Logical characterizations of phonological well-formedness

Adam Jardine

February 18, 2016

1 Introduction

- What is the character of phonological constraints? One potential focus is the phonetic nature of constraints (see, e.g., Hayes et al., 2004). Another equally valid question is, what is the *computational* nature of constraints?
- What are the well-formedness generalizations in the following tone patterns? High-tones are marked with an acute accent on the vowel (á), low tones unmarked.

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

a.	hána	‘nose’	e.	haná	‘flower’
b.	sakúra	‘cherry blossom’	f.	usagí	‘rabbit’
c.	kagaríbi	‘watch fire’	g.	kakimonó	‘document’
d.	kagaribí-ga	“ + NOM	h.	kakimono-gá	“ + NOM

(2) Arigibi (New Guinea; Donohue, 1997)

a.	naː	‘finish’	e.	umú	‘dog’	h.	olaʔolá	‘red’
b.	tutuː	‘long’	f.	ními	‘louse’	i.	tuniʔʌ́ʔʌ	‘all’
c.	vovoʔo	‘bird’	g.	mudɛbɛ́	‘claw’	j.	idómai	‘eye’
d.	ɛlaila	‘hot’	f.	ivío	‘sun’	k.	núʔʌtama	‘bark’
			g.	ŋgíʔɛpu	‘heart’			

(3) Hirosaki Japanese (somewhat simplified; Haraguchi, 1977)

a.	é	‘handle’	d.	niwatorí	‘chicken’
b.	amé	‘candy’	e.	kudamóno	‘fruit’
c.	áki	‘autumn’	f.	toránku	‘trunk’
			g.	kóomori	‘bat’

- The above patterns can be translated into the following **sets** of well-formed strings of H and L TBUs:

set - collection of unique objects

(4) Kagoshima Japanese:

Well-formed	Ill-formed
H, HL, LH,	L, LL, HH,
LHL, LLH,	LLL, LHH, HLL, HHL, HHH,
LLHL, LLLH, . . . ,	LLLL, LLHH, LHLL, LHLH, LHHL, LHHH, HLLL, . . . ,
LLLLHL, LLLLLH, . . .	LLLHLL, LLLHLH, LLLHHL, LLLHHH, LLHLLL, . . .

(5) Arigibi:

Well-formed	Ill-formed
H, L, LL, LH, HL,	HH, LHH, HLH, HHL,
LLL, LLH, LHL, HLL,	LLHH, LHLH, LHHL, LHHH,
LLLL, LLLH, LLHL, LHLL, HLLL, . . .	HLLH, HLHL, HLHH, HHLL, . . .

(6) Hirosaki Japanese:

Well-formed	Ill-formed
H, HL, LH,	L, LL, HH,
LHL, LLH, HLL,	LLL, LHH, HLH, HHL, HHH,
LLLH, LLHL, LHLL, HLLL,	LLLL, LLHH, LHLH, LHHL, LHHH, HLLH, . . . ,
LLLLH, LLLHL, LLHLL, . . .	LLLLL, LLLHH, LLHLH, . . . , HLLLH, . . .

- Here are a couple of other logically possible well-formedness patterns.

(7) Language X

Well-formed	Ill-formed
L, H, LL, HH,	HL, LH,
LLL, LHL, HLH, HHH	LLH, LHH, HLL, HHL,
LLLL, LLHL, LHLL, LHHL,	LLLH, LLHH, LHHH, LHLH,
HLLH, HLHH, HHLH, HHHH,	HLLL, HLHL, HHLL,
LLLLL, LLLHL, LLHLL, LLHHL,	LLLLH, LLLHH, LLHLH, LLHHH,
LHLLL, LHLHL, LHHHL, HLLLH, . . .	LHLLH, . . . , HLLLL, HLLHL, . . .

(8) Language Y

Well-formed	Ill-formed
L, LL, HH, LLL, LHH, HLH, HHL,	H, HL, LH, HLL, LHL, LLH, HHH,
LLLL, LLHH, LHLH, LHHL,	LLLH, LLHL, LHLL,
LHHL, HLHL, HHLL, HHHH,	LHHH, HLLL, HHHL,
LLLLL, LLLHH, LLHLH, LLHHL,	LLLLH, LLLHL, LLHLL, LLHHH,
LHLLH, LHLHL, LHHLL, LHHHH,	LHLLL, LHLHH, LHHLH, LHHHL,
HLHHH, HHLHH, HHHLH, . . .	HLLLL, HLLHH, HLHHL, . . .

- Why are (4)–(6) attested but not Language X and Y?
- This talk is on the basics of applying **logic** to develop an *explicit* and *restrictive* theory of constraints based on their computational properties.
  - We get a clear understanding that *representation really matters* and a precise way of studying the relationship between representation and expressivity
  - These characterizations are related to other computational characterizations, and thus can inform explicit and restrictive theories of phonological transformations and phonological learning.

2 Basic string models

2.1 String sets

- A **string** is a sequence of symbols taken from an alphabet Σ. Let Σ^* denote the set of *all possible* strings over Σ. This set includes λ, the **empty string**, or string of length 0. Let $w \cdot v$ denote the concatenation of two strings w and v.

string, Σ, Σ^*
empty string (λ)
$w \cdot v$

- All of the strings in (6) through (8) are from Σ^* where $\Sigma = \{\text{H, L}\}$. The patterns of *well-formed* strings in (6) through (8) are **subsets** of Σ^*. The problem, then, is how to distinguish the set of well-formed strings from the rest of Σ^* (i.e., the possible, yet ill-formed, strings). Here, we will see how to do this with logical statements referring to properties of the strings.

subset - a set X is a subset of Y if all the members of X are also members of Y

- We will need to use the word boundary symbols $\rtimes$ and $\ltimes$ to mark the beginning and ends of strings. Let $\rtimes\Sigma^*\ltimes$ refer to the strings in Σ^* delineated with $\rtimes$ and $\ltimes$. For example, LLHL is a string in Σ^*, so $\rtimes$LLHL$\ltimes$ is a string in $\rtimes\Sigma^*\ltimes$.

$\rtimes$, $\ltimes$
$\rtimes\Sigma^*\ltimes$

2.2 Substructures

- The basic property of a **structure** we will use to create a logical language is whether or not it contains a particular **substructure**. We'll look at two kinds of substructures over strings. The first are **substrings**.

structure - a set of elements and relations between those elements
substructure

Definition 1 (substring) *A string u is a* substring *of another string w iff $w = v_1 \cdot u \cdot v_2$, where v_1 and v_2 are any other two strings (including λ).*

substring

- In other words, a substring is a contiguous 'chunk' of another string. For example, $\rtimes$L is a substring of $\rtimes$LLHL$\ltimes$, because $\rtimes$LLHL$\ltimes$= $\lambda\cdot\rtimes$L$\cdot$LHL$\ltimes$.

(9) What are some other substrings of $\rtimes$LLHL$\ltimes$?

- A theory of tonal well-formedness based on *substrings* of strings over $\{H, L\}$ claims that humans pay attention to contiguous chunks of TBUs of some finite size, focusing on whether each TBU is H or L.

3 A simple logic

3.1 Conjunctions of negative literals

- A **logic** is a set of statements built through taking basic statements referring to properties of structures and connecting with logical connectives (e.g., $\wedge$, $\vee$, $\neg$) and, for some logics, quantifiers. **logic**
- A logical statement can be interpreted as a *grammar* specifying a subset of structures that satisfy the statement.
- We'll focus on **conjunctions of negative literals**, or logical statements which specify a list of banned substructures. This very restrictive kind of logic is a sublogic of full propositional logic over structures, in which the basic statements or **literals** are substructures. An example where the substructures are substrings is given in (10).

(10) $\neg$HL$\wedge\neg$HH

A more rigorous definition of literals using **model theory** are given in the Appendix, but for the current purposes an informal definition is sufficient.

Definition 2 (literal) *Given a set S of structures, a statement ϕ is a* literal *if it is a substructure of some structure in S.* **literal**

- This is a very general definition, and we'll see in a moment how to vary the kinds of structures and the kinds of substructures we talk about. First, let's focus on strings and substrings.

Definition 3 (substring literal) *Given an alphabet Σ, a statement ϕ is a* substring literal *if it is a substring of some string in $\rtimes\Sigma^*\ltimes$.* **substring literal**

(11) Some substring literals assuming again the alphabet is {L,H} are...

Definition 4 (Satisfaction of a literal) *For a literal ϕ, a structure s* satisfies *ϕ (written $s \models \phi$) if ϕ is a substructure of s.*

satisfaction ($\models$)

(12) What are some substring literals that $\rtimes$LLHL$\ltimes$ satisfies? What are some literals that it doesn't satisfy?

Definition 5 (Conjunction of negative literals) *A statement ϕ is a* conjunction of negative literals (CNL) *if it is of the form*

conjunction of negative literals (CNL)

$$\neg\ell_1 \wedge \neg\ell_2 \wedge \neg\ell_3 \wedge ... \wedge \neg\ell_n$$

where each ℓ_i is a literal.

- The symbols $\neg$ and $\wedge$ are the standard Boolean 'not' and 'and' connectives. A statement $\neg\ell_1 \wedge \neg\ell_2 \wedge \neg\ell_3 \wedge ... \wedge \neg\ell_n$ thus means "Don't contain any of s_1 through s_n as substructures." Formally,

Definition 6 (Satisfaction of a CNL) *For a CNL $\phi = \neg\ell_1 \wedge \neg\ell_2 \wedge \neg\ell_3 \wedge ... \wedge \neg\ell_n$ and a structure s, $s \models \phi$ iff for each ℓ_i, $s \not\models \ell_i$ (s does* not *satisfy ℓ_i).*

(13) What is a CNL that *all* and *only* well-formed strings in Kagoshima Japanese satisfy?

(14) Is there a CNL that describes *all* and *only* the well-formed strings in Language X?

3.2 CNLs and computation

- A CNL thus can serve as a *grammar* which picks out, from a universal set of structures, a subset of structures that are well-formed according to the CNL. With substring literals, the sets of strings describable by CNLs are the *Strictly Local* (SL) formal languages (McNaughton and Papert, 1971). SL is a *natural* class of patterns in that there are other characterizations of them. For example, the SL languages are describable by tiling grammars (Rogers et al., 2013) and finite-state acceptors whose states correspond to substrings of size $k-1$ for some k (McNaughton and Papert, 1971).
- Furthermore, there is a clear cognitive interpretation of evaluating the well-formedness of a string with respect to a CNL: scanning through the string with a window of a fixed size (Rogers et al., 2013), as illustrated in Fig. 1. It is this nature that allowes SL languages to be learned from positive data (García et al., 1990; Heinz, 2010b).

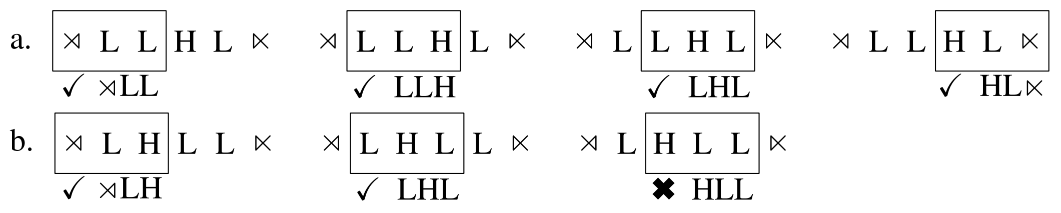
a. ⋊ L L H L ⋉ ⋊ L L H L ⋉ ⋊ L L H L ⋉ ⋊ L L H L ⋉
✓ ⋊LL
✓ LLH
✓ LHL
✓ HL⋉
b. ⋊ L H L L ⋉ ⋊ L H L L ⋉ ⋊ L H L L ⋉
✓ ⋊LH
✓ LHL
✖ HLL

Figure 1: Evaluating ⋊LLHL⋉ and ⋊LHLL⋉

- The above characteristics draw from the *restrictiveness* of CNLs as a logical language. CNLs are at the least expressive of a long list of logical grammars, each more powerful than the next.
- For example, **propositional logic**, which allows literals to be combined with the full set of boolean connectives ($\wedge, \vee, \neg, \rightarrow$), can capture patterns that CNLs cannot. The language X pattern is one of those patterns; its propsitional statement is given in (15). **propositional logic**

(15) $\big((\rtimes\text{H} \rightarrow \text{H}\ltimes) \wedge (\text{H}\ltimes \rightarrow \rtimes\text{H})\big) \wedge \big((\rtimes\text{L} \rightarrow \text{L}\ltimes) \wedge (\text{L}\ltimes \rightarrow \rtimes\text{L})\big)$

- We can thus imagine CNLs—and the patterns they describe—at the bottom of an axis of logical expressiveness, as Figure 2 begins to depict. However, it must be emphasized that we are still talking about *substring* literals, and that expressivity also changes when we vary the structure. Figure 2 will be filled out in more detail momentarily, but first, let us take what we've learned about CNLs and expressiveness and apply it to phonology.

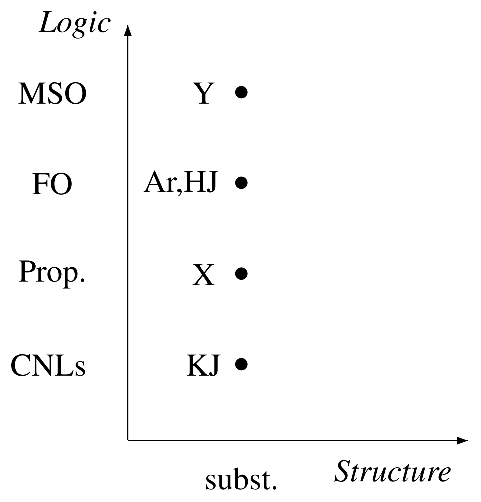

Figure 2: Logics and representations (I)

3.3 CNLs and phonology

- Comparing CNLs to propositional logic teaches us that *negative* constraints—in the sense of forbidding substructures—are less expressive than positive ones (for more discussion see Jardine and Heinz, to appear). As a theory of phonological well-formedness, CNLs thus provide a *restrictive* theory of constraints that also comes with models of cognition and learning.
- The Kagoshima Japanese pattern was one example of a well-formedness generalization that can be captured with CNLs. Many 'local' phonological patterns can be captured with these kind of constraints (Heinz, 2007, 2009; Rogers et al., 2013).

(16) What are some other well-formedness generalizations or constraints that can be described with CNLs (you can vary the alphabet)?

- However, CNLs cannot capture 'long-distance' patterns like those of Arigibi or Hirosaki Japanese.

(17) What is the intuition behind why these generalizations can't be captured by CNLs? How is it different than why Language X and Language Y can't be captured by CNLs?

- We want to capture Arigibi and Hirosaki Japanese without letting in patterns like Language X (or Language Y). Thus, the best way to do this is not increase the power of the logic but *enrich the structure* (Jardine and Heinz, to appear).

4 Varying structure

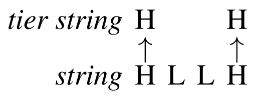

Fig. 3: A tier for H TBUs

4.1 Substructures over tiers

- One way of doing this is to use the linguistic insight of *relativized locality* (Nevins, 2010), or the idea that humans pay attention to the relative order of units sharing certain characteristics, ignoring units which do not.
- To formalize this, we can define a **tier** T as a subset of the alphabet Σ. To formalize the idea of considering information 'relative' to the tier we can define a function $erase_T$ which takes a string and removes all symbols not on T.

tier

Definition 7 **(**$erase_T$**)** *(Heinz et al., 2011) Given* Σ *and* T*, for a string* w *in* Σ^**,*

$erase_T$

$$erase_T(w) \overset{def}{=} \begin{cases} \lambda & \text{if } w = \lambda \\ erase_T(u) \cdot t & \text{if } w = u \cdot t \ \ (u \text{ is in } \Sigma^* \text{ and } t \text{ is in } T) \\ erase_T(u) & \text{if } w = u \cdot v \ \ (v \text{ contains no members of } T) \end{cases}$$

- Essentially, $erase_T(w)$ represents w with all of the non-tier information ignored. For example, if $T = \{\text{H}\}$, then $erase_T(\text{HLLH}) = \text{HH}$.
- We can then define **tier substrings** as substrings relative to this tier.

Definition 8 (tier-substring) *Given an alphabet* Σ *and a tier* T*, a a string* u *in* $\rtimes T^* \ltimes$ *is a* tier substring *of a string* w *in* Σ^* *iff* u *is a substring of* $\rtimes \cdot erase_T(w) \cdot \ltimes$.

tier substring

- Essentially, u is a tier-substring of w if it is a substring of w when everything in w not in T is ignored. For example, if $\Sigma = \{\text{H}, \text{L}\}$ and $T = \{\text{H}\}$, then $\rtimes$H is a tier-substring of LLHL because $erase_T(\text{LLHL}) = \rtimes \text{H} \ltimes$, of which $\rtimes$H is a substring. Importantly, note that if $T = \Sigma$, then tier substrings are just substrings (thus, tier substrings are a generalization of substrings).

(18) Let $T = \{\text{H}\}$. What are the tier substrings of LHLHLH?

- A theory of tonal well-formedness based on *tier substrings* of strings over {H, L} when the tier is {H} claims that humans pay attention to the sequences of H TBUs, ignoring the intervening material.

Definition 9 (tier-substring literal) *For a tier T, a statement ϕ is a* tier-substring literal *if it is a substring of some string in $\rtimes T^* \ltimes$.* **tier substring literal**

(19) Some substring literals assuming again the alphabet is {L,H} are...

- We can then use the definition of satisfaction of literals and CNLs as we did before, but use *tier substrings* (instead of substrings) as the relevant substructure.

(20) How can we describe the Arigibi pattern with CNLs over the tier $T = \{\text{H}\}$?

(21) How can we describe the Hirosaki Japanese pattern?

4.2 Tier CNLs, computation, and phonology

- CNLs where the literals are tier substrings describe the *Tier-based Strictly Local* languages (TSL; Heinz et al., 2011; Jardine and Heinz, accepted). Like the Strictly Local languages, the Tier-based Strictly Local languages have a clear cognitive interpretation (scanning over a tier), and they are also learnable, even if the tier is not known to the learner in advance (Jardine and Heinz, accepted).
- With tier substrings, we get a much neater picture of phonological well-formedness conditions: attested patterns are describable by CNLs, while the unattested patterns still require more complex logics. Figure 4 fills out this chart. Note that Arigibi (Ar) and Hirosaki Japanese (HJ) require the very powerful First Order (FO) logic when considering strict locality, but they can be described with CNLs when we allow for tier locality.
- In this way, a simple logic with an enriched representation gives the best fit to phonology. In fact, at least for segmental phonology, we get a very good fit to the typology (Heinz et al., 2011; McMullin and Hansson, to appear).

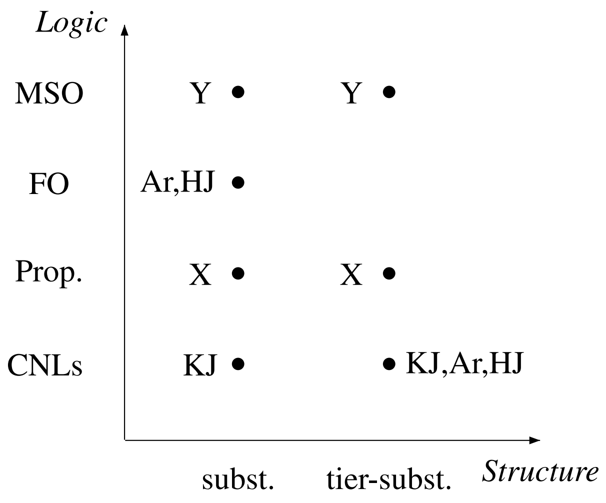

Figure 4: Logics and representations (II)

4.3 Varying the structure further

- Tone, however, is more complex than segmental phonology (Jardine, to appear). An example is the full Hirosaki Japanese pattern, given below in (3).

(22) Hirosaki Japanese (Haraguchi, 1977, pp. 76–7)

Noun	Isolation	+NOM	Noun	Isolation
a. ‘handle’	é	e-gá	f. ‘chicken’	niwatorí
	H	LH		LLLH
b. ‘picture’	ê	é-ga	g. ‘lightening’	kaminarî
	F	HL		LLLF
c. ‘candy’	amé	ame-gá	h. ‘fruit’	kudamóno
	LH	LLH		LLHL
d. ‘rain’	amê	amé-ga	i. ‘trunk’	toránku
	LF	LHL		LHLL
e. ‘autumn’	áki	áki-ga	j. ‘bat’	kóomori
	HL	HLL		HLLL

(23) What is the generalization in (22)? Is this describable with CNLs where the literals are tier-substrings? Why or why not?

- A major goal of my disseration (and the point of my talk tomorrow) is to define another function like $erase_T$ that lets us look at the autosegmental structure of tones, and to study how expressivity of CNLs changes when we consider the literals to be substructures of autosegmental representations. Spoiler: CNLs over autosegmental structures can also capture patterns like Hirosaki Japanese, and in general give a very good fit to tonal typology.

5 To review

- Logics and representations define a space of constraint definition languages (de Lacy, 2011) that are well-defined and whose relative expressivity can be studied.
- CNLs represent a type of constraint that is *local* in a well-defined way. Increasing the logical power very quickly increases our expressive power (c.f. propositional logic).
- The nature of phonological well-formedness is thus predominantly *local*—CNLs, whether over strings or tiers, do a very good job of describing the attested variation. When we run into patterns that we cannot describe, *it is a more restrictive choice to vary the representation rather than the logical power*.

Future work

- While our understanding of varying the logic is well-understood (Büchi, 1960; Rogers et al., 2013; Thomas, 1982), much more could be understood about varying the representation.
- We have left out maps – what is the nature of FAITHFULNESS constraints?
  - Automata-theoretic characterizations of SL maps exist (Chandlee, 2014; Chandlee et al., 2014)
  - Engelfriet and Hoogeboom (2001) and Courcelle et al. (2012) give a model-independent way of specifying maps using MSO. What does the chart in Fig. 4 look like using their maps? How do we define CNL maps?
  - Another approach is to encode faithfulness directly in the model (Jardine, 2016; Potts and Pullum, 2002)
- Previous work has shown how CNLs are learnable (García et al., 1990; Heinz, 2010a, 2011; Jardine and Heinz, accepted). What is the model-general CNL learning algorithm (Jardine and Heinz, LSA)?
- How can *weighted logics* be applied to gradient phonological generalizations?

References

Büchi, J. R. (1960). Weak second-order arithmetic and finite automata. *Zeitschrift für Mathematische Logik und Grundlagen der Mathmatik*, 6:66—92.

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

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

Courcelle, B., Engelfriet, J., and Nivat, M. (2012). *Graph structure and monadic second-order logic: A language-theoretic approach*. Cambridge University Press.

de Lacy, P. (2011). Markedness and faithfulness constraints. In Oostendorp, M. V., Ewen, C. J., Hume, E., and Rice, K., editors, *The Blackwell Companion to Phonology*. Blackwell.

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

Engelfriet, J. and Hoogeboom, H. J. (2001). MSO definable string transductions and two-way finite-state transducers. *ACM Transations on Computational Logic*, 2:216–254.

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.

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

Hayes, B., Kircher, R., and Steriade, D. (2004). *Phonetically based phonology*. Cambridge, UK: Cambridge University Press.

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

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

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

Heinz, J. (2010b). String extension learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 897–906. Association for Computational Linguistics.

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.

Jardine, A. (2016). *Locality and non-linear representations in tonal phonology*. PhD thesis, University of Delaware. In preparation.

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

Jardine, A. and Heinz, J. (accepted). Learning tier-based strictly 2-local languages. *Transactions of the Association of Computational Linguistics*.

Jardine, A. and Heinz, J. (to appear). Markedness constraints are negative: an autosegmental constraint definition language. In *Proceedings of the 51st Annual Meeting of the Chicago Linguistics Society (CLS 2015)*.

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

McMullin, K. and Hansson, G. O. (to appear). Long-distance phonotactics as Tier-Based Strictly 2-Local languages. In *Proceedings of the Annual Meeting on Phonology 2015*.

McNaughton, R. and Papert, S. (1971). *Counter-Free Automata*. MIT Press.

Nevins, A. (2010). *Locality in Vowel Harmony*. Number 55 in Linguistic Inquiry Monographs. MIT Press.

Potts, C. and Pullum, G. K. (2002). Model theory and the content of OT constraints. *Phonology*, 19:361–393.

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.

Thomas, W. (1982). Classifying regular events in symbolic logic. *Journal of Computer and Systems Sciences*, 25:360—376.

Appendix

1 Models

- A **model** is an explicit representation of the information in a structure. Strings over $\Sigma = \{\text{H,L}\}$ can be described with models of the **signature**

model

signature - general shape of a set of models

$\mathcal{S}^{\lhd}$

(24) $$\mathcal{S}^{\lhd} = \langle D, \lhd, P_{\mathrm{H}}, P_{\mathrm{L}} \rangle$$

where D is a set of positions, $\lhd$ is a binary **successor** relation, and P_{H} and P_{L} are unary relations (=sets) showing which positions are occupied by a H symbol and which are occupied by a L symbol, respectively. Let's assume for now that P_{H} and P_{L} are disjoint but that every member of D is in either P_{H} or P_{L}.

successor ($\lhd$) - total order relating each object with the immediately following object

For example, the following is a model of string LLHL:

(25) $$\Big\langle \ \{1,2,3,4\}_D,\ \{(1,2),(2,3),(3,4)\}_{\lhd},\ \{3\}_{P_{\mathrm{H}}},\ \{1,2,4\}_{P_{\mathrm{L}}} \ \Big\rangle$$

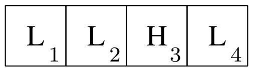

Fig. 5: The information in string HLHL

- *Any* string in Σ^* can be modeled by some model with the signature $\mathcal{S}^{\lhd}$ in (24), and *every* model of the signature $\mathcal{S}^{\lhd}$ in (24) describes some string over Σ^*.
- As a theory of representation, (24) claims that, for at least tone patterns, humans pay attention to whether each TBU is H or L, and the immediate successor relation between each TBU in a word.
- We're also going to want to refer to word edges. Let's revise $\mathcal{S}^{\lhd}$ to include the boundary symbols $\rtimes$ and $\ltimes$. So HLHL is now $\rtimes$HLHL$\ltimes$, whose model is given below in (27.)

(26) $$\mathcal{S}^{\lhd} = \langle D, \lhd, P_{\mathrm{H}}, P_{\mathrm{L}}, P_{\rtimes}, P_{\ltimes} \rangle$$

(27) $$\Big\langle \ \{0,1,2,3,4,5\}_D,\ \{(0,1),(1,2),(2,3),(3,4),(4,5)\}_{\lhd}, \\ \{1,3\}_{P_{\mathrm{H}}},\ \{2,4\}_{P_{\mathrm{L}}},\ \{0\}_{P_{\rtimes}}, \{5\}_{P_{\ltimes}} \ \Big\rangle$$

2 Substructures

- For a structure s, another structure r is a **substructure** of s if there is a mapping from the domain of r to the domain of s such that all of the relations in the signature are maintained. For example, the string LH is a substructure of LLHL because the positions in LH can be mapped by f as defined below to positions 3 and 4 in ⋊LLHL⋉, which have the same properties.

substructure

(28) LH = $\big\langle \{0,1\}_D, \{(0,1)\}_{\triangleleft}, \{1\}_{P_{\mathrm{H}}}, \{0\}_{P_{\mathrm{L}}}, \{\}_{P_{\rtimes}}, \{\}_{P_{\ltimes}} \big\rangle$

(29) $f(0) = 2, f(1) = 3$

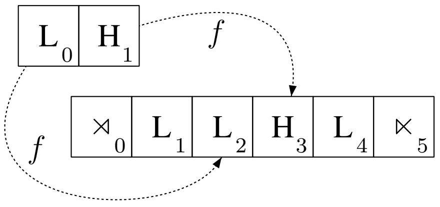

Fig. 6: Mapping LH to LLHL

A **substring** is a substructure of a string when modeled with the signature $\mathcal{S}^{\triangleleft}$.

substring

3 Tiers

- In model theory, we can define tier-substring literals through a second **tier successor** ordering relation $\triangleleft_T$ that keeps track of the precedence only between members of the tier. Let us call the new signature with this relation $\mathcal{S}^{\triangleleft_T}$.

tier successor ($\triangleleft_T$)

(30) $$\mathcal{S}^{\triangleleft_T} = \langle D, \triangleleft, \triangleleft_T, P_{\mathrm{H}}, P_{\mathrm{L}}, P_{\rtimes}, P_{\ltimes} \rangle$$

$\mathcal{S}^{\triangleleft_T}$

- For example, if $T = \{\mathrm{H}\}$, then in the string ⋊HLLHL⋉, the first H precedes the second H with respect to $\triangleleft_T$.

(31) $\big\langle \{0,1,2,3,4,5,6\}_D, \{...\}_{\triangleleft}, \{(0,1),(1,4),(4,6)\}_{\triangleleft_T}, \{1,4\}_{P_{\mathrm{H}}}, \{2,3,5\}_{P_{\mathrm{L}}}, \{0\}_{P_{\rtimes}}, \{6\}_{P_{\ltimes}} \big\rangle$

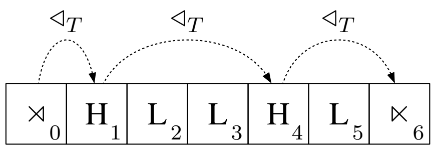

Figure 7: ⋊HLLHL⋉ as a model in $\mathcal{S}^{\triangleleft_T}$

- Tier substrings then include only the $\triangleleft_T$, and not $\triangleleft$. Under this notion, when $T = \{\mathrm{H}\}$, ⋊H and HH count as substructures, but LH does not. The models for ⋊H and HH are depicted in Fig. 8.

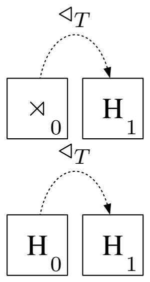

Fig. 8: ⋊H and HH as substructures for $\mathcal{S}^{\triangleleft_T}$