Sour Grapes is phonotactically complex

Overview

- Sour Grapes is an unattested spreading/harmony process
- Heinz & Lai (2013) conjecture its non-existence reflects its computational complexity as a function
- I show that Sour Grapes is not an exceptional function, and
- I show that licit strings derived from Sour Grapes have exceptionally complex phonotactics

What is Sour Grapes?

Unattested pattern predicted in parallel OT with Agree
- A feature spreads completely through some domain, or,
- If complete harmony would be blocked, the feature does not spread at all (Wilson, 2003; 2006; McCarthy, 2010)

Running example of progressive nasal spreading

M is a trigger A is a blocker S is a blocker

- Spreading when no fricative is present satisfies Agree
- Initially: /MA\^w/ \rightarrow [MM\^M] Finally: /MA\^w/ \rightarrow [MM\^M]

<table>
<thead>
<tr>
<th>/mawam/</th>
<th>AGREE</th>
<th>IDENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. mawam W 2 L</td>
<td></td>
<td></td>
</tr>
<tr>
<td>b. máwam W 2 L 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>c. máwàm W 2 L 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>→ d. máwàm 3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Spreading over fricatives blocked by *NASALFRICATIVE
- IDENT prefers no spreading at all to partial spreading

<table>
<thead>
<tr>
<th>/mawasa/</th>
<th>*NASALFRICATIVE</th>
<th>AGREE</th>
<th>IDENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>→ a. mawasa 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>b. mawàsa 1 W 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>c. máwása 1 W 2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>d. máwàsà 1 W 3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>e. máwása W 1 1 W 4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>f. máwása L W 5</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Long strings may have multiple domains of spreading:

/SOURGRAPESAMSAMAAA/ \rightarrow [MMMMAMAMAMAM]

Sour Grapes is a weakly deterministic function

- Heinz & Lai (2013) prove Sour Grapes is not subsequential
- They conjecture that it is not weakly deterministic, and explain its non-existence in terms of functional complexity
- I show that Sour Grapes is in fact weakly deterministic
- Strategy similar to other unbounded circumbinant processes (O'Hara & Smith, 2018a,b; McCollum et al., 2018)

L-SFST: Spread locally, markup unbounded contexts with MAM

R-SFST: Spread only if MAM markup is present

Analysis confirmed using XFST (Beesley & Karttunen, 2003)
- Tested all strings [M, A, S]^19 in Python (1,743,392,199 total)
- Sour Grapes is not exceptionally complex as a function

Sour Grapes defines a non-counting language

- Sour Grapes is a function with an associated language
- Define language as those strings that map onto themselves
  ⇒ e.g., /MAAMASMAAA/ \rightarrow [MMMAMAMM]

- Set of phonotactically licit strings is first-order definable with precedence; belongs to the non-counting class of subregular languages (McNaughton & Papert, 1971):

∀ A : [¬∃M < A] \lor [∃M < A \land [∃S > M \land ∀M' : [M' > M \lor S < M']]]

- This is the least expressive class that models Sour Grapes

Copperbelt Bemba Turkana

High tone spreading in Copperbelt Bemba (Bickmore & Kula, 2013; Kula & Bickmore, 2015; Jardine, 2016a)
⇒ /báka-mu-londolol-a/ \rightarrow [bákámu-lóndololá]
⇒ /báka-londolol-a-kó/ \rightarrow [bákáloándólolákó]

Licit words are locally testable (Rogers & Pullum, 2011):

(S \land (∼H)) \lor (∼H \land (∼LHHL \land (∼L)L))

- Sour Grapes is not Autosegmental Strictly Local (Jardine, 2016b, 2017, 2018; Blum, 2018)
  ⇒ ASL languages ban a finite set of connected k-subgraphs
  ⇒ There is no way to capture the unbounded contexts

Copperbelt Bemba is Autosegmental Strictly Local