Using Output Strict Locality to Model and Learn Long-distance Processes

Jane Chandlee

University of Delaware & Nemours Biomedical Research

November 15, 2014

Main objectives

- Define *Output Strict Locality*, a strong computational property that holds of the mapping from UR to SR.
- Show how the output-oriented nature of this property is needed to model spreading processes.
- Analyze various long-distance processes with a component of OSL spreading.

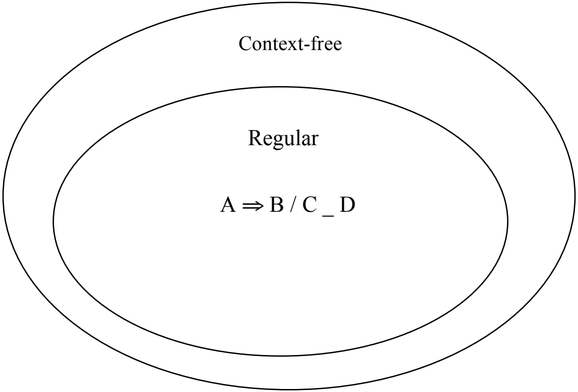
Computational properties

- Phonological processes are *maps* (i.e., functions) from an input (UR) to an output (SR).
- The nature of these maps helps us characterize possible phonological processes.

  (1) Tesar (2008, 2012, 2014) characterizes certain maps as *output-driven*.

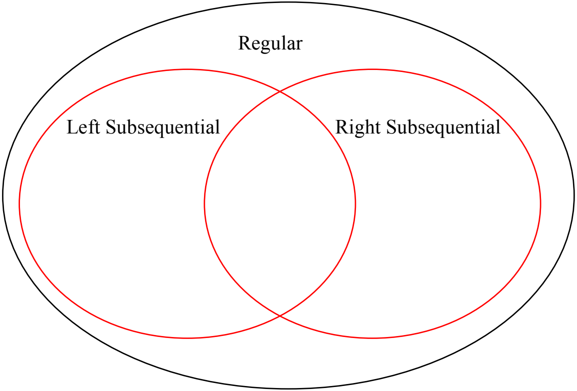
- Also aids in learning, when the learner can use the computational property as an inductive principle or bias.

Phonological maps are REGULAR



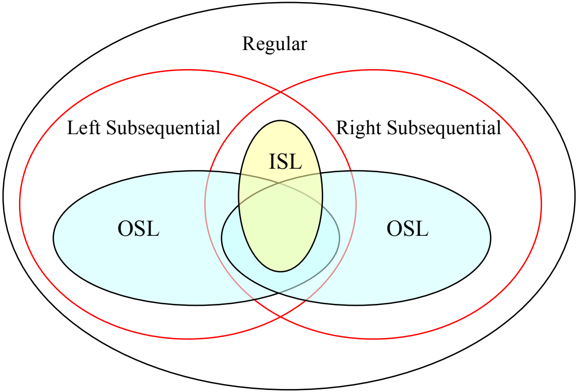
Johnson (1972); Kaplan and Kay (1994); Koskenniemi (1983)

Phonological maps are SUBSEQUENTIAL



Mohri (1997); Chandlee et al. (2012); Gainor et al. (2012); Heinz and Lai (2013); Jardine (2013); Luo (2013); Payne (2013); Chandlee and Heinz (2012)

Phonological maps are STRICTLY LOCAL



Chandlee (2014); Chandlee et al. (2014)

Tails

For any function $f : \Sigma^* \rightarrow \Gamma^*$ and $x \in \Sigma^*$, let the *tails* of x with respect to f be defined as

$$\texttt{tails}_f(x) = \left\{(y, v) \in \Sigma^* \times \Gamma^* \mid f(xy) = uv \wedge u = \texttt{lcp}(f(x\Sigma^*))\right\} .$$

Tails

Post-nasal obstruent voicing (Quechua, Orr (1962); Rice (1993); Pater (2010))

(2) f(kampa) = kamba

(3) $\texttt{tails}_f$(kam) = {(p,b),(m,m),(a,a),(k,k),(d,t)...}

Input Strict Locality

Definition (Input Strictly Local Function)

A function f is Input Strictly Local (ISL) if there is a k such that for all $u_1, u_2 \in \Sigma^*$, if $\texttt{Suff}^{k-1}(u_1) = \texttt{Suff}^{k-1}(u_2)$ then $\texttt{tails}_f(u_1) = \texttt{tails}_f(u_2)$.

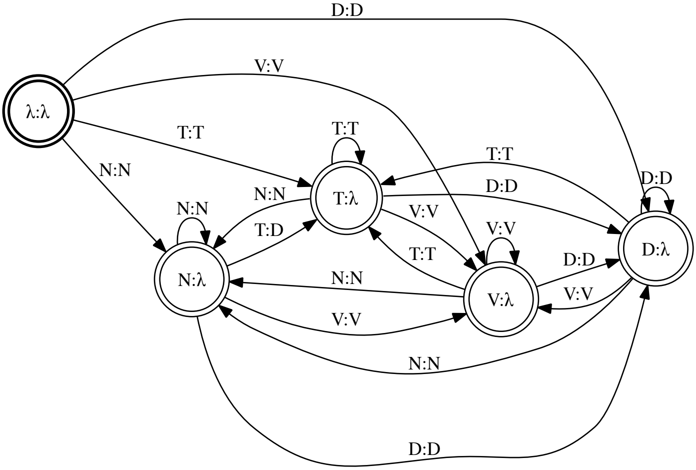
Input Strict Locality

Definition (Input Strictly Local Function)

A function f is Input Strictly Local (ISL) if there is a k such that for all $u_1, u_2 \in \Sigma^*$, if $\texttt{Suff}^{k-1}(u_1) = \texttt{Suff}^{k-1}(u_2)$ then $\texttt{tails}_f(u_1) = \texttt{tails}_f(u_2)$.

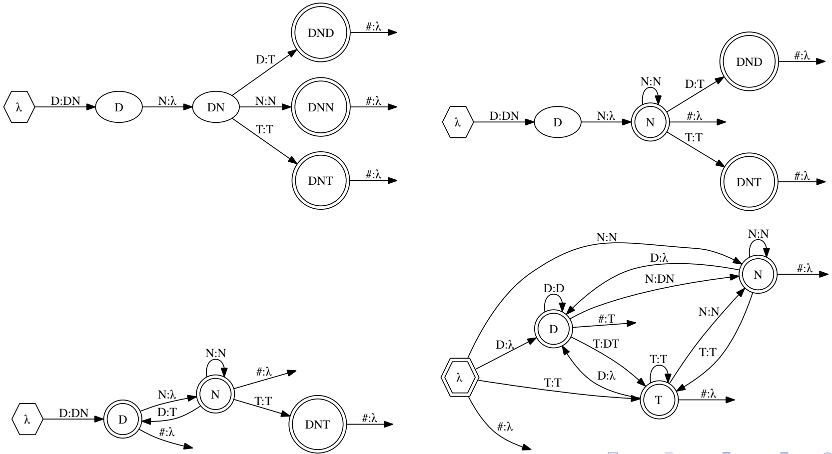
(4) $\texttt{tails}_f(\text{kam}) = \texttt{tails}_f(\text{tam}) = \texttt{tails}_f(\text{kamam}) = \texttt{tails}_f(\text{kamamam}) = \{(\text{p,b}),(\text{m,m}),(\text{a,a}),(\text{k,k}),(\text{d,t})...\}$

ISL FST characterization



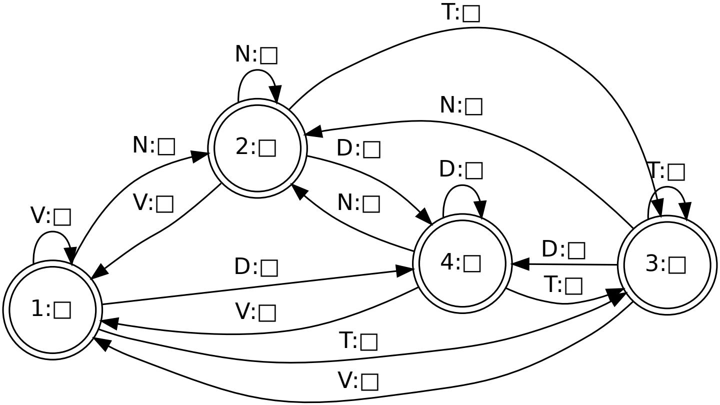
Learning ISL

ISLFLA: state merging algorithm (Chandlee 2014; Chandlee et al. 2014)



Learning ISL

SOSFIA (Jardine et al. 2014)



Empirical coverage of ISL

(5) A $\rightarrow$ B / C __ D

- If CAD is a finite language and the rule applies simultaneously, the mapping is ISL (see Kaplan and Kay, 1994; Hulden, 2009; Chandlee, 2014).
- This includes epenthesis, deletion, local substitution, metathesis, local partial reduplication, general affixation (approx. 94% of the processes in P-Base (v.1.95, Mielke (2004))).

Non-ISL processes

Because ISL functions only pay attention to the input, the trigger of the process must be present underlyingly.

Nasal spreading

(6) Johore Malay (Onn 1980)

a. [-nasal] $\rightarrow$ [+nasal] / [+nasal] __

b. pəŋawasan $\mapsto$ pəŋãw̃ãsan, ‘supervision’

Nasal spreading

(7) f(pəŋawasan) = pəŋãw̃ãsan

(8) a. pəŋaw ↦ pəŋãw̃

b. paw ↦ paw

(9) a. $\mathtt{tails}_f$(pəŋaw) = {(p,p),(a,ã),(w,w̃),...}

b. $\mathtt{tails}_f$(paw) = {(p,p),(a,a),(w,w), ...}

Output Strict Locality

Definition (Output Strictly Local Function)

A function f is Output Strictly Local (OSL) if there is a k such that for all $u_1, u_2 \in \Sigma^*$, if $\texttt{Suff}^{k-1}(f(u_1)) = \texttt{Suff}^{k-1}(f(u_2))$ then $\texttt{tails}_f(u_1) = \texttt{tails}_f(u_2)$.

Output Strict Locality

(10) a. pəŋa ↦ pəŋã
b. pəŋawa ↦ pəŋãw̃ã
c. papa ↦ papa

(11) $\texttt{tails}_f$(pəŋa) = $\texttt{tails}_f$(pəŋawa) = {(p,p),(a,ã),(w,w̃),...}

(12) $\texttt{tails}_f$(papa) = {(p,p),(a,a),(w,w),...}

Output Strict Locality

$$\text{OSL} = \text{transitive closure of ISL} \circ \text{ISL}$$

Output Strict Locality

- Conjecture: the ISL functions of interest to phonology are also OSL.

(13) Final devoicing: bad $\mapsto$ bat

(14) Place assimilation: inpossible $\mapsto$ impossible

- Some processes are *only* OSL.

Schwa deletion

(15) French (Dell 1973, 1980, Noske 1993)

a. $ə \rightarrow \emptyset$ / VC__CV

(16) tu devenais, ‘you became’

a. ty dəvənɛ $\mapsto$ ty dvənɛ

b. ty dəvənɛ $\mapsto$ ty dəvnɛ

c. ty dəvənɛ $\mapsto$ *ty dvnɛ

Long Distance Processes

Spreading: trigger arbitrarily far from target, but iteration makes the process local.

Long Distance Processes

Harmony: spreading but with certain segments skipped.

Harmony

Vowel harmony without transparent vowels.

(17) $\mathsf{V} \rightarrow [+\alpha] \; / \; \mathsf{V}_{+\alpha}\mathsf{C}_0^{k-2}\_\_$

k = length of maximum coda + maximum onset + 2

Harmony

Vowel harmony *with* transparent vowels.

There is no k such that the target and trigger vowel will form a contiguous substring bounded by k.

Harmony

Vowel harmony with transparent vowels:

1) spread harmonizing feature to all vowels ✓ OSL

2) remove feature from transparent vowels ✓ OSL

(Clements 1977; Vago 1980)

Vowel transparency

(18) Khalkha Mongolian rounding harmony, [i] transparent (Svantesson et al. 2005; Gafos and Dye 2011)

a. poor-ig-o, ‘kidney.ACC.RFL’

b. teeʒ-ig-ɔ, ‘food.ACC.RFL’

(1) V $\rightarrow$ [+round] / $V_{[+round]}$ C_0^{k-2} __
(2) y $\rightarrow$ i

poor-ig-O $\mapsto_1$ poor-yg-o $\mapsto_2$ poor-ig-o

LD assimilation

(19) Kikongo nasal assimilation (Rose and Walker 2004)

a. /tu-kun-idi/ $\mapsto$ [tukunini] 'we planted'

b. /tu-nik-idi/ $\mapsto$ [tunikini] 'we ground'

(1) [+voice] $\rightarrow$ [+nasal] / [+voice] T_0^{k-1} __

(2) [-cons] $\rightarrow$ [-nasal]

$$\text{tu-nik-idi} \mapsto_1 \text{tu-n}\tilde{\text{i}}\text{k-}\tilde{\text{i}}\text{n}\tilde{\text{i}} \mapsto_2 \text{tu-nik-ini}$$

LD displacement

(20) Colville (Mattina 1979)

a. pˤáw 'he ran down'

b. pw-ən-cˤát-əlx 'they make noise running down'

pˤáw-ən-cat-əlx ↦ pˤaˤwˤ-əˤnˤ-cˤát-əlx ↦ paw-ən-cˤát-əlx

OSL composition

OSL functions are not closed under composition!

OSL composition

Theorem (Elgot and Mezei 1965)

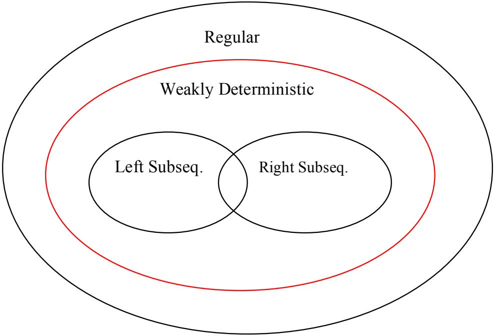
A function $f : X^* \mapsto Y^*$ is a regular relation iff there exists a left subsequential function $g : X^* \mapsto Z^*$ and a right subsequential function $h : Z^* \mapsto Y^*$, with $X \subseteq Z$, such that $f = g \circ h$.

Weakly Deterministic Function

Definition (Weakly Deterministic Function, (Heinz and Lai 2013))

A regular function f is weakly deterministic iff there exists a left subsequential function $g : X^* \mapsto X^*$ and a right subsequential function $h : X^* \mapsto X^*$ such that g is not length-increasing and $f = g \circ h$.

Weakly Deterministic Function



(Heinz and Lai 2013)

How much markup?

Will constraints on markup likewise restrict the power of $OSL \circ OSL$? If so, what are those constraints?

Two options for LD processes

Option 1

- One computational property, OSL, characterizes both local and long-distance processes.
- Like ISL, the OSL property can also be used to learn these maps (stay tuned!).

Two options for LD processes

Option 2

- Local processes are characterized as those OSL maps that *are* closed under composition (i.e., Structure Preservation, Kiparsky 1985)).
- Long distance processes require a distinct, to-be-defined property (e.g., Strictly Piecewise, Tier-Based Strictly Local).

Acknowledgments

Thanks to Jeffrey Heinz, Rémi Eyraud, Adam Jardine, and James Rogers for valuable contributions and discussion.

Optimization

Nonregular mappings can be obtained through the interaction of simple markedness constraints and standard faithfulness constraints (Riggle 2004; Gerdemann and Hulden 2012).

Optimization

$\textsc{Ident}, \textsc{Dep} \gg *ab \gg \textsc{Max}$

Non-regular relation:

$a^n b^m \mapsto a^n$, if $m < n$

$a^n b^m \mapsto b^m$, if $n < m$

Strict Locality

- A *Strictly k-Local formal language* is one for which the following property holds:

Theorem (Suffix Substitution Closure)

(Rogers and Pullum 2011) $\mathcal{L}$ *is Strictly Local iff for all strings* u_1, v_1, u_2, v_2, *there exists* $k \in \mathbb{N}$ *such that for any string* x *of length* $k-1$, *if* $u_1 x v_1, u_2 x v_2 \in \mathcal{L}$, *then* $u_1 x v_2 \in \mathcal{L}$.

Strict Locality

Post-nasal obstruent voicing

(21) Quechua (Orr 1962; Rice 1993; Pater 2010)

a. sinik-pa ‘porcupine’s’

b. kam-ba ‘yours’

Strict Locality

Post-nasal obstruent voicing: Valid surface strings are a SL-2 language.

TVNDV in the language.
NVNTV in the language.

TVNTV *necessarily* in the language.

Strict Locality

- Surface constraints expressible with a contiguous substring of bounded length can be are Strictly *k*-Local, where *k* is the length of the illicit substring (**?**).
- For *processes* (i.e., UR $\mapsto$ SR), we need *functions*.