Tier-based Strictly Local Constraints for Phonology[1]

Jeffrey Heinz[1] Chetan Rawal[2] Herbert G. Tanner[2]

[1]Department of Linguistics and Cognitive Science

[2]Department of Mechanical Engineering

Association for Computational Linguistics
Portland, Oregon, June 21, 2011

[1]Supported by NSF under CPS#1035577

Allomorphy in Phonology

Latin Liquid Dissimilation (Jensen 1974, Odden 1994). Which morpheme do the stems take: [aris] or [alis]?

a.	nav-alis	‘naval’	d.	sol-aris	‘solar’
b.	episcop-alis	‘episcopal’	e.	lun-aris	‘lunar’
c.	infiti-alis	‘negative’	f.	milit-aris	‘military’

What’s happening here?

g.	flor-alis	‘floral’	*flor-aris
h.	sepulkr-alis	‘funereal’	*sepulkr-aris
i.	litor-alis	‘of the shore’	*litor-aris

Allomorphy in Phonology

Latin Liquid Dissimilation (Jensen 1974, Odden 1994). Which morpheme do the stems take: [aris] or [alis]?

a.	nav-alis	‘naval’	d.	sol-aris	‘solar’
b.	episcop-alis	‘episcopal’	e.	lun-aris	‘lunar’
c.	infiti-alis	‘negative’	f.	milit-aris	‘military’

What’s happening here?

g.	flor-alis	‘floral’	*flor-aris
h.	sepulkr-alis	‘funereal’	*sepulkr-aris
i.	litor-alis	‘of the shore’	*litor-aris

Allomorphy in Phonology

Latin Liquid Dissimilation (Jensen 1974, Odden 1994). Which morpheme do the stems take: [aris] or [alis]?

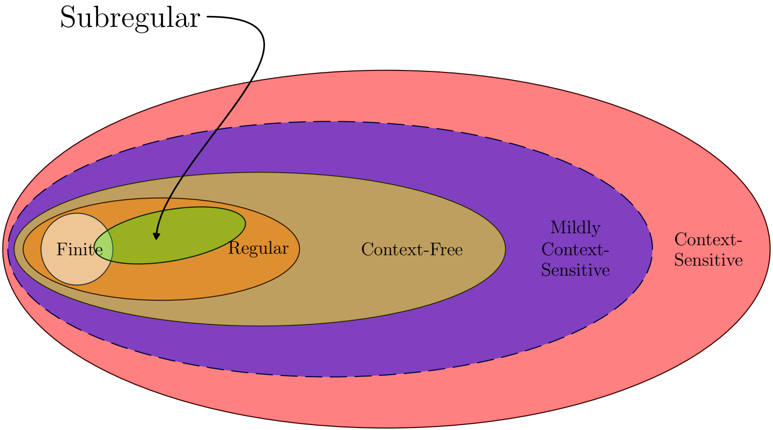
a.	nav-alis	'naval'	d.	sol-aris	'solar'
b.	episcop-alis	'episcopal'	e.	lun-aris	'lunar'
c.	infiti-alis	'negative'	f.	milit-aris	'military'

What's happening here?

g.	flor-alis	'floral'	*flor-aris
h.	sepulkr-alis	'funereal'	*sepulkr-aris
i.	litor-alis	'of the shore'	*litor-aris

1. Theories of phonological tiers
   (Goldsmith 1976, Clements 1976, McCarthy1979, Poser 1982, Prince 1984, Mester 1988, Odden 1994, Archangeli and Pulleyblank 1994, Clements 1995)
2. Such constraints can be be captured by subregular languages

What is subregular?



There is room at the bottom

1. Better characterizations of phonological patterns.
   - Many regular patterns are not phonological: words must have an even number of sibilants, etc.
   - But virtually all phonological patterns are regular! (Johnson 1972, Kaplan and Kay 1994)
2. Factoring and composition with lower complexity
   - When intersecting *arbitrarily many* arbitrary regular sets, complexity grows exponentially.
   - What about intersection of arbitrarily many sets *from some well-defined subregular region*? (cf. Eisner 1997)
3. Learning
   - Under many definitions of "learning", there is either no algorithm which can learn any regular set or only NP-hard ones which can (Vapnik 1998, Jain et al. 1999).
   - What about learning only the *sets in some well-defined subregular region*?

Tiers: Ignoring inconsequential events

1. A tier T is a subset of Σ.
2. Latin Allomorphy: Ignoring all the non-liquid sounds, $l\ l$ and $r\ r$ sequences are forbidden.

Definition

The erasing (projection) function:

$$E_T(\sigma_1 \cdots \sigma_n) = u_1 \cdots u_n$$

where $u_i = \sigma_i$ iff $\sigma_i \in T$ and $u_i = \lambda$ otherwise

Example

If $\Sigma = \{a, b, c\}$ and $T = \{b, c\}$ then

$$E_T(aabaaacaaabaa) = bcb$$

Tiers: Ignoring inconsequential events

1. A tier T is a subset of Σ.
2. Latin Allomorphy: Ignoring all the non-liquid sounds, $l\ l$ and $r\ r$ sequences are forbidden.

Definition

The erasing (projection) function:

$$E_T(\sigma_1 \cdots \sigma_n) = u_1 \cdots u_n$$

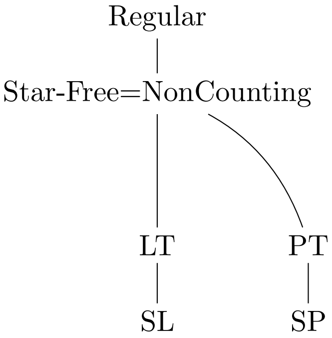
where $u_i = \sigma_i$ iff $\sigma_i \in T$ and $u_i = \lambda$ otherwise

Example

If $\Sigma = \{a, b, c\}$ and $T = \{b, c\}$ then

$$E_T(aabaaacaaabaa) = bcb$$

Interesting subregular classes



Proper inclusion relationships among language classes (indicated from top to bottom).

LT	Locally Testable	PT	Piecewise Testable
SL	Strictly Local	SP	Strictly Piecewise

(McNaughton and Papert 1971, Simon 1975, Rogers and Pullum 2007, in press, Rogers et al. 2010)

Locally Testable Languages

Factors

$$F_k(w) = \begin{cases} v \in \Sigma^k \ \mid \ w = uvx;\ u, v \in \Sigma^*, & |w| \geq k \\ w & \text{otherwise} \end{cases}$$

Example: $\alpha = abbcac;\ F_2(\alpha) = \{ab,\ bb,\ bc,\ ca,\ ac\}$.

Strictly Local (SL) Languages

$$L \in SL \iff \exists G \subseteq F_k(\Sigma^*) \left[\forall w \in \Sigma^* \left[w \in L \ \Leftrightarrow \ F_k(w) \subseteq G \,\right] \right]$$

Example: If $G = \{ab,\ bb,\ bc,\ ca\}$ then $\alpha \notin L(G)$.

Locally Testable (LT) Languages

$$L \in LT \iff \forall w, v \in \Sigma^* \left[F_k(w) = F_k(v) \Rightarrow \left[\, w \in L \Leftrightarrow v \in L \,\right] \right]$$

Locally Testable Languages

Factors

$$F_k(w) = \begin{cases} v \in \Sigma^k \ \mid \ w = uvx; \ u, v \in \Sigma^*, & |w| \geq k \\ w & \text{otherwise} \end{cases}$$

Example: $\alpha = abbcac$; $F_2(\alpha) = \{ab, \ bb, \ bc, \ ca, \ ac\}$.

Strictly Local (SL) Languages

$$L \in SL \iff \exists G \subseteq F_k(\Sigma^*) \left[\forall w \in \Sigma^* \left[w \in L \ \Leftrightarrow \ F_k(w) \subseteq G \right] \right]$$

Example: If $G = \{ab, \ bb, \ bc, \ ca\}$ then $\alpha \notin L(G)$.

Locally Testable (LT) Languages

$$L \in LT \iff \forall w, v \in \Sigma^* \left[F_k(w) = F_k(v) \Rightarrow \left[w \in L \Leftrightarrow v \in L \right] \right]$$

Piecewise Testable Languages

Subsequences

$$P_k(w) = \Big\{\sigma_1 \cdots \sigma_n \in \Sigma^* \;\mid\; w \in \Sigma^*\sigma_1\Sigma^* \cdots \Sigma^*\sigma_n\Sigma^*;\; n \leq k\Big\}$$

Example: $\alpha = abcd;\; P_2(\alpha) = \{\lambda, a, b, c, d, ab, ac, ad, bc, bd, cd\}$.

Strictly Piecewise (SP) Languages

$$L \in SP \iff \exists G \subseteq P_k(\Sigma^*) \left[\forall w \in \Sigma^* \left[w \in L \;\Leftrightarrow\; P_k(w) \subseteq G \right] \right]$$

Example: If $G = \{\lambda, a, b, c, d, ab, ac, bc, bd, cd\}$ then $\alpha \notin L(G)$.

Piecewise Testable (LT) Languages

$$L \in LPT \iff \forall w, v \in \Sigma^* \left[P_k(w) = P_k(v) \Rightarrow \left[w \in L \Leftrightarrow v \in L \right] \right]$$

Piecewise Testable Languages

Subsequences

$$P_k(w) = \left\{ \sigma_1 \cdots \sigma_n \in \Sigma^* \;\mid\; w \in \Sigma^* \sigma_1 \Sigma^* \cdots \Sigma^* \sigma_n \Sigma^*;\; n \leq k \right\}$$

Example: $\alpha = abcd$; $P_2(\alpha) = \{\lambda, a, b, c, d, ab, ac, ad, bc, bd, cd\}$.

Strictly Piecewise (SP) Languages

$$L \in SP \iff \exists G \subseteq P_k(\Sigma^*) \left[\forall w \in \Sigma^* \left[w \in L \Leftrightarrow P_k(w) \subseteq G \right] \right]$$

Example: If $G = \{\lambda, a, b, c, d, ab, ac, bc, bd, cd\}$ then $\alpha \notin L(G)$.

Piecewise Testable (LT) Languages

$$L \in LPT \iff \forall w, v \in \Sigma^* \left[P_k(w) = P_k(v) \Rightarrow \left[w \in L \Leftrightarrow v \in L \right] \right]$$

Tier-based Strictly Local Languages

$$L \in TSL$$

$$\Updownarrow$$

$$\exists T \subseteq \Sigma,\ G \subseteq F_k(T^*)$$

$$\left[\ \forall w \in \Sigma^* \ \left[w \in L \ \Leftrightarrow \ F_k(E_T(w)) \subseteq G\ \right]\ \right]$$

Example

Let $T = \{l, r\}$ and $G = \{lr, rl\}$ and $k = 2$.

Then $floralis \in L(G)$ because $F_2(E_T(floralis)) = F_2(lrl)$ and $F_2(lrl) = \{lr, rl\} \subseteq G$.

But $floraris \notin L(G)$ since $F_2(E_T(floraris)) = F_2(lrr)$ and $F_2(lrr) = \{lr, rr\} \nsubseteq G$.

Theorems

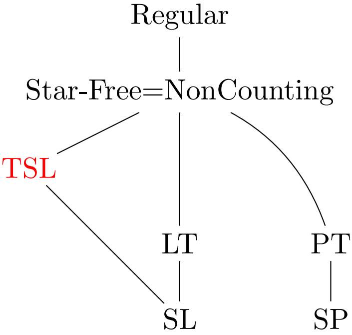
Theorem 1. SL $\subset$ TSL.

Theorem 2. TSL $\subset$ Star-free.

Theorem 3. TSL $\not\subseteq$ Locally Testable.

Theorem 4. TSL $\not\subseteq$ Piecewise Testable.

Generalizes Strictly Local languages



Proper inclusion relationships among language classes (indicated from top to bottom).

LT	Locally Testable	PT	Piecewise Testable
SL	Strictly Local	SP	Strictly Piecewise

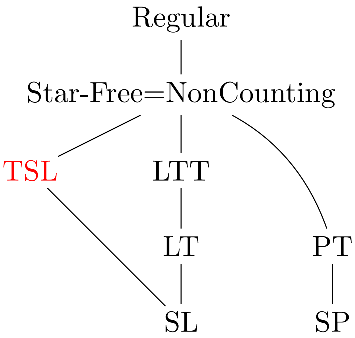
TSL Tier-based Strictly Local

Adequately Expressive for Phonology?

Phonotactic Patterns

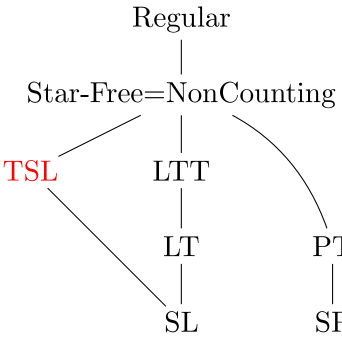
- ✓ Adjacency constraints (Strictly Local)
- ✓ Consonantal harmony
- ✓ Consonantal disharmony
- ✓ Vowel harmony without neutral vowels
- ✓ Vowel harmony with opaque vowels
- ✓ Vowel harmony with transparent vowels

Conclusions and future work



1. Automata-theoretic and algebraic characterizations
2. Learning the tier from positive evidence
3. Bounding the complexity of various product operations
4. Extending to relations

Conclusions and future work



1. Automata-theoretic and algebraic characterizations
2. Learning the tier from positive evidence
3. Bounding the complexity of various product operations
4. Extending to relations

Thank you