Investigating phonological typology: a computational approach

Jane Chandlee

Department of Computer Science
Haverford College

February 17, 2016
Collaborators

- Jeffrey Heinz, University of Delaware
- Adam Jardine, University of Delaware
- Rémi Eyraud, Laboratoire d’Informatique Fondamentale
- Jim Rogers, Earlham College
Objectives

What factors delimit the set of possible phonological mappings?

The approach:

- Characterize phonological \( \text{UR} \rightarrow \text{SR} \) maps using methodology from theoretical computer science.
- Establish the computational complexity class these maps belong to (i.e., what is the upper bound?).
More specifically, many phonological maps are shown to belong to a **subregular** class of functions called the Input Strictly Local functions.

These functions are computationally restrictive, which in part accounts for why certain logically possible maps that do not belong to the class are also unattested.

Belonging to this class also has an advantage when it comes to learnability.
Phonological functions

underlying form $\mapsto$ surface form

(1) German final devoicing
/bad/ $\mapsto$ [bat], ‘bath’
Why functions?

(2) \([–son] \rightarrow [–voice] / \_\_\_ \#\)

(3) *D# \gg IDENT(Voice)

(Baković 2013)
Chomsky hierarchy

Finite \quad \text{Regular} \quad \text{Context-Free} \quad \text{Context-Sensitive} \\
Computationally Enumerable
Restricting phonology

- Phonological rules of the form $A \rightarrow B/C \rightarrow D$ are regular relations provided the rule doesn’t re-apply to its own structural change. (Johnson 1972, Kaplan & Kay 1994)
Restricting phonology further

Phonological maps \( CAD \mapsto CBD \) are **subregular** provided \( CAD \) is a finite set. (Chandlee 2014, Chandlee et al. 2014)

(4) Final devoicing
\[
D\# \mapsto T\#
\]
Subregular hierarchy of languages

Successor
- Regular

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

Monadic
- Second Order
  - First Order
    - Propositional
      - Conjunctions of Negative Literals

Jane Chandlee
Investigating Phonological Typology
Subregular hierarchy of languages

- As a hierarchy of formal languages (i.e., stringsets), the subregular hierarchy has been used to study phonotactics (Heinz 2009, 2010, Heinz et al. 2011, Rogers et al. 2013).
- Phonotactic constraints appear to be restricted to the SL and SP regions.
- What about phonological maps?
Subregular hierarchy of functions

Successor       Precedence

Regular

Input Strictly Local

Jane Chandlee
Investigating Phonological Typology
(5) Quechua post-nasal obstruent voicing
a. /kampa/ $\mapsto$ [kamba] ‘yours’

(Pater 2004)
ISL function: Example 1

# k a m p a #
ISL function: Example 1

# k a m p a #

#
ISL function: Example 1

# k a m p a #

k a
ISL function: Example 1

# k a m p a #

k a m
ISL function: Example 1

# k a m p a #
k a m b
ISL function: Example 1

# k a m p a #

k a m b a
The ‘window’ size is the length of the targeted sequence: e.g., the length of $NC_\emptyset$.

This length is the $k$-value of the function: Post-nasal obstruent voicing is 2-ISL.
(6) Rotuman \((CV\# \mapsto VC\#)\)

\begin{align*}
\text{a. } & \text{hosa } \mapsto \text{ hoas} & \text{‘flower’} \\
\text{b. } & \text{hula } \mapsto \text{ hual} & \text{‘moon’} \\
\text{c. } & \text{tiko } \mapsto \text{ tiok} & \text{‘flesh’}
\end{align*}

(Churchwood 1940)
ISL function: Example 2

# h u l a #
ISL function: Example 2

# h u l a #
ISL function: Example 2

# h u l a #

hu
ISL function: Example 2

# h u l a #

hu
ISL function: Example 2

# h u l a #

hu al
Computing Phonology
ISL functions
Long-distance phonology
Logical characterizations

ISL functions: automata characterization

Jane Chandlee
Investigating Phonological Typology
Jane Chandlee
Investigating Phonological Typology
ISL functions: automata characterization

Jane Chandlee

Investigating Phonological Typology
Computing Phonology
ISL functions
Long-distance phonology
Logical characterizations

ISL functions: automata characterization

Jane Chandlee
Investigating Phonological Typology
ISL functions: automata characterization
ISL functions: automata characterization

k a m p a
k a m b a
FSTs are primarily for **classification**, but they are also useful for proving the class is learnable.

Two provably correct algorithms for learning ISL functions:

1. ISLFLA (Chandlee et al. 2014)
2. SOSFIA (Jardine et al. 2014)
Given this data: \{(kampa, kamba), (kamka, kamga), (kam, kam)\} ...

How can we learn the function (i.e., the FST)?
Computing Phonology
ISL functions
Long-distance phonology
Logical characterizations

Jane Chandlee
Investigating Phonological Typology
Inductive principle: if the target function is ISL-2, then all that matters is the previous input symbol.
Computing Phonology
ISL functions
Long-distance phonology
Logical characterizations

ISLFLA

Jane Chandlee
Investigating Phonological Typology
How much of phonology is ISL?

- A review of P-Base (v1.95, Mielke 2008), which includes approximately 5500 patterns from 500 languages, revealed that 95% are ISL functions (Chandlee 2014, Chandlee & Heinz to appear).

- This includes local substitution, plus (all?) deletion, (all?) epenthesis, and all synchronic metathesis.
Non-ISL maps

(7) Sarcee
/naʃyatʃ/ \(\mapsto\) [naʃyatʃ] ‘I killed them again’

(8) Kikongo
/tu-nik-idi/ \(\mapsto\) [tunikini] ‘we ground’

(Cook 1984, Bennett 2013, Rose & Walker 2004)
Non-ISL maps

- Long-distance substitution corresponds to markedness constraints against non-contiguous sequences:

  (9)  
  a. *s...ʃ  
  b. *n...d

- Work on long-distance phonotactics indicates that these patterns are still computationally restricted when locality is interpreted as *precedence* (Heinz 2010).
Subregular hierarchy of languages

Successor
- Regular
  - Locally Threshold Testable
    - Locally Testable
      - Strictly Local
    - Piecewise Testable
      - Strictly Piecewise
  - Non-Counting
    - Locally Threshold Testable
  - Proprietary
    - Conjunctions of Negative Literals

Precedence
- First Order
  - Monadic Second Order

Finite

Jane Chandlee
Investigating Phonological Typology
Subregular hierarchy of functions

Successor

Precedence

Regular

Input Strictly Local

Strictly Piecewise
SP functions?

\[ \lambda \]

\[ T: T, V: V, D: D \quad T: T, V: V, N: N, D: N \]

\[ N: N \quad N \]
Computing Phonology
ISL functions
Long-distance phonology
Logical characterizations

Subregular hierarchy of languages

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

Precedence
- Non-Counting
  - Piecewise Testable
    - Strictly Piecewise

Monadic
  - Second Order
    - First Order
      - Propositional
        - Conjunctions of Negative Literals

Investigating Phonological Typology

Jane Chandlee
Logical characterization of ISL

\[ g \]
\[ \begin{array}{c}
V \\
0
\end{array} \rightarrow \begin{array}{c}
N \\
1
\end{array} \rightarrow \begin{array}{c}
C \\
2
\end{array} \rightarrow \begin{array}{c}
V \\
3
\end{array} \]

\[ \tau(g) \]
\[ \begin{array}{c}
V \\
0
\end{array} \rightarrow \begin{array}{c}
N \\
1
\end{array} \rightarrow \begin{array}{c}
N \\
2
\end{array} \rightarrow \begin{array}{c}
V \\
3
\end{array} \]

\[ a. \quad NC(x) = (\exists y)[N(y) \land C(x) \land y \triangleleft x] \]

\[ b. \quad \varphi_0^N(x) = N(x) \lor NC(x) \]

\[ c. \quad \varphi_0^C(x) = C(x) \land \neg NC(x) \]

\[ d. \quad \varphi_0^V(x) = true \]
a. \( NC(x) = (\exists y)[N(y) \land C(x) \land y \prec x] \)

b. \( \varphi^0_N(x) = N(x) \lor NC(x) \)

c. \( \varphi^0_C(x) = C(x) \land \neg NC(x) \)

d. \( \varphi^0_V(x) = true \)
Logical characterization

Question: what logic is sufficiently expressive for ISL and SP functions?
Subregular hierarchy of functions

- **Successor**
  - Regular
  - Input Strictly Local
- **Precedence**
  - Regular
  - Strictly Piecewise
- **Monadic Second Order**
- **First Order**

Jane Chandlee
Investigating Phonological Typology
Logical characterizations

- The distinction between local and long-distance phonological maps is in the representation, not the computational complexity.

- Beyond strings, the same types of logical formulae can also be defined for other representations, such as autosegmental representations (Jardine 2016) and metrical trees.
Conclusions

- Locality has been an implicit guideline for phonological formalisms (rules and constraints), but it is also a defining property of the phonological mappings themselves.

- Identifying a restrictive computational complexity class for phonology has implications for both typology and learning.

- The use of logical characterizations in phonology provides a unified analytical framework for studying the computational nature of a range of phenomena.
Thank you!
References

References


References


