The Role of Computation in Phonological Typology and Learning

Jane Chandlee

Haverford College

Dartmouth College
January 31, 2017

Computational Linguistics

Computational linguists pursue a variety of research goals:

- Algorithms and methods for handling natural language data.
  - Siri, Google Translate, Amazon Echo, etc.
- Using the study of computation to understand what language *is*.
  - Computational theory of language

Levels of language

Phonetics	Production and perception of speech sounds
Phonology	Sound patterns
Morphology	Word formation processes
Syntax	Sentence structure
Semantics	Meaning
Pragmatics	Social/cultural conventions

Levels of language

Phonetics	Production and perception of speech sounds
Phonology	**Sound patterns**
Morphology	Word formation processes
Syntax	Sentence structure
Semantics	Meaning
Pragmatics	Social/cultural conventions

Computational nature of phonology

- Central question: what is the nature of the computations involved in phonological systems?
- Main result: phonology is quite **restrictive** in its computational complexity, and this restrictiveness gives us insight into both **cognition** and **language learning**

Phonological patterns

Phonotactics	Processes
German: [zaːk] (*zaːg), ‘say’	/zaːg/ $\mapsto$ [zaːk]
English: [gɹeɪps] (*gɹeɪpz), ‘grapes’	/gɹeɪpz/ $\mapsto$ [gɹeɪps]

Phonotactics

Attested	Don't end a word with sound x Don't start a word with sound x Don't allow sequences of sound x followed by sound y etc.
Unattested	Don't have an even/odd number of sound x in a word If a word starts with sound x it can't end with sound y A word can have up to 3 sound x's, but no more etc.

Phonotactics

Attested	Don’t end a word with sound *x* Don’t start a word with sound *x* Don’t allow sequences of sound *x* followed by sound *y* etc.
Unattested	Don’t have an even/odd number of sound *x* in a word If a word starts with sound *x* it can’t end with sound *y* A word can have up to 3 sound *x*’s, but no more etc.

Phonotactics

Attested	Don't end a word with sound *x* Don't start a word with sound *x* Don't allow sequences of sound *x* followed by sound *y* etc.
Unattested	Don't have an even/odd number of sound *x* in a word If a word starts with sound *x* it can't end with sound *y* A word can have up to 3 sound *x*'s, but no more etc.

Goal: Explain this boundary in terms of computational complexity.

Phonotactics as formal languages

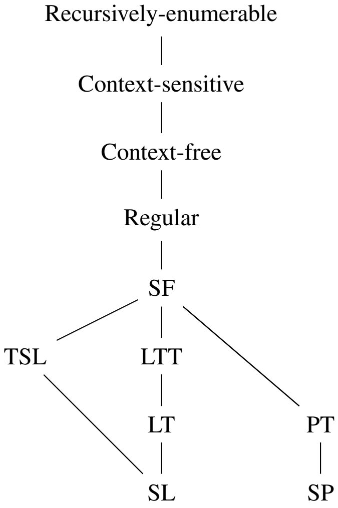
- A formal language is a set of strings built from an alphabet, or set of symbols, Σ

(1) English: Σ = { p, t, k, b, d, g, m, n, ŋ, s, z, ʃ, ʒ, ... }

- A phonotactic constraint can be modeled with the set of strings that do **not** violate it.

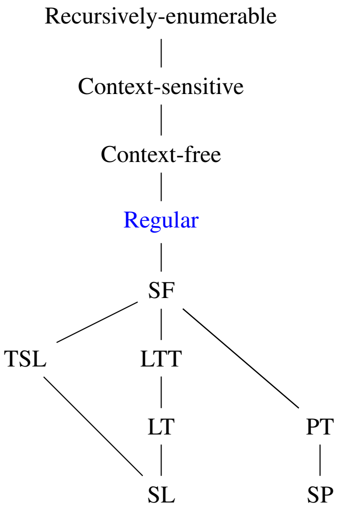
(2) { gɹeɪps, æpl̩z, fɪgz, æpɹɪkɑts, ips, ... }

Classifying formal languages



(Chomsky, 1956; Rogers and Pullum, 2011; Rogers et al., 2013)

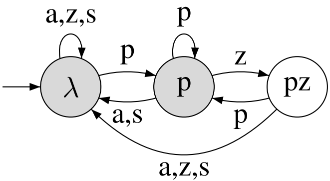
Hypothesis: phonotactics are regular



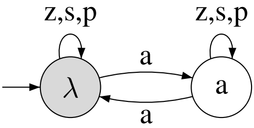
(Chomsky, 1956; Rogers and Pullum, 2011; Rogers et al., 2013)

Hypothesis: phonotactics are regular

Don't end in [pz].
$\Sigma = \{p, z, s, a\}$

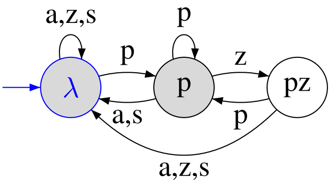


Don't have an odd number of [a]'s.
$\Sigma = \{p, z, s, a\}$

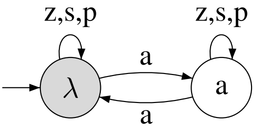


Hypothesis: phonotactics are regular

Don't end in [pz].
$\Sigma = \{p, z, s, a\}$



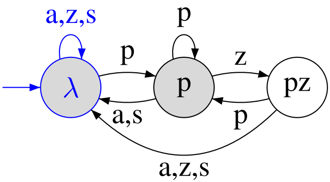
Don't have an odd number of [a]'s.
$\Sigma = \{p, z, s, a\}$



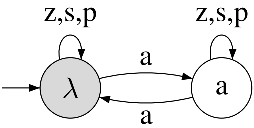
z a p z

Hypothesis: phonotactics are regular

Don't end in [pz].
$\Sigma = \{p, z, s, a\}$



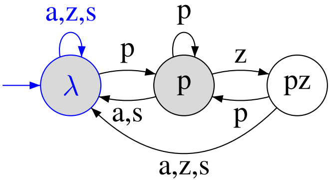
Don't have an odd number of [a]'s.
$\Sigma = \{p, z, s, a\}$



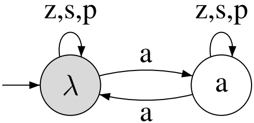
z a p z

Hypothesis: phonotactics are regular

Don't end in [pz].
$\Sigma = \{p, z, s, a\}$



Don't have an odd number of [a]'s.
$\Sigma = \{p, z, s, a\}$

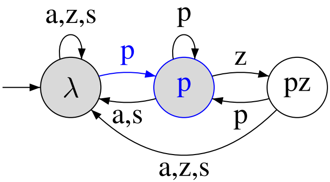


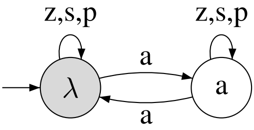
z a p z

Hypothesis: phonotactics are regular

Don't end in [pz].
$\Sigma = \{p, z, s, a\}$

Don't have an odd number of [a]'s.
$\Sigma = \{p, z, s, a\}$

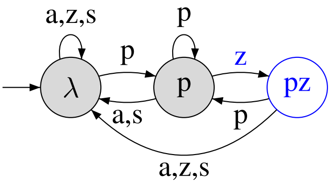




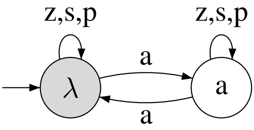
z a p z

Hypothesis: phonotactics are regular

Don't end in [pz].
$\Sigma = \{p, z, s, a\}$



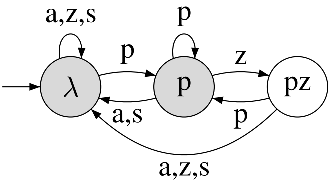
Don't have an odd number of [a]'s.
$\Sigma = \{p, z, s, a\}$



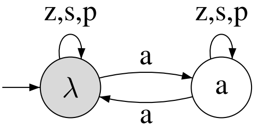
z a p z

Hypothesis: phonotactics are regular

Don't end in [pz].
$\Sigma = \{p, z, s, a\}$



Don't have an odd number of [a]'s.
$\Sigma = \{p, z, s, a\}$

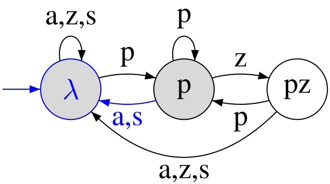


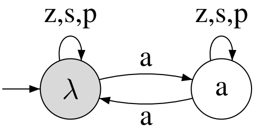
z a p z
X

Hypothesis: phonotactics are regular

Don't end in [pz].
$\Sigma = \{p, z, s, a\}$

Don't have an odd number of [a]'s.
$\Sigma = \{p, z, s, a\}$



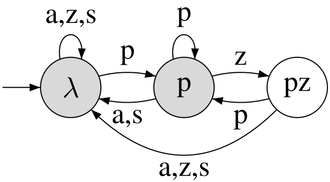


z a p s

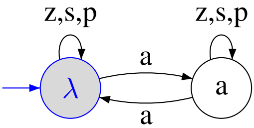
✓

Hypothesis: phonotactics are regular

Don’t end in [pz].
$\Sigma = \{p, z, s, a\}$



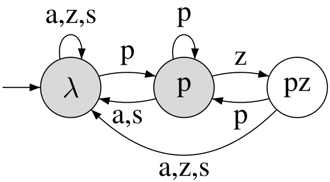
Don’t have an odd number of [a]’s.
$\Sigma = \{p, z, s, a\}$



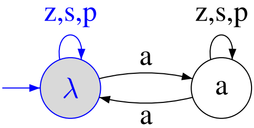
z a p s

Hypothesis: phonotactics are regular

Don't end in [pz].
$\Sigma = \{p, z, s, a\}$



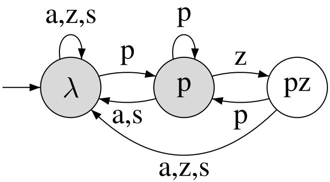
Don't have an odd number of [a]'s.
$\Sigma = \{p, z, s, a\}$



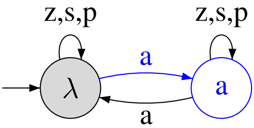
z a p s

Hypothesis: phonotactics are regular

Don't end in [pz].
$\Sigma = \{p, z, s, a\}$



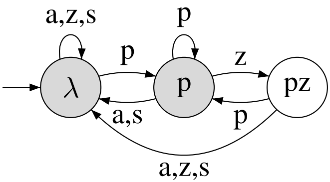
Don't have an odd number of [a]'s.
$\Sigma = \{p, z, s, a\}$



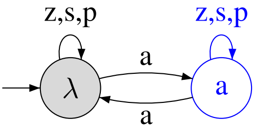
z a p s

Hypothesis: phonotactics are regular

Don't end in [pz].
$\Sigma = \{p, z, s, a\}$



Don't have an odd number of [a]'s.
$\Sigma = \{p, z, s, a\}$

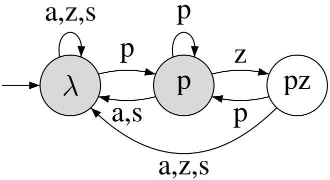


z a p s

Hypothesis: phonotactics are regular

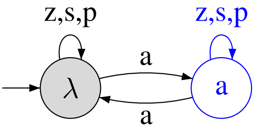
Don’t end in [pz].

$\Sigma = \{p, z, s, a\}$



Don’t have an odd number of [a]’s.

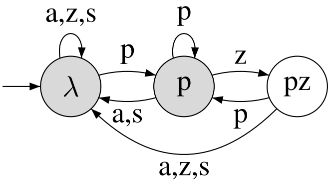
$\Sigma = \{p, z, s, a\}$



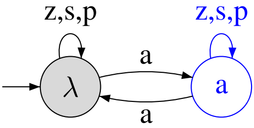
z a p s

Hypothesis: phonotactics are regular

Don't end in [pz].
$\Sigma = \{p, z, s, a\}$



Don't have an odd number of [a]'s.
$\Sigma = \{p, z, s, a\}$

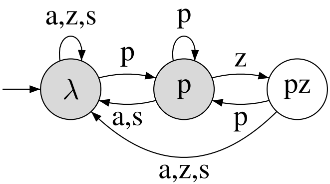


z a p s

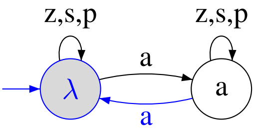
X

Hypothesis: phonotactics are regular

Don't end in [pz].
$\Sigma = \{p, z, s, a\}$



Don't have an odd number of [a]'s.
$\Sigma = \{p, z, s, a\}$



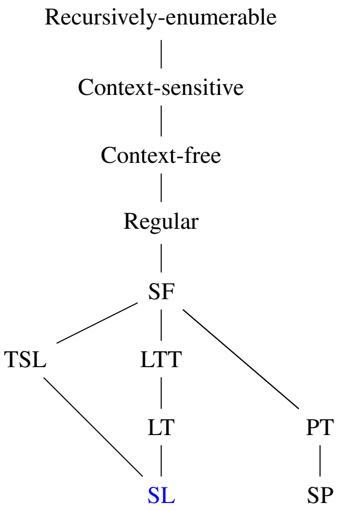
z a p a

✓

✓ Hypothesis: phonotactics are regular

However,...

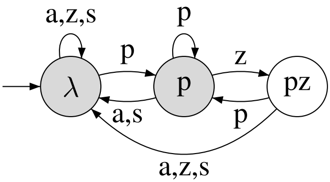
Hypothesis: phonotactics are *subregular*



(Chomsky, 1956; Heinz, 2007; Rogers and Pullum, 2011; Rogers et al., 2013)

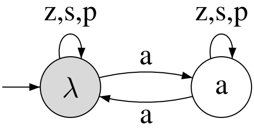
Strictly Local FSAs

Don't end in [pz].
$\Sigma = \{p, z, s, a\}$



States represent
last segment(s) seen.

Don't have an odd number of [a]'s.
$\Sigma = \{p, z, s, a\}$



States represent even/odd [a]'s.

Phonological processes

- Assumption: the English plural suffix is /z/, but in some cases it is pronounced [s].

bags	bægz
chips	tʃɪps

- To avoid sequences of [pz], we have a *process* that changes /z/ in this context to [s].

$$\text{tʃɪpz} \mapsto \text{tʃɪps}$$

Phonological processes as functions

- A processes can be represented with a *function* that maps tʃɪpz to tʃɪps
- A function is a set of string pairs:

  (3) { (tʃɪpz, tʃɪps), (bægz, bægz), ... }

- I'll call these phonological maps (see also Tesar (2012)).

Complexity of phonological maps

Regular relations (Johnson, 1972; Kaplan and Kay, 1994)

$\downarrow$

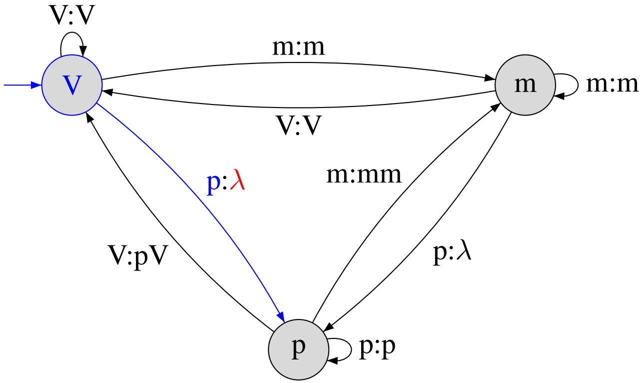
Subsequential functions (Mohri, 1997)

$\downarrow$

Strictly Local functions (Chandlee, 2014)

Strictly Local function

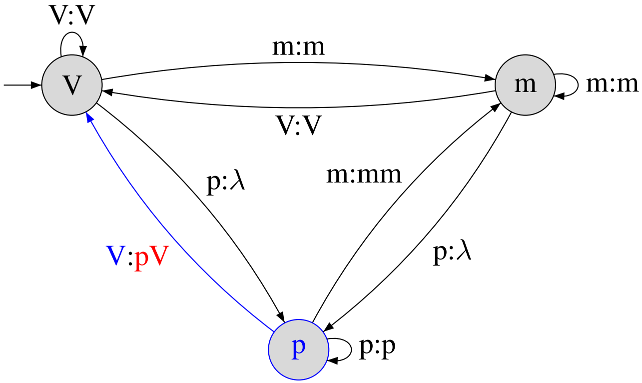
(4) Korean (Lee and Pater, 2008)
/pa**pm**ul/ $\mapsto$ [pa**mm**ul] ‘rice water’



$\rtimes$ p a p m u l $\ltimes$
λ

Strictly Local function

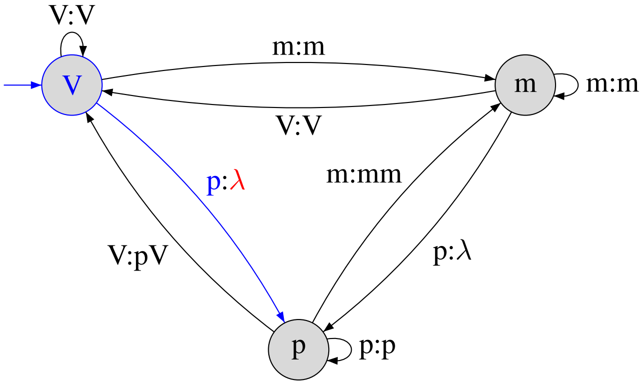
(4) Korean (Lee and Pater, 2008)
/pa**pm**ul/ ↦ [pa**mm**ul] ‘rice water’



⋊	p	a	p	m	u	l	⋉
	λ	pa					

Strictly Local function

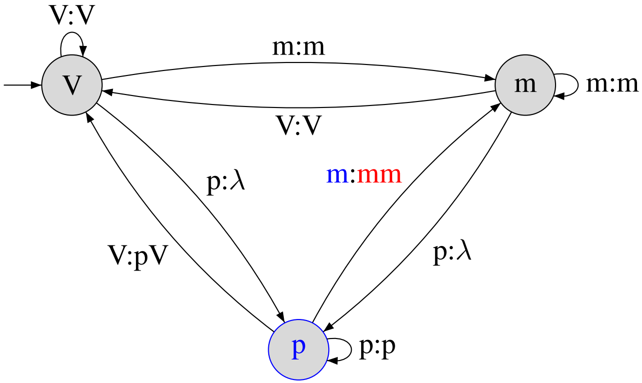
(4) Korean (Lee and Pater, 2008)
/pa**pm**ul/ $\mapsto$ [pa**mm**ul] ‘rice water’



$\rtimes$	p	a	p	m	u	l	$\ltimes$
	λ	pa	λ				

Strictly Local function

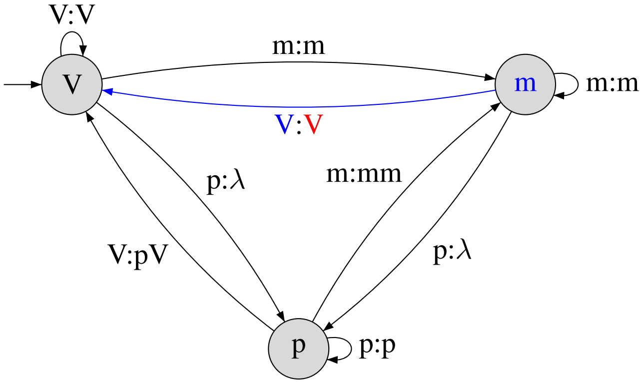
(4) Korean (Lee and Pater, 2008)
/pa**pm**ul/ $\mapsto$ [pa**mm**ul] 'rice water'



⋊	p	a	p	m	u	l	⋉
	λ	pa	λ	mm			

Strictly Local function

(4) Korean (Lee and Pater, 2008)
/pa**pm**ul/ $\mapsto$ [pa**mm**ul] 'rice water'



⋊	p	a	p	m	u	l	⋉
	λ	pa	λ	mm	u		

Complexity of phonological maps

- Local phonological processes are Strictly Local functions (Chandlee, 2014)

REGULAR RELATIONS

↓

SUBSEQUENTIAL FUNCTIONS

↓

STRICTLY LOCAL FUNCTIONS

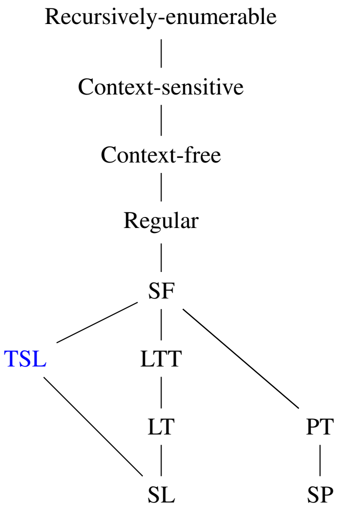
Long-distance phonology

(5) Kikongo (Meinof, 1932; Odden, 1994; Rose and Walker, 2004)

/tu**n**ik-i**d**i/ $\mapsto$ [tu**n**ik-i**n**i] ‘we ground’

- SL version of this phonotactic constraint: don’t have [d] after [niki]

Long-distance phonotactics are TSL

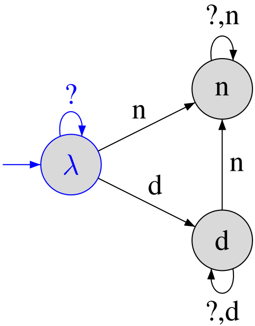


(Heinz et al., 2011; McMullin, 2016)

Tier-based Strictly Local Languages

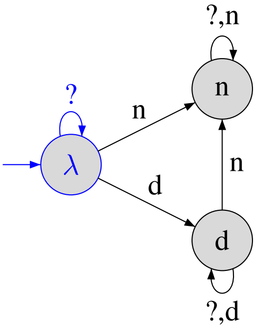
- First designate a subset of the alphabet, called the *tier*:
  T = {n, d}
- Ignoring non-tier symbols, the constraint is:
  ‘Don’t have [d] after [n].’

Tier-based Strictly Local FSA



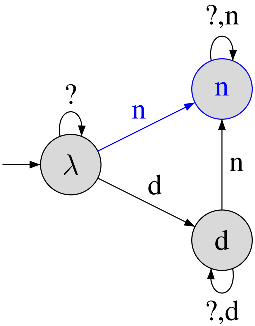
t u n i k i d i

Tier-based Strictly Local FSA



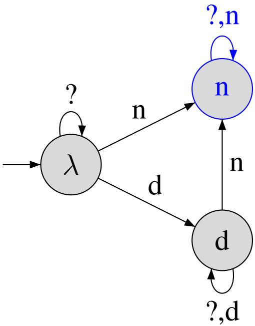
t u n i k i d i

Tier-based Strictly Local FSA



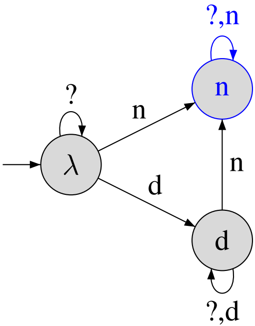
t u n i k i d i

Tier-based Strictly Local FSA



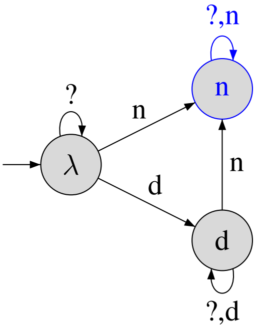
t u n i k i d i

Tier-based Strictly Local FSA



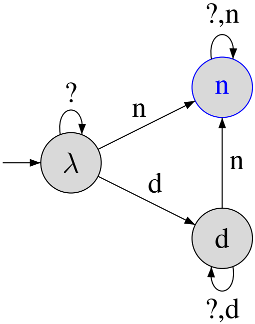
t u n i k i d i

Tier-based Strictly Local FSA



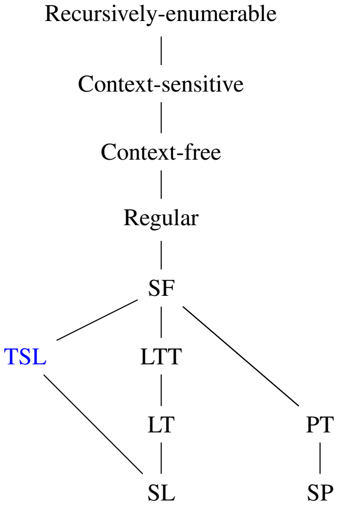
t u n i k i d i

Tier-based Strictly Local FSA



t u n i k i d i

Long-distance phonotactics are TSL (and therefore subregular)



(Heinz et al., 2011; McMullin, 2016)

Long-distance processes

What about long-distance maps?

Hierarchy of maps

REGULAR RELATIONS

↓

SUBSEQUENTIAL FUNCTIONS

↓

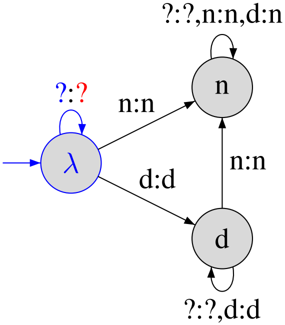
TIER-BASED STRICTLY LOCAL FUNCTIONS

↓

STRICTLY LOCAL FUNCTIONS

Tier-based Strictly Local functions

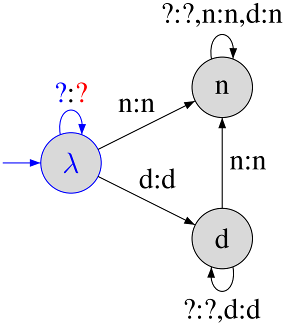
(6) /tu**n**iki**d**i/ $\mapsto$ [tu**n**iki**n**i]



$\rtimes$	t	u	n	i	k	i	d	i	$\ltimes$
	t								

Tier-based Strictly Local functions

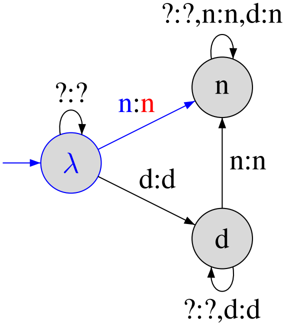
(6) /tu**n**iki**d**i/ $\mapsto$ [tu**n**iki**n**i]



⋊	t	u	n	i	k	i	d	i	⋉
	t	u							

Tier-based Strictly Local functions

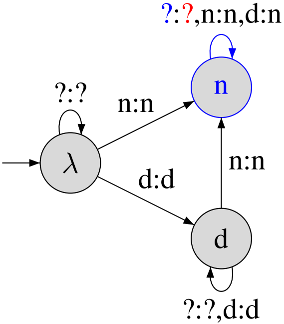
(6) /tu**n**iki**d**i/ $\mapsto$ [tu**n**iki**n**i]



$\rtimes$	t	u	n	i	k	i	d	i	$\ltimes$
	t	u	n						

Tier-based Strictly Local functions

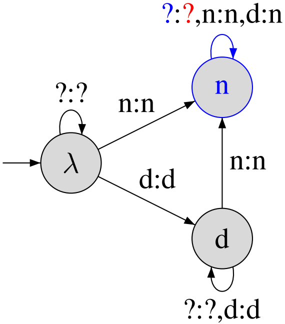
(6) /tu**n**iki**d**i/ $\mapsto$ [tu**n**iki**n**i]



$\rtimes$	t	u	n	i	k	i	d	i	$\ltimes$
	t	u	n	i					

Tier-based Strictly Local functions

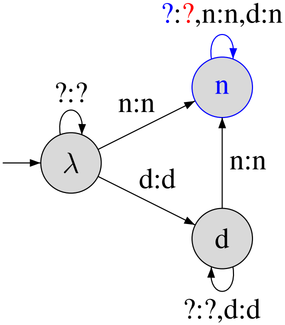
(6) /tu**n**iki**d**i/ $\mapsto$ [tu**n**iki**n**i]



$\rtimes$	t	u	n	i	k	i	d	i	$\ltimes$
	t	u	n	i	k				

Tier-based Strictly Local functions

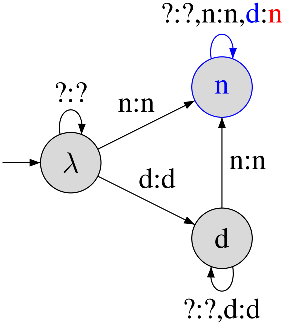
(6) /tu**n**iki**d**i/ $\mapsto$ [tu**n**iki**n**i]



$\rtimes$	t	u	n	i	k	i	d	i	$\ltimes$
	t	u	n	i	k	i			

Tier-based Strictly Local functions

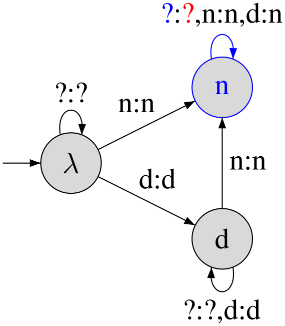
(6) /tu**n**iki**d**i/ $\mapsto$ [tu**n**iki**n**i]



⋊	t	u	n	i	k	i	d	i	⋉
	t	u	n	i	k	i	n		

Tier-based Strictly Local functions

(6) /tu**n**iki**d**i/ $\mapsto$ [tu**n**iki**n**i]



⋊	t	u	n	i	k	i	d	i	⋉
	t	u	n	i	k	i	n	i	

Complexity of phonological maps

- Long-distance phonological processes are conjectured to be Tier-based Strictly Local functions (Chandlee et al., 2017)

REGULAR RELATIONS

↓

SUBSEQUENTIAL FUNCTIONS

↓

TIER-BASED STRICTLY LOCAL FUNCTIONS

↓

STRICTLY LOCAL FUNCTIONS

Main result

- Both types of phonological patterns (phonotactics and processes) belong to subregular classes of formal languages and functions.
  - SL or TSL
- These classes provide a better fit to the typology than the regular languages and relations.

Implications for phonological learning

- The regular relations are not learnable from positive data...
- but the SL languages and functions are (Chandlee et al., 2014; Jardine et al., 2014)!

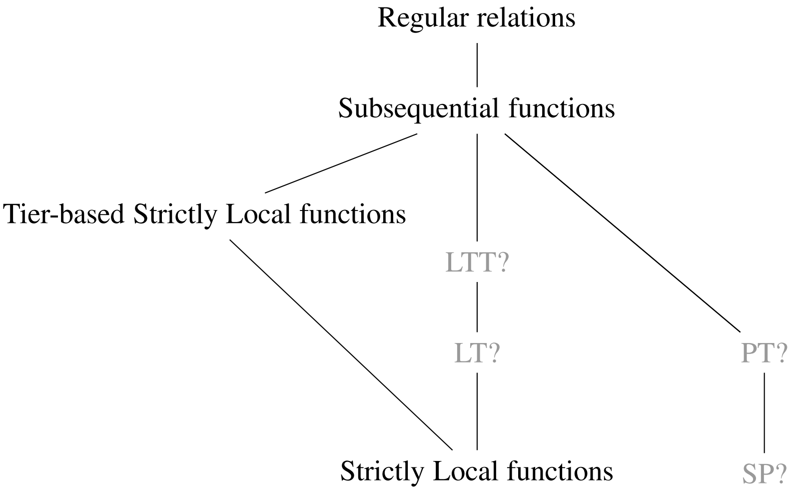
Implications for cognition

- What kind of information must we keep track of when performing phonological computations?
- Subregular analyses suggest it's very limited.

Future work and open questions

- Fill out the hierarchy of subregular functions.

Subregular hierarchy of maps



Future work and open questions

- Fill out the hierarchy of subregular functions.
- Identify logical characterizations of the various classes.
- Test whether subregular classes of FSTs improve efficiency of various NLP/HLT algorithms:
  - grapheme-to-phoneme conversion
  - pronunciation variation
  - etc.

References I

Chandlee, J. (2014). *Strictly Local Phonological Processes*. PhD thesis, University of Delaware.

Chandlee, J., Heinz, J., and Eyraud, R. (2014). Learning strictly local subsequential functions. *Transactions of the Association for Computational Linguistics*, 2:491–503.

Chandlee, J., Heinz, J., Jardine, A., and McMullin, K. (2017). Modeling long-distance alternations with tier-based strictly local functions. Talk given at LSA 2017.

Chomsky, N. (1956). Three models for the description of language. *IRE Transactions on Information Theory 113124*, (IT-2).

Heinz, J. (2007). *The Inductive Learning of Phonotactic Patterns*. PhD thesis, University of California, Los Angeles.

References II

Heinz, J., Rawal, C., and Tanner, H. G. (2011). Tier-based Strictly Local constraints for phonology. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 58–64, Portland, Oregon, USA. Association for Computational Linguistics.

Jardine, A., Chandlee, J., Eyraud, R., and Heinz, J. (2014). Very efficient learning of structured classes of subsequential functions from positive data. In *Proceedings of the 12th International Conference on Grammatical Inference (ICGI 2014)*, JMLR Workshop Proceedings, pages 94–108.

Johnson, C. (1972). *Formal Aspects of Phonological Description*. Mouton, The Hague.

Kaplan, R. and Kay, M. (1994). Regular models of phonological rule systems. *Computational Linguistics*, (20):371–387.

References III

Lee, S. and Pater, J. (2008). Phonological inference and word recognition: Evidence from korean. Ms., Korea University and University of Massachusetts, Amherst.

McMullin, K. (2016). *Tier-based Locality in Long-distance Phonotactics: Learnability and Typology*. PhD thesis, University of British Columbia.

Meinof, C. (1932). *Introduction to the phonology of the Bantu languages*. Berlin: Dietrich Reimer/Ernst Vohsen. Trans. by N. J. van Warmelo.

Mohri, M. (1997). Finite-state transducers in language and speech processing. *Computational Linguistics*, (23):269–311.

Odden, D. (1994). Adjacency parameters in phonology. *Language*, 70(2):289–330.

References IV

Rogers, J., Heinz, J., Fero, M., Hurst, J., Lambert, D., and Wibel, S. (2013). Cognitive and sub-regular complexity. In Morrill, G. and Nederhof, M.-J., editors, *Formal Grammar, Lecture Notes in Computer Science*, volume 8036, pages 90–108. Springer.

Rogers, J. and Pullum, G. (2011). Aural pattern recognition experiments and the subregular hierarchy. *Journal of Logic, Language and Information*, (20):329–342.

Rose, S. and Walker, R. (2004). A typology of consonant agreement as correspondence. *Language*, 80:475–531.

Tesar, B. (2012). Learning phonological grammars for output-driven maps. In Lima, S., Mullin, K., and Smith, B., editors, *NELS 39: Proceedings of the 39th Annual Meeting of the North Eastern Linguistic Society*, pages 785–798, University of Massachusetts Amherst. GLSA.