Morphological epenthesis in Serbo-Croatian noun inflection

Andrija Petrovic

SBU/RU MLRG, Mar 2 2021

Introduction

• Class I:	zavod institute.NOM.SG.M	zavod-a institute.GEN.SG.M
	sel-o village.NOM.SG.N	sel-a village.GEN.SG.N
	tele calf.NOM.SG.N	tele**t**-a calf.GEN.SG.N
• Class II:	ʒen-a woman.NOM.SG.F	ʒen-e woman.GEN.SG.F
• Class III:	reʧ word.NOM.SG.F	reʧ-i word.GEN.SG.F

Introduction

• Class I:	zavod institute.NOM.SG.M	zavod-a institute.GEN.SG.M
	sel-o village.NOM.SG.N	sel-a village.GEN.SG.N
	tele calf.NOM.SG.N	tele**t**-a calf.GEN.SG.N
• Class II:	ʒen-a woman.NOM.SG.F	ʒen-e woman.GEN.SG.F
• Class III:	reʧ word.NOM.SG.F	reʧ-i word.GEN.SG.F

Introduction

• Class I:	zavod institute.NOM.SG.M	zavod-a institute.GEN.SG.M
	sel-o village.NOM.SG.N	sel-a village.GEN.SG.N
	tele calf.NOM.SG.N	tele**t**-a calf.GEN.SG.N
• Class II:	ʒen-a woman.NOM.SG.F	ʒen-e woman.GEN.SG.F
• Class III:	reʧ word.NOM.SG.F	reʧ-i word.GEN.SG.F

morphologically conditioned consonant insertion
–
repairing vowel-final stems

Introduction

- part of a wider algorithmic process of noun inflection
    - Boolean Monadic Recursive Schemes (BMRSs)
    - analyzed as logical transductions on strings
        - stem + [Case Num] → inflected word   ~ realization rules
    - hierarchical ordering of more specific and less specific blocking and licensing structures
        - ~ realization rules intrinsically ordered by Pāṇini's principle (cf. Stump 2001)
    - possibility to compose BMRSs into a complete inflectional system
        - algorithm of inflectional class assignment

Morphologically conditioned consonant insertion

- SC stems are normally C-final; those that are not are repaired via consonant insertion

  → **if another suffix follows immediately**

otherwise the word is vowel-final (bare stem) and considered well-formed (as the nom.sg. suffix is *-o*/*-e*)

	tele	‘calf.NOM.SG’
	tele*t*-a	‘calf.GEN.SG.
/tele + ji/	[teletɕi]	‘calf-like(ADJ.)’

Morphologically conditioned consonant insertion

- *t* is not the general epenthetic consonant in this language

/zaova/	[zaova]	'sister-in-law'
/violina/	[vijolina]	'violin'

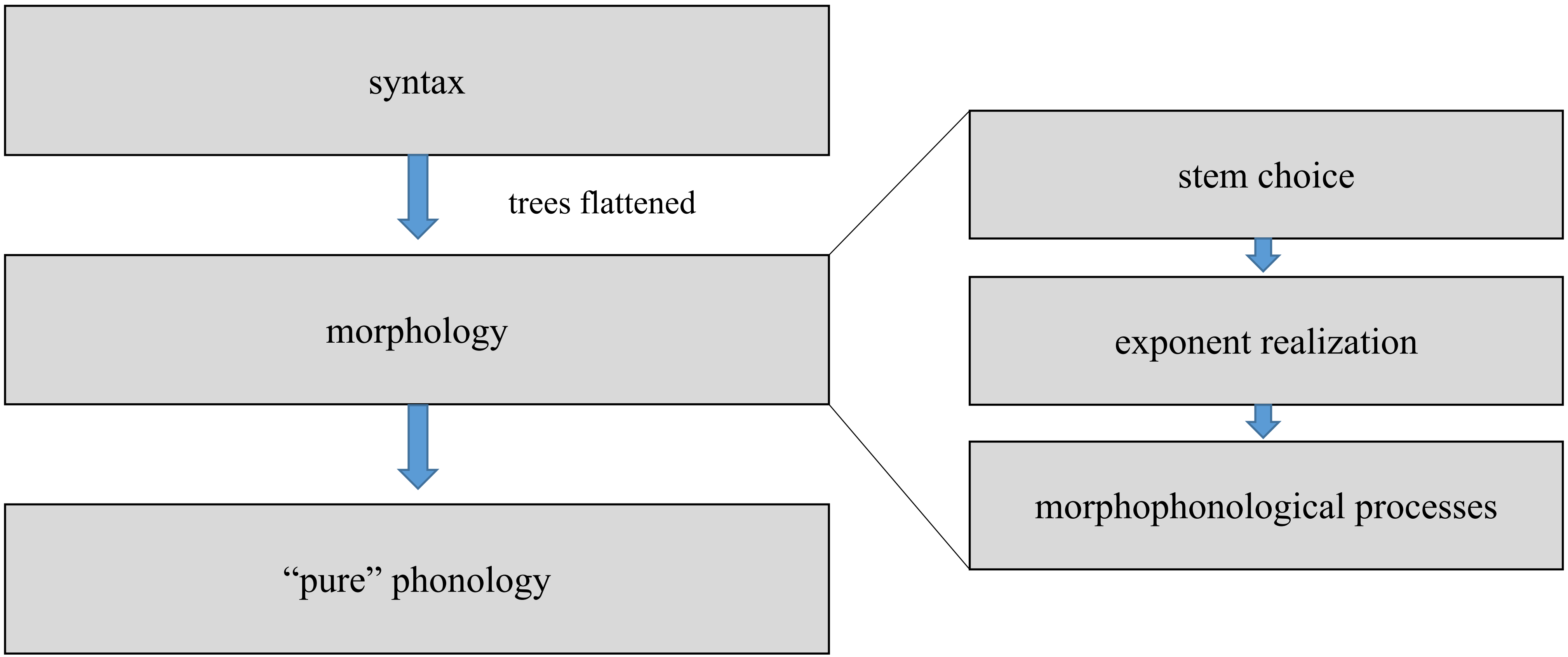
- *t* epenthesized only to repair vowel-final stems, does not depend on vowel hiatus

/tele + ji/	[teletɕi] 'calf-like'	vs.	/koz + ji/	[kozji] 'goat-like'
/sirtɕe + ni/	[sirtɕetni] 'acetic'	vs.	/plod + ni/	[plodni] 'fertile'

The nature of morphological processes

- morphological processes are regular (Karttunen et al. 1992), most often subregular (Chandlee 2017)
- DM is a tree-based theory, but can be formalized over strings (Ermolaeva & Edmiston 2018)
  - yield function flattens trees, retaining necessary boundary symbols
- In terms of computational power, W&P morphology is equivalent to a collection of regular relations (Karttunen 2003)

Proposed model



Analysis

- *Boolean Monadic Recursive Schemes* (BMRS)
  - incorporates the observation that phonology is regular (Johnson 1972; Kaplan & Kay 1994; Heinz 2018)
  - morphology, too, is regular (Karttunen et al. 1992), most often subregular (Chandlee 2017)
  - unlike FSTs, captures linguistically significant generalizations – we can use representations that are common in linguistics
    - phonological, morphosyntactic features
    - Pāṇini’s Principle (= Elsewhere condition) has no obvious finite-state implementation (Karttunen 2003), whereas it is directly captured with BMRS’s ‘if...then...else’ syntax

Analysis

- primitives:
  - *boolean values* ⊤ and ⊥
  - *monadic predicates* $P(t)$ – take a *single* argument *t,* and return ⊤ or ⊥

$\mathbb{V}$ = {a, e, i, o, u}
$\mathbb{C}$ = {b, ʦ, ʧ, ʨ, d, ʤ, ʥ, f, g, h, j, k, l, ʎ, m, n, ɲ, p, r, s, ʃ, t, v, z, ʒ}
$\mathbb{M}$ = {[nom], [gen], [dat], [acc], [voc], [ins], [loc], [sg], [pl]}

$\Sigma = \mathbb{V} \cup \mathbb{C} \cup \mathbb{M}$ $\qquad$ $\Sigma_{\bowtie} = \Sigma \cup \{\rtimes, \ltimes, +\}$

Analysis

- primitives:
  - *boolean values* ⊤ and ⊥
  - *monadic predicates* $P(t)$ – take a *single* argument *t,* and return ⊤ or ⊥

$\mathbb{V}$ = {a, e, i, o, u}
$\mathbb{C}$ = {b, ʦ, ʧ, ʨ, d, ʤ, ʥ, f, g, h, j, k, l, ʎ, m, n, ɲ, p, r, s, ʃ, t, v, z, ʒ}
$\mathbb{M}$ = {[nom], [gen], [dat], [acc], [voc], [ins], [loc], [sg], [pl]}

$\Sigma = \mathbb{V} \cup \mathbb{C} \cup \mathbb{M}$ $\Sigma_{\bowtie} = \Sigma \cup \{\rtimes, \ltimes, +\}$

$\mathcal{I}$ = {a(t), …, ʒ(t), [nom](t), …, [pl](t), ⋊(t), ⋉(t), +(t)}
$\mathcal{O}$ = {$\mathrm{a}_{\mathrm{o}}(t)$, …, $ʒ_{\mathrm{o}}(t)$, $[\mathrm{nom}]_{\mathrm{o}}(t)$, …, $[\mathrm{pl}]_{\mathrm{o}}(t)$, $\rtimes_{\mathrm{o}}(t)$, $\ltimes_{\mathrm{o}}(t)$, $+_{\mathrm{o}}(t)$}

Analysis

- no right word boundary symbol → stems are (contiguous) strings of characters occupying indices belonging to a *countably infinite* set of indices

$\mathbf{S} = \langle D; \sigma_1, \sigma_2, \ldots, p, s \rangle$

$D = \{1, 2, \ldots\}$

- the right word boundary is assigned at the end of the word only after the inflectional exponents are realized (look out for **⋉(*x*)** in the following slides!)
- (I do this to avoid using copy sets, which might not be a great idea.)

Analysis

- In BMRS, we can refer to both input and output predicates, and define new ones

input

1	2	3	4	5	6	7	…
⋊	t	e	l	e	+	[dat sg]	…

output

	t	e	l	e	t	u	

- Examples of defining new input predicates: case/number feature combinations, syncretism, defining the class of FCs

Analysis

- necessary combinations of case and number features available for neuter nouns:

`[nom/acc/voc sg]`(x) = if `[nom/acc/voc]`(x) then `[sg]`(x) else ⊥
`[dat/loc sg]`(x) = if `[dat/loc]`(x) then `[sg]`(x) else ⊥
`[ins sg]`(x) = if `[ins]`(x) then `[sg]`(x) else ⊥
`[nom/acc/voc pl]`(x) = if `[nom/acc/voc]`(x) then `[pl]`(x) else ⊥
`[gen pl]`(x) = if `[gen]`(x) then `[pl]`(x) else ⊥
`[dat/loc pl]`(x) = if `[dat/loc]`(x) then `[pl]`(x) else ⊥
`[ins pl]`(x) = if `[ins]`(x) then `[pl]`(x) else ⊥

(one would expect 14 – 7 cases × 2 numbers – but we do not have 14 distinct phonological forms of overt suffixes)

Analysis

- (non-directional) syncretism

`[dat/loc]`(x) = if `[dat]`(x) then ⊤ else `[loc]`(x)
`[nom/acc/voc]`(x) = if `[nom]`(x) then ⊤ else
if `[acc]`(x) then ⊤ else `[voc]`
`[gen/ nom/acc/voc pl]`(x) = if `[gen]`(x) then ⊤
else `[nom/acc/voc pl]`(x)
`[dat/loc pl/ins pl]`(x) = if `[dat/loc pl]`(x) then ⊤
else `[ins pl]`(x)

Analysis

- effect of fronting consonants (i.e. fronting of o to e after stem-final FCs)

(FC = C[cor, -ant] + ʦ)

[FC](*x*) = if ʦ(*x*) then ⊤ else
if ʧ(*x*) then ⊤ else
if ʨ(*x*) then ⊤ else
if ʤ(*x*) then ⊤ else
if ʥ(*x*) then ⊤ else
if ɟ(*x*) then ⊤ else
if ʎ(*x*) then ⊤ else
if ɲ(*x*) then ⊤ else
if ʃ(*x*) then ⊤ else ʒ(*x*)

Analysis

- *t*-insertion

$$\begin{aligned}\mathtt{outseg}(x) = {} & \text{if } \mathtt{a}_o(x) \text{ then } \top \text{ else} \\ & \text{if } \mathtt{b}_o(x) \text{ then } \top \text{ else} \\ & \text{if } \mathtt{c}_o(x) \text{ then } \top \text{ else} \\ & (...) \\ & \text{if } \mathtt{z}_o(x) \text{ then } \top \text{ else } \mathtt{ʒ}_o(x)\end{aligned}$$

$$\begin{aligned}\mathtt{t}_o(x) = {} & \text{if } +(x) \text{ then} \\ & \quad \text{if } \mathtt{e}(p(x)) \text{ then } \mathtt{outseg}(s(x)) \text{ else } \bot \\ & \text{else } \mathtt{t}(x)\end{aligned}$$

Analysis

- Output predicates: (~ realization rules)

$\mathtt{u}_o(x)$ = if $\mathtt{[dat/loc\ sg]}(x)$ then $\top$ else $\mathtt{u}(x)$
$\mathtt{a}_o(x)$ = if $\mathtt{[gen/\ nom/acc/voc\ pl]}(x)$ then $\top$ else
if $\mathtt{[gen\ pl]}(p(x))$ then $\top$ else
if $\mathtt{[ins\ pl]}(p(p(x)))$ then $\top$ else $\mathtt{a}(x)$
$\mathtt{o}_o(x)$ = if $\mathtt{[nom/acc/voc\ sg]}(x)$ then $\top$ else
if $\mathtt{[ins\ sg]}(x)$ then $\top$ else $\mathtt{o}(x)$
$\mathtt{m}_o(x)$ = if $\mathtt{[ins]}(p(x)))$ then $\top$ else $\mathtt{m}(x)$
$\mathtt{i}_o(x)$ = if $\mathtt{[ins\ pl]}(x)$ then $\top$ else $\mathtt{i}(x)$
$\mathtt{e}_o(x)$ = if $\mathtt{o}_o(x)$ then
if $+(p(x))$ then $\mathtt{[pal]}(p(p(x)))$ else $\bot$
else $\mathtt{e}(x)$

$\mathtt{l}_o(x) = \mathtt{l}(x)$
$\mathtt{b}_o(x) = \mathtt{b}(x)$
(etc., for {ʦ, ʧ, ʨ, d, ʤ, ʥ, f, g, h, j, k, l, ʎ, n, ɲ, p, r, s, ʃ, v, z, ʒ})

Analysis

	1	2	3	4	5	6	7	8	9
input	⋊	t	e	l	e	+	[ins sg]		…
output		t	e	l	e	t	o	m	

- Output predicates:

$\texttt{u}_o(x)$ = if $\texttt{[dat/loc sg]}(x)$ then $\top$ else $\texttt{u}(x)$
$\texttt{a}_o(x)$ = if $\texttt{[gen/ nom/acc/voc pl]}(x)$ then $\top$ else
  if $\texttt{[gen pl]}(p(x))$ then $\top$ else
  if $\texttt{[ins pl]}(p(p(x)))$ then $\top$ else $\texttt{a}(x)$
$\texttt{o}_o(x)$ = if $\texttt{[nom/acc/voc sg]}(x)$ then $\top$ else
  if $\texttt{[ins sg]}(x)$ then $\top$ else $\texttt{o}(x)$
$\texttt{m}_o(x)$ = if $\texttt{[ins]}(p(x)))$ then $\top$ else $\texttt{m}(x)$
$\texttt{i}_o(x)$ = if $\texttt{[ins pl]}(x)$ then $\top$ else $\texttt{i}(x)$
$\texttt{e}_o(x)$ = if $\texttt{o}_o(x)$ then
  if $+(p(x))$ then $\texttt{[pal]}(p(p(x)))$ else $\bot$
  else $\texttt{e}(x)$

$\texttt{l}_o(x) = \texttt{l}(x)$
$\texttt{b}_o(x) = \texttt{b}(x)$
(etc., for {ʦ, ʧ, ʨ, d, ʤ, ʥ, f, g, h, j, k, l, ʎ, n, ɲ, p, r, s, ʃ, v, z, ʒ})

Analysis

$\ltimes(x)$ = if `[nom sg]`(x) then `e`$(p(x))$ else
if `[nom/acc/voc sg]`$(p(x))$ then $\top$ else
if `[dat/loc sg]`$(p(x))$ then $\top$ else
if `[gen/ nom/acc/voc pl]`(x) then $\top$ else
if `[ins sg]`$(p(p(x)))$ then $\top$ else
if `[dat/loc pl/ins pl]`$(p(p(p(x))))$ then $\top$ else $\bot$

out(x) = if $+(x)$ then $\mathtt{t}_o(x)$ else
if $\rtimes(x)$ then $\bot$ else
if $\ltimes(x)$ then $\bot$ else $\top$

Analysis

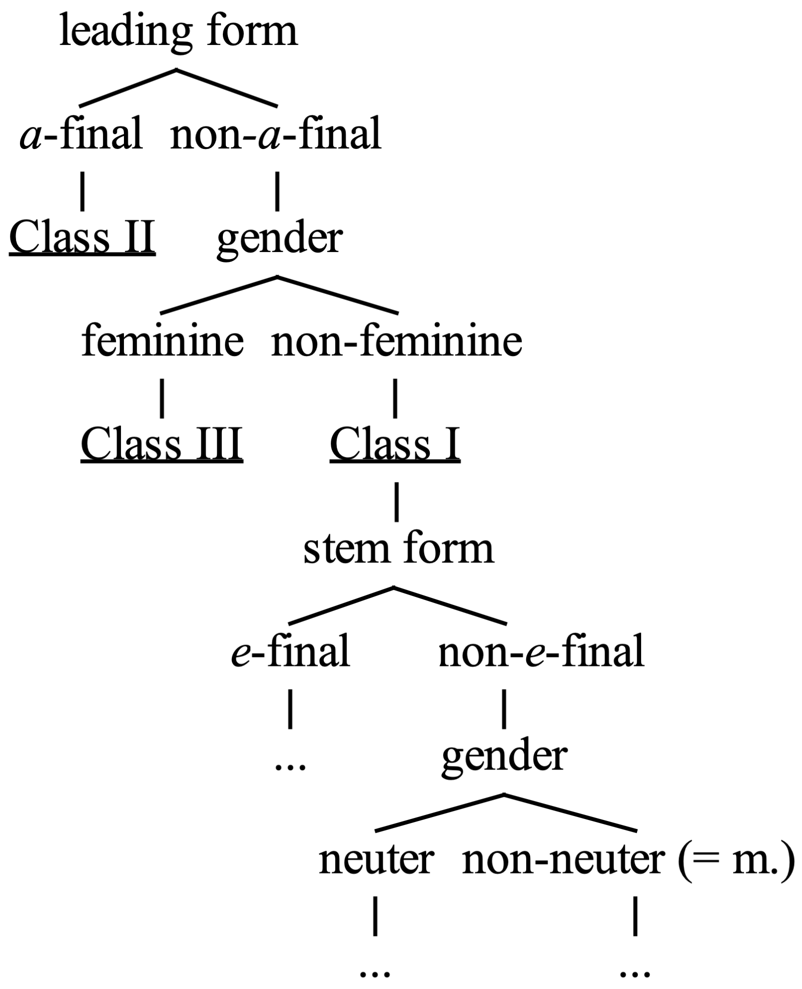
- no copy sets = no epenthesis/insertion of material longer that 1 symbol ☹

- is there a smart way to do this (+ more → non-concatenative morphology) without copy sets? The non-computationally minded often find them counterintuitive

Class membership assignment in SC

- Class membership assignment in SC can be modeled as a predictable process if we take into account a *leading form*/*base* (Wurzel 1990, McCarthy 2005, Albright 2008)
- This I argue to be the nominative singular: it is the most frequent form (subject case), and it ranks highest in the Case Hierarchy (Blake 2001).
- loanwords do not (necessarily) have an oblique form – they are incorporated into the declension system on the basis of their input form (which is borrowed as nominative singular)
- so, lexical entry = {stem, leading form, morphosyntactic info}, but the algorithm should work with a deficient lexical entry too

Class membership assignment in SC



Conclusions

- Inserting *t* repairs aberrant, V-final stems (whenever an overt suffix is attached)
- This happens regardless of whether that suffix is vowel- or consonant-initial
- A means to avoid stipulating listed stems – assuming unpredictable stem allomorphs would basically reduce the phenomenon to an accident
- Assuming morphological epenthesis enables us to express the generalizations explicitly and overtly

Conclusions

- BMRSs can directly capture morphological and phonological generalizations, retaining the computationally restrictive nature of such processes
- Intuitive, easily implementable, extendable to a wider range of phenomena
- Future work: noun class membership assignment base on ‘if…then…else’ syntax; class membership does not have to be listed information

References

• Albright, A. (2008). Inflectional Paradigms Have Bases Too. Arguments from Yiddish. In: A. Bachrach & A. Nevins (Eds.), *The Bases of Inflectional Identity*. Oxford: Oxford University Press.

• Bhaskar, S., Chandlee, J., Jardine, A., & Oakden, C. (2020). Boolean monadic recursive schemes as a logical characterization of the subsequential functions. In *International Conference on Language and Automata Theory and Applications* (pp. 157-169). Springer, Cham.

• Blake, B. J. (2001). *Case*. Cambridge: Cambridge University Press.

• Chandlee, J. (2017). Computational locality in morphological maps. *Morphology*, 27:599– 641.

• Chandlee, J., & Jardine, A. (2020). Recursive Schemes for Phonological Analysis. Unpublished MS. Retrieved from http://adamjardine.net/files/chandleejardineBMRSms.pdf

• Ermolaeva, M., & Edmiston, D. (2018). Distributed morphology as a regular relation. *Proceedings of the Society for Computation in Linguistics*, *1*(1), 178-181.

• Heinz, J. (2018). The computational nature of phonological generalizations. In Hyman, L. and Plank, F., editors, *Phonological Typology,* Phonetics and Phonology, chapter 5, pages 126–195. De Gruyter Mouton.

• Johnson, C. D. (1972). *Formal aspects of phonological description*. Mouton.

• Kaplan, R. and Kay, M. (1994). Regular models of phonological rule systems. *Computational Linguistics*, 20:331–78.

• Karttunen, L. (2003). Computing with realizational morphology. In *International Conference on Intelligent Text Processing and Computational Linguistics* (pp. 203-214). Springer, Berlin, Heidelberg.

• Karttunen, L., Kaplan, R. M., & Zaenen, A. (1992). Two-level morphology with composition. In *COLING 1992 Volume 1: The 15th International Conference on Computational Linguistics.*

• McCarthy, J. (2005). Optimal Paradigms. In: L. Downing, T. Hall & R. Raffelsiefen (Eds.), *Paradigms in Phonological Theory*. Oxford: Oxford University Press, pp. 170–210.

• Moradi, S. (2017). *Non-canonical Epenthesis: Epenthetic quality and the role of morphonology* (ms). Stony Brook University.

• Repetti, L., Moradi, S. & Aronoff, M. (2018). *Epenthesis and Morphology in Romance.* Presented at LSRL 48, York University, Toronto.

• Stump, G. T. (2001). *Inflectional morphology: A theory of paradigm structure*. Cambridge: Cambridge University Press.

• Wurzel, W. U. (1990). The Mechanism of Inflection: Lexical Representation, Rules, and Irregularities. In: W. U. Dressler (Ed.), *Contemporary Morphology*. Berlin: Mouton de Gruyter.