Representing and Learning Opaque Maps with Strictly Local Functions

Jane Chandlee

Jeffrey Heinz

Adam Jardine

UNIVERSITY OF DELAWARE

CNRS, Paris, France

April 18, 2015

GLOW 38: Workshop on Computational Phonology

Representing and Learning Opaque Maps

1. Chandlee 2014 defines and studies *input strictly local* functions which are a class of string-to-string maps (see also Chandlee and Heinz, in revision).
2. Here we show that many **opaque** phonological patterns can be represented and modeled with these functions **without** any additional modifications.
3. This matters because input strictly local functions are **efficiently learnable** from positive evidence (Chandlee 2014, Chandlee et al. 2014, and Jardine et al. 2014).
4. Other reasons why it matters—and the implications for phonological theory more generally—will also be discussed.

Part I

Studying the nature of phonological maps

Transformations in Generative Phonology

- Theories of generative phonology concern transformations, or **maps** from abstract underlying representations to surface phonetic representations.
- A truism about maps:

  Different grammars may generate the same map.

  Such grammars are *extensionally equivalent*.
- Grammars are finite, *intensional* descriptions and maps are their (possibly infinite) *extensions*
- Maps may have properties largely independent of their grammars. . .
  - output-driven maps (Tesar 2014)
  - regular maps (Elgot and Mezei 1956, Scott and Rabin 1959)
  - subsequential maps (Oncina et al. 1993, Mohri 1997, Heinz and Lai 2013)
  - . . .

Input Strict Locality: Main Idea

These maps are Markovian in nature.

$$x_0\ x_1\ \ldots\ x_n$$

$$\downarrow$$

$$u_0\ u_1\ \ldots\ u_n$$

where

1. Each x_i is a single symbol ($x_i \in \Sigma_1$)
2. Each u_i is a *string* ($u_i \in \Sigma_2^*$)
3. There exists a $k \in \mathbb{N}$ such that for all input symbols x_i its output string u_i depends only on x_i and the $k-1$ elements immediately preceding x_i.

   (so u_i is a function of $x_{i-k+1}x_{i-k+2}\ldots x_i$)

Input Strict Locality: Main Idea in a Picture

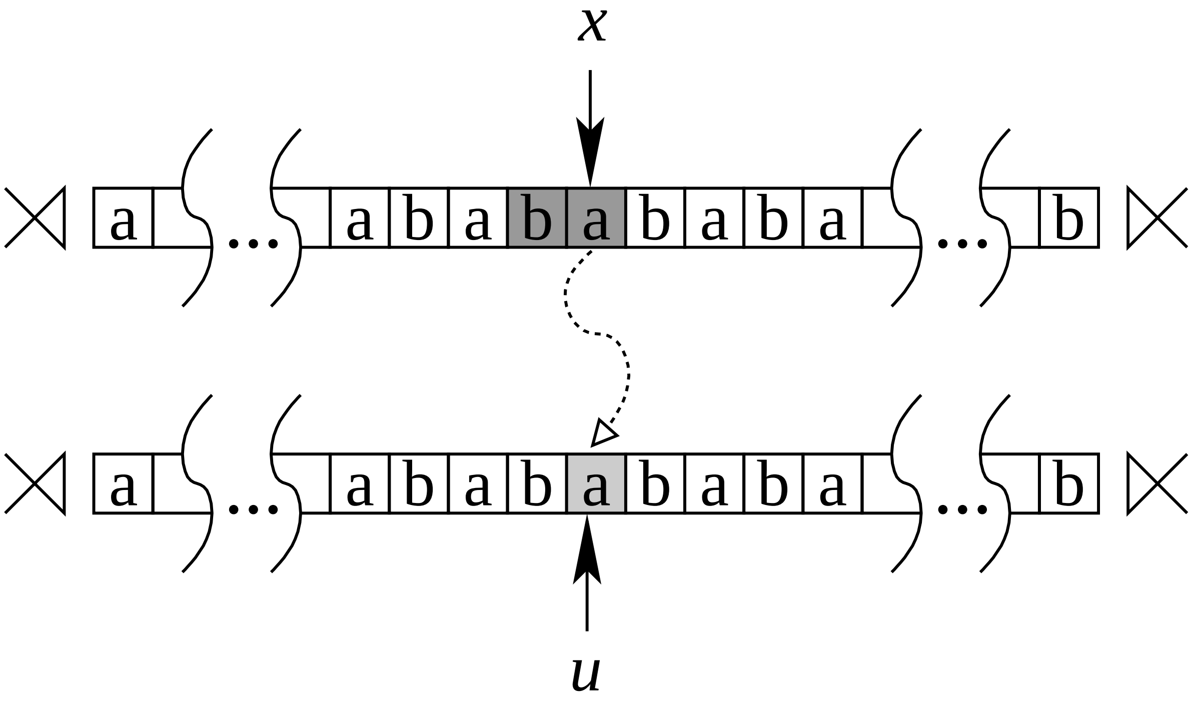


Figure 1: For every Input Strictly 2-Local function, the output string u of each input element x depends only on x and the input element previous to x. In other words, the contents of the lightly shaded cell only depends on the contents of the darkly shaded cells.

Example: Nasal Place Assimilation is ISL with $k = 2$

/inpɚfɛkt / ↦ [impɚfɛkt]

input:	⋊	ɪ	n	p	ɚ	f	ɛ	k	t	⋉
output:	⋊	ɪ	λ	mp	ɚ	f	ɛ	k	t	⋉

Example: Nasal Place Assimilation is ISL with $k = 2$

$$/\text{inpɚfɛkt}\ / \mapsto [\text{impɚfɛkt}]$$

input:	$\rtimes$	ɪ	n	p	ɚ	f	ɛ	k	t	$\ltimes$
output:	$\rtimes$	ɪ	λ	mp	ɚ	f	ɛ	k	t	$\ltimes$

Example: Nasal Place Assimilation is ISL with $k = 2$

/inpɚfɛkt / ↦ [impɚfɛkt]

input:	⋊	ɪ	n	p	ɚ	f	ɛ	k	t	⋉
output:	⋊	ɪ	λ	mp	ɚ	f	ɛ	k	t	⋉

Example: Nasal Place Assimilation is ISL with $k = 2$

/inpɚfɛkt / ↦ [impɚfɛkt]

input:	⋊	ɪ	n	p	ɚ	f	ɛ	k	t	⋉
output:	⋊	ɪ	λ	mp	ɚ	f	ɛ	k	t	⋉

What can be modeled with ISL functions?

1. A significant range of individual phonological processes such as local substitution, deletion, epenthesis, and synchronic metathesis
2. Approximately 95% of the individual processes in P-Base (v.1.95, Mielke (2008))
3. Opaque maps! (This talk, in a moment)

(Chandlee 2014, Chandlee and Heinz, in revision)

What cannot be modeled with ISL functions

1. progressive and regressive spreading
2. long-distance (unbounded) consonant and vowel harmony

(Chandlee 2014, Chandlee and Heinz, in revision)

What about spreading and long-distance phonology?

- ISL functions naturally generalize Strictly Local (SL) stringsets in Formal Language Theory.
- SL stringsets model local *phonotactics* and ISL functions model phonological maps with local triggers.
- *Output* SL functions are another generalization of SL stringsets which precisely models spreading (Chandlee et al. under review).
- Stringly-Piecewise (SP) and Tier-based Strictly Local (TSL) stringsets model *long-distance phonotactics* (Heinz 2010, Heinz et al. 2011).
- We expect functional characterizations of SP and TSL stringsets will model long-distance maps (work-in-progress).

Automata characterization of k-ISL functions

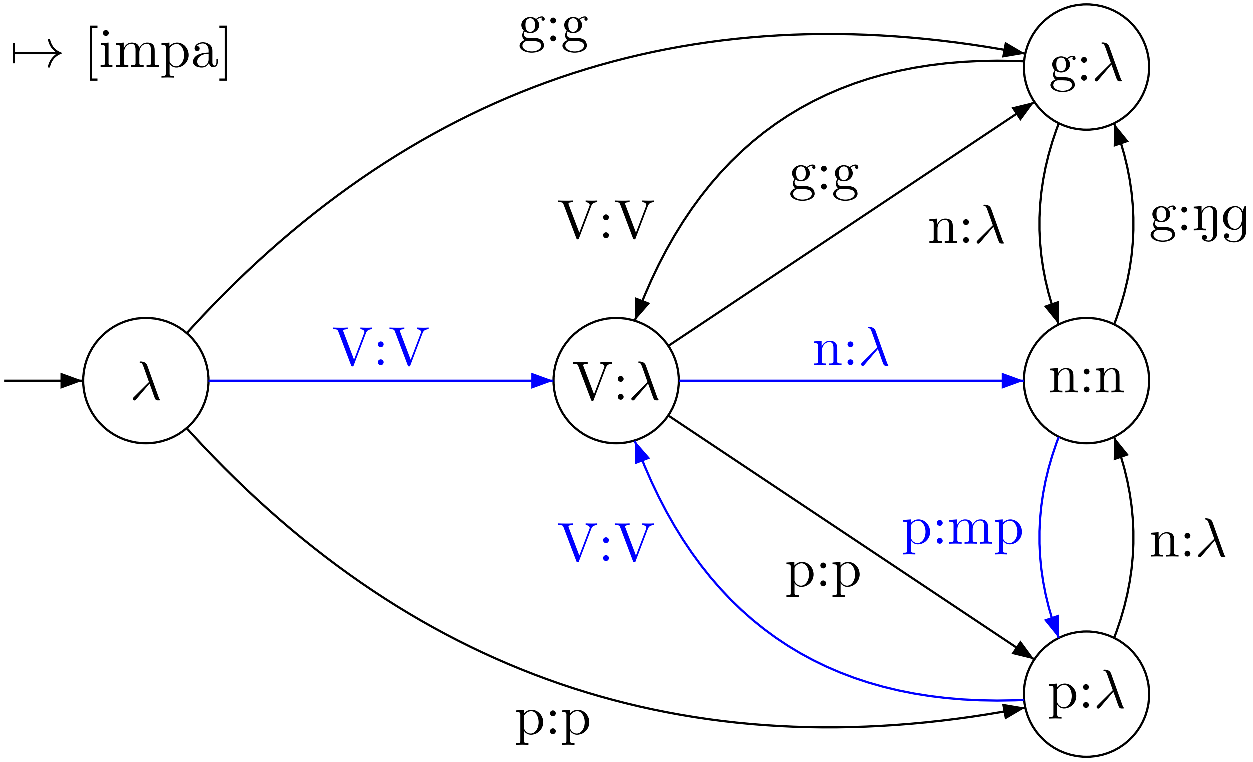
Theorem Every k-ISL function can be modeled by a *k-ISL transducer* and every *k-ISL transducer* represents a k-ISL function.

The state space and transitions of these transducers are organized such that two input strings with the same $k-1$ suffix always lead to the same state.

(Chandlee 2014, Chandlee et. al 2014)

Example: Fragment of k-ISL transducer for NPA

/inpa/ ↦ [impa]



Not all transitions and states are shown, and vowel states and transitions are collapsed. The nodes are labeled `name:output_string`.

Part II

Studying Opacity in Phonology

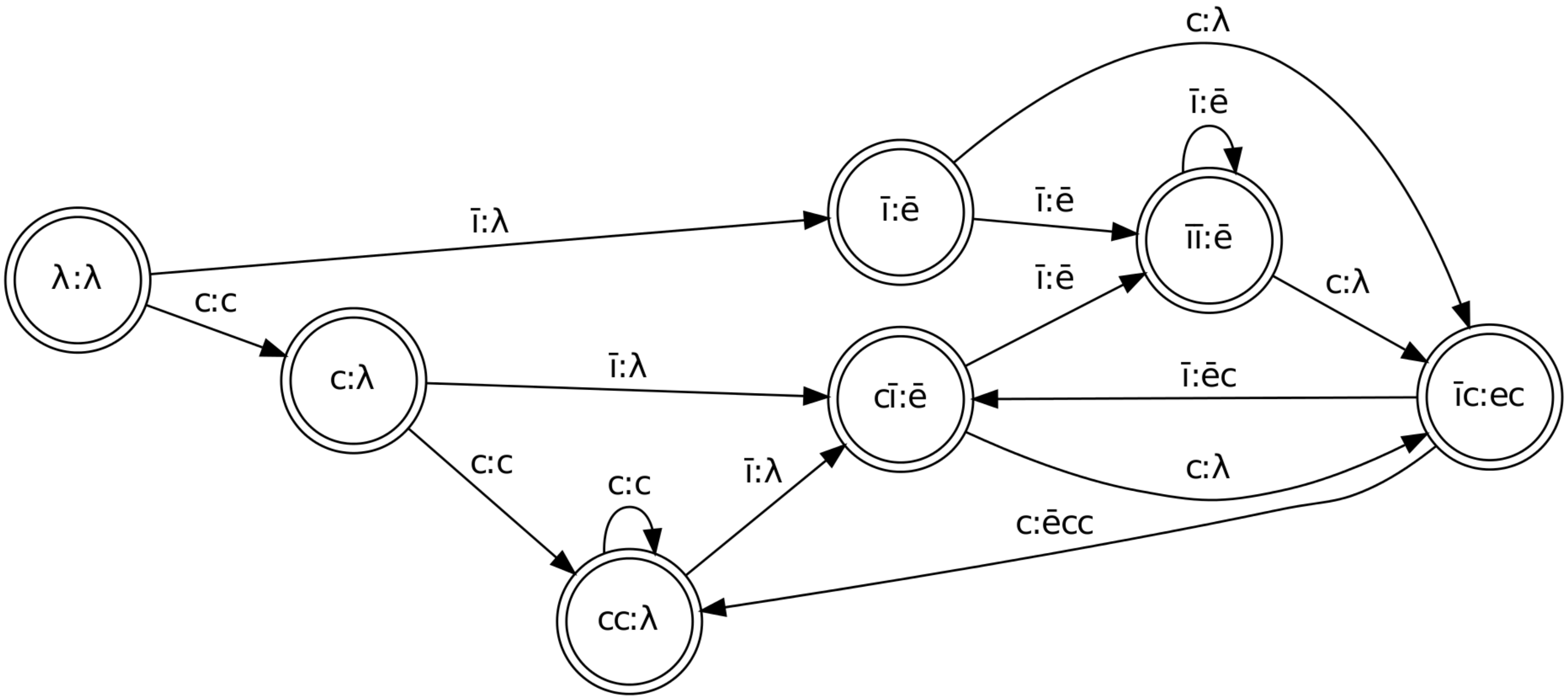
Defining Opaque maps

- Opaque maps are defined as the extensions of particular rule-based grammars (Kiparsky 1971, McCarthy 2007).
- Baković (2007) provides a typology of opaque maps.
  - Counterbleeding
  - Counterfeeding on environment
  - Counterfeeding on focus
  - Self-destructive feeding
  - Non-gratuitous feeding
  - Cross-derivational feeding
- Subsequent examples are drawn from this paper.

Counterbleeding in Yokuts

	‘might fan’
	/ʔili:+l/
[+long] → [-high]	ʔile:l
V → [-long] / __ C #	ʔilel
	[ʔilel]

Fragment of a 3-ISL transducer for Yokuts



l	ī	l	⋉
l	λ	λ	el

Counterfeeding-on-environment in Bedouin Arabic

Bedouin Arabic

	‘Bedouin’
	/badw/
$\text{a} \rightarrow \text{i} \; / \; \underline{\quad} \; \sigma$	—
$\text{G} \rightarrow \text{V} \; / \; \text{C} \; \underline{\quad}\#$	badu
	[badu]

This is 4-ISL.

The other examples

- Counterfeeding on focus in Bedouin Arabic (3-ISL)
- Self-destructive feeding in Turkish (5-ISL)
- Non-gratuitous feeding in Classical Arabic (5-ISL)
- Cross-derivational feeding in Lithuanian (4-ISL)

What is k?

- If a function described by a rewrite rule A $\longrightarrow$ B / C ___ D is ISL then k will be the length of the longest string in the structural description CAD.
- For opaque maps describable as the composition of two rewrite rules, it is approximately the length of the longest string in the structural description of either rule (it may be a little longer).
- A k-value of 5 appears sufficient for the examples in Baković's paper.

Part III

Learning ISL functions

ISLFLA: Input Strictly Local Function Learning Algorithm

- The input to the algorithm is k and a finite set of (u, s) pairs.
- ISLFLA builds a input prefix tree transducer and merges states that share the same $k - 1$ prefix.
- Provided the sample data is of sufficient quality, ISLFLA provably learns any function k-ISL function in quadratic time.
- Sufficient data samples are quadratic in the size of the target function.

(Chandlee et al. 2014, TACL)

SOSFIA: Structured Onward Subsequential Function Inference Algorithm

SOSFIA takes advantage of the fact that every k-ISL function can be represented by an *onward* transducer with the *same* structure (states and transitions).

- Thus the input to the algorithm is k-ISL transducer with empty output transitions, and a finite set of (u, s) pairs.
- SOSFIA calculates the outputs of each transition by examining the longest common prefixes of the outputs of prefixes of the input strings in the sample (onwardness).
- Provided the sample data is of sufficient quality, SOSFIA provably learns any function k-ISL function in *linear* time.
- Sufficient data samples are *linear* in the size of the target function.

(Jardine et al. 2014, ICGI)

Part IV

Implications for a theory of phonology

Some reasons why Classic OT has been influential

- Offers a theory of typology.
- Comes with learnability results.
- Solves the duplication/conspiracy problems.

What have we shown?

1. Many attested phonological maps, including many opaque ones, are k-ISL for a small k.
2. k-ISL functions make strong typological predictions.
   (a) No non-regular map is k-ISL.
   (b) Many regular maps are not k-ISL.
   $\Rightarrow$ So they are *subregular*.
3. k-ISL functions are efficiently learnable.

How does this relate to traditional phonological grammatical concepts?

1. Like OT, k-ISL functions do not make use of intermediate representations.
2. Like OT, k-ISL functions separate marked structures from their repairs (Chandlee et al. to appear, AMP 2014).
   - k-ISL functions are sensitive to all and only those markedness constraints which could be expressed as $*x_1x_2\ldots x_k$, ($x_i \in \Sigma$).
   - In this way, k-ISL functions model the "homogeneity of target, heterogeneity of process" (McCarthy 2002)

Where is the Optimality?

- Paul Smolensky asked this question to Jane Chandlee at the 2013 AMP at UMass where ISL functions were first introduced.
- The answer is “Over there.”
- Perhaps the question ought to be “How does Optimality Theory account for the typological generalization that so many phonological maps are ISL?”

Undergeneration in Classic OT

- It is well-known that classic OT cannot generate opaque maps (Idsardi 1998, 2000, McCarthy 2007, Buccola 2013)

  (though Baković 2007, 2011 argues for a more nuanced view).

- Many, many adjustments to classic OT have been proposed.
  - constraint conjunction (Smolensky), sympathy theory (McCarthy), turbidity theory (Goldrick), output-to-output representations (Benua), stratal OT (Kiparsky, Bermudez-Otero), candidate chains (McCarthy), harmonic serialism (McCarthy), targeted constraints (Wilson), contrast preservation (Łubowicz) comparative markedness (McCarthy) serial markedness reduction (Jarosz), . . .

See McCarthy 2007, *Hidden Generalizations* for review, meta-analysis, and more references to these earlier attempts.

Adjustments to Classic OT

- constraint conjunction (Smolensky), sympathy theory (McCarthy), turbidity theory (Goldrick), output-to-output representations (Benua), stratal OT (Kiparsky, Bermudez-Otero), candidate chains (McCarthy), harmonic serialism (McCarthy), targeted constraints (Wilson), contrast preservation (Łubowicz) comparative markedness (McCarthy) serial markedness reduction (Jarosz), . . .

These approaches invoke different representational schemes, constraint types and/or architectural changes to classic OT.

- The typological and learnability ramifications of these changes is not yet well-understood in many cases.
- On the other hand, *no special modifications are needed* to establish the ISL nature of the opaque maps we have studied.

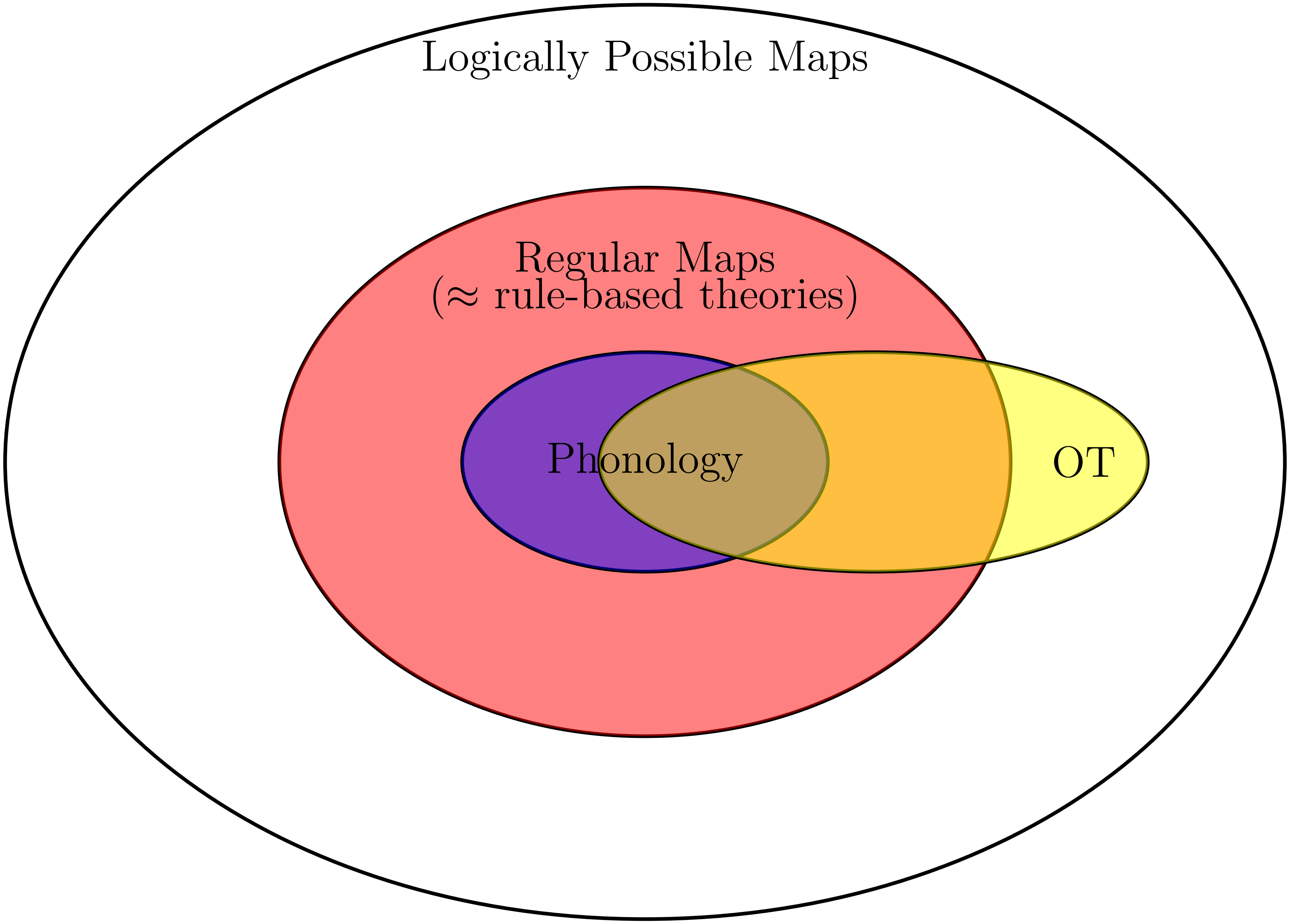
Overgeneration in Classic OT

- It is not controversial that classic OT generates non-regular maps with simple constraints (Frank and Satta 1998, Riggle 2004, Gerdemann and Hulden 2012, Heinz and Lai 2013)

OT's greatest strength is its greatest weakness.

- The signature success of a successful OT analysis is when complex phenomena are understood as the interaction of simple constraints.
- But the overgeneration problem is precisely this problem: complex—but weird—phenomena resulting from the interaction of simple constraints (e.g. Hansson 2007 on ABC).
- As for the undergeneration problem, opaque candidates are not optimal in classic OT.

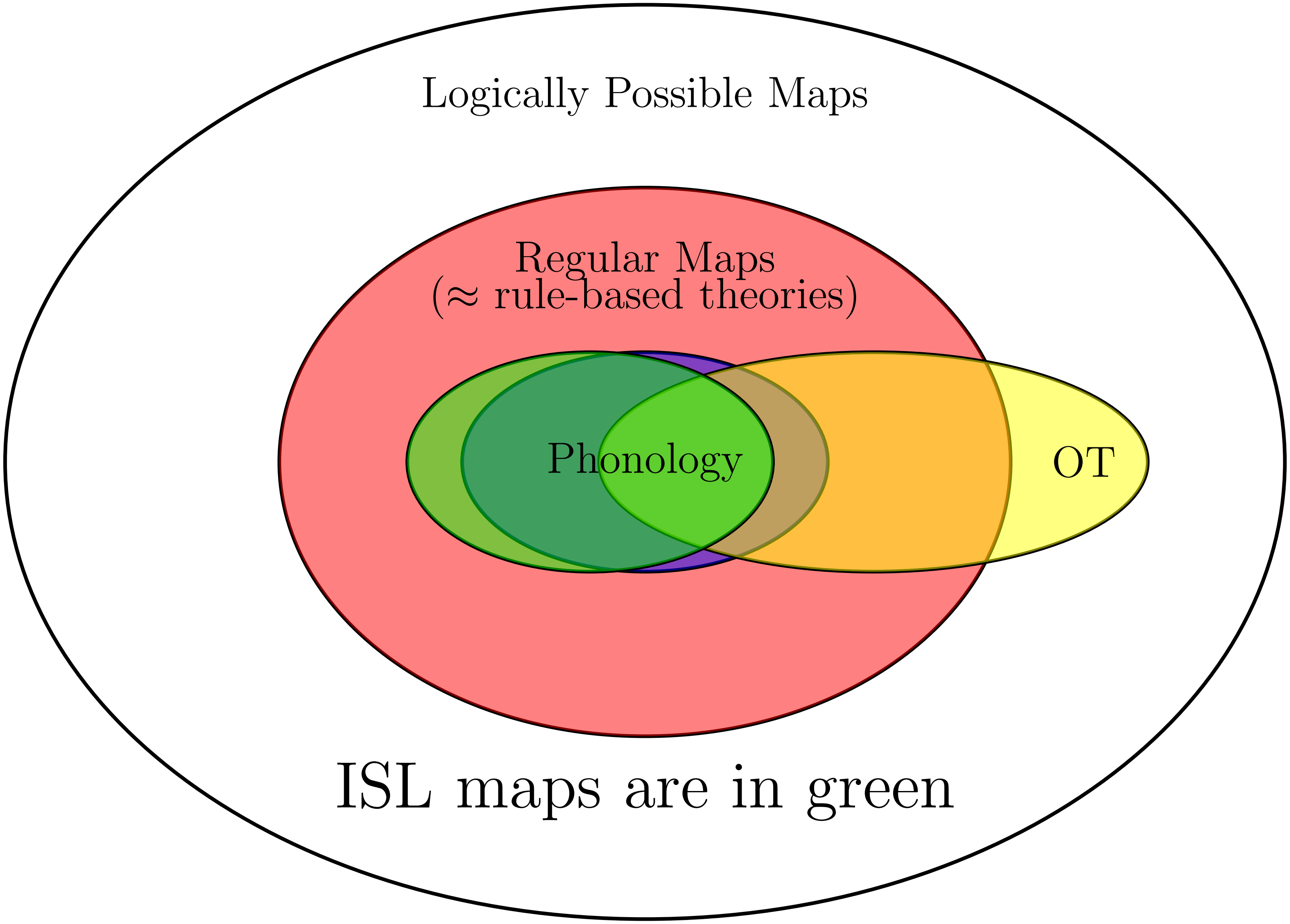
In a picture


Logically Possible Maps
Regular Maps
(≈ rule-based theories)
Phonology
OT

Conclusion

- Despite some limitations, k-ISL functions characterize well the nature of phonological maps.
  - Many phonological maps, **including opaque ones**, can be expressed with them.
  - k-ISL functions provide both a more expressive and restrictive theory of typology than classic OT, which we argue **better matches** the attested typlogy.
- Like classic OT, there are no intermediate representations, and k-ISL functions can express the "homogeneity of target, heterogeneity of process" which helps address the conspiracy and duplication problems.
- k-ISL functions are learnable.
- Unlike OT, *subregular computational properties* like ISL—and not optimization—form the core computational nature of phonology.

Conclusion in a picture



ISL maps are in green

Questions

1. What is k? Can k be learned?
2. What about phonetics?
3. How do you learn underlying representations?

Acknowledgments

Thank You.

*Special thanks to Rémi Eyraud, Bill Idsardi, and Jim Rogers for valuable discussion. We also thank Iman Albadr, Hyun Jin Hwangbo, Taylor Miller, and Curt Sebastian for useful comments and feedback.

Appendix

Some extra slides as needed

Formal Language Theory

- ISL functions naturally extend SL stringsets in Formal Language Theory.
- For SL stringsets, well-formedness is determined by examining windows of size k.

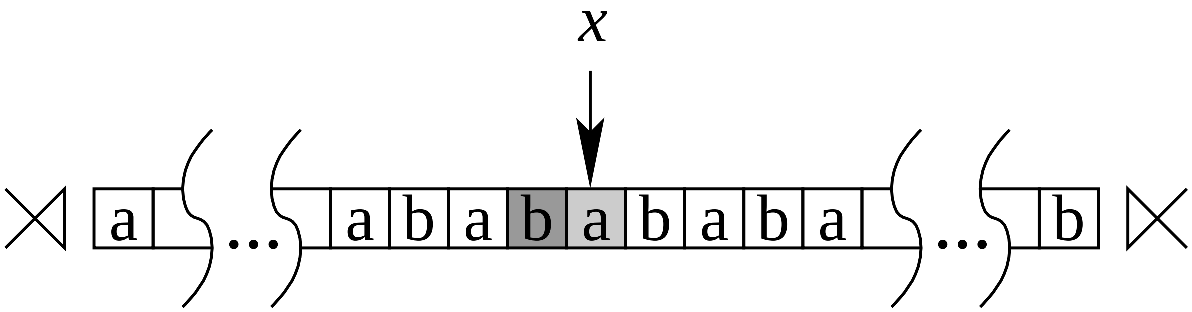


Figure 2: A stringset is Strictly 2-Local if the well-formedness of each word can be determined by checking whether each symbol x in the word is licensed by the preceding symbol.

(McNaughton and Papert 1971, Rogers and Pullum 2011)

Subregular Hierarchies for Stringsets

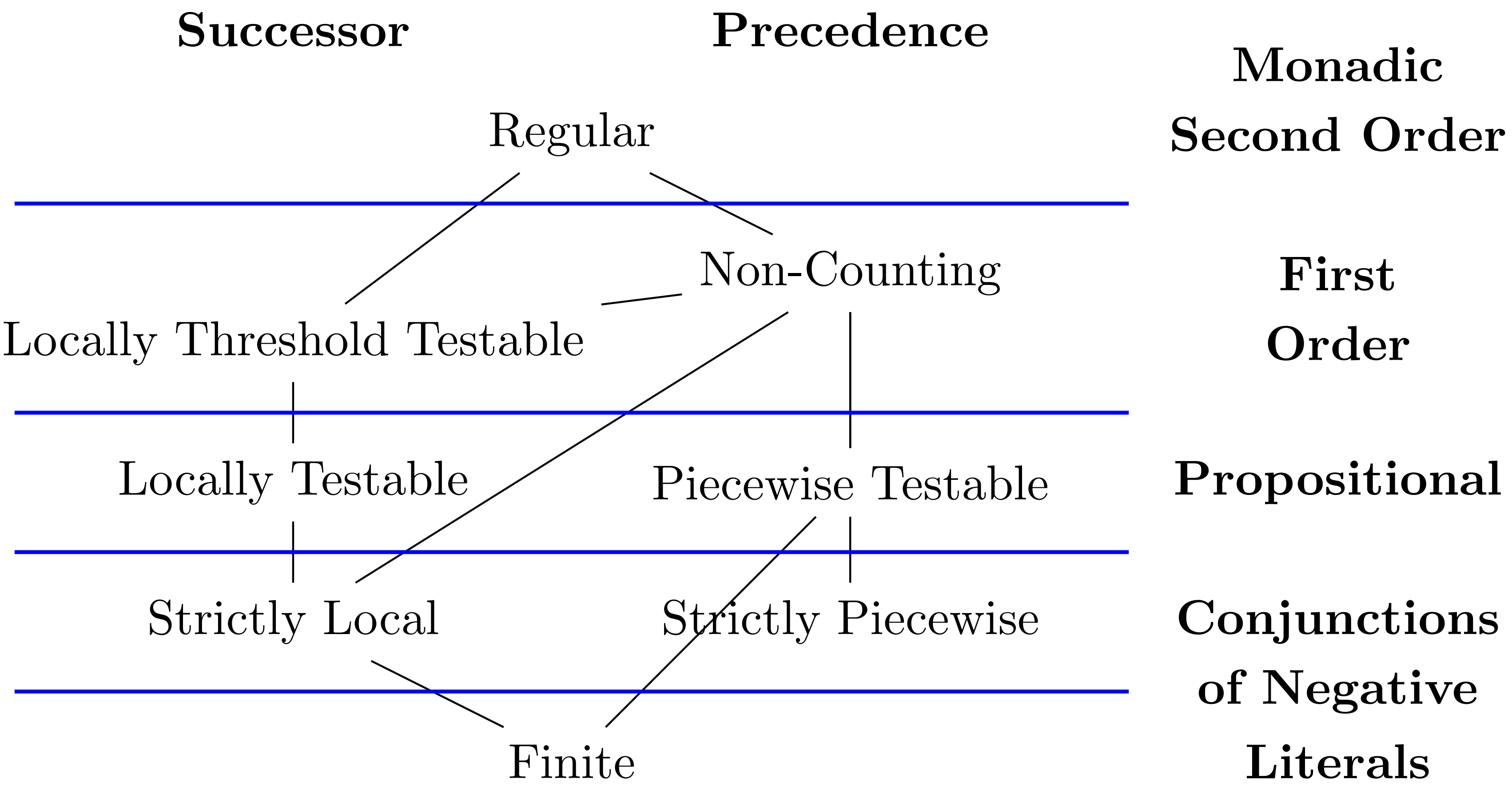
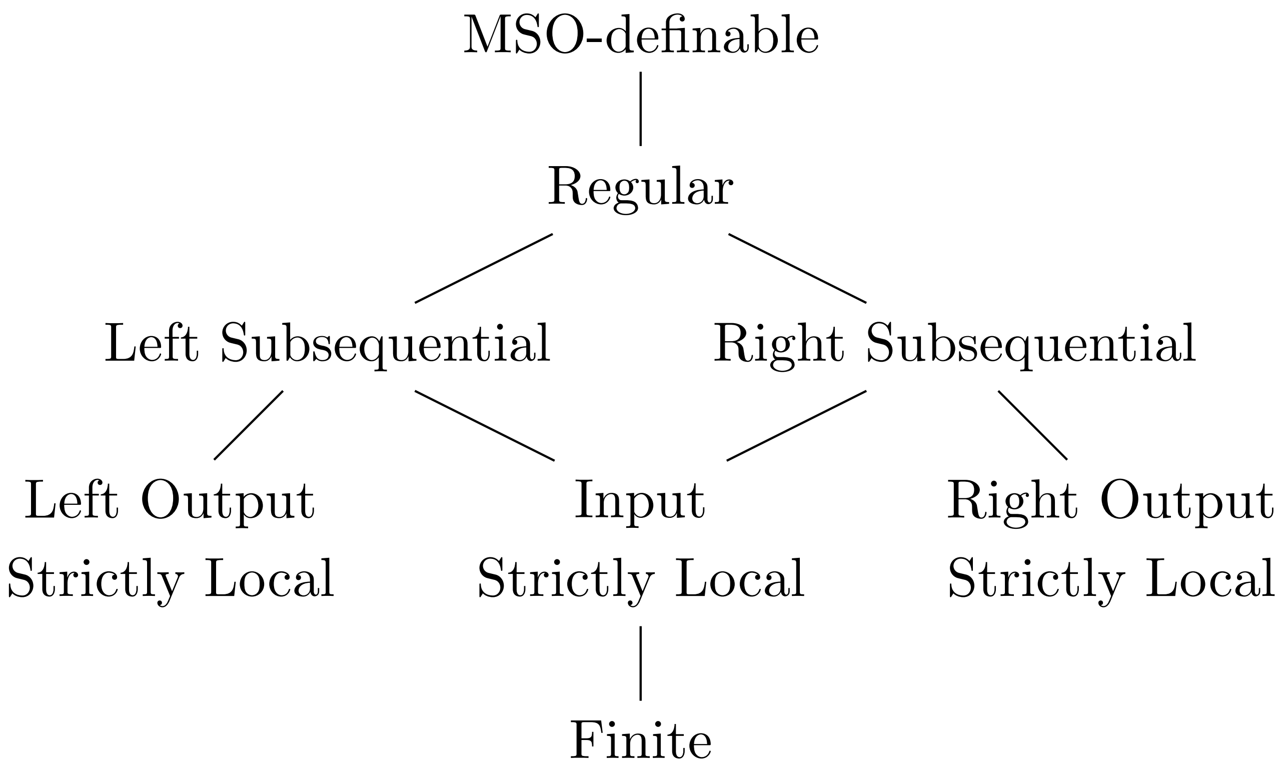


Figure 3: Subregular hierarchies of stringsets.

Subregular Hierarchies for Maps



MSO-definable

Regular

Left Subsequential

Right Subsequential

Left Output Strictly Local

Input Strictly Local

Right Output Strictly Local

Finite

Figure 4: Subregular hierarchies of maps (so far).

- We are currently generalizing other subregular classes of stringsets to maps.

Formal Definition of ISL

Definition (Residual Function) Given a function f and input string x, the residual function of f w.r.t. x is

$$r_f(x) \overset{\text{def}}{=} \Big\{(y,v) \mid (\exists u)\big[f(xy) = uv \wedge u = \texttt{lcp}(f(x\Sigma^*))\big]\Big\}$$

where $\texttt{lcp}(S)$ is the *longest common prefix* of all strings in S.

Definition (k-ISL Function) A function (map) f is Input Strictly k-Local if for all input strings w, v:

$$\text{if } \texttt{Suff}^{k-1}(w) = \texttt{Suff}^{k-1}(v) \text{ then } r_f(w) = r_f(v).$$

Note this definition is *agrammatical*!

Counterfeeding-on-focus

Bedouin Arabic

	‘he wrote’
	/katab/
$\text{i} \rightarrow \emptyset \;/\; \underline{\hspace{1em}}\; \sigma$	katab
$\text{a} \rightarrow \text{i} \;/\; \underline{\hspace{1em}}\; \sigma$	kitab
	[kitab]

This is 3-ISL.

Self-destructive feeding

Turkish

	‘your baby’
	/bebek+n/
$\emptyset \rightarrow \mathrm{i}$ / C ___ C #	bebekin
$\mathrm{k} \rightarrow \emptyset$ /V ___ +V	bebein
	[bebein]

This is 5-ISL.

Non-gratuitous feeding

Classical Arabic

	‘write.MS’
	/ktub/
$\varnothing \rightarrow \mathrm{V}_i \ / \ \# \ \_\_ \ \mathrm{CCV}_i$	uktub
$\varnothing \rightarrow ʔ \ / \ \# \ \_\_ \ \mathrm{V}$	ʔuktub
	[ʔuktub]

This is 5-ISL.

Cross-derivational feeding

Lithuanian

	‘to strew all over’
	/ap-berti/
$\varnothing \rightarrow \mathrm{i} \,/\, \mathrm{C}\ \underline{\quad}\ \mathrm{C}$	apiberti
where Cs are homorganic stops	
$[-\mathrm{voice}] \rightarrow [+\mathrm{voice}] \,/\ \underline{\quad}\ \mathrm{D}$	apiberti
	[apiberti]

This is 4-ISL.

Simple constraints in OT generate non-regular maps

$$\textsc{Ident}, \textsc{Dep} >> \text{*ab} >> \textsc{Max}$$

$$a^n b^m \mapsto a^n, \text{ if } m < n$$

$$a^n b^m \mapsto b^m, \text{ if } n < m$$

(Gerdemann and Hulden 2012)