Locality and relativized locality as essential organizing principles of phonology

Jeffrey Heinz

Locality Workshop
Stony Brook University
2025 04 11

Today

1. I will argue that particular, subregular, notions of locality are central organizing principles of phonological generalizations.

Today

2. These concepts provide a way to understand

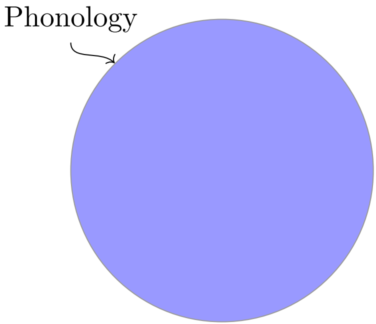
   1. the extensive variation cross-linguistically, and
   2. how these patterns can be acquired from examples, and
   3. the important role played by representation in grammar.

Today

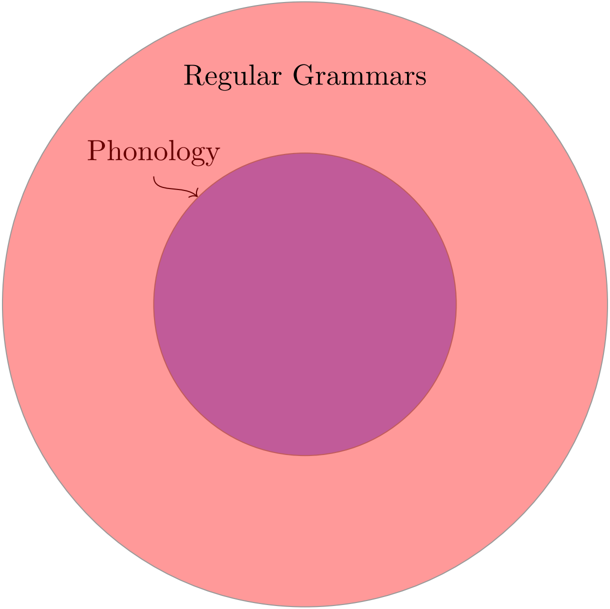
3. Formalizing these insights directly with logic and explicit representations provides a better theory of phonology than theories

1. based on global optimization like Optimality Theory, or
2. based on serial rule application.

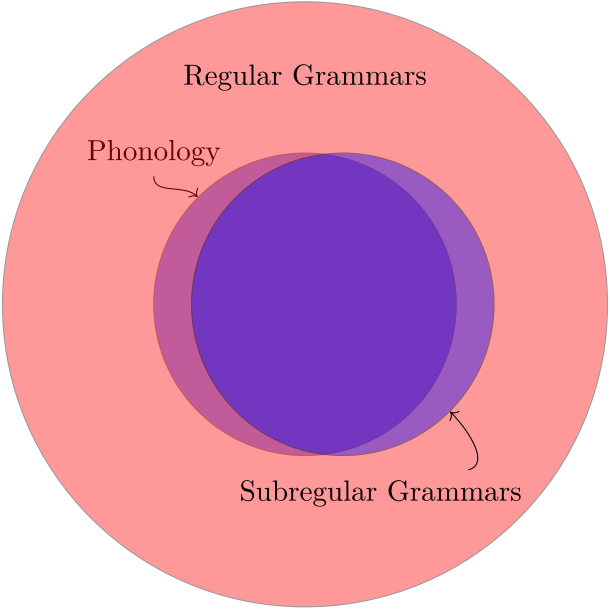
Typological Argument



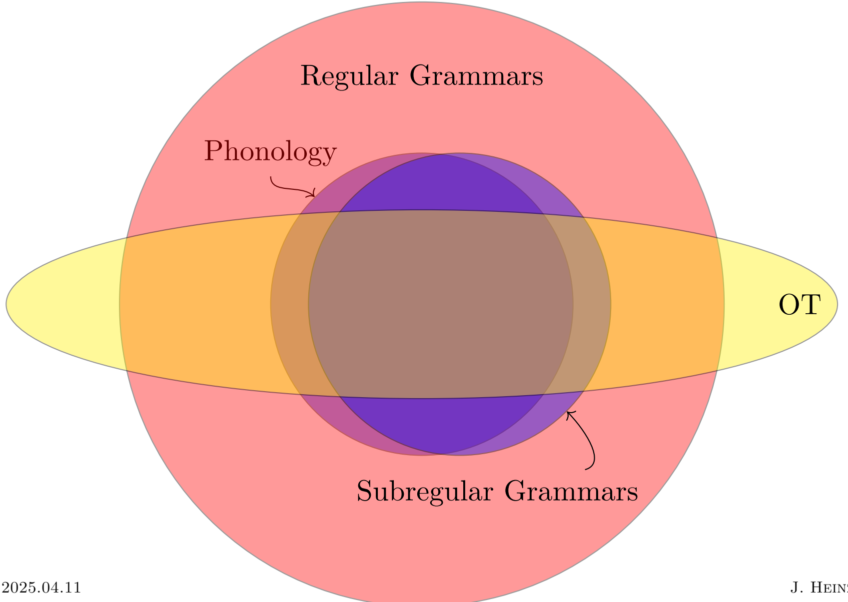
Typological Argument



Typological Argument



Typological Argument



Outline

1. Locality as adjacency and relativized adjacency
2. Mathematical and computational analysis
3. Typological and learning results
4. Implications for phonological theories
5. Considerations of grammar more broadly (syntax)

Locality as an Adjacency Window

Word-final /e/ raising in Finnish (Uralic, Finland)

/ # r e k e # /

[# r e k **i** #]

'sledge (nom. sg.)'

Odden 2014

Word-final /e/ raising in Finnish (Uralic, Finland)

/ # r . . . # /

[# r . . . #]

‘sledge (nom. sg.)’

Odden 2014

Word-final /e/ raising in Finnish (Uralic, Finland)

/ # . e k . # /

[# . e . . #]

'sledge (nom. sg.)'

Odden 2014

Word-final /e/ raising in Finnish (Uralic, Finland)

/ # . . k . # /

[# . . k . #]

'sledge (nom. sg.)'

Odden 2014

Locality as an Adjacency Window

Word-final /e/ raising in Finnish (Uralic, Finland)

/ # r e k e # /

[# . . . **i** #]

'sledge (nom. sg.)'

Odden 2014

Word-final /e/ raising in Finnish (Uralic, Finland)

/ # r e k e # /

[# r e k **i** #]

'sledge (nom. sg.)'

Odden 2014

Locality as an Adjacency Window

Word-final /e/ raising in Finnish (Uralic, Finland)

/ # r e k e # /

[# r e k **i** #]

'sledge (nom. sg.)'

The max window size is called the 'k' parameter. Here k=2.
/e/ becomes [i] if its neighbor to the right is the boundary.

Odden 2014

Long-distance nasal agreement in Kikongo (Bantu, Congo)

/ k u - d u m u k - i s - i l a /

[k u - d u m u k - i s - i **n** a]

'to cause to jump for'

Piggot 1996

Long-distance nasal agreement in Kikongo (Bantu, Congo)

/ . . . . a /

[. . . . . **a**]

‘to cause to jump for’

Piggot 1996

Long-distance nasal agreement in Kikongo (Bantu, Congo)

/ . . . . . m . . . . l . /

[. . . . . **n** .]

'to cause to jump for'

Piggot 1996

Locality as Relativized Adjacency

Long-distance nasal agreement in Kikongo (Bantu, Congo)

/ k u - d u m u k - i s - i l a /

[k u - d u m u k - i s - i **n** a]

'to cause to jump for'

The window is on a 'tier' including nasals and laterals (and k=2).
/l/ becomes [n] if its neighbor to the left on the tier is a nasal.

Piggot 1996, Goldsmith 1979, Lambert 2023, and others

Mathematical/Computational Analysis of Locality in String Transformations

Extensions of grammars are infinite objects like circles.

Word-final /e/ raising | **Circle with radius 1**

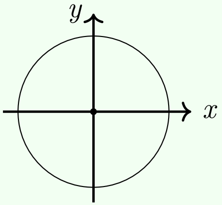
Intensional Descriptions

1. e ⟶ [+high] / __ #
2. *e# ≫ Ident(high)

1. Cartesian: $x^2 + y^2 = 1$
2. Polar: $r = 1$

Extensional Descriptions

(ove, ovi), (yoke, yoki), ...
(tukki, tukki), (kello, kello), ...
(manilabanile, manilabanili),
...



Truisms about grammars

1. Different grammars may generate the same constraints and transformations just like different equations can realize the same functions.
2. Grammars may have properties largely independent of grammatical particulars.
   - regular and rational functions (Kleene 1956, Scott and Rabin 1959, Elgot and Mezei 1965, Filiot and Reynier 2016)
   - output-driven maps (Tesar 2014)

WORD-FINAL /e/ RAISING

Phonological Map

/reke/ → [reki]
/ove/ → [ovi]
/muuri/ → [muuri]
...

Algebra

	x	e
x	x	e
e	x	e

Rewrite Rules

/e/ → [+high] / __ #

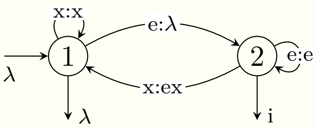
Optimality Theory

*e# ≫ IDENT(HIGH)

Logic

$$\text{high}_o(x) := \text{high}(x) \vee \Big(\text{front}(x) \wedge \neg\text{low}(x) \wedge \text{boundary}(\text{next}(x))\Big)$$

State Machine

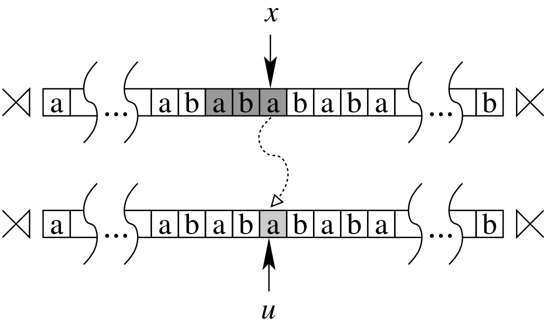


Questions about Locality

1. How can we determine whether a phonological transformation only uses information...
   - in adjacency windows (bounded windows of size k)?
   - in relativized adjacency windows (bounded windows of size k on a tier)?
2. How much attested phonology is covered by such local functions?
3. How can such functions be learned or acquired from examples?
4. How can these ideas be extended to non-linear phonological representations such as syllable structure, morphological structure, autosegmental representations, and others?

Locality in Terms of Adjacency Windows

Input Strictly Local functions



The output produced by position x only depends on the symbol at x and the previous $k-1$ symbols.

Chandlee 2014, Chandlee and Heinz 2018

Locality in Terms of Adjacency Windows (continued)

Input Strictly Local functions

A function f is Input Strictly Local if there is a $k \in \mathbb{N}$ such that for any strings $u1, u2$ which share a suffix v of length k then the behavior of f on any extension w beyond v is identical.

Chandlee 2014, Chandlee and Heinz 2018

Locality in Terms of Adjacency Windows (continued)

Input Strictly Local functions

A function f is Input Strictly Local if there is a $k \in \mathbb{N}$ such that for any strings $u1, u2$ which share a suffix v of length k then the behavior of f on any extension w beyond v is identical.

	u1
input:	blahblah...
output:	

	u2
input:	goblgobl...
output:	

Chandlee 2014, Chandlee and Heinz 2018

Locality in Terms of Adjacency Windows (continued)

Input Strictly Local functions

A function f is Input Strictly Local if there is a $k \in \mathbb{N}$ such that for any strings $u1, u2$ which share a suffix v of length k then the behavior of f on any extension w beyond v is identical.

	u1
input:	blahblah...
output:	blehbleh...

	u2
input:	goblgobl...
output:	geblgebl...

Chandlee 2014, Chandlee and Heinz 2018

Locality in Terms of Adjacency Windows (continued)

Input Strictly Local functions

A function f is Input Strictly Local if there is a $k \in \mathbb{N}$ such that for any strings $u1, u2$ which share a suffix v of length k then the behavior of f on any extension w beyond v is identical.

	u1	v (k=4)
input:	blahblah...	zuba
output:	blehbleh...	
	u2	
input:	goblgobl...	zuba
output:	geblgebl...	

Chandlee 2014, Chandlee and Heinz 2018

Locality in Terms of Adjacency Windows (continued)

Input Strictly Local functions

A function f is Input Strictly Local if there is a $k \in \mathbb{N}$ such that for any strings $u1, u2$ which share a suffix v of length k then the behavior of f on any extension w beyond v is identical.

	u1	v (k=4)
input:	blahblah...	zuba
output:	blehbleh...	beha

	u2	v (k=4)
input:	goblgobl...	zuba
output:	geblgebl...	goba

Chandlee 2014, Chandlee and Heinz 2018

Locality in Terms of Adjacency Windows (continued)

Input Strictly Local functions

A function f is Input Strictly Local if there is a $k \in \mathbb{N}$ such that for any strings $u1, u2$ which share a suffix v of length k then the behavior of f on any extension w beyond v is identical.

	u1	v (k=4)	w
input:	blahblah...	zuba	ahmaahma...
output:	blehbleh...	beha	
	u2	v (k=4)	w
input:	goblgobl...	zuba	ahmaahma...
output:	geblgebl...	goba	

Chandlee 2014, Chandlee and Heinz 2018

Locality in Terms of Adjacency Windows (continued)

Input Strictly Local functions

A function f is Input Strictly Local if there is a $k \in \mathbb{N}$ such that for any strings $u1, u2$ which share a suffix v of length k then the behavior of f on any extension w beyond v is identical.

	u1	v (k=4)	w
input:	blahblah...	zuba	ahmaahma...
output:	blehbleh...	beha	ahemahem...
	u2	**v (k=4)**	**w**
input:	goblgobl...	zuba	ahmaahma...
output:	geblgebl...	goba	ahemahem...

Chandlee 2014, Chandlee and Heinz 2018

Other Characterizations

Logic

Quantifier Free: Formulas defining the logical transduction can refer to **next** and **previous** elements, but cannot use the quantifiers $\exists$ and $\forall$.

$$\text{high}_o(x) :=$$
$$\text{high}(x) \vee \big(\ \text{front}(x) \wedge \neg\text{low}(x) \wedge \text{boundary}(\textbf{next}(x))\ \big)$$

Nesting gives larger windows: **next**(**next**(x)) and so on.

Lindell and Chandlee 2016

Logic (continued). Compare:

1. $P_o(x) := Q(x) \wedge (\exists y)\big[R(y)\big]$ (First Order Definable)

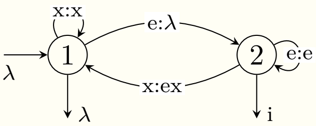
   Requires scanning whole word for such a y!!

2. $P_o(x) := Q(x) \wedge R(x)$ (Quantifier-free Definable)

   Information to decide P is local to x in the input!!

Finite-state Machine

When processing a string, the state of the transducer corresponds to the last $k-1$ symbols read.



Perles et al. 1963, Vaysse 1986, Chandlee et al. 2014

Algebra

Definiteness: All elements y, z in the map's syntactic semigroup satisfy the equation $yz^\omega = z^\omega$.

	x	e
x	x	e
e	x	e

Almeida 1995, Lambert 2022, to appear, Lambert and Heinz 2023

Algebra

Definiteness: All elements y, z in the map's syntactic semigroup satisfy the equation $yz^\omega = z^\omega$.

	x	e
x	x	e
e	x	e

Given a string-to-string MSO-definable logical transduction, or arbitrary finite-state transducer, it is decidable whether they represent an ISL/definite/QF function!

Almeida 1995, Lambert 2022, to appear, Lambert and Heinz 2023

Typological Coverage

1. Maps describable with SPE-style rules $A \longrightarrow B \;/\; C \_\_ D$ which apply simultaneously, provided there is a longest string in CAD.
2. Approximately 95% of the individual processes in P-Base (v.1.95, Mielke 2008), including local substitution, deletion, epenthesis, and synchronic metathesis
3. Many *opaque* transformations without any special modification.

> The necessary and sufficient information to decide the output for many phonological maps, including many opaque ones, is contained within adjacency windows of bounded length on the input side.

Chandlee 2014, Chandlee and Heinz 2018, Chandlee, Heinz and Jardine 2018

Logical transformations with nonlinear representations

Logical transductions, and the Quantifier Free notion of locality, extend to **any** linguistic representation!

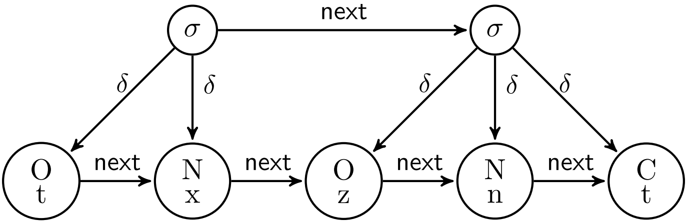
- Segments
- Unary, Binary, and Scalar Features
- Elements
- Syllabic Roles
- Metrical Feet or the Grid
- Autosegmental Representations
- Articulatory Scores, Coupling Graphs
- Morphological Structure
- Derivation trees
- ...

Logical transformations with nonlinear representations

Strother-Garcia (2018) shows that the process of syllabification in Imdlawn Tashlhiyt Berber (Dell and Elmedlaoui 1985, Prince and Smolensky 1993) is Quantifier Free.

/ t x z n t /

maps to



Logical transformations with nonlinear representations

Strother-Garcia (2018) concludes *"...syllabification in ITB can be represented by a QF [logical] transduction, a formalism restricted to substantially lower computational complexity than [traditional] phonological grammars...Establishing that ITB syllabification is QF highlights an insight not apparent from [those traditional] grammatical formalisms..."*

Logical transformations with nonlinear representations

Chandlee and Jardine (2019) examine tonal processes using QF transductions over strings and over Autosegmental Representations.

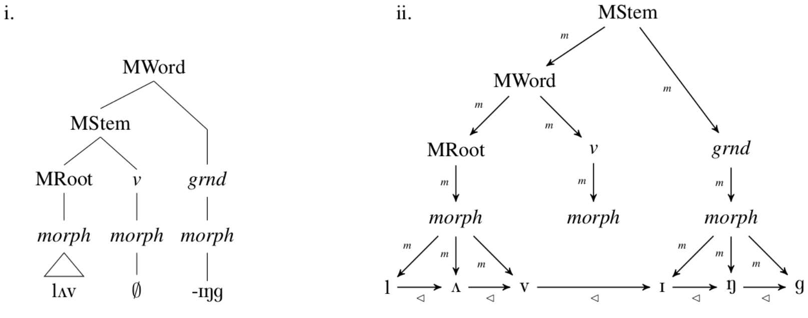
Pattern	**Language**	**ISL**	**A-ISL**
Bounded shift (§4.1, 5.2)	Rimi	✓	✓
Bounded spread (§6.1)	Bemba	✓	✓
Bounded Meussen's Rule (§6.3)	Luganda	✓	✗
Unbounded shift (§4.2,5.3)	Zigula	✗	✓
Unbounded deletion (§6.2)	Arusa	✗	✓
Alternating Meussen's Rule (§6.4)	Shona	✗	✗
Unbounded spread (§6.5)	Ndebele	✗	✗

Table 1: Summary of analyses.

Logical transformations with nonlinear representations

Dolatian (2020) examines the phonology-morphology interface in light of Quantifier Free logical transductions. He concludes

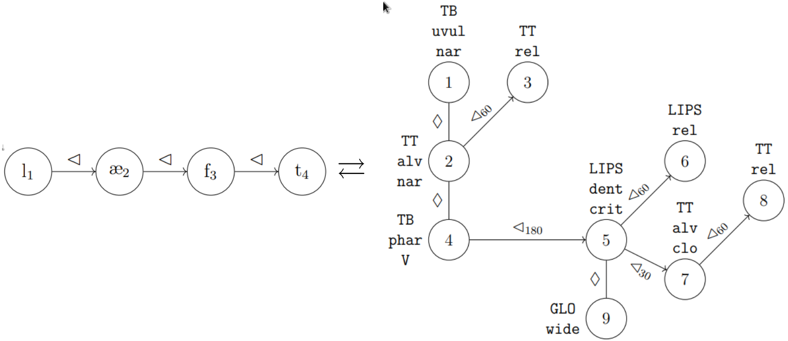
“the bulk of the morphology-phonology interface requires local computation, not global computation.”



“loving”

Logical transformations with nonlinear representations

Nelson (2022, 2024) examines the phonetics-phonology interface, and shows how to convert coupling graphs in Articulatory Phonology to familiar segmental representations and vice versa using first order (not QF) logic.



Phonology that is Not ISL/Definite/QF

- Progressive and regressive long-distance harmony (as in Kikongo) and disharmony
- Progressive and regressive iterative spreading patterns
- Certain kinds of suprasegmental processes (stress, tone)

Chandlee 2014, Chandlee and Heinz 2018, Chandlee and Jardine 2019

Locality as Relativized Adjacency

Long-distance nasal agreement in Kikongo (Bantu, Congo)

/ k u - d u m u k - i s - i l a /

[k u - d u m u k - i s - i **n** a]

'to cause to jump for'

The window is on a 'tier' including nasals and laterals (and k=2).
/l/ becomes [n] if it's neighbor to the left on the tier is the a nasal.

Piggot 1996, Goldsmith 1979, Lambert 2023, and others

Long-distance Harmony

Phonological Map

/tamok-ila/ → [tomokina]
/nakap-ila/ → [nakapina]
/takop-ila/ → [takopila]
...

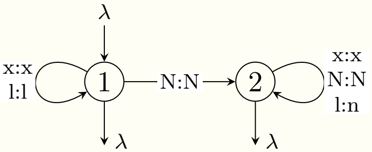
Algebra

	x	n
x	x	n
n	n	n

Logic

$$\text{nasal}_o(x) := \text{nasal}(x) \vee \big(\text{lateral}(x) \wedge \text{nasal}(\text{prev\_sonorant}(x))\big)$$

State Machine



Burness et al. 2021, Lambert and Heinz 2024

Long-distance Harmony

Phonological Map

/tamok-ila/ → [tomokina]
/nakap-ila/ → [nakapina]
/takop-ila/ → [takopila]
...

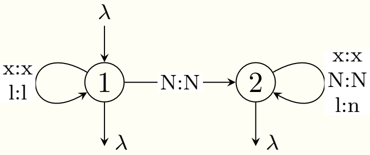
Algebra

	x	n
x	x	n
n	n	n

Logic

$$\text{nasal}_o(x) := \text{nasal}(x) \lor \big(\text{lateral}(x) \land \text{nasal}(\text{prev\_sonorant}(x))\big)$$

State Machine



Observe that the transitions with /x, l/ as inputs never change state. This means they are invisible, i.e. off the tier!

Tier Definite Maps

- A map is Tier Definite (generally Tier X) if, upon removal of the non-tier elements, the resulting map belongs to the Definite class (generally to the X class).
- In the state machine, the non-tier elements are the ones which self-loop on every state.
- Algebraically, the non-tier elements correspond to the identity.

Lambert 2022, 2023, to appear

Tier Definite Maps

- A map is Tier Definite (generally Tier X) if, upon removal of the non-tier elements, the resulting map belongs to the Definite class (generally to the X class).
- In the state machine, the non-tier elements are the ones which self-loop on every state.
- Algebraically, the non-tier elements correspond to the identity.
- Kikongo nasal harmony is Tier Definite!

Lambert 2022, 2023, to appear

Reverse Definite and Tier Reverse Definite

Algebra

Reverse Definiteness: All elements y, z in the map's syntactic semigroup satisfy the equation $z^\omega y = z^\omega$.

	x	e
x	x	x
e	e	e

1. Informally, the behavior of the process is determined according to a shared prefix.

Almeida 1995

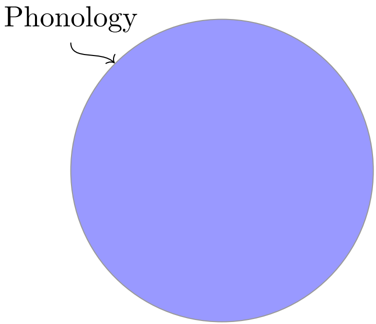
Long-distance harmony patterns and iterative spreading

Algebraic classification for left-to-right ($\rightarrow$) and right-to-left ($\leftarrow$) processing. **D** is Definite, **K** is Reverse Definite, **N** is both Definite and Reverse Definite, and $[\![\cdot]\!]_T$ means on a tier.

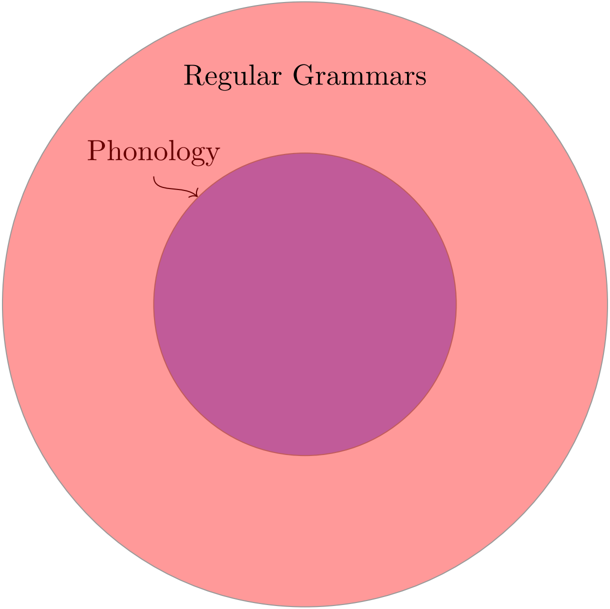
Pattern	$\rightarrow$	$\leftarrow$
Post-Nasal Voicing	**D**	**D**
Prog. Iterative Spreading	$[\![\mathbf{D}]\!]_T$	–
Reg. Symmetric Harmony	–	$[\![\mathbf{K}]\!]_T$
Reg. Asymmetric Harmony	–	$[\![\mathbf{N}]\!]_T$
Pre-Nasal Voicing	**D**	**D**
Reg. Iterative Spreading	–	$[\![\mathbf{D}]\!]_T$
Prog. Symmetric Harmony	$[\![\mathbf{K}]\!]_T$	–
Prog. Asymmetric Harmony	$[\![\mathbf{N}]\!]_T$	–

Lambert and Heinz 2024:Table 1

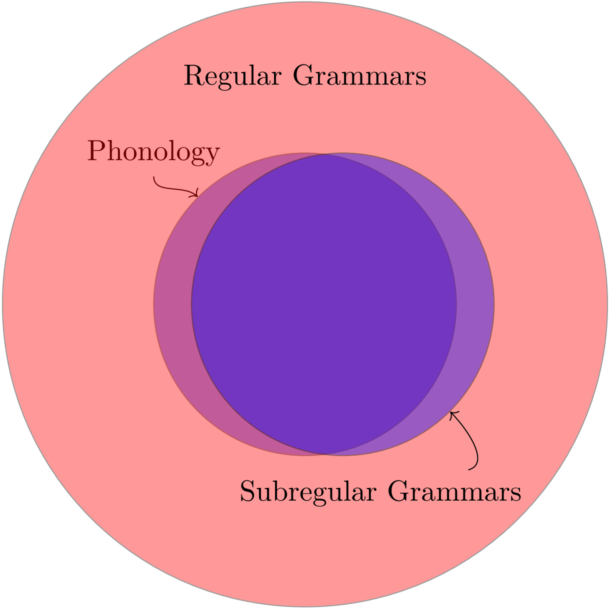
Typological Argument



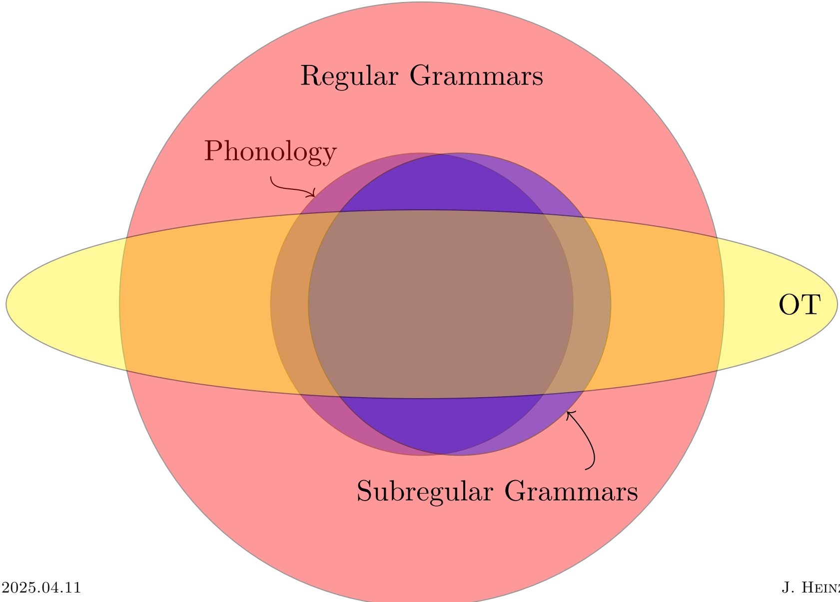
Typological Argument



Typological Argument



Typological Argument



Learnability

1. String-to-string ISL/Definite/QF transformations are learnable from positive examples up to a given k.

Algorithm	Time and Data Complexity	Notes
OSTIA	$O(n^3)$	does not need k
ISLFLA	$O(n^2)$	needs (max) k
SOSFIA	$O(n^1)$	needs (max) k

2. Markowska (in progress) has developed these algorithms to work with featural representations to dramatically reduce the amount of data needed (from many thousands to hundreds).

Oncina et al. 1993, Chandlee et al. 2014, Jardine et al. 2014

Learning Tiers

Tiers can be learned from observations!

Theory

- Jardine and Heinz 2016
- Jardine and McMullin 2017
- Burness and McMullin 2019, 2021 (for maps!)
- De Santo and Aksënova 2021 (multiple tiers!)
- Lambert 2021 (online learning!)
- Swanson 2024 (over trees!)

Practice

- Gouskova and Gallagher 2020
- Belth 2024 (for maps!)

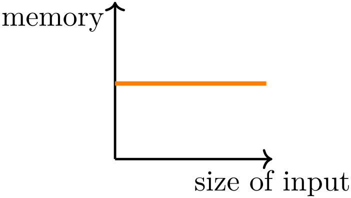
Regular Grammars and OT

Regular grammars

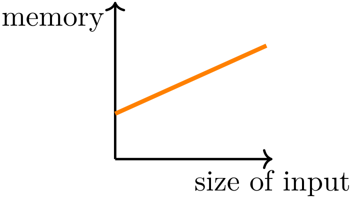
1. Monadic Second Order (MSO) logic
2. Regular expressions
3. Finite-state automata

What "Regular" means

A grammar is regular provided **the memory required to assemble the computation is bounded by a constant,** ***regardless of the size of the input.***



Regular



Non-Regular

Example: Vowel Harmony

Progressive

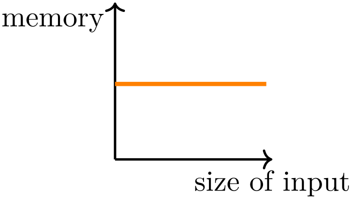
Vowels harmonize in backness with the first vowel in the underlying representation.

Majority Rules

Vowels harmonize in backness with the majority of vowels in the underlying representation.

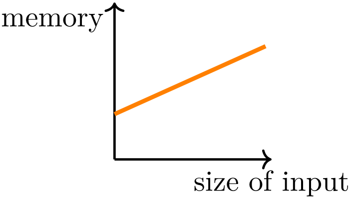
UR	Progressive	Majority Rules
/nokelu/	nokolu	nokolu
/nokeli/	nokolu	nikeli
/pidugo/	pidige	pudugo
/pidugomemi/	pidigememi	pidigememi

(Bakovic 2000, Finley 2008, 2011, Heinz and Lai 2013)



Regular

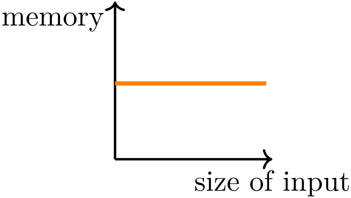
Progessive



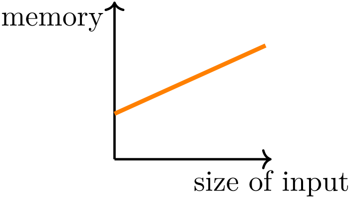
Non-Regular

Majority Rules

Progressive and Majority Rules Harmony



Regular



Non-Regular

Progessive

Majority Rules

/p i d u g o m e m i/

Some Perspective

Typological

Majority Rules is unattested. (Bakovic 2000, Karakas and Dolatian, in progress)

Psychological

Human subjects fail to learn Majority Rules in artificial grammar learning experiments, unlike progressive harmony.
(Finley 2008, 2011)

Computational

Majority Rules is not regular. (Riggle 2004, Heinz and Lai 2013)

Optimality Theory

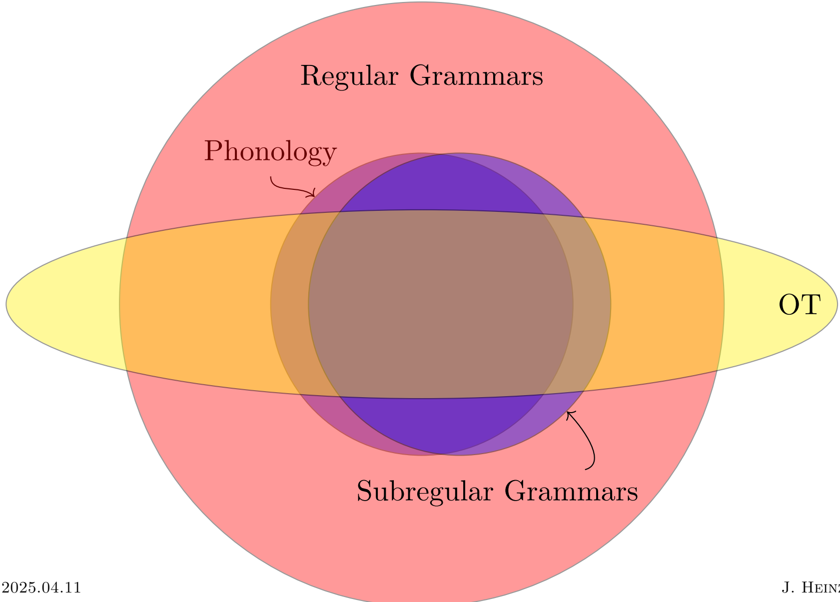
1. There exists a CON and ranking over it which generates Majority Rules: Agree(back)>>IdentIO[back].
2. Some believe changing CON may resolve this, but such an approach fails to recognize the essence of the problem...
3. which is global optimization itself.
4. There are many reasons to think CON cannot be so changed!
   (Hao 2019, 2024; Lamont 2021, 2022)

Evidence supporting the hypothesis that phonological generalizations are regular originate with Johnson (1972) and Kaplan and Kay (1994), who showed how to translate any ordered sequence of SPE-style rewrite rules into a finite-state automaton.

Consequently:

1. Constraints on well-formed surface and underlying representations are regular (since the image and pre-image of finite-state functions are finite-state, Rabin and Scott 1959)
2. Since virtually any phonological grammar can be expressed as an ordered sequence of SPE-style rewrite rules, this means "being regular" is a property of the functions that *any* phonological grammar defines.

Typological Argument



Beyond Phonology

Subregular syntax

Regular tree grammars are sufficient to describe syntactic structures (Stabler 2019).

What about subregular grammars?

Subregular syntax

Graf (2022:146) writes

> *I will discuss: how subregular linguistics provides a unified view of phonology, morphology, syntax, and possibly even semantics, allowing us to transfer insights between linguistic domains; how it can be combined with learnability considerations to derive typological restrictions of harmony systems; how it interacts with seemingly minor differences in the analysis of inflectional markers in Noon; and how it can derive the existence of islands directly from the computational nature of movement.*
>
> *A lot has been accomplished since the beginnings of the program 10 years ago, but even more remains to be done. Theoretical linguists and formal grammarians alike are already in an excellent position to contribute to this program.*

Graf 2018 et seq., Graf and Shafiei 2019, Vu et al. 2019, Graf and Kostyszyn 2021, Hanson 2023 et seq.

Case and Agreement

Hanson (to appear, in progress) shows that agreement and case relations can be understood as satisfying particular locality conditions on the paths between the relevant units in the syntactic derivation trees of these languages, once irrelevant material has been projected away.

Hanson to appear

Case and Agreement

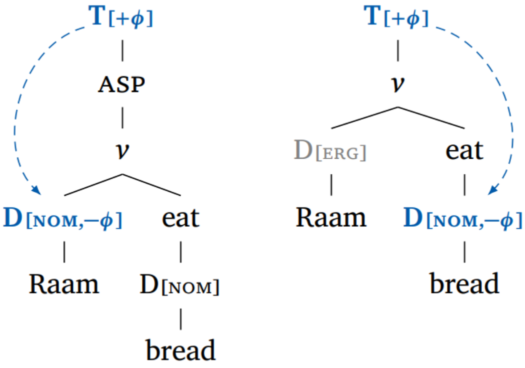


Figure 7: Case-sensitive agreement in Hindi. Ergative DPs are invisible, causing the object to agree instead of the subject

---

Hanson to appear

Case and Agreement

	Tier projection	Tier constraints	Phenomenon
a.	All $\pm\phi$ elements	Strict matching of $+\phi/-\phi$	Minimality
b.	Some D heads do not project	(as in (a))	Invisibility
c.	Some non-agreeing items project	(as in (a))	Blocking
d.	(as in (a))	Reverse order of $+\phi/-\phi$	Upward agreement
e.	(as in (a))	Allow sequential $+\phi$	Chain agreement
f.	(as in (a))	Allow sequential $-\phi$	Multiple agreement

Hanson to appear

Conclusion

Mathematical and computational analysis reveals

1. precise definitions of locality that inform linguistic study,
2. that much (all?) of phonology is local with the right representations (tiers and nonlinear representations),
3. that the computations are mostly (all?) subregular (Quantifier-Free),
4. which information what must be attended to when learning or acquiring phonology

Conclusion

Mathematical and computational analysis reveals

1. precise definitions of locality that inform linguistic study,
2. that much (all?) of phonology is local with the right representations (tiers and nonlinear representations),
3. that the computations are mostly (all?) subregular (Quantifier-Free),
4. which information what must be attended to when learning or acquiring phonology

A theory of phonological grammar organized around these locality principles compares favorably to ones based on global optimization, or serially ordered rules.

Current and Future Work

1. Examine and compare more representations, for instance element theory/government phonology,
2. Further study of particularly complex phenomena
3. Examine more logics including Boolean Monadic Recursive Schemes (Chandlee and Jardine 2021)
4. Learning lexicons, grammars, exceptions, variation
5. Learning transformations over non-linear representations
6. Developing tools for the facilitation of, verification of, and automatic discovery/acquisition of analyses

THANK YOU!

References I

Almeida, Jorge. 1995. *Finite Semigroups and Universal Algebra*, vol. 3 of *Series in Algebra*. Singapore: World Scientific.

Baković, Eric. 2000. Harmony, dominance and control. Doctoral dissertation, Rutgers University.

Burness, Phillip, and Kevin McMullin. 2019. Efficient learning of output tier-based strictly 2-local functions. In *Proceedings of the 16th Meeting on the Mathematics of Language*, 78–90. Toronto, Canada: Association for Computational Linguistics.

Burness, Phillip, and Kevin McMullin. 2021. Learning multiple independent tier-based processes. In *Proceedings of the Fifteenth International Conference on Grammatical Inference*, edited by Jane Chandlee, Rémi Eyraud, Jeff Heinz, Adam Jardine, and Menno van Zaanen, vol. 153 of *Proceedings of Machine Learning Research*, 66–80. PMLR.

URL
https://proceedings.mlr.press/v153/burness21a.html

Burness, Phillip A., Kevin J. McMullin, and Jane Chandlee. 2021. Long-distance phonological processes as tier-based strictly local functions. *Glossa: a Journal of General Linguistics* 6.

Chandlee, Jane. 2014. Strictly local phonological processes. Doctoral dissertation, The University of Delaware.

Chandlee, Jane, Rémi Eyraud, and Jeffrey Heinz. 2014. Learning strictly local subsequential functions. *Transactions of the Association for Computational Linguistics* 2:491–503.

Chandlee, Jane, and Jeffrey Heinz. 2018. Strict locality and phonological maps. *Linguistic Inquiry* 49:23–60.

Chandlee, Jane, Jeffrey Heinz, and Adam Jardine. 2018. Input strictly local opaque maps. *Phonology* 35:171–205.

References III

Chandlee, Jane, and Adam Jardine. 2019. Autosegmental input-strictly local functions. *Transactions of the Association for Computational Linguistics* 7:157–168.

Chandlee, Jane, and Adam Jardine. 2021. Computational universals in linguistic theory: Using recursive programs for phonological analysis. *Language* 93:485–519.

De Santo, Aniello, and Alëna Aksënova. 2021. Learning interactions of local and non-local phonotactic constraints from positive input. In *Proceedings of the Society for Computation in Linguistics 2021*, edited by Allyson Ettinger, Ellie Pavlick, and Brandon Prickett, 167–176. Online: Association for Computational Linguistics. URL https://aclanthology.org/2021.scil-1.16/

References IV

Dolatian, Hossep. 2020. Computational locality of cyclic phonology in Armenian. Doctoral dissertation, Stony Brook University.

Elgot, C. C., and J. E. Mezei. 1965. On relations defined by generalized finite automata. *IBM Journal of Research and Development* 9:47–68.

Finley, Sara. 2008. The formal and cognitive restrictions on vowel harmony. Doctoral dissertation, Johns Hopkins University, Baltimore, MD.

Finley, Sara. 2011. The privileged status of locality in consonant harmony. *Journal of Memory and Language* 65:74–83.

Goldsmith, John. 1979. *Autosegmental phonology*. New York: Garland.

References V

Gouskova, Maria, and Gillian Gallagher. 2020. Inducing nonlocal constraints from baseline phonotactics. *Natural Language and Linguistic Theory* 38:77–116.

Graf, Thomas. 2018. Why movement comes for free once you have adjunction. In *Proceedings of CLS 53*, edited by Daniel Edmiston, Marina Ermolaeva, Emre Hakgüder, Jackie Lai, Kathryn Montemurro, Brandon Rhodes, Amara Sankhagowit, and Miachel Tabatowski, 117–136.

Graf, Thomas. 2022. Subregular linguistics: bridging theoretical linguistics and formal grammar. *Theoretical Linguistics* 48:145–184.

Graf, Thomas, and Kalina Kostyszyn. 2021. Multiple wh-movement is not special: The subregular complexity of persistent features in Minimalist Grammars. In *Proceedings of the Society for Computation in Linguistics 2021*, edited by Allyson Ettinger, Ellie Pavlick, and Brandon Prickett, 275–285. Online: Association for Computational Linguistics. URL https://aclanthology.org/2021.scil-1.25/

Graf, Thomas, and Nazila Shafiei. 2019. C-command dependencies as TSL string constraints. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2019*, edited by Gaja Jarosz, Max Nelson, Brendan O'Connor, and Joe Pater, 205–215.
URL https://aclanthology.org/W19-0121/

References VII

Hanson, Kenneth. 2023. A TSL analysis of Japanese case. In *Proceedings of the Society for Computation in Linguistics 2023*, edited by Tim Hunter and Brandon Prickett, 15–24. Amherst, MA: Association for Computational Linguistics. URL https://aclanthology.org/2023.scil-1.2/

Hanson, Kenneth. to appear. Tier-based strict locality and the typology of agreement. *JOurnal of Language Modelling* .

Hao, Sophie. 2024. Universal Generation for Optimality Theory Is PSPACE-Complete. *Computational Linguistics* 1–35.

Hao, Yiding. 2019. Finite-state optimality theory: non-rationality of harmonic serialism. *Journal of Language Modelling* 7:49–99.

Heinz, Jeffrey, and Regine Lai. 2013. Vowel harmony and subsequentiality. In *Proceedings of the 13th Meeting on the Mathematics of Language (MoL 13)*, edited by Andras Kornai and Marco Kuhlmann, 52–63. Sofia, Bulgaria.

Jardine, Adam, Jane Chandlee, Rémi Eyraud, and Jeffrey Heinz. 2014. Very efficient learning of structured classes of subsequential functions from positive data. In *Proceedings of the Twelfth International Conference on Grammatical Inference (ICGI 2014)*, edited by Alexander Clark, Makoto Kanazawa, and Ryo Yoshinaka, vol. 34, 94–108. JMLR: Workshop and Conference Proceedings.

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

Jardine, Adam, and Kevin McMullin. 2017. Efficient learning of tier-based strictly k-local languages. In *Language and Automata Theory and Applications, 11th International Conference*, edited by Frank Drewes, Carlos Martín-Vide, and Bianca Truthe, Lecture Notes in Computer Science, 64–76. Springer.

Johnson, C. Douglas. 1972. *Formal Aspects of Phonological Description*. The Hague: Mouton.

Kaplan, Ronald, and Martin Kay. 1994. Regular models of phonological rule systems. *Computational Linguistics* 20:331–378.

Karakas, Ayla, and Hossep Dolatian. In revision. Computationally, reduplication and pseudo-reduplication are different .

Kleene, S.C. 1956. Representation of events in nerve nets. In *Automata Studies*, edited by C.E. Shannon and J. McCarthy, 3–40. Princeton University. Press.

Lambert, Dakotah. 2021. Grammar interpretations and learning tsl online. In *Proceedings of the Fifteenth International Conference on Grammatical Inference*, edited by Jane Chandlee, Rémi Eyraud, Jeff Heinz, Adam Jardine, and Menno van Zaanen, vol. 153 of *Proceedings of Machine Learning Research*, 81–91. PMLR. URL https://proceedings.mlr.press/v153/lambert21a.html

References XI

Lambert, Dakotah. 2022. Unifying classification schemes for languages and processes with attention to locality and relativizations thereof. Doctoral dissertation, Stony Brook University.
URL
https://vvulpes0.github.io/PDF/dissertation.pdf/

Lambert, Dakotah. 2023. Relativized adjacency. *Journal of Logic Language and Information* 32:707–731.

Lambert, Dakotah. to appear. Multitier phonotactics. *Phonology* .

Lambert, Dakotah, and Jeffrey Heinz. 2023. An algebraic characterization of total input strictly local functions. In *Proceedings of the Society for Computation in Linguistics*, vol. 6, 25–34.

References XII

Lamont, Andrew. 2021. Optimizing over subsequences generates context-sensitive languages. *Transactions of the Association for Computational Linguistics* 9:528–537.

Lamont, Andrew. 2022. Optimality theory implements complex functions with simple constraints. *Phonology* 38:729–740.

Lindell, Steven, and Jane Chandlee. 2016. A logical characterization of input strictly local functions Presented at the Fourth Workshop on Natural Language and Computer Science, in affiliation with LICS 2016.

Nelson, Scott. 2022. Are representations in articulatory and generative phonology so different? Poster presented at the 2022 Annual Meeting on Phonology.

Nelson, Scott. 2024. The computational structure of phonological and phonetic knowledge. Doctoral dissertation, Stony Brook University.

References XIII

Odden, David. 2014. *Introducing Phonology*. 2nd ed. Cambridge University Press.

Oncina, José, Pedro García, and Enrique Vidal. 1993. Learning subsequential transducers for pattern recognition tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15:448–458.

Perles, Micha A., Michael Oser Rabin, and Eliahu Shamir. 1963. The theory of definite automata. *IEEE Transactions on Electronic Computers* 12:233–243.

Piggot, Glynn. 1996. Implications of consonant nasalization for a theory of harmony. *Canadian Journal of Linguistics* 41.

Riggle, Jason. 2004. Generation, recognition, and learning in finite state Optimality Theory. Doctoral dissertation, University of California, Los Angeles.

Scott, Dana, and Michael Rabin. 1959. Finite automata and their decision problems. *IBM Journal of Research and Development* 5:114–125.

Stabler, Edward P. 2019. Three mathematical foundations for syntax. *Annual Review of Linguistics* 5:243–260.

Strother-Garcia, Kristina. 2018. Imdlawn Tashlhiyt Berber syllabification is quantifier-free. In *Proceedings of the Society for Computation in Linguistics*, vol. 1. Article 16.

Strother-Garcia, Kristina. 2019. Using model theory in phonology: A novel characterization of syllable structure and syllabification. Doctoral dissertation, University of Delaware.

Swanson, Logan. 2024. Syntactic learning over tree tiers. In *Preliminary Proceedings of the ESSLLI 2024 Student Session*, edited by Francesco Di Cosmo and John Lindqvist.

Tesar, Bruce. 2014. *Output-driven Phonology*. Cambridge University Press.

Vaysse, Odile. 1986. Addition molle et fonctions *p*-locales. *Semigroup Forum* 34:157–175.

Vu, Mai Ha, Nazila Shafiei, and Thomas Graf. 2019. Case assignment in TSL syntax: a case study. In *Proceedings of the Society for Computation in Linguistics (SCiL) 2019*, edited by Gaja Jarosz, Max Nelson, Brendan O'Connor, and Joe Pater, 267–276.
URL https://aclanthology.org/W19-0127/