Overview

- **Goal:** Computational characterizations lead to restrictive, testable, and learnable theories of phonology (Heinz, 2018).
- **Question:** what are the computational requirements of iterative prosody?
- **Result:** Local (with recursion)!

Logical maps

- Logic: output defined over input (Courcelle, 1994)
  \[ c(x) \equiv b(x) \land a(p(x)) \]
  \[ \begin{array}{c|c}
  a & \begin{array}{c}
  p \\
  b \\
  \end{array} \\
  \hline
  \begin{array}{c}
  a \\
  p \\
  b \\
  c \\n  \end{array} & \begin{array}{c}
  a \\
  p \\
  b \\
  c \\n  \end{array} \\
  \end{array} \]
- Local = Quantifier Free: no \( \exists, \forall \) (Chandlee and Lindell, forthcoming)
- Recursion = Least Fixed Point (LFP) operators (Libkin 2004)
- Local recursion = QF-LFP reference information from the output string
- Implicit Recursion = implicit definitions; Rogers 1997
- Use either predecessor (p) or successor (s)

Least Fixed Point: Local recursion

Example: \( baaa \rightarrow bbbb \)
- \( a \)'s following \( a \) \( b \) are outputted as \( b \)
- Not local on the input
- Local on output: \( a \rightarrow b \) when \( b \) before \( a \) on output

Implicit definition: \( b'(x) = b(x) \lor b'(p(x)) \)
- Given an input element \( x \)
  - it is mapped to \( a \) in the output
  - when it is \( a \) in the input or
  - it is preceded by \( a \) in the output.

Iterative stress

- Murinbata: stress 1st \( \sigma \) and every other \( \sigma \)
  \[ \sigma \sigma \sigma \sigma \sigma \rightarrow \sigma \delta \sigma \delta \sigma \delta \]
- Formalize...
  \[ \delta(x) \equiv first(x) \lor \delta(p(p(x))) \]
  \[ \sigma \sigma \sigma \sigma \sigma \rightarrow \sigma \delta \sigma \delta \sigma \delta \]

Iterative syllabification

Arabic dialects: different epenthesis sites in CC* clusters (Ito, 1989)
- 3C: insert \( V \) after \( C_1 \) in Iraqi, and \( C_2 \) in Cairene.
- 4C: insert \( V \) after \( C_2 \).

Why?
- Iraqi syllabifies R-to-L, while Cairene L-to-R \( + V \) is added based on a CVC template.

 Iraqi
 (R-to-L) \(<katap-tl-u><katap-tl-ha>\)
 (L-to-R) \(<katap-tl-u><katap-tl-ha>\)

Cairene
 (R-to-L) \(<katap-tl-u><katap-tl-ha>\)
 (L-to-R) \(<katap-tl-u><katap-tl-ha>\)

QFLFP Characterization

- \( L'(x) \) and \( R'(x) \) determine L- and R-edges of \( \sigma \)'s before resyllabification.
- Resyllabification is only apparent in L-to-R.
- We only show R-to-L parsing

\[
L'(x) \equiv \begin{cases} 
C(x) \land V(s(x)) \lor \sigma & \text{select } C \text{ in } CV \\
C(x) \land C(s(x)) \land L(s(s(x))) & \text{select } C \text{ in } CC_C 
\end{cases}
\]

For example...

<table>
<thead>
<tr>
<th>Input</th>
<th>k</th>
<th>a</th>
<th>t</th>
<th>a</th>
<th>b</th>
<th>t</th>
<th>l</th>
<th>u</th>
</tr>
</thead>
<tbody>
<tr>
<td>( L' ) is true at...</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Iteration 0</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Iteration 1</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interim Output:</td>
<td>( i'(x_2) )</td>
<td>( k_0 )</td>
<td>( a )</td>
<td>( t_L )</td>
<td>( a )</td>
<td>( b_L )</td>
<td>( t_L )</td>
<td>( u )</td>
</tr>
<tr>
<td>Output:</td>
<td>( k )</td>
<td>( a )</td>
<td>( a )</td>
<td>( b )</td>
<td>( t )</td>
<td>( l )</td>
<td>( u )</td>
<td></td>
</tr>
</tbody>
</table>

Discussion

- Provides a testable hypothesis for iterative phonological functions based on computational power
- Highlights output orientation of iterative functions
- What about feet? What about more patterns?

Select References
Chandlee, Jane and Lindell, Steven (forthcoming). A logical characterization of strictly local functions. In Heinz, Jeffrey, editor, Computational Phonology, DOI

Acknowledgements Thanks to Facebook messenger for enabling collaboration.