not disjoint, and the distinction is often irrelevant.
- GOAL: isolate those cases for which the distinction does matter, i.e., those cases for which input-based computation is necessary to prevent otherwise inescapable iteration.
Significance

1. Evidence that a computational(ly-informed) theory of phonology needs access to both input and output structure.

2. More thorough understanding of when each type of computation is needed and why.

3. More precise definitions of *iterative* and *noniterative* that aren’t tied to a particular theory’s intensional descriptions.
Rules as functions

- Zooming in for now on rules of this form:

  (5)  
  a. $A \rightarrow B / C$  
  b. $A \rightarrow B / \_ C$  
  c. where $A$ and $C$ are finite sets of strings

- In other words, ‘local processes’
- Doing so enables the use of more restrictive finite-state models that make the need for input/output structure more clear.
- Will return to rules with two-sided contexts later.
**ISL and OSL**

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

**Input Strictly Local**

\[ \lambda \]

\[ \begin{array}{c}
  b : b \\
  a : b \\
  a : a \\
  \end{array} \]

\[ \begin{array}{c}
  b : b \\
  b : b \\
  a : a \\
  \end{array} \]

\[ \text{baaa} \rightarrow \text{bbaa} \]

**(Left) Output Strictly Local**

\[ \lambda \]

\[ \begin{array}{c}
  b : b \\
  a : b \\
  a : a \\
  \end{array} \]

\[ \begin{array}{c}
  b : b \\
  b : b \\
  b : b \\
  \end{array} \]

\[ \text{baaa} \rightarrow \text{bbbb} \]
Relationship among classes

Chandlee (2014); Chandlee et al. (2015)
Substitution (non-dissimilatory)

\[
[-\text{cons}] \rightarrow [+\text{nasal}] \slash [+\text{nasal}] \\
\text{+iterative} \quad \quad \quad \text{—iterative}
\]

NVVV → ÑVṼ \quad NVVV → ÑVVV

LOSL? \quad ISL?
Substitution (non-dissimilatory)

\[-\text{cons}] \rightarrow [+\text{nasal}] / [+\text{nasal}] \quad __

+\text{iterative} \quad \quad \quad \quad \quad \quad -\text{iterative}

NVVV \rightarrow \tilde{N}V\tilde{V}V \quad NVVV \rightarrow \tilde{N}V\tilde{V}V

LOSL \quad \quad \quad \quad \quad \quad ISL \cap LOSL
OSL and noniterativity

OSL (+iterative)

OSL (−iterative)
Let \textsc{Triggers} be the set of segments that trigger a process, and let \textsc{Outputs} be the set of segments that result from that process.

A phonological map is

1. \textit{output-nondistinct} if \textsc{Outputs} \cap \textsc{Triggers} \neq \emptyset.
2. \textit{output-distinct} if \textsc{Outputs} \cap \textsc{Triggers} = \emptyset.
Triggers and Outputs

\[\text{NVVV} \rightarrow \text{n}\tilde{V}\tilde{V}\tilde{V}\tilde{V}\]
\[\text{NVVV} \rightarrow \text{n}\tilde{V}\tilde{V}\tilde{V}\]

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

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

output-nondistinct
output-distinct
Bengali ATR harmony

\{ε, ɔ\} → \{e, o\} before a high vowel (Mahanta, 2007).

\begin{align*}
\text{kōth̩a} & \quad \text{‘spoken words’} & \text{kọlpo} & \quad \text{‘resembling’} \\
\text{kōṭ̩ito} & \quad \text{‘uttered’} & \text{kọlpito} & \quad \text{‘invented’} \\
\text{kōthoniyo} & \quad \text{‘speakable’} & \text{kọlponiyo} & \quad \text{‘imaginable’}
\end{align*}

/ɔɕi/ → [oei]
Bengali ATR harmony

- **Triggers** = \{i, u\} (only high vowels can trigger harmony)
- **Outputs** = \{e, o\}
- Output-distinct
- **ISL \cap ROSL**
Central Crimean Tatar labial harmony


/tuz-luwγ-ų/ [tuz-łuγ-ų] ‘salt’
/kyz-lig-i/ [kız-lyg-i] ‘autumn’
/toz-luwγ-ų/ [toz-łuγ-ų] ‘dust’
/køz-lig-i/ [køz-lyg-i] ‘eye’

All forms marked for Nmzr-Poss.3S.
Crimean Tatar labial harmony

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

- **Triggers** = \{\#y, \#ø, \#u, \#o\} (only initial vowels are triggers)
- **Outputs** = \{y, ø, u, o\}
- Output-distinct
- ISL \cap LOSL
Crimean Tatar labial harmony

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

*\# [+rd, -rd] (McCollum and Kavitskaya, 2018)
Akan serial verbs

[+ATR] spreads leftward to the next vowel, but no further (Ampofo and Rasin, 2021).

/tɔ fa di/ [tɔ fæ di] ‘buy, take, and eat’
/tɔ fa bo di/ [tɔ fa bo di] ‘buy, take, crack, and eat’
/tɔ di su/ [to di su] ‘buy, eat, and cry’
Akan serial verbs

- **Triggers = Outputs** = \{i, e, æ, o, u\}
- Output-nondistinct
- ISL only
## Substitution (non-dissimilatory)

<table>
<thead>
<tr>
<th>ISL</th>
<th>ISL $\cap$ OSL</th>
<th>OSL</th>
</tr>
</thead>
<tbody>
<tr>
<td>output-nondistinct</td>
<td>output-distinct</td>
<td>output-nondistinct</td>
</tr>
<tr>
<td>single target</td>
<td></td>
<td>multiple targets</td>
</tr>
</tbody>
</table>
(6) \( L \rightarrow R / L \) __

- **TRIGGERS** = \{L\}
- **OUTPUTS** = \{R\}
- Output-distinct
- Predicts both ISL and OSL will produce a noniterative map.
Dissimilation

(7) \( L \rightarrow R / L \)

- ISL (and simultaneous application): \( LLLL \rightarrow LRRR \)
- LOSL (and iterative application): \( LLLL \rightarrow LRLR \)
Tianjin tone sandhi (Chen, 1986, 2000)

\[ R \rightarrow H / \_ \_ R \quad RRR \rightarrow HHR \]

\[ F \rightarrow L / \_ \_ F \quad FFF \rightarrow FLF \]

\[ L \rightarrow R / \_ \_ L \quad LLL \rightarrow LRL \]
Rhythmic law

(8) \( V:\rightarrow V / V: C_0 \) __

Slovak (Indo-European; Slovakia)

/ˇci:t-a:m/ \( [\tilde{c}i:t-am] \) ‘read-1S’
/ˇci:t-a:v-a:m/ \( [\tilde{c}i:t-av-am] \) (frequentive)

Githabal (Australian)

/nu:n-da:j/ \( [nu:n-da\ddot{a}] \) ‘too hot’
/djalum-ba:da:j-be:/ \( [djalum-ba:\ddot{a}da\ddot{a}-be:] \) ‘is certainly right on the fish’

(Kenstowicz and Kisseberth, 1979)
Dissimilation (summary)

<table>
<thead>
<tr>
<th>ISL</th>
<th>ISL \cap OSL</th>
<th>OSL</th>
</tr>
</thead>
<tbody>
<tr>
<td>identical span</td>
<td>alternating pattern</td>
<td></td>
</tr>
</tbody>
</table>

- Output-distinctness is irrelevant; these rules are self-\textit{bleeding}.
- What is relevant is whether the rule generates an alternating pattern or a span of identical segments.
- ‘Forced choice’ assumption of (non)iterativity, though we often won’t have the data to tell.
‘Anchored’ deletion

\[(9) \quad V \rightarrow \emptyset / \_ \; \#\]

- ISL (and simultaneous application): \(\text{CVVV} \rightarrow \text{CVV}\)
- ROSL (and iterative application): \(\text{CVVV} \rightarrow C\)
‘Overlap’ deletion

(10) \[ V \rightarrow \emptyset / \_ \_ \_ V \]

ISL (simultaneous) \hspace{1cm} \text{aeiou} \rightarrow u

LOSL (left-to-right iter) \hspace{1cm} \text{aeiou} \rightarrow \text{eiou} \rightarrow \text{iou} \rightarrow \text{ou} \rightarrow u

ROSL (right-to-left iter) \hspace{1cm} \text{aeiou} \rightarrow \text{aeiu} \rightarrow \text{aeu} \rightarrow \text{au} \rightarrow u
## Deletion (summary)

<table>
<thead>
<tr>
<th>ISL</th>
<th>ISL $\cap$ OSL</th>
<th>OSL</th>
</tr>
</thead>
<tbody>
<tr>
<td>anchored</td>
<td>overlap</td>
<td>anchored</td>
</tr>
<tr>
<td>single target</td>
<td></td>
<td>multiple targets</td>
</tr>
</tbody>
</table>
## Interim summary

<table>
<thead>
<tr>
<th></th>
<th>ISL</th>
<th>ISL ∩ OSL</th>
<th>OSL</th>
</tr>
</thead>
<tbody>
<tr>
<td>substitution</td>
<td>output-nondist</td>
<td>output-distinct</td>
<td>output-nondist</td>
</tr>
<tr>
<td></td>
<td>single target</td>
<td></td>
<td>multiple targets</td>
</tr>
<tr>
<td>dissimilation</td>
<td>identical span</td>
<td></td>
<td>alternating pattern</td>
</tr>
<tr>
<td>deletion</td>
<td>anchored</td>
<td>overlap</td>
<td>anchored</td>
</tr>
<tr>
<td></td>
<td>single target</td>
<td></td>
<td>multiple targets</td>
</tr>
</tbody>
</table>
Rules with two-sided contexts

(11) \[ a \rightarrow b / a \_\_\_ a \]

\[
\begin{array}{c}
\lambda \\
a \\
aa \\
\end{array}
\]

\[
\begin{array}{c}
b:b \\
a:a \\
\# : a \\
\end{array}
\]

\[
\begin{array}{c}
a : b \\
a : \lambda \\
b : ab \\
b : b \\
\end{array}
\]

aaaa → abba
Rules with two-sided contexts

- These rules are necessarily ISL, but do they ever iterate?
- Survey of PBase (Mielke, 2008):

<table>
<thead>
<tr>
<th></th>
<th>Potentially iterative</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Substitution</td>
<td>8</td>
<td>672</td>
</tr>
<tr>
<td>Deletion</td>
<td>37</td>
<td>207</td>
</tr>
<tr>
<td>Epenthesis</td>
<td>—</td>
<td>287</td>
</tr>
</tbody>
</table>

(12) Iterative epenthesis

a. $\emptyset \rightarrow ab / a \_ b$
Potentially iterative, but...

(13) Mundari (Austro-Asiatic; India; Cook, 1965, pg. 61)

a. \{u, o\} → [w] / V — V
b. /kiũa/ [kiwa] ‘chin’
c. /heẽa/ [hewa] ‘accustom’

- Need to see what happens to /euua/
Rhythmic syncope

(14) \( V \rightarrow \emptyset \) / VC \_ CV

Maithili (Indo-European; India; Yadav, 1996, pg. 51)

/\text{nik}_\text{el-\text{et-ah}}/ [\text{nik}_\text{el-tah}] \sim [\text{nik}_\text{l-\text{etah}}], *[\text{nik}_\text{l-tah}] \quad \text{‘will come out’}

/\text{l}_\text{et-\text{ek-\text{el-\text{e}uk}}}/ [\text{l}_\text{et-\text{k-el-e}uk}] \sim [\text{l}_\text{et-\text{ek-l-e}uk}], *[\text{l}_\text{et-k-l-e}uk] \quad \text{‘(your) hung’}
Rhythmic syncope

- Definitely iterates, but is it two-sided?
- \((CV)^n\), if \(n\) is odd...

<table>
<thead>
<tr>
<th>C</th>
<th>V</th>
<th>C</th>
<th>V</th>
<th>C</th>
<th>V</th>
<th>C</th>
<th>V</th>
<th>C</th>
<th>V</th>
</tr>
</thead>
<tbody>
<tr>
<td>V</td>
<td>→</td>
<td>∅</td>
<td>/</td>
<td>VC</td>
<td>→</td>
<td>CV</td>
<td>∅</td>
<td>_</td>
<td>∅</td>
</tr>
<tr>
<td>V</td>
<td>→</td>
<td>∅</td>
<td>/</td>
<td>VC</td>
<td>→</td>
<td>_</td>
<td>∅</td>
<td>_</td>
<td>_</td>
</tr>
</tbody>
</table>
Rhythmic syncope

- Definitely iterates, but is it two-sided?
- \((CV)^n\), if \(n\) is even...

\[
\begin{array}{cccccc}
C & V & C & V & C & V \\
V \rightarrow \emptyset & / & VC \rightarrow & CV \\
V \rightarrow \emptyset & / & VC \rightarrow & \emptyset & \_ & \emptyset
\end{array}
\]
Two-sided iterative rules

- Neither ISL nor OSL, because they need both.
- See Hao and Bowers (2019); Bowers and Hao (2020) for one solution: tier-based synchronized strictly $k$-local ($k$-TSSL) functions
Conclusions

• In rule-based phonology, *iterative* = a rule that reapplies to its own output. If all rules apply this way, then all maps are iterative.

• In OT, *noniterative* = epiphenomenal, or maybe ‘potentially problematic’.

• Once we step away from rules/constraints, we can reserve the terms *iterative* and *noniterative* for those cases that necessitate output- or input-based computation, respectively.
  • Or, we could just dispense with these terms entirely!
Conclusions

- The results presented in this talk identify when we need each computation type and why.
- The original goal of having only one rule application mode was motivated by a need for restrictiveness.
- Having access to both input and output structure does not necessarily compromise that goal, as subregularity ensures restrictiveness.
References I


Cook, W. A. (1965). A descriptive analysis of Mundari: A study of the structure of the Mundari language according to the methods of the linguistic science, with particular attention to the units of sound, the units of meaning, the units of grammar, and their mutually contrastive arrangement patterns. PhD thesis, Georgetown University.


References II


Thank you!
Appendix: Maps that are $+/-$ iterative?

- In vowel harmony with *icy targets* (Jurgec, 2011), a vowel undergoes the harmony but then blocks it.

(15) Karajá (Macro-Jê; Central Brazil; Ribeiro, 2003)

a. /bɛdɔ-dĩ/ [bedoni] ‘a type of *filhote* (fish species)’

b. /krɔbi-dĩ/ [krɔbini] ‘a type of monkey’
Appendix: Maps that are $+/−$ iterative?

- Input \textit{tier-based} strictly local function (Burness et al., 2021)
- Alphabet: $\{C, \varsigma, \varepsilon, i, u, o, e, i, u\}$
- Tier: $\{o, e, i, u, i, u\}$

\[
\lambda: i, u, \varepsilon, \varsigma, C
\]
\[
B: i, u, \varepsilon, \varsigma, C
\]