The Role of Strict Locality in Computing Phonology: Implications for Typology and Learning

Jane Chandlee

University of Delaware  |  Nemours Biomedical Research

April 15, 2015
Objectives

- Present some basics of phonology and introduce a learning problem.
- Provide a computational solution to this problem.
- Discuss some of the larger implications for natural language phonology.
Phonological Processes

- English plural suffix: /z/
  - shoes [ʃuz]
  - lids [lɪdz]
- Sometimes pronounced [s]
  - hats [hæts]
How should we describe this fact of English?

- Option 1: English has more than one plural.
- Option 2: English has one plural and a process for determining how it is pronounced.
Option 2: English has one plural and a process for determining how it is pronounced in any given word.

Generative linguistics argues that the best explanation for systematic variation in pronunciations is to posit an abstract, underlying form (e.g., /z/) and a **transformation** to a surface, pronounced form (e.g., /z/ → [s]).
Phonological Processes

hats \ [hæts] 
lids \ [lidz] 

Generalization:

- When the preceding sound is voiced, /z/ \to [z].
- When the preceding sound is voiceless, /z/ \to [s].
Phonological Processes

- A process can be represented with a function:

  (1) a. \( f(\text{l\text{idz}}) = \text{l\text{idz}} \)
  b. \( f(\text{h\text{ætz}}) = \text{h\text{æts}} \)
  c. \( f(\text{f\text{uz}}) = \text{f\text{uz}} \)

- This function is infinite and will be defined for all possible words, including those the speaker has no prior experience with.
Phonological Functions

- Functions can be grouped into classes that are distinguished by their computational complexity.
- How can we define the class of *phonological* functions?
Chomsky Hierarchy

- Recursively enumerable
- Context-sensitive
- Context-free
- Regular
- Finite
Phonological transformations are regular relations (Johnson 1972, Koskenniemi 1983, Kaplan & Kay 1994).

This means they can be represented with Finite State Transducers (FSTs).
Regular Relations

Jane Chandlee

Strict Locality and Phonology
(2) $f(\text{datz}) = \text{dats}$

- Does the FST accept the pair (datz, dats)?
Regular Relations

Phonology
(Sub)Regular Relations
Learning Technique
Contributions

Jane Chandlee
Strict Locality and Phonology
Regular Relations

V:V, d:d, z:z

1

V:V, d:d

t:t

2

V:V, d:d, z:z

t:t

3

z:s

t:t

4

z:z

t:t

V:V, d:d, z:z

d a t z

d a t s

Jane Chandlee
Strict Locality and Phonology
Regular Relations

1. V:V, d:d, z:z
2. t:t
3. z:z
4. V:V, d:d, z:z

Diagram:

- Node 1: V:V, d:d, z:z
- Node 2: t:t
- Node 3: V:V, d:d, z:z
- Node 4: z:s

Connections:
- 1 to 2: t:t
- 2 to 3: z:z
- 3 to 4: z:s

Symbols:
- d
- a
- t
- z
- s

Jane Chandlee
Strict Locality and Phonology
Regular Relations

V:V, d:d, z:z

1

V:V, d:d

2

t:t

3

V:V, d:d, z:z

t:t

4

z:s

z:z

t:t

\textbf{d a t z}

\textbf{d a t s}

Jane Chandlee

Strict Locality and Phonology
Regular Relations

1. V:V, d:d, z:z
2. t:t
3. V:V, d:d
4. t:t
5. z:s
6. z:z

V:V, d:d, z:z

d a t z
d a t s
Regular Relations

1. V:V, d:d, z:z
2. t:t
3. V:V, d:d
4. z:s

1
V:V, d:d, z:z 2t:t
V:V, d:d 3z:z
V:V, d:d, z:z 4t:t

Jane Chandlee
Strict Locality and Phonology
Phonological functions are regular relations, but...

...two indications that we can ‘do better’:

1. The class of regular relations cannot be learned from positive data in the sense of (Gold 1967).
2. Predicts the existence of processes that are intuitively ‘odd’ based on what we know about phonology.
Non-phonological regular relations

(3) ‘odd’ phonological function: 
\( z \mapsto s \), but only at the end of a word that contains an odd number of t’s

- Still a regular relation!
Non-phonological regular relations

V:V, d:d, z:z

1

V:V, d:d

2 t:t

V:V, d:d, z:z, t:t

3

V:V, d:d, z:z

4 z:s

z:z

Jane Chandlee

Strict Locality and Phonology
Non-phonological regular relations

Jane Chandlee

Strict Locality and Phonology
Non-phonological regular relations

\[
\begin{align*}
V:V,d:d,z:z & \quad 2 \quad t:t \\
V:V,d:d & \quad 4 \quad z:s \\
V:V,d:d,z:z,t:t & \quad 3 \quad z:z \\
V:V,d:d & \quad 2 \\
& \quad 1 \\
\end{align*}
\]
Non-phonological regular relations

1. $V:V, d:d, z:z$
2. $t:t, d:d$
3. $z:z$
4. $V:V, d:d, z:z, t:t$

Diagram:

- Node 1: $V:V, d:d, z:z$
- Node 2: $t:t, d:d$
- Node 3: $z:z$
- Node 4: $V:V, d:d, z:z, t:t$

Symbols:
- $d$
- $a$
- $t$
- $z$
- $s$
Non-phonological regular relations

1. V:V, d:d, z:z
2. t:t
3. V:V, d:d
4. z:s

\begin{align*}
V:V, d:d, z:z, t:t
\end{align*}
Non-phonological regular relations
Non-phonological regular relations

\[ V:V, d:d, z:z \]
\[ t:t \]
\[ t:t \]
\[ V:V, d:d \]
\[ z:s \]
\[ z:z \]
\[ V:V, d:d, z:z, t:t \]
Non-phonological regular relations

V:V,d:d,z:z
2
V:V,d:d
4
z:s
z:z
V:V,d:d,z:z,t:t
Phonological functions are properly *subregular*. 
What subregular class corresponds to phonological functions?

How do humans learn these functions from positive data?
Questions

1. What subregular class corresponds to phonological functions?
2. How do humans learn these functions from positive data?
State Merging

\[
\begin{align*}
0 & \xrightarrow{a} 1 \\
1 & \xrightarrow{a} 2 \\
0 & \xrightarrow{a} 1,2
\end{align*}
\]

\[
\{aa\} \quad aa^*\]

Demonstration: German final devoicing

(4)  
   a. /bad/ $\mapsto$ [bat], ‘bath’
   b. /sag/ $\mapsto$ [sak], ‘say’
The Learning Problem

Given this data: \{(dad, dat), (dat, dat), (daa, daa)\}

How can we learn the function (i.e., the FST)?
State Merging

\[
\begin{align*}
\lambda & \xrightarrow{d:a} d & \xrightarrow{a:\lambda} da \\
d & \xrightarrow{d:t} da \\
& \xrightarrow{a:\lambda} daa \\
& \xrightarrow{t:t} dat \\
& \xrightarrow{#:\lambda} dad \\
& \xrightarrow{#:\lambda} daa \\
& \xrightarrow{#:\lambda} dat
\end{align*}
\]
What states should be merged?
State Merging

What states should be merged?

Hypothesis: phonological computations require only ‘local’ information.
What states should be merged?

Hypothesis: phonological computations require only ‘local’ information.

How can we quantify ‘local’?
The Strictly Local (SL) Languages are a subregular class of formal languages.

The constraints on the strings in a SL-\(k\) language are expressible as substrings of length \(k\).

(5) German is SL-2, since \(|d\#| = 2\).

A SL-\(k\) language can be learned by merging states that have the same suffix of length \(k - 1\) (Heinz 2009).
Strictly Local State Merging

\[ \lambda \rightarrow d:da \rightarrow d \rightarrow a:\lambda \rightarrow da \rightarrow d:t \rightarrow daa \rightarrow #:\lambda \rightarrow dat \rightarrow #:\lambda \rightarrow \]

Jane Chandlee

Strict Locality and Phonology
Strictly Local State Merging

\[
\lambda \xrightarrow{d:d\alpha} d \xrightarrow{\alpha:\lambda} a \xrightarrow{\alpha:\lambda} a \xrightarrow{\#:\lambda} d \xrightarrow{d:t} a \xrightarrow{t:t} t \xrightarrow{\#:\lambda} #:
\]

Jane Chandlee  Strict Locality and Phonology
Strictly Local State Merging

Jane Chandlee

Strict Locality and Phonology
Strictly Local State Merging

\[ \lambda \]

\[ d \text{ : } \lambda \]

\[ d : d \]

\[ t : dt \]

\[ t : t \]

\[ \lambda \text{ : } \lambda \]

\[ # : \lambda \]

\[ \alpha : \alpha \]

\[ d : \lambda \]

\[ a : da \]

\[ a : a \]

\[ # : \lambda \]

\[ \alpha : \alpha \]

\[ d : \lambda \]

\[ a : a \]

\[ # : \lambda \]
What kind of function is this (i.e., what can we learn this way)?
Definition

A function is Input Strictly $k$-Local if the output of every input string $a_0a_1\cdots a_n$ is $u_0u_1\cdots u_n$ where $a_i \in \Sigma$ and $u_i$ is a string which only depends on $a_i$ and the $k-1$ symbols before $a_i$ in the input string.

(Chandlee 2014, Chandlee & Heinz to appear)
ISL function learning algorithm (ISLFLA)

- ISLFLA can learn all and only $k$-ISL functions, in quadratic time and data (Chandlee et al. 2014).
- ISL is a *proper* subclass of regular relations, which rules out many non-phonological relations.

(Chandlee et al. 2014)
Consider an alternative version of German final devoicing:
- Devoicing final /d/ only if the word contains an even number of d’s.
This FST cannot distinguish /dad/ (even number of d’s) from /dadad/ (odd number of d’s).
Phonological typology

How many phonological processes are ISL functions?
A review of the phonological pattern database P-Base (v1.95, Mielke 2008) revealed that 95% are ISL.

This includes substitution processes like voicing assimilation and final devoicing, as well as insertion, deletion, and metathesis.
Phonological typology

(6) Dutch epenthesis ($k = 2$)
a. mɛlk $\mapsto$ mɛlɛk, ‘milk’

(7) Tagalog deletion ($k = 4$)
a. bukasin $\mapsto$ buksin, ‘was opened’

(8) Rotuman metathesis ($k = 3$)
a. hosa $\mapsto$ hoas, ‘flower’
Non-ISL phonological processes

What phonological processes aren’t ISL functions?
Two main categories:

1. Iterative processes:

   (9) Johore Malay nasal spreading
   a. \( \text{pəŋawasan} \rightarrow \text{pəŋ\text{\text}ã\text{\text}ã\text{\text}âsan} \), ‘supervision’

2. Long-distance processes:

   (10) Kikongo nasal assimilation
   a. \( \text{tunikidi} \rightarrow \text{tunikini} \), ‘we ground’

(Onn 1980, Rose & Walker 2004)
What processes aren’t ISL functions?

- Iterative processes are Output Strictly Local (OSL) functions (Chandlee et al. 2015).
- Long-distance processes are likely still subregular, but require a different notion of locality (Heinz 2010).
Subregular hierarchy of languages

- Regular
- Star-Free
- TSL
- LTT
- PT
- LT
- SP
- SL
ISL functions are based on SL languages and correspond to SL constraints: *tz#, *CV#, *lk, etc.

Long-distance processes correspond to constraints like *n...d, which are SP constraints.
Beyond phonology

The ISL subclass may lead to more efficient algorithms for other string-to-string transductions:

- **Grapheme-to-Phoneme conversion**
  
  (11) knight $\mapsto [n\alpha i t]$

- **Pronunciation variation in speech recognition and synthesis**
  
  (12) ‘bouquet’
  a. boukei $\mapsto$ buwkei
The ISL functions formalize the role of locality in the input-output transformations allowed in phonology. These functions are expressive enough to model a significant range of phonological processes and restrictive enough to be efficiently learned from positive data. Current and future work is focused on addressing certain types of non-ISL phonological processes and exploring HLT applications of the ISL class.
Selected references

Selected references