Logical Characterizations of Local and Long-distance Phonological Agreement

Jane Chandlee

Department of Computer Science
Haverford College

May 7, 2016
Characterizing phonological maps

- **Goal**: identify the most restrictive computational complexity class that is sufficiently expressive for the set of possible phonological maps.
- Known computational complexity classes have multiple, converging characterizations.
  - finite state automata
  - formal language theory
  - algebraic
  - logical
Phonological maps are **regular relations** (Johnson 1972, Kaplan & Kay 1994)


...but not all classes are currently well understood.
Subregular hierarchies

Formal Languages

- Regular
- Locally Threshold Testable
- Locally Testable
- Strictly Local
- Finite

String-to-string maps

- Regular Relations
- Subsequential Functions
- Strictly Local Functions

(Mohri 1997, Rogers & Pullum 2011, Rogers et al. 2013, Chandlee 2014)
Phonotactics

- Local phonotactics are **SL languages** (Heinz 2007, Heinz 2009)

  (1) In a language with final devoicing:
  
  a. Allowable substrings: \( \{ \times V, VV, V\times, \times D, \times T, \) DV, DD, TV, ... \}
  
  b. Prohibited substrings: \( \{ D\times \} \)

- Long-distance phonotactics are **SP** (Heinz 2010) or **TSL languages** (Heinz et al. 2011)

  (2) In a language with sibilant harmony:

  a. Allowable subsequences: \( \{ s...s, \int...\int, ... \} \)
  
  b. Prohibited subsequences: \( \{ s...\int, \int...s \} \)
Maps

- Local phonological maps are **SL functions** (Chandlee 2014, Chandlee et al. 2015)

\[(3) \quad D \times \mapsto T \times\]

- FST characterization
- Language-theoretic characterization

- **Hypothesis**: Long-distance maps are **SP functions**

\[(4) \quad s...\mathcal{I} \mapsto s...s\]
Why use logic?

- Goal is to build on what is known of SL functions to identify the class of SP functions.
- Logic provides a natural extension from local to long-distance.
Examples of agreement

• Adjacent

(5) Korean (Lee and Pater 2008)
   /papmul/ ↦ [pammul] ‘rice water’

• Non-adjacent but bounded

   a. /kunila/ ↦ [kunina] ‘sow for’
   b. /nikila/ ↦ [nikila] ‘season for’

• Unbounded

   /kudumukisila/ ↦ [kudumukisina] ‘to cause to jump for’
String graphs

\[ \text{pampm}u \leftrightarrow \text{pamm}u \]

\[ g \]

\[ \tau(g) \]
String graphs

\[ \text{papmul} \leftrightarrow \text{pammul} \]

- Set of nodes: \{0, 1, 2, 3, 4, 5\}
- Successor relation: \(S(0) = 1, S(1) = 2\), etc.
  (also written \(1 \bowtie 2\))
String graphs

\[ p \leftarrow p \mapsto m \mapsto u \mapsto l \]

\[ \tau(g) \]

- Labeling function: \( \ell(0) = p, \ell(1) = a, \text{etc.} \)
Natural class predicates

- \( V(x) = \text{TRUE} \) iff \( \ell(x) = a \lor \ell(x) = u \lor \ldots \)
- \( N(x) = \text{TRUE} \) iff \( \ell(x) = n \lor \ell(x) = m \lor \ldots \)
- \( C(x) = \text{TRUE} \) iff \( \neg N(x) \land \neg V(x) \)
Substring predicates

- $CN(x) = (\exists y) [C(x) \land N(y) \land x \triangleleft y]$
Graph transduction

- Predicates define the nodes and labels of the output graph in terms of the input graph (Engelfriet and Hoogeboom 2001)

\[ \varphi^0_N(x) = N(x) \lor CN(x) \]

\[
\begin{array}{cccccc}
 g & p & a & p & m & u & l \\
 0 & 1 & 2 & 3 & 4 & 5 \\

t(\tau(g)) & m & m & & & \\
 0 & 1 & 2 & 3 & 4 & 5 \\
\end{array}
\]
Graph transduction

$$\varphi^0_C(x) = C(x) \land \neg CN(x)$$
Graph transduction

\[ \varphi_0^V(x) = V(x) \]

\[ g \]

\[ \tau(g) \]
Non-adjacent but bounded

kunila $\mapsto$ kunina

$\text{NVL}(x) = (\exists y, z) \ [N(y) \land V(z) \land L(x) \land y \triangleleft z \land z \triangleleft x]$
Non-adjacent but bounded

\[ \varphi_0^N(x) = N(x) \lor NVL(x) \]
Non-adjacent but bounded

\[ \varphi^0_L(x) = L(x) \land \neg \text{NVL}(x) \]

\[ \varphi^0_V(x) = V(x) \]

\[ \varphi^0_D(x) = D(x) \]
Unbounded

\[ \text{tukunidi} \implies \text{tukunini} \]

\[ \text{NVL}(x) = (\exists y, z) \ [N(y) \land V(z) \land L(x) \land y \triangleleft z \land z \triangleleft x] \]
Unbounded

\[\text{tunikidi} \mapsto \text{tunikini}\]

\[N \longrightarrow L(x) = (\exists v, w, y, z) [N(y) \land L(x) \land y \triangleleft z \land z \triangleleft w \land w \triangleleft v \land v \triangleleft x]\]
kudumukisila $\leftrightarrow$ kudumukisina

$$N \rightarrow \rightarrow L(x) = (\exists s, t, u, v, w, y, z) \left[N(y) \land L(x) \land y \triangleleft z \land z \triangleleft w \land w \triangleleft v \land v \triangleleft u \land u \triangleleft t \land t \triangleleft s \land s \triangleleft x\right]$$
Instead of successor, **precedence**

- **Successor**: $x \triangleleft y$
  - $S(0) = 1$, $S(1) = 2$, ...
- **Precedence**: $x < y$
  - $P(1) = \{0\}$, $P(2) = \{0, 1\}$, ..., $P(10) = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
Instead of substring, **subsequence**

\[ k\text{udu}m\text{ukis}i\text{a} \leftrightarrow k\text{udu}m\text{ukisi}n\text{a} \]

\[ NL(x) = (\exists y) [N(y) \land L(x) \land y < x] \]
## Categories

<table>
<thead>
<tr>
<th>Local</th>
<th>Long-distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adjacent</td>
<td>Non-adjacent</td>
</tr>
<tr>
<td>Bounded</td>
<td>Unbounded</td>
</tr>
<tr>
<td>SL</td>
<td>SP</td>
</tr>
</tbody>
</table>
More examples of bounded LDA

(8) **Transvocalic**: Anywa, Basaa, Bemba, Bukusu, Dime, Gaagudju, Herero, Ila, Kipare, Koyra, Kwanyama, Lamba, Luba, Mayali, Mwiini, Nyangumarta, Pangwa, Pare, Pende, Punu, Ruund, Sawai, Shambaa, Sudanese, Suku, Tonga, Yabem, Zayse, ...

(9) **Co-occurrence restrictions within roots or morphemes**: Adhola, Alur, Anywa, Basque, Chaha, Chol, Ganda, Hausa, Ineseno Chumash, Izere, Kalasha, Karaim, Komi-Permyak, Kukuya, Luo, Malto, Ndebele, Ngizim, Pengo, Pohnpeian, Shilluk, Siwi, Teke-Gabon, Tiene, Yucatec Mayan, Zulu, ...

(see Shaw (1991), Odden (1994), Hansson (2001), Rose & Walker (2004), and references therein)
Domain restrictions

(10) Malto (Mahapatra 1979, Hansson 2001): \{t, d\} and \{t, d\} cannot co-occur in tautomorphemic CVC

- SL function that targets substrings of length 3?
  \[
  \begin{align*}
  t\text{a}t &\mapsto t\text{a}t \\
  t\text{a}t &\mapsto t\text{a}t \\
  d\text{a}d &\mapsto d\text{a}d \\
  d\text{a}d &\mapsto d\text{a}d
  \end{align*}
  \]

- Or SP function that targets subsequences of length 2 that will never ‘see’ inputs longer than 3?
  \[
  \begin{align*}
  t\text{a} \mapsto t\text{a} \\
  d\text{a} \mapsto d\text{a} \\
  t\text{a}t &\mapsto t\text{a}t \\
  d\text{a}d &\mapsto d\text{a}d
  \end{align*}
  \]
Subregular hierarchies

Formal Languages

- Regular
- Non-Counting
- Locally Threshold Testable
- Locally Testable
- Piecewise Testable
- Strictly Local
- Strictly Piecewise
- Finite

String-to-string maps

- Regular Relations
- Subsequential Functions
- Strictly Local Functions
Subregular hierarchy of languages

(Rogers & Pullum 2011, Rogers et al. 2013)
Subregular hierarchy of functions

- **Successor**
  - Regular
  - Strictly Local

- **Precedence**
  - Regular
  - Strictly Piecewise

- \(\approx\) **Monadic Second Order**

- **First Order**
First Order Logic

- $\forall, \exists$
- $\land, \lor, \lnot$
- $\rightarrow, \leftrightarrow$
- $x, y, z, ...$
- $P(x), R(x, y), ...$
‘Sub’-First Order Logic

Proposal: SL functions are quantifier-free, unary, monotone graph transductions (Chandlee & Lindell, in prep)

- $\forall, \exists$
- $\land, \lor, \neg$
- $\rightarrow, \leftrightarrow$
- $x, y, z, \ldots$
- $P(x), R(x, y), \ldots$
Next steps

• Extend to dissimilation maps
• Logical and FST characterizations of SP functions
• Interactions of SL and SP maps?
Conclusions

- Characterizing computational complexity classes from a variety of perspectives leads to a more complete understanding of what the class is.
- Identifying the right class (or combination of classes) for phonological maps likewise leads to insights into what phonology is (and isn’t).
- From this perspective, the distinction between local and long-distance phonology is one of representation, not computational complexity.
Acknowledgments

For ongoing collaboration and inspiration, thanks to:

- Jeffrey Heinz (University of Delaware)
- Adam Jardine (University of Delaware)
- Rémi Eyraud (Aix-Marseilles University)
- Jim Rogers (Earlham College)
- Steven Lindell (Haverford College)
References

References

SP FSTs?

(11) Benchnon sibilant harmony (Hayward 1988, Hansson 2001)

a. /s\textipa{\textipa{i}ap-s}/ \mapsto [s\textipa{i}aps] ‘make wet’
b. /\textipa{sir}-s/ \mapsto [\textipa{sir}\textipa{s}] ‘bring near’

\[ \begin{align*}
V:V, s:s, C:C \quad & C:C, V:V, \int:\int, s:\int \\
0 \quad & \rightarrow \quad 1
\end{align*} \]