Sanskrit n-Retroflexion is Input-Output Tier-Based Strictly Local

Thomas Graf[1] • **Connor Mayer**[2]

[1]Department of Linguistics, Stony Brook University • [2]Department of Linguistics, UCLA

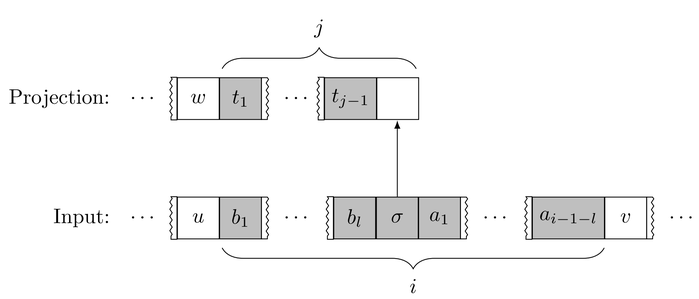
Subregular phonology and *nati*

- Segmental phonology is *regular*, but regular languages generate patterns unattested in natural language.
- *Subregular phonology* looks for subclasses of the regular languages that:
  1. Can generate natural language phenomena.
  2. Generate only attested patterns.
- Provides a tighter typology, insights into learnability and computational universals of language [1].
- Sanskrit /n/-retroflexion, or *nati*, is subregular, but highly complex.
  - Cannot be handled by established classes like *strictly local* (SL) or *tier-based strictly local* (TSL).

Contribution: *Nati* can be captured by an extension of TSL: *input-output TSL* (IO-TSL), providing a tighter bound on the complexity of segmental phonology.

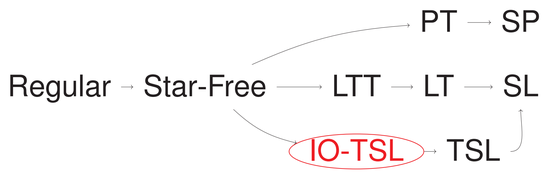
Input-Output Tier-based Strictly Local

- SL-k grammars require that strings do not contain any of a set of illegal k-grams.
  - Handles local assimilation, word-final devoicing, etc.
- TSL-k grammars ban a set of k-grams, but candidate strings are first filtered by a *projection function* [2].
  - The projection function erases certain symbols.
  - Handles culminativity, long-distance harmony, etc.
- IO-TSL-(i, j, k) grammars extend TSL-k by modifying the projection function (IOSL-(i, j)).
  - There is still a set of illicit k-grams.
  - Consider a window of length $i - 1$ around the symbol in the input string when choosing whether to project.
  - We can also look at the preceding $j - 1$ symbols that have already been projected.
- IO-TSL grammars are sensitive to the structure of both the input and output when constructing a tier.



Relation to other subregular languages

- IO-TSL is conjectured to be a proper subset of the *star-free languages*.
- IO-TSL contains the TSL languages.



The *nati* pattern [3]

- Basic pattern: /n/ becomes retroflex [ɳ] when preceded in the word by a non-lateral retroflex continuant.

No *nati*		*nati*	
káːm-eːna	‘by desire’	naɻ-eːɳa	‘by man’
joːg-eːna	‘by means’	manuʂj-eːɳa	‘by human’

- Coronal blocking: Coronals that intervene between the trigger and target block *nati*.

ɻátʰ-eːna	‘by chariot’	pɻa-ɳinaːja	‘lead forth’
gaɻuḍ-eːna	‘by Garuḍa’	vaɻɳ-anaːnam	*no gloss*

- Sonorant adjacency: The /n/ target must be immediately followed by a non-liquid sonorant.

bɻaɦman ‘brahman’	caɻ-a-**n**-ti ‘wander (3Pl)’

- Velar/labial blocking: Preceding velar and labial plosives can block *nati*, but only when:
  - They occur immediately before the target /n/; and
  - There is a left root boundary between target and trigger.

No blocking		Blocking	
ʊɻtɻa-√ɦáɳa	‘Vr̥tra-killing’	pɻ-√aːp-noː-ti	‘attains (3s)’
√ɻug-ɳá	‘break (PP)’	pɻa-√bʱag-na	‘broken’

- Retroflex blocking: *nati* is blocked when a retroflex occurs to the right of the target, but only when there is:
  - A left root boundary between target and trigger.
  - No coronal between /n/ and the blocking retroflex.

No blocking		Blocking	
pɻa-√ɳeː-tɻ	‘leader’	pɻa-√nakʂ-	‘approach’
√pɻ-ɳa-k-ʂi	‘unite (2s)’	pɻa-√nɻt-	‘dance forth’

Formal analysis

Nati is (3,2,3)-IO-TSL.

- I-TSL projects symbols that matter in a specific context.
- O-TSL omits segments that do not affect grammatically given what we already know about the string.
- Only a bounded number of symbols form a “*nati* zone” on the tier, so these can be regulated by k-grams.

R: Retroflex triggers	S: Non-liquid sonorants
C: Coronals	P: Labial and velar plosives

Projection function:

- Always project R (*IOSL*-(1, 1))
- Project S if it is immediately preceded by [n] in the input (*IOSL*-(2, 1))
- Project $\surd$ if the previous tier symbol is R (*IOSL*-(1, 2))
- Project P if the previous tier symbol is $\surd$ and the next two input symbols are [n] and S (*IOSL*-3, 2))
- Project C if previous tier symbol is R, $\surd$, or S, unless C is [n] and next input symbol is S (*IOSL*-(2, 2))

Banned substrings:

- *RS*
- $\surd SX$ (where X is $\ltimes$, C, or S)

Example	Tier	Forbidden substring
naɻeːɳa	aɻɳ⋉	–
*naɻeːna	aɻa⋉	ɻa
pɻaɲinaːja	ɻɳa⋉	–
caɻanti	ɻt⋉	–
ʊɻtɻa√ɦáɳa	ɻtɻ√ɳ⋉	–
*ʊɻtɻa√ɦána	ɻtɻ√a⋉	√a⋉
pɻ√aːpnoːti	ɻ√poːt⋉	–

Example	Tier	Forbidden substring
√ɻugná	ɻɳ⋉	–
*√ɻugná	ɻá⋉	ɻá
pɻa√ŋeːtɻ	ɻ√ɳɻ⋉	–
*pɻa√neːtɻ	ɻ√eːtɻ⋉	√eːt
√bɻaːfimaɳéːʂu	ɻɳʂ⋉	–
*√bɻaːfimanéːʂu	ɻéːʂ⋉	ɻéː
pɻa√nakʂ	ɻ√aʂ⋉	–

Discussion and future work

- IO-TSL is a natural upper bound on the complexity of *nati* when construed as a phonotactic constraint.
- But IO-TSL is too powerful (first-last harmony)
- Open questions: Is IO-TSL learnable? Is IO-TSL FO-definable? How can we restrict IO-TSL without losing empirical coverage?

Selected references

[1] **Heinz**, J. 2018. The computational nature of phonological generalizations. In L. Hyman and F. Plank, eds, *Phonological Typology*, Phonetics and Phonology, ch. 5, 126-195. Mouton De Gruyter. • [2] **Heinz**, J. 2011. Tier-based strictly local constraints in phonology. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, 58-64. • [3] **Ryan**, K. 2017. Attenuated spreading in Sanskrit retroflex harmony. *Linguistic Inquiry*, 48(2):299-340.