Graph pattern learning for long-distance phonotactics

Adam Jardine

Dept. of Linguistics & Cognitive Science
University of Delaware

NAPhC 9
May 7, 2016

Introduction

- Languages have long-distance phonotactic patterns, especially in tone (Yip, 2002; Hyman, 2011)
- Dependencies between non-adjacent units can make the learning problem difficult (Hayes and Wilson, 2008)
- How are these patterns learned?

Introduction

- Computationally local string learners form a strong theory of phonotactic learning, including many long-distance patterns (Heinz, 2009, 2010; Jardine and Heinz, 2016)
- Some tone patterns are beyond these learners

Introduction

- These can be learned with a local **autosegmental** learner
- Idea: learn banned **subgraphs**
- Local autosegmental learning provides a strong theory of tone learning
- May be extended to long-distance segmental phonology as well

Computational locality

- **Strictly Local (SL)** stringsets are those describable by a finite set of *banned substrings* (Rogers and Pullum, 2011)

Computational locality

- Kagoshima Japanese
  (Hirayama, 1951; Haraguchi, 1977; Kubozono, 2012)

a.	hána	‘nose’	HL
b.	sakúra	‘cherry blossom’	LHL
c.	kagaríbi	‘watch fire’	LLHL
d.	kagaribí-ga	‘watch fire’ + NOM	LLLHL
			...
e.	haná	‘flower’	LH
f.	usagí	‘rabbit’	LLH
g.	kakimonó	‘document’	LLLH
h.	kakimono-gá	‘document’ + NOM	LLLLH
			...

Computational locality

$$KJ = \{ \begin{array}{ll} \text{\#HL\#}, & \text{\#LH\#}, \\ \text{\#LHL\#}, & \text{\#LLH\#}, \\ \text{\#LLHL\#}, & \text{\#LLLH\#}, \\ \ldots & \end{array} \}$$

- $G_{KJ} = \{\text{HLL, HH, HLH, LL\#}\}$

Computational locality

$$KJ = \{ \begin{array}{ll} \#HL\#, & \#LH\#, \\ \#LHL\#, & \#LLH\#, \\ \#LLHL\#, & \#LLLH\#, \\ \ldots & \end{array} \}$$

- $G_{KJ} = \{$HLL, HH, HLH, LL#$\}$

*#HLLLL#, *#HLLHL#

Computational locality

$$KJ = \{ \#HL\#, \quad \#LH\#, \quad \#LHL\#, \quad \#LLH\#, \quad \#LLHL\#, \quad \#LLLH\#, \quad ... \}$$

- $G_{KJ} = \{$HLL, HH, HLH, LL#$\}$

*#HLLLL#, *#HLLHL#, *#LLHHL#

Computational locality

$$KJ = \{ \begin{array}{ll} \#HL\#, & \#LH\#, \\ \#LHL\#, & \#LLH\#, \\ \#LLHL\#, & \#LLLH\#, \\ \ldots & \end{array} \}$$

- $G_{KJ} = \{$HLL, HH, HLH, LL#$\}$

*#HLLLL#, *#HLLHL#, *#LLHHL#, *#LHLHL#

Computational locality

$$KJ = \{ \begin{array}{ll} \#HL\#, & \#LH\#, \\ \#LHL\#, & \#LLH\#, \\ \#LLHL\#, & \#LLLH\#, \\ ... & \end{array} \}$$

- G_{KJ} = {HLL, HH, HLH, LL#}

*#HLLLL#, *#HLLHL#, *#LLHHL#, *#LHLHL#, *#LLLLL#, ...

Computational locality

Learning SL patterns

- Fix k

Computational locality

Learning SL patterns

- Fix k

$$\texttt{substr}_k(w) = \{u \mid u \text{ is a } k\text{-substring of } w\}$$

Computational locality

Learning SL patterns

- Fix k

$$\texttt{substr}_k(w) = \{u \mid u \text{ is a } k\text{-substring of } w\}$$

- $\texttt{substr}_3(\text{\#LLHL\#}) = \{\text{\#LL, LLH, LHL, HL\#}\}$

Computational locality

Learning SL patterns

- Fix k

$$\texttt{substr}_k(w) = \{u \mid u \text{ is a } k\text{-substring of } w\}$$

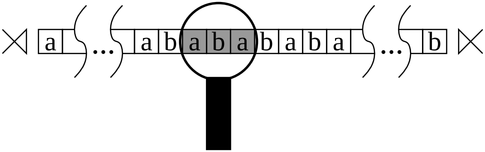
$$\texttt{substr}_k(D) = \bigcup_{w \in D} \texttt{substr}_k(w)$$

- $\texttt{substr}_3(\text{\#LLHL\#}) = \{\text{\#LL, LLH, LHL, HL\#}\}$

Computational locality

Learning SL patterns

- `substr`$_k$ scans each input string with a window of size k



Rogers and Pullum (2011); Rogers et al. (2013)

Computational locality

Learning SL patterns

- Let S_k be the set of all possible k-substrings
- Learner: $G_0 = S_k$
  $G_n = S_k - \texttt{substr}_k(\{d_1, d_2, ..., d_n\})$

Computational locality

Learning SL patterns

- Let S_k be the set of all possible k-substrings
- Learner: $G_0 = S_k$
  $G_n = S_k - \texttt{substr}_k(\{d_1, d_2, ..., d_n\})$
- *KJ*: $k = 3$

  $G_0 =$ { #LL, LLL, LLH, LHL, HLL, HLH, ..., LH#}

Computational locality

Learning SL patterns

- Let S_k be the set of all possible k-substrings
- Learner: $G_0 = S_k$
  $G_n = S_k - \texttt{substr}_k(\{d_1, d_2, ..., d_n\})$
- *KJ*: $k = 3$

G_1 = { #LL, LLL, LLH, LHL, HLL, HLH, ..., LH#}

Time	Datum	$\texttt{substr}_3$
1	#LLHL#	{#LL, LLH, LHL, HL#}

Computational locality

Learning SL patterns

- Let S_k be the set of all possible k-substrings
- Learner: $G_0 = S_k$
  $G_n = S_k - \texttt{substr}_k(\{d_1, d_2, ..., d_n\})$
- *KJ*: $k = 3$

$G_2 = \{$ #LL, LLL, LLH, LHL, HLL, HLH, ..., LH# $\}$

Time	Datum	$\texttt{substr}_3$
1	#LLHL#	{#LL, LLH, LHL, HL#}
2	#LLLH#	{#LL, LLL, LLH, LH#}

Computational locality

Learning SL patterns

- Let S_k be the set of all possible k-substrings
- Learner: $G_0 = S_k$
  $G_n = S_k - \texttt{substr}_k(\{d_1, d_2, ..., d_n\})$
- *KJ*: $k = 3$

$G_3 = G_2 = \{$ #LL, LLL, LLH, LHL, HLL, HLH, ..., LH#$\}$

Time	Datum	$\texttt{substr}_3$
1	#LLHL#	{#LL, LLH, LHL, HL#}
2	#LLLH#	{#LL, LLL, LLH, LH#}
3	#LLLLHL#	{#LL, LLL, LLH, LHL, HL#}

Computational locality

Learning SL patterns

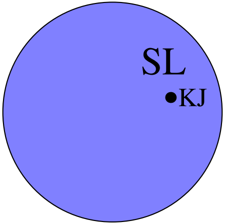
- Let S_k be the set of all possible k-substrings
- Learner: $G_0 = S_k$
  $G_n = S_k - \texttt{substr}_k(\{d_1, d_2, ..., d_n\})$
- *KJ*: $k = 3$

  $G_3 = G_2 = \{$ #LL, LLL, LLH, LHL, HLL, HLH, ..., LH#$\}$

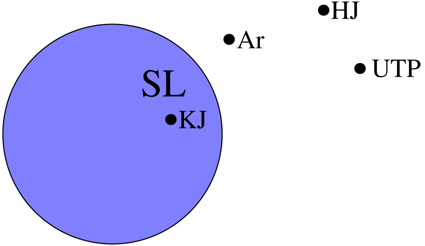
Time	Datum	$\texttt{substr}_3$
1	#LLHL#	{#LL, LLH, LHL, HL#}
2	#LLLH#	{#LL, LLL, LLH, LH#}
3	#LLLLHL#	{#LL, LLL, LLH, LHL, HL#}
...		

- There will always be some n for which G_n describes *KJ* (García et al., 1990; Heinz, 2010, 2011)

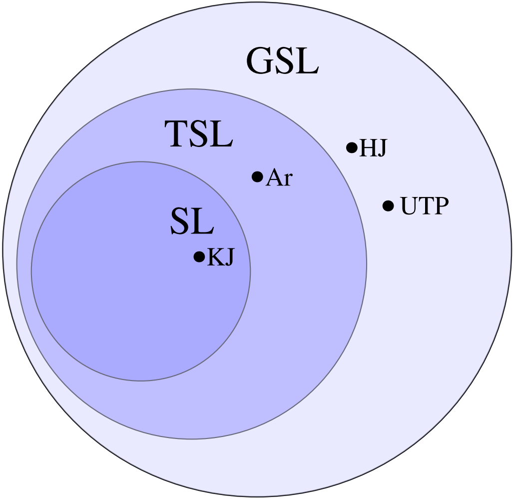
Tone patterns and stringset classes



Tone patterns and stringset classes



Tone patterns and stringset classes



Non-local tone patterns

- Arigibi (Donohue, 1997): **At most one H**

a.	naː L	'finish'	e.	umú LH	'dog'	j.	olaʔolá LLLH	'red'
b.	tutuː LL	'long'	f.	nímo HL	'louse'	k.	tuniʔʌ́ʔʌ LLHL	'all'
c.	vovoʔo LLL	'bird'	g.	mudɛbɛ́ LLH	'claw'	l.	idómai LHLL	'eye'
d.	ɛlaila LLLL	'hot'	h.	ivío LHL	'sun'	m.	núʔʌtama HLLL	'bark'
			i.	ŋgíʔɛpu HLL	'heart'			

Non-local tone patterns

- Arigibi (Donohue, 1997): **At most one H**

a.	naː L	'finish'	e.	umú LH	'dog'	j.	olaʔolá LLLH	'red'
b.	tutuː LL	'long'	f.	nímo HL	'louse'	k.	tuniʔʌ́ʔʌ LLHL	'all'
c.	vovoʔo LLL	'bird'	g.	mudɛbɛ́ LLH	'claw'	l.	idómai LHLL	'eye'
d.	ɛlaila LLLL	'hot'	h.	ivío LHL	'sun'	m.	núʔʌtama HLLL	'bark'
			i.	ŋgíʔɛpu HLL	'heart'			

- **Not** SL
- *H...H; {HH, HLH, HLLH, HLLLH, ...}

Non-local tone patterns

- Arigibi: **At most one H**
- Hirosaki J. (Haraguchi, 1977):
  Exactly one H or F; F word-final

a.	‘chicken’	niwatorí	LLLH
b.	‘lightening’	kaminarî	LLLF
c.	‘fruit’	kudamóno	LLHL
d.	‘trunk’	toránku	LHLL
e.	‘bat’	kóomori	HLLL

Non-local tone patterns

- Arigibi: **At most one H**
- Hirosaki J. (Haraguchi, 1977):
  Exactly one H or F; F word-final

a.	‘chicken’	niwatorí	LLLH
b.	‘lightening’	kaminarî	LLLF
c.	‘fruit’	kudamóno	LLHL
d.	‘trunk’	toránku	LHLL
e.	‘bat’	kóomori	HLLL

- *H...H, *H...F, *L^n, *FL, etc.

Non-local tone patterns

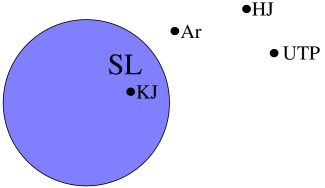
- Arigibi: **At most one H**
- Hirosaki J.: **Exactly one H or F; F word-final**
- Unbounded Tone Plateauing (Hyman, 2011; Jardine, to appear):
  At most one plateau of H

a.	‘chopper’	mutéma	LHL
b.	‘log’	kisikí	LLH
c.	‘log chopper’	mutémá+bísíkí	LHHHHH
d.	″ ″	*mutéma+bisikí	*LHLLLH

(Luganda; Hyman, 2011; Hyman and Katamba, 2010)

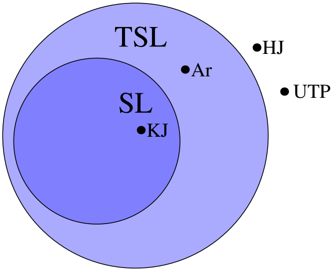
Non-local tone patterns

- Arigibi: **At most one H** (*Ar*)
- Hirosaki J.: **Exactly one H or F; F word-final** (*HJ*)
- Unbounded Tone Plateauing: **At most one plateau of H** (*UTP*)



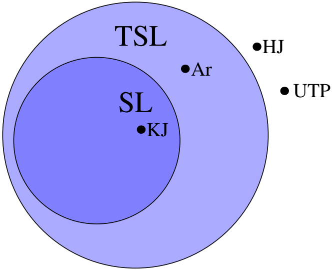
Non-local tone patterns

- **Tier-based SL** (Heinz et al., 2011): Banned substrings evaluated over one subset or **tier** of alphabet
- Arigibi: $G = \{\text{HH}\}$ when L is ignored;
  *HLH, *LLHLLLH, ...



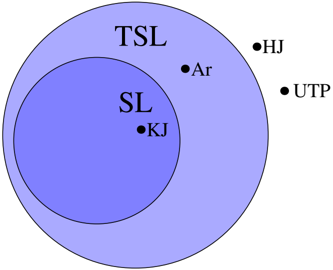
Non-local tone patterns

- **Tier-based SL** (Heinz et al., 2011): Banned substrings evaluated over one subset or **tier** of alphabet
- Arigibi: $G = \{\text{HH}\}$ when L is ignored;
  *HLH, *LLHLLLH, ...



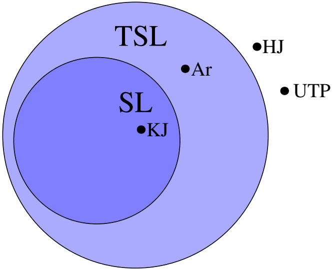
Non-local tone patterns

- **Tier-based SL** (Heinz et al., 2011): Banned substrings evaluated over one subset or **tier** of alphabet
- Arigibi: $G = \{\text{HH}\}$ when L is ignored;
  *HLH, *LLHLLLH, ...



Non-local tone patterns

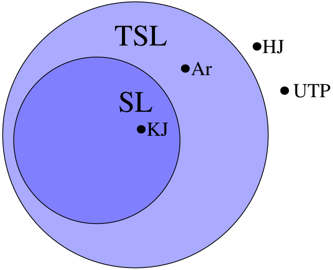
- **Tier-based SL** (Heinz et al., 2011): Banned substrings evaluated over one subset or **tier** of alphabet
- Arigibi: $G = \{\text{HH}\}$ when L is ignored;
  *HLH, *LLHLLLH, ...



- Tier and grammar are learnable (Jardine and Heinz, 2016)

Non-local tone patterns

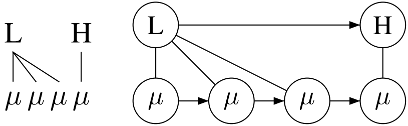
- For *HJ* (exactly one H; F word-final), can't ignore Ls: *FLL...
- For *UTP* (at most one H plateau), can't posit *HH



Computational locality for autosegmental representations

- Autosegmental representations are **graphs** (Goldsmith, 1976; Coleman and Local, 1991)

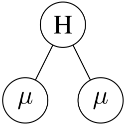
olaʔolá LLLH 'red'
(Arigibi)

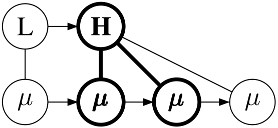


- We can instead consider banned sub**graph** grammars

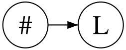
Computational locality for autosegmental representations

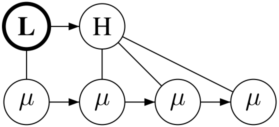
- Let a **subgraph** be some finite, connected piece of a graph





- Subgraphs may refer to boundaries on each tier (not depicted in full graphs)

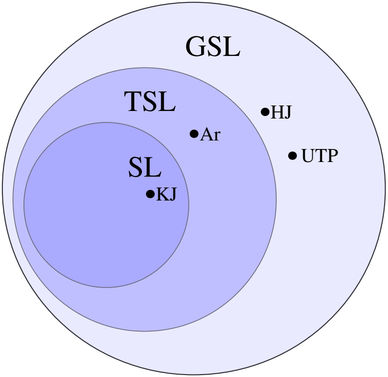




- k is the number of nodes

Computational locality for autosegmental representations

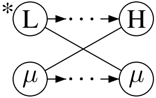
- **Graph SL**: Describable by banned subgraph grammars



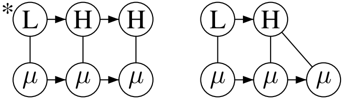
- These grammars are learnable in a similar way to SL grammars

Some assumptions

- Association preserves precedence relations **(the No-Crossing Constraint (NCC))**



- Adjacent nodes on tonal tier cannot be identical **(the Obligatory Contour Principle (OCP)**

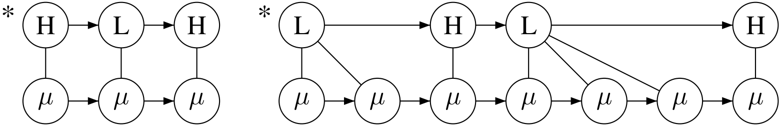


- Such representations can be generated from strings (Jardine and Heinz, 2015)

Banned subgraph analyses

Arigibi: At most one H

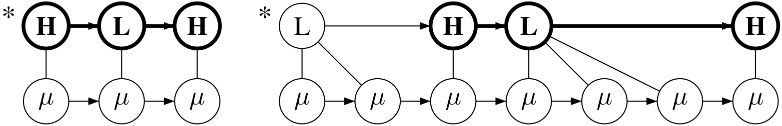
- *HLH, *HLLH, *LLHLLLH, ...

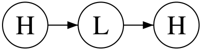


Banned subgraph analyses

Arigibi: At most one H

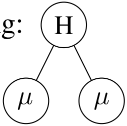
- *HLH, *HLLH, *LLHLLLH, ...

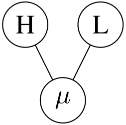


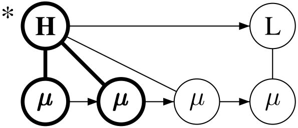
- First banned subgraph: 

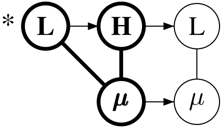
Banned subgraph analyses

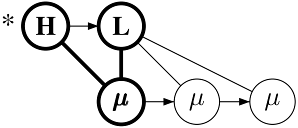
Arigibi: At most one H

- Spreading: 

- Contours: 

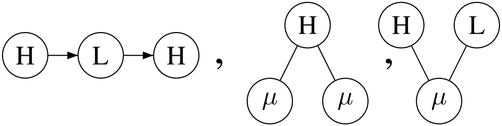
*HHHL 

*RL 

*FLLL 

Banned subgraph analyses

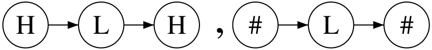
Arigibi: At most one H

- $G_{Ar} = \{$ 

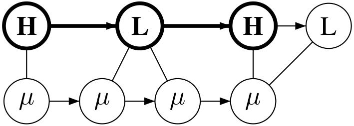
$\}$

Banned subgraph analyses

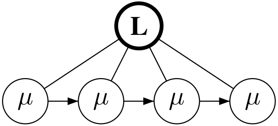
Hirosaki Japanese: Exactly one H or F; F is word-final

- *HLLF, *LLLL
- 

*

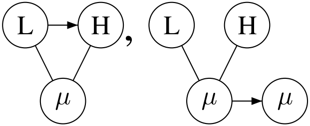


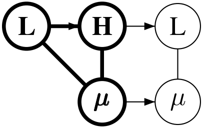
*

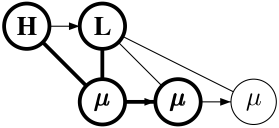


Banned subgraph analyses

Hirosaki Japanese: Exactly one H or F; F is word-final

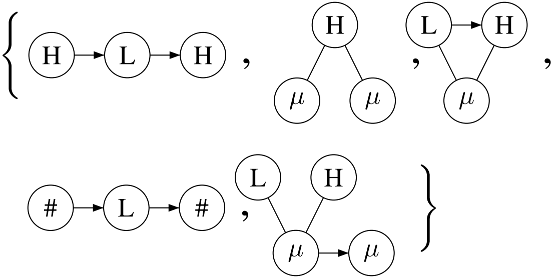
- Contours: 

*RL * 

*FLLL * 

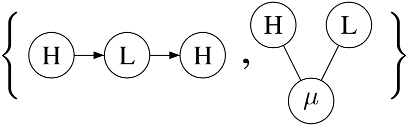
Banned subgraph analyses

Hirosaki Japanese: Exactly one H or F; F is word-final

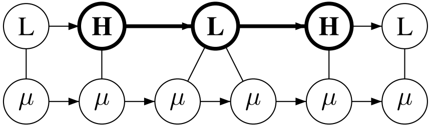
- $G_{HJ} =$ 

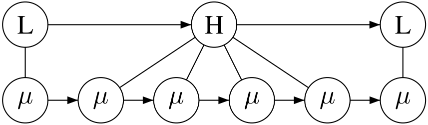
Banned subgraph analyses

UTP: At most one plateau of H

- $G_{UTP} =$ 

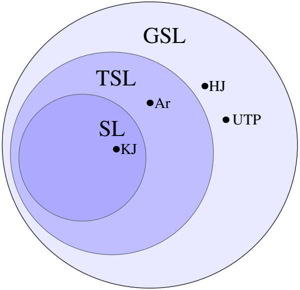
- *LHLLHL, LHHHHL

* 



Banned subgraph analyses

- Arigibi: **At most one H** (*Ar*)
- Hirosaki J.: **Exactly one H or F; F word-final** (*HJ*)
- Unbounded Tone Plateauing: **At most one plateau of H** (*UTP*)



Learning GSL

- Fix k

Learning GSL

- Fix k

$$\texttt{subg}_k(g) = \{s |\ s \text{ is a } k\text{-subgraph of } g\}$$

Learning GSL

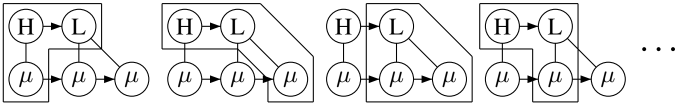
- Fix k

$$\texttt{subg}_k(g) = \{s |\ s \text{ is a } k\text{-subgraph of } g\}$$

$$\texttt{subg}_k(D) = \bigcup_{g \in D} \texttt{subg}_k(g)$$

Learning banned GSL

- Cognitive interpretation of $\texttt{subg}_k$ is the same as $\texttt{substr}_k$: scan input structures, remembering substructures of size k



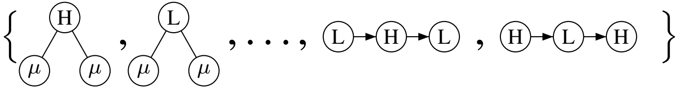
Learning GSL

- Let S_k be the set of all possible k-sub**graphs**
- Learner: $G_0 = S_k$
  $G_n = S_k - \texttt{subg}_k(\{d_1, d_2, ..., d_n\})$

Learning GSL

- Let S_k be the set of all possible k-sub**graphs**
- Learner: $G_0 = S_k$
  $G_n = S_k - \texttt{subg}_k(\{d_1, d_2, ..., d_n\})$
- *Ar*: $k = 3$

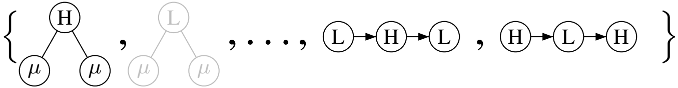
$G_0 =$

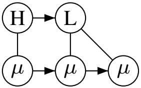
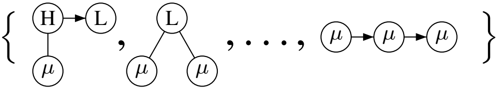


Learning GSL

- Let S_k be the set of all possible k-sub**graphs**
- Learner: $G_0 = S_k$
  $G_n = S_k - \texttt{subg}_k(\{d_1, d_2, ..., d_n\})$
- *Ar*: $k = 3$

$G_1 =$

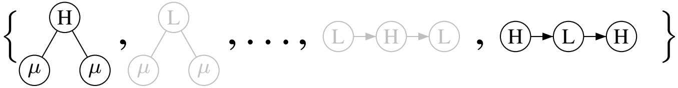


Time	Datum	$\texttt{subg}_3$
1		

Learning GSL

- Let S_k be the set of all possible k-sub**graphs**
- Learner: $G_0 = S_k$

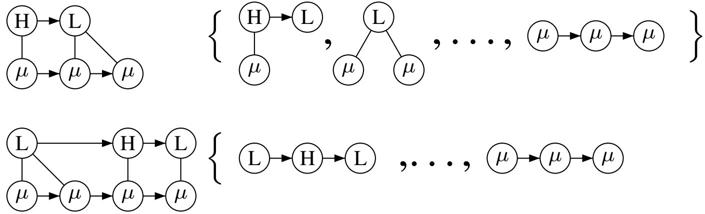
  $G_n = S_k - \texttt{subg}_k(\{d_1, d_2, ..., d_n\})$
- *Ar*: $k = 3$

$G_2 =$ 

Time | Datum | $\texttt{subg}_3$

1

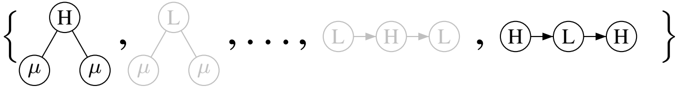
2

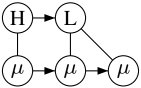
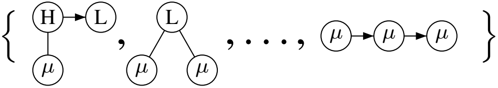
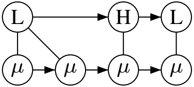


Learning GSL

- Let S_k be the set of all possible k-sub**graphs**
- Learner: $G_0 = S_k$

  $G_n = S_k - \texttt{subg}_k(\{d_1, d_2, ..., d_n\})$
- *Ar*: $k = 3$

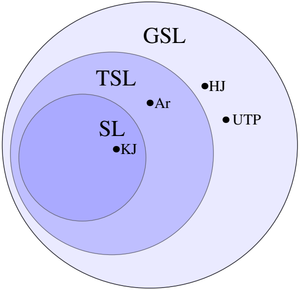
$G_2 =$ 

Time	Datum	$\texttt{subg}_3$
1		
2		

...

Learning GSL

- There will always be some n for which G_n describes Ar
- Searching for k connected subgraphs is tractable (Ferreira, 2013)
- Can learn from string input (Jardine and Heinz, 2015)



Conclusions

- Tone includes many long-distance patterns, including some that are outside of the range of established string-based learners
- A graph-based learner can learn these patterns
- This is thanks to a computational notion of **locality** extended to autosegmental representations
- Future work: how can this result be extended to representations in other domains (segmental, metrical)?

Acknowledgments

Thanks to Jeff Heinz, Rémi Eyraud, and the members of the UD Speech, Phonetics, and Phonology Lab

References I

Coleman, J. and Local, J. (1991). The “No Crossing Constraint” in autosegmental phonology. *Linguistics and Philosophy*, 14:295–338.

Donohue, M. (1997). Tone systems in New Guinea. *Linguistic Typology*, 1:347–386.

Ferreira, R. (2013). *Efficiently Listing Combinatorial Patterns in Graphs*. PhD thesis, Università degli Studi di Pisa.

García, P., Vidal, E., and Oncina, J. (1990). Learning locally testable languages in the strict sense. In *Proceedings of the Workshop on Algorithmic Learning Theory*, pages 325–338.

Goldsmith, J. (1976). *Autosegmental Phonology*. PhD thesis, Massachussets Institute of Technology.

Haraguchi, S. (1977). *The Tone Pattern of Japanese: An Autosegmental Theory of Tonology*. Kaitakusha.

Hayes, B. and Wilson, C. (2008). A maximum entropy model of phonotactics and phonotactic learning. *Linguistic Inquiry*, 39:379–440.

References II

Heinz, J. (2009). On the role of locality in learning stress patterns. *Phonology*, 26:303–351.

Heinz, J. (2010). Learning long-distance phonotactics. *Linguistic Inquiry*, 41:623–661.

Heinz, J. (2011). Computational phonology part I: Foundations. *Language and Linguistics Compass*, 5(4):140–152.

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.

Hirayama, T. (1951). *Kyuusyuu hoogen Onchoo no Kenkyuu (Studies on the Tone of the Kyushu Dialects)*. Tokyo: Gakkai no shinshin-sha.

Hyman, L. (2011). Tone: Is it different? In Goldsmith, J. A., Riggle, J., and Yu, A. C. L., editors, *The Blackwell Handbook of Phonological Theory*, pages 197–238. Wiley-Blackwell.

References III

Hyman, L. and Katamba, F. X. (2010). Tone, syntax and prosodic domains in Luganda. In Downing, L., Rialland, A., Beltzung, J.-M., Manus, S., Patin, C., and Riedel, K., editors, *Papers from the Workshop on Bantu Relative Clauses*, volume 53 of *ZAS Papers in Linguistics*, pages 69–98. ZAS Berlin.

Jardine, A. (to appear). Computationally, tone is different. *Phonology*.

Jardine, A. and Heinz, J. (2015). A concatenation operation to derive autosegmental graphs. In *Proceedings of the 14th Meeting on the Mathematics of Language (MoL 2015)*, pages 139–151, Chicago, USA. Association for Computational Linguistics.

Jardine, A. and Heinz, J. (2016). Learning tier-based strictly 2-local languages. *Transactions of the Association for Computational Linguistics*, 4:87–98.

Kubozono, H. (2012). Varieties of pitch accent systems in Japanese. *Lingua*, 122:1395–1414.

References IV

Rogers, J., Heinz, J., Fero, M., Hurst, J., Lambert, D., and Wibel, S. (2013). Cognitive and sub-regular complexity. In *Formal Grammar*, volume 8036 of *Lecture Notes in Computer Science*, 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.

Yip, M. (2002). *Tone*. Cambridge University Press.