Linearization and a Local Syntax-Phonology Interface

Vincent Czarnecki

September 28, 2024

Main Questions

- How can we understand syntax-phonology interaction as an *interpretation* of syntactic representation through the lens of phonological representation?

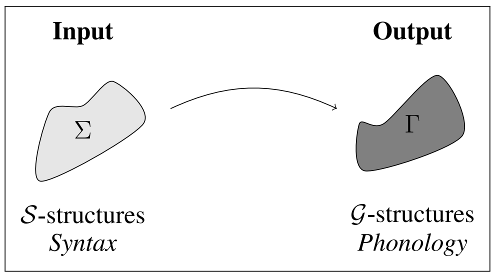

Figure 1: Sketch of Linerization as Logical Interpretation

- How can this help us understand the typology phrasal phonological patterns in a computationally restrictive manner?

Introduction

- A central question in the syntax-phonology interface literature is how to address mismatches between syntactic and prosodic structure in phrasal phonology (*chunks*$\neq$*chunks*).

Syntactic Domains $\neq$ Phonological Domains

Example: Chimwiini Right Alignment

As Kalivoda (2018) explains, words receive an accent at the right edges of XP's (in **bold**) but otherwise do not, which is argued to indicate the right edge of a phonological phrase φ:

(1) *(om**á**ri)*$_\varphi$ *(liweele kuwa ham**á**di)*$_\varphi$ *(uzile g**á**ari)*$_\varphi$
(omari)$_\varphi$ (forgot that hamadi)$_\varphi$ (bought car)$_\varphi$
'omari forgot that hamadi bought a car'

Spell-out is triggered (indicated by dotted line) at phase heads (**bolded**), and the Spell-out domain is the complement of the phase head.

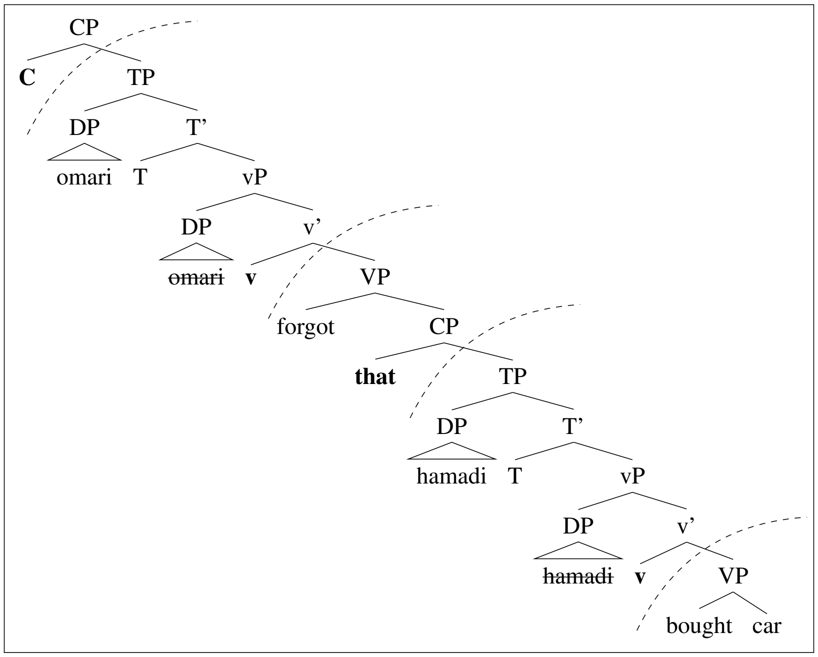

Figure 2: Tree for Chimwiini "omari forgot that hamadi bought a car."

If we take a purely Spell-out driven φ-domain formation approach, then the following phonological phrasing is predicted:

[**omari** [**forgot that** [**hamadi** [**bought car**]]]]

$$\downarrow$$

$$\boxed{(\text{omari})_\varphi\ (\text{forgot that})_\varphi\ (\text{hamadi})_\varphi\ (\text{bought car})_\varphi}$$

- *Close, but no cigar!* Purely Spell-out based φ-domain formation makes the wrong prediction; however, note that it is only *off by a little bit* when compared to the prediction above.

  Picture a rule: If a single-word φ-domain is immediately preceded by a two or more-word φ-domain, incorporate them into a single domain: $\boxed{\omega\ \omega)_\varphi(\omega)_\varphi \rightarrow \omega\ \omega\ \omega)_\varphi}$

  This gives precisely the attested phrasing: $\boxed{(\text{omari})_\varphi\ (\text{forgot that hamadi})_\varphi\ (\text{bought car})_\varphi}$

Main Intuition

Syntax-prosody mismatches occur *at or very near to* phase boundaries, where *at or very near to* is formally captured using Input Strict Local (ISL) restructuring maps at the linearized, morphological level.

- This opposes more mainstream approaches such as MATCH Theory which see the interface as optimization between recursive syntactic and prosodic structures (Lee and Selkirk, 2022; Kratzer and Selkirk, 2020; Selkirk, 2011).

  Instead, the view taken here aligns with claims that there is no recursion in the phonological module (Idsardi and Raimy, 2013; Idsardi, 2018; Scheer, 2023).

  "Recursive structure in phonology only exists in order to satisfy some theoretical purpose... If the theory were different, i.e. if there were no principle of foot binarity or the idea that phonological structure mimics morpho-syntactic structure, there would be no recursion. In other words, alleged phonological recursion is always theory-born: it is created by the analyst in order to satisfy some specific assumptions." (Scheer, 2023)

- Also knowing that unbounded prosodic recursion is non-finite state (Dolatian et al., 2021), this is a big theoretical buy-in because when invoked, prosodic recursion is usually restricted to two or three levels.

  Additionally, there must also be an external motivation for positing a maximum recursion depth.

- The temporal nature of the speech stream requires linearity anyway, why not assume that trees are flattened right out of the syntax such that *the narrow syntax requires hierarchy but not linearity, and post-syntax requires linearity but lacks hierarchy*.

- For these reasons, I am assuming the following architecture:

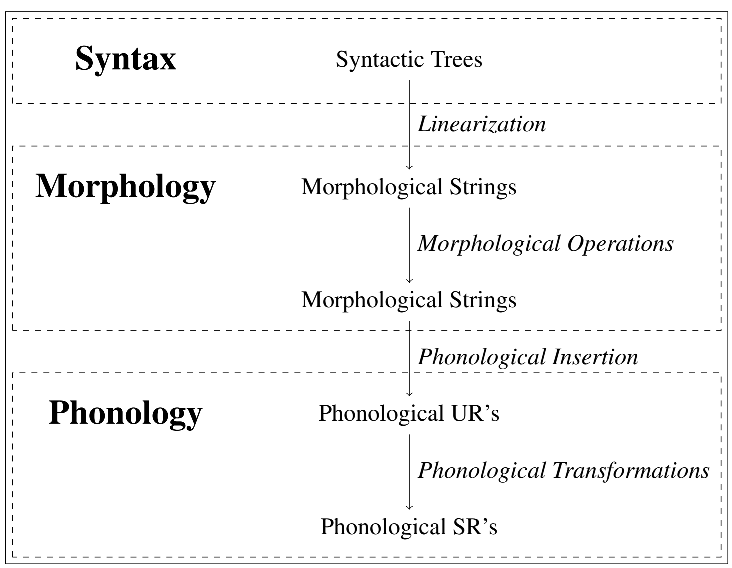

Figure 3: Proposed Schema of Syntax-Phonology Interaction

Model Theoretic Analysis

Relational Tree Models

- Relational tree models will be defined in the following way[1]: $\langle D \mid \triangleleft^*, \prec, \sigma_i \in \Sigma \rangle$, where:
  - $D \subsetneq \mathbb{N}$ is a domain of nodes over the tree
  - $\triangleleft^*(x, y)$ is the binary general dominance relation
  - $\prec (x, y)$ is the binary syntactic selection relation
  - $\sigma_i(x) \in \Sigma$ are unary relations over an alphabet Σ of morphosyntactic labels/features

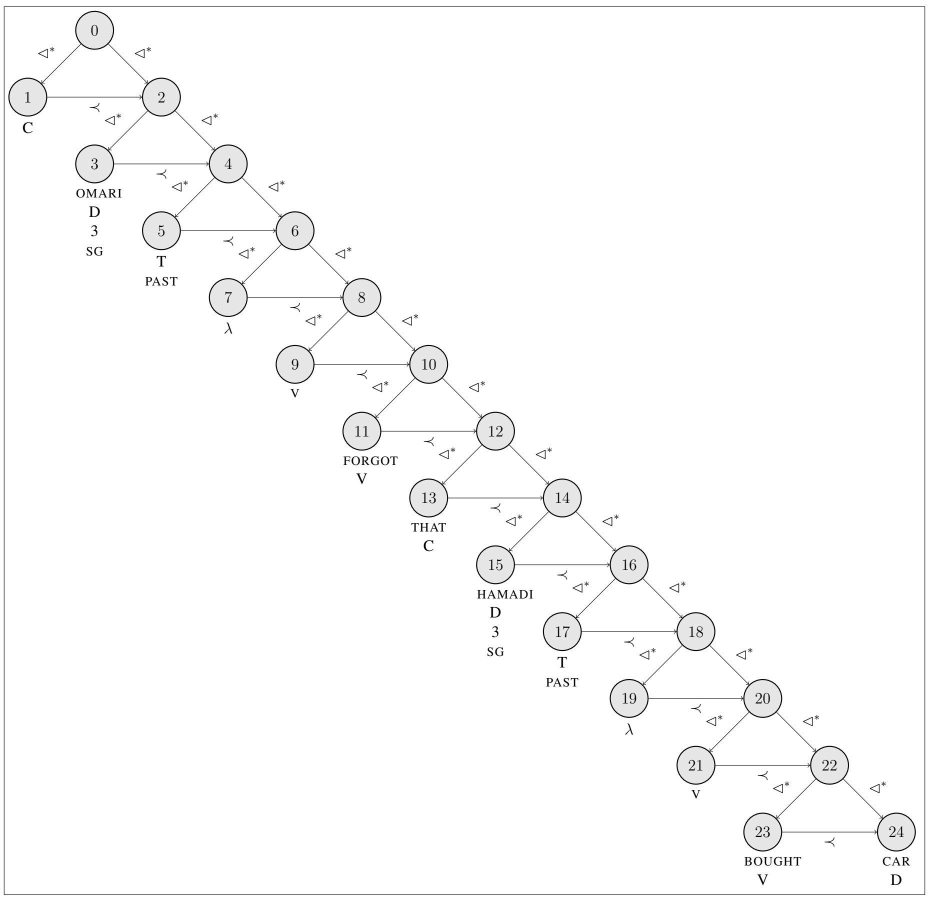

Figure 4: Tree Model for Chimwiini "Omari forgot that Hamadi bought car"

[1]I assume that there is no linear order in the syntactic representation, but representations in this vein that have been standard in Model Theoretic Syntax (Rogers and Nordlinger, 1998). Currently I am considering replacing selection $\prec (x, y)$ with a symmetric Merge relation $\mathcal{M}(x, y)$. In this sense, trees are unordered and selection can be inferred by featural composition of nodes, not changing any of the linearization facts introduced below. Thus, precedence is inferred from the syntax for linearization purposes but it is unnecessary for narrow syntactic computation.

- The goal now is to determine how to linearize structures like this, which is intuitively simple but is a nontrivial task for a few reasons:
  - Trees can have left/right branches of arbitrary length.
  - We don't think of linearization as one big tree-flattening event, it proceeds in chunks.

  Crucially, the linearized string will be defined in terms of **strict precedence**, not general precedence (in principle, differing from the Linear Correspondence Axiom (Kayne, 1994)).[2]

  We will take a declarative approach that specifies:

  $\boxed{\text{My output string will look like } Y \text{ if my input tree looks like } X.}$

Linearization as an Interpretation (Prelim.)

- Let $\Sigma = \{\text{A,B,C,D,E,F,G,H,I,J,K}\}$ be an alphabet of abstract symbols, and consider the input tree over Σ and its full linearization below.

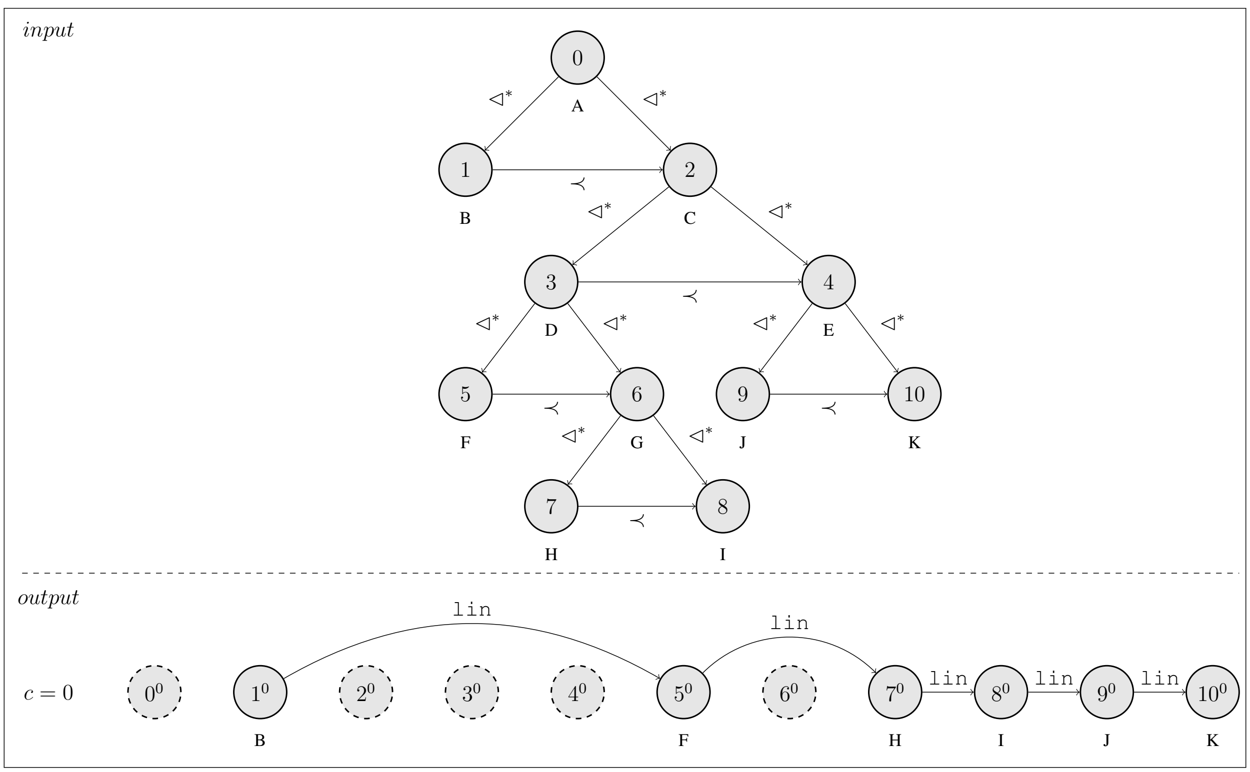

Figure 5: Linearization Toy Example

[2]This also "bakes in" the importance of locality in morphological processes Embick and Noyer (1999); Aksënova and De Santo (2018); Chandlee (2017).

- To understand how this works, let's define some helper predicates:
- A node is a leaf if there are no nodes that it dominates:

$$\texttt{leaf}(x) := \neg\exists z[\triangleleft^*(x,z)]$$

- With this, we can say that in the output string, a node will be labeled with $\sigma \in \Sigma$ iff it was a leaf labeled σ in the input tree:

$$\begin{aligned}
\mathrm{A}^0(x) &:= \mathrm{A}(x) \wedge \texttt{leaf}(x)\\
\mathrm{B}^0(x) &:= \mathrm{B}(x) \wedge \texttt{leaf}(x)\\
&\vdots\\
\mathrm{K}^0(x) &:= \mathrm{K}(x) \wedge \texttt{leaf}(x)
\end{aligned}$$

- This covers the labeling relations for the output, but now the ordering relations are shown.
- For clarity, the following predicate indicates that a node y is dominated by x and dominates z and so we say that y is *between* x and z in the tree:

$$\texttt{between}(x,y,z) := \triangleleft^*(x,y) \wedge \triangleleft^*(y,z)$$

- A *left-leaf* is a leaf that has nothing selecting it, and a *right-leaf* is a leaf that selects nothing:

$$\begin{aligned}
\texttt{left-leaf}(x) &:= \texttt{leaf}(x) \wedge \neg\exists y[\prec(y,x)]\\
\texttt{right-leaf}(x) &:= \texttt{leaf}(x) \wedge \neg\exists y[\prec(x,y)]
\end{aligned}$$

- For a given node, whichever node is the (unique!) leaf-node below it such that there is nothing further left is its leftmost leaf.

$$\texttt{lml}(x,y) := \forall z[(\triangleleft^*(y,z) \wedge \texttt{left-leaf}(z) \wedge \forall s[\texttt{between}(y,s,z) \wedge \neg\exists t[\prec(t,s)]]) \leftrightarrow z = x]$$

- For a given node, whichever node is the (unique!) leaf-node below it such that there is nothing further right is its rightmost leaf.

$$\texttt{rml}(x,y) := \forall z[(\triangleleft^*(y,z) \wedge \texttt{right-leaf}(z) \wedge \forall s[\texttt{between}(y,s,z) \wedge \neg\exists t[\prec(s,t)]]) \leftrightarrow z = x]$$

- Note that any leaf node is its own leftmost and rightmost leaf.
- Looking back at the tree, why are these notions relevant?
- *Fast Forward*: entire state of affairs can be encapsulated by the following condition: x is the rightmost leaf of a node t, **and** y is the leftmost leaf of a node s, **and** t selects s, shown pictorially below:

$$\texttt{lin}(x, y) := \exists t \exists s[\texttt{rml}(t, x) \wedge \texttt{lml}(s, y) \wedge \prec (t, s)]$$

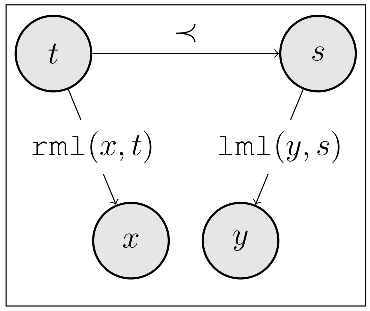

Figure 6: Input Tree Conditions for Strict Precedence in Output String

Incorporating Phases

- At this point, we have a reliable tree flattener; however, if we think that the derivation proceeds in chunks, we need some way of constraining when this will apply.
- According to standard assumptions in the syntactic literature, the sister of a (strong) phase head undergoes Spell-out (Chomsky, 2001).

  Consider the example below in Figure 7 (modified from Dobashi (2016)) where the heads W and Y are phase heads (in **bold**).

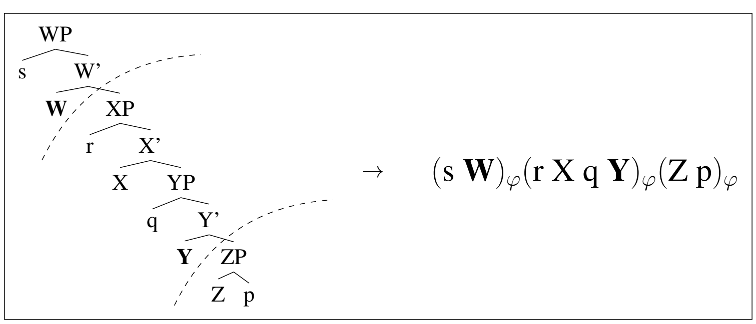

Figure 7: Linearization by Phase Toy Exmaple

- Spell-out occurs at each phase head. Thus, once the derivation reaches Y, its sister ZP containing Z and p is spelled out and inaccessible to the remainder of the derivation.

Once the next phase head W is reached in the derivation, Spell-out is triggered and its sister XP containing r, X, q, and Y is spelled out and inaccessible to the remainder of the derivation.

- Observe the following visual representation of the interpretation (definition in Appendix):

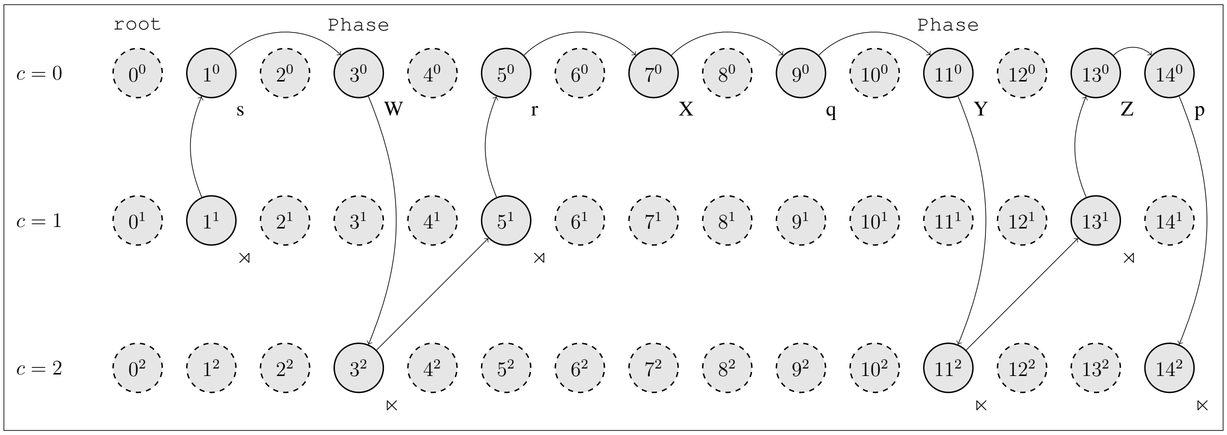

Figure 8: Linearization by Phase as a Logical Transduction

The Morphological Level

- For simplicity of presentation the entire graphical representations of the models are not shown but rather a reduced version is shown.

  This reduced version contains the lexical label a node bears (OMARI, FORGOT, HAMADI, etc.), the syntactic category (C, T, V, etc.) if there is no lexical label, the boundary symbols (⋊, ⋉) or the empty string λ.

- Consider the following series of mappings on that original string:

⋊C⋉ ⋊OMARI T λ V⋉ ⋊FORGOT THAT⋉ ⋊HAMADI T λ V⋉ ⋊BOUGHT CAR⋉

$\downarrow_{\lambda\text{-deletion}}$

⋊C⋉ ⋊OMARI T V⋉ ⋊FORGOT THAT⋉ ⋊HAMADI T V⋉ ⋊BOUGHT CAR⋉

$\downarrow_{\text{non-label deletion}}$

⋊OMARI⋉ ⋊FORGOT THAT⋉ ⋊HAMADI⋉ ⋊BOUGHT CAR⋉

$\downarrow_{\text{restructuring}}$

⋊OMARI⋉ ⋊FORGOT THAT HAMADI⋉ ⋊BOUGHT CAR⋉

There are all sorts of mappings that can occur at this morphological string level; however, note that the restructuring map is ISL!

This depends on a *fixed-length bounded window* of prosodic words and boundaries in the input morphological string.

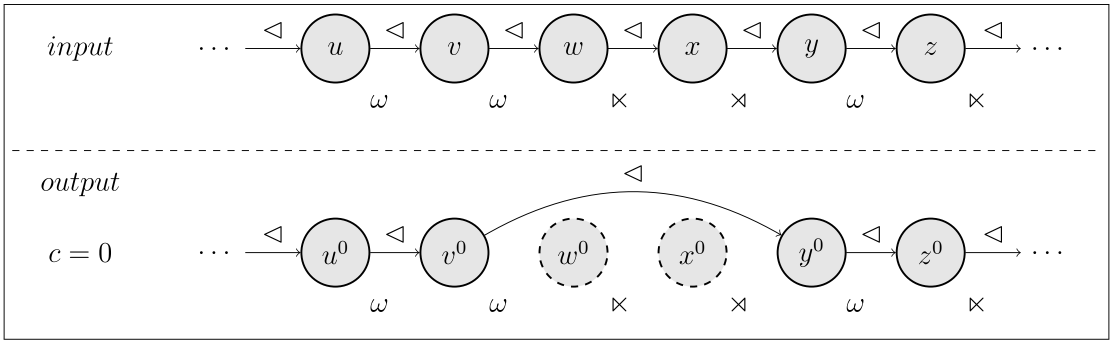

Figure 9: Domain Reconstruction as a Logical Transduction

The input configuration that leads to knowing whether a node surfaces as a left or right boundary symbol is given by the following:

$$\exists u, v, w, x, y, z[u \lhd v \lhd w \lhd x \lhd y \lhd z \wedge \omega(u) \wedge \omega(v) \wedge \ltimes(w) \wedge \rtimes(x) \wedge \omega(y) \wedge \ltimes(z)]$$

Recall the main intuition, that typology of phrasal patterns falls out from this!

Typology

- In Dobashi (2003) and subsequent work (Dobashi, 2016, 2019), the following typology of phonological phrasing patterns in SVO constructions is given:

Ewe	$(\mathrm{S})_\varphi(\mathrm{V})_\varphi(\mathrm{O})_\varphi$
Italian	$(\mathrm{S})_\varphi(\mathrm{V})\ _\varphi(\mathrm{O})_\varphi$ or $(\mathrm{S})_\varphi(\mathrm{V\ O})_\varphi$ if O is non-branching
Kimatuumbi	$(\mathrm{S})_\varphi(\mathrm{V\ O})_\varphi$
Kinyambo	$(\mathrm{S})_\varphi(\mathrm{V\ O})_\varphi$ or $(\mathrm{S\ V})_\varphi$ if S is non-branching

  This typological variation is accounted for by a combination of syntactic structure and phonological restructuring.

  The present work is a computational formalization and extension of the sorts of restructuring discussed in this work.

- Specifying branchingness (as in Italian and Kinyambo) is merely specifying conditions of a substring of morphological information regarding how many words are in a given domain.

  To tell whether a constituent was branching in the syntax is to tell whether it contains more than one word when linearized.

Further Ideas/Directions/Questions

- Empirical
  - Can optimization generate unwanted patterns of "long-distance" restructuring?
  - Can full vs. partial reduplication be actualized as reduplication at the linear morphological vs. morpho-phonological level,
- Theoretical
  - Movement Deletion: TSL mapping on linear morphological string on a movement feature F-tier?
  - Are there benefits to defining linearization functionally instead of relationally? (Given the relationship between QF transductions and ISL)
  - Can boundaries be encoded *relationally* as opposed to using symbols $\rtimes, \ltimes$? Something like $\varphi_{init}(x)$ and $\varphi_{fin}(x)$ (Would be better from a strict modularity perspective)
  - Set Merge vs. Pair Merge: irrelevant if $\prec$ can be inferred by featural composition?

References

Alëna Aksënova and Aniello De Santo. Strict locality in morphological derivations. In Proceedings of the 53rd meeting of Chicago Linguistics Society (CLS 53), 2018.

Jane Chandlee. Computational locality in morphological maps. Morphology, 27(4):599–641, 2017.

Noam Chomsky. Derivation by phase. An Annotated Syntax Reader, 482, 2001.

Yoshihito Dobashi. Phonological phrasing and syntactic derivation. Cornell University, 2003.

Yoshihito Dobashi. A prosodic domain= a spell-out domain? Phonological externalization, 1: 11–22, 2016.

Yoshihito Dobashi. Externalization: Phonological interpretations of syntactic objects. Routledge, 2019.

Hossep Dolatian, Aniello De Santo, and Thomas Graf. Recursive prosody is not finite-state. In Proceedings of the Seventeenth SIGMORPHON Workshop on Computational Research, 2021.

David Embick and Rolf Noyer. Locality in post-syntactic operations. MIT working papers in linguistics, 34(265-317), 1999.

William Idsardi and Eric Raimy. Three types of linearization and the temporal aspects of speech. Challenges to linearization, 1:31–56, 2013.

William J. Idsardi. Why Is Phonology Different? No Recursion, pages 212–223. Cambridge University Press, 2018.

Nicholas Kalivoda. Syntax-prosody mismatches in Optimality Theory. University of California, Santa Cruz, 2018.

Richard S Kayne. The antisymmetry of syntax, volume 25. MIT press, 1994.

Angelika Kratzer and Elisabeth Selkirk. Deconstructing information structure. Glossa: a journal of general linguistics, 5(1), 2020.

Seunghun J. Lee and Elisabeth Selkirk. Xitsonga tone: The syntax-phonology interface. In Prosody and Prosodic Interfaces. Oxford University Press, 05 2022.

James Rogers and Rachel Nordlinger. A descriptive approach to language-theoretic complexity. Citeseer, 1998.

Tobias Scheer. Recursion in phonology: Anatomy of a misunderstanding. Representing phonological detail: Segmental structure and representations, pages 265–287, 2023.

Elisabeth Selkirk. The syntax-phonology interface. The handbook of phonological theory, pages 435–484, 2011.

Appendix

Labeling and Precedence Relations for Linearization By Phase:

Labeling Relations:

- A node will bear σ iff it was a leaf node in the tree.

  $\sigma^0(x) := \texttt{leaf}(x)$

- A node will bear $\rtimes$ iff there's a phase p present and it met the conditions for linearization with x, or it is at the root's leftmost leaf.

  $\rtimes^1(x) := \exists p[(\texttt{Phase}(p) \wedge \texttt{lin}(p,x))] \vee \exists r[\texttt{root}(r) \wedge \texttt{lml}(r,x)]$

- A node will bear $\ltimes$ iff there's a phase p present, or it is at the root's rightmost leaf.

  $\ltimes^2(x) := \texttt{Phase}(x) \vee \exists r[\texttt{root}(r) \wedge \texttt{rml}(r,x)]$

- No other nodes will bear symbols.

  all others $\bot$

Precedence Relations:

- A node x will precede a node y iff they met the conditions for linearization and the first is not a phase.

  $\triangleleft_{\mu}^{0,0}(x, y) := \mathtt{lin}(x, y) \wedge \neg\mathtt{Phase}(x)$

- A node x will precede a right boundary y iff there's a phase p present, or it is at the root's rightmost leaf.

  $\triangleleft_{\mu}^{0,2}(x, y) := x = y \wedge (\mathtt{Phase}(x) \vee \exists r[\mathtt{root}(r) \wedge \mathtt{rml}(r, x)])$

- A left boundary x will precede a node y iff there's a phase p present and p meets the conditions of linearization with y, or it is at the root's leftmost leaf.

  $\triangleleft_{\mu}^{1,0}(x, y) := x = y \wedge (\exists p[\mathtt{Phase}(p) \wedge \mathtt{lin}(p, y)] \vee \exists r[\mathtt{root}(r) \wedge \mathtt{lml}(r, y)])$

- A left boundary will precede a right boundary iff x is at a phase and x met the conditions for linearization with y.

  $\triangleleft_{\mu}^{2,1}(x, y) := \mathtt{Phase}(x) \wedge \mathtt{lin}(x, y)$

- Precedence will not hold for any other combinations of nodes and boundaries.

  all others $\bot$