Logical Transductions for the Typology of Ditransitive Prosody

Mai Ha Vu
Dept. of Linguistics
and Scandinavian Studies
University of Oslo, Oslo, Norway
m.h.vu@iln.uio.no

Aniello De Santo
Department of Linguistics
University of Utah
Salt Lake City, Utah, USA
aniello.desanto@utah.edu

Hossep Dolatian
Department of Linguistics
Stony Brook University
Stony Brook, NY, USA
hossep.dolatian@stonybrook.edu

Abstract

Given the empirical landscape of possible prosodic parses, this paper examines the computations required to formalize the mapping from syntactic structure to prosodic structure. In particular, we use logical tree transductions to define the prosodic mapping of ditransitive verb phrases in SVO languages, building off of the typology described in Kalivoda (2018). Explicit formalization of syntax-prosody mapping revealed a number of unanswered questions relating to the fine details of theoretical assumptions behind prosodic mapping.

1 Introduction

Within computational and mathematical phonology, there is ample work on formalizing segmental and suprasegmental phonological processes that are word-bounded, such as by using finite state acceptors (FSAs) and transducers (FSTs) (Kaplan and Kay, 1994; Roche and Schabes, 1997; Hulden, 2009; Chandlee, 2014; Heinz, 2018), or using equivalent logical transductions (Potts and Pullum, 2002; Jardine, 2016; Strother-Garcia, 2019; Dolatian, 2020; Dolatian et al., 2021b).

Until recently however, there was little work on the computational machinery required by sentence-level or phrase-level phonology (prosodic phonology). This gap may be because early work on prosodic phonology found that some common aspects of prosody were computationally regular over strings, and can be formalized with FSAs (Pierrehumbert, 1980). However, the abstract representations that are the target of prosodic processes are subject to extensive debates in the linguistic literature, and they play a crucial role for questions about the nature of the linguistic phenomena at the phonology-syntax interface (Nespor and Vogel, 1986; Selkirk, 1982, 2011; Yu, 2021).

It is an established fact that phonological processes can refer to domains larger than a word. These domains form hierarchical layers: the prosodic word (w or PW), the prosodic phrase (p or PPh)[1], and the intonational phrase (i or iP). These prosodic constituents show systemic relations with syntactic constituents. However, such relations have been argued not to be strictly isomorphic — that is, prosodic constituency cannot be read directly from syntactic constituency. The characteristics of the *mapping* between syntactic structure and prosodic structure are important to theoretical approaches that consider prosodic constituency to be relevant for phonological generalizations. In this sense, distransitive constructions — verbs with multiple *internal arguments* (e.g. *gave Mary books*) — are a core example of prosodic-syntax mismatches cross-linguistically.

Building on the systematic report of such mismatches in SVO languages provided by Kalivoda (2018), this paper works out a formalization of the typology of attested syntax-prosodic mappings for ditransitive constructions in terms logical transductions (Courcelle, 1994; Courcelle and Engelfriet, 2012). In other linguistic domains, the rigor provided by computational/mathematical formalization has helped researchers commit to details of their theoretical assumptions, and fully understand the impact of particular representational choices. In line with this observation, this paper contributes to recent work laying the ground for mathematical investigations of the syntax-prosody interface (Yu, 2017, 2022, 2021; Dolatian et al., 2021a). These first steps already shed light on how a variety of theoretical details often unspecified in the literature need further clarification before extensive logical formalization of the syntax-prosody interface can be achieved.

The paper is organized as follows. Section 2 goes over the basic empirical typology of ditransitive prosody. Section 3 presents the formal preliminaries for the logical notation. Section 4 formally defines the

[1] Although a prosodic phrase is traditionally marked as ϕ, in what follows we will use p. We will instead use ϕ to indicate logical predicates.

bulk of syntactic information relevant for ditransitive prosody. Section 5 shows how such information can be used to formally define the mapping from syntax to prosody. We then discuss (§6) and conclude (§7).

2 Typology of ditransitive prosody

In prosodic phonology, syntactic constituents (e.g. XP's) are said to map onto prosodic constituents (e.g. prosodic phrases). These two types of constituents are often mis-aligned, meaning that an XP can be larger or smaller than its corresponding prosodic phrase. Unsurprisingly, different languages have different rules for how XPs are mapped. In this paper, we focus on a formal exploration of the prosody of ditransitive sentences in SVO languages, given that there is data available on their typology (Dobashi, 2003; Kalivoda, 2018).

2.1 What is prosodic structure

In a ditransitive sentence, the verb phrase includes two internal arguments: colloquially, the direct object and the indirect object. Cross-linguistically, ditransitive sentences can have different types of prosodic phrasings (Dobashi, 2003). In some SVO languages like English, a typical phrasing is to make the verb be in the same prosodic phrase p as the first object, while the second object is a separate prosodic phrase (Kalivoda 2018, 46 citing Selkirk (2000); examples are our own).

1. ($_p$ she gave a book) ($_p$ to Mary)
   ($_p$ she gave Mary) ($_p$ books)

Note that throughout the paper, we only focus on the mapping of syntactic constituents to prosodic constituents (= prosodic phrases). Within a given language, the edges of these prosodic constituents should be retrievable from the acoustic signal, such as via some language-specific phonological or phonetic rule that references these edges.

2.2 Types of ditransitive phrasings

For a language like English, ditransitive verb phrases are phrased as two separate prosodic phrases: (VN)(N). In a survey of work on ditransitive prosody, Kalivoda (2018, 38) finds that SVO languages can prosodically parse ditransitive phrases in one of four ways.[2] The names of the distinct 'prosodic types' we refer to throughout the paper are our own (see Table 1).

[2]For SOV languages like Korean, Kalivoda (2018) finds only one possible phrasing: (N)(NV). They acknowledge though that the SOV gaps may be accidental gaps that are due to the smaller number of studied SOV languages. We set aside SOV languages from our current formalization.

Table 1: Kalivoda (2018)'s typology of prosodic phrasing in ditransitives

Syntax	Prosodic Type	Phrasing	Language
SVO	separated	(V) (N) (N)	Ewe
SVO	closest-merged	(V N) (N)	Chimwiini
SVO	recursive	((V N) N)	Kimatuumbi
SVO	all-merged	(V N N)	Zulu

In a language like Ewe, the verb and two objects are each phrased separately: (V)(N)(N). In Chimwini, the verb and closest noun are phrased together, while the second object is phrased separately, like English: (VN)(N). In Kimatuumbi, the VOO sequence is phrased recursively: ((VN)N). In Zulu, all three items are phrased together: (VNN).

2.3 Syntactic structure of ditransitives

For the input syntactic structure of the verbal cluster that we want to map to the output prosodic structure, we follow (Kalivoda, 2018). As consistent with most modern generative work, we assume that a surface VOO sequence is made up of two VP-like layers (*VP shell*, Larson, 1988; Aoun and Li, 1989; Harley, 2002, a.o.). The lower VP layer consists of the two objects: the first object in spec-VP and the second object in the complement of VP. The verb undergoes head-movement from its base position within VP to adjoin to v in the higher layer. We illustrate this in Figure 1.

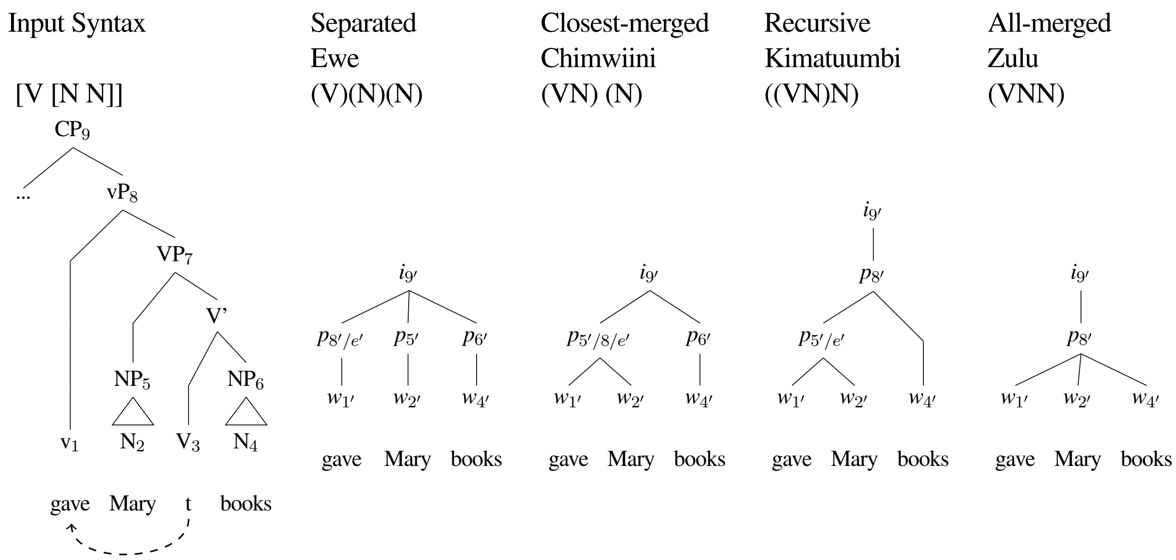
For illustration, assume that the subject is in a higher position in the clause (TP or CP). The CP is mapped to an intonational phrase, while intermediate functional levels are ignored (Dobashi, 2003). The intonational phrase dominates the prosodic phrases of the VP. We omit the subject's prosodic phrase because it is irrelevant to the issue of correctly mapping the verb + objects cluster into prosodic constituents.

2.4 Formal relationship between syntax and prosody

Given this set of relations between the input syntax and the output prosodic representation (Figure 1), different analyses can be given for the correspondence of individual syntactic phrases with specific prosodic phrases. Indexes on each tree in Figure 1 illustrate these possible associations. These indexes can be thought of as numeral shorthand for the Gorn addresses of nodes in the syntactic tree. For instance, the CP node at index 9 is mapped to the intonational phrase at index $9'$. Overt terminal nodes $(1,2,4)$ each get mapped to a prosodic word $(1',2',4')$.

Crucially, there is ambiguity in the literature about

Figure 1: Syntactic and prosodic structure of a ditransitive phrase in an SVO language



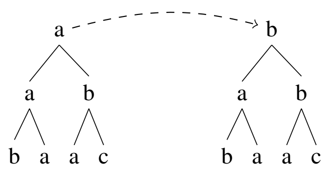
the exact input-output correspondences for prosodic phrases. In the Ewe (V)(N)(N) system, for example, the two noun phrases $(5,6)$ each get mapped to a prosodic phrase $(5',6')$. As for the verb, its surface prosodic phrase can be argued to either be a) epenthesized/inserted or created from no existing syntactic phrase (e'), or b) derived from the vP (index 12). In the latter case, the vP is phrased to a small prosodic phrase that excludes its arguments; such mismatches in the size of an XP and its prosodic phrase have been called underparsing or undermatch in the literature (Elfner, 2015; Guekguezian, 2017, 2021).

3 Logical Tree Transductions

In this section, we illustrate the use of Monadic Second Order (MSO) logic to define tree-to-tree transductions. MSO transductions are equivalent to regular functions (Filiot, 2015), and have been commonly employed to model both segmental and autosegmental phonological processes (Jardine, 2016; Chandlee and Jardine, 2019a; Strother-Garcia, 2018). For the current discussion, we assume familiarity with logic (boolean connectives, first-order quantification, etc.) and set notation on the reader's part.

With logical transductions, the input tree model is defined in terms of a signature $\langle D,R \rangle$. The segments are defined in terms of a set of domain elements D taken from the set of positive integers. For tree models, the common practice is to use Gorn-addresses. The domain elements satisfy a set of relations R which can be unary or binary. Unary relations designate the labels L of these domain elements, e.g. the label $\mathsf{V}(x)$ designates domain elements which are nodes labeled V (for verb). Domain elements are connected via

Figure 2: Example tree transduction

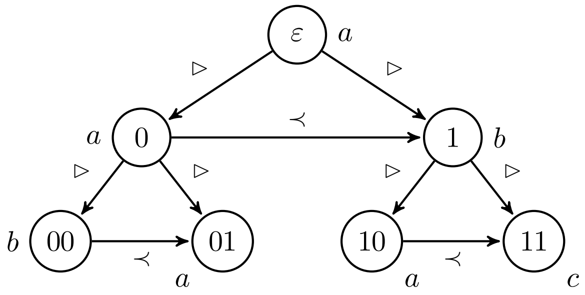


binary relations. Two binary relations are standardly considered to be relevant for trees, immediate dominance $\triangleleft(x,y)$ and left-of $\prec(x,y)$. In our current discussion, only immediate dominance will be used.

As a toy example, take a tree transduction that changes root nodes that are labeled *a* into root nodes that are labeled *b* (Figure 2). We first illustrate the logical definition of a tree for the input tree in this example transduction, with more extended illustration of each logical statement in Equation 2 in Figure 3. The model definition first establishes the domain of the structure, here using Gorn addresses. Each unary relation corresponds to labels and is the set of nodes for whom that label applies. For instance, the set for *a*(x) are the nodes which are labeled *a*: these are the nodes with Gorn addreses ε,0,1,00,01,10,11, as can also be seen in Figure 3. Each binary relation is a set of pairs for which the binary relation holds: Equation 2 thus states that the dominance relation $\triangleleft$ holds for nodes ϵ and 0, meaning that the node with addres ϵ dominates the node with address 0, and so on. Proper dominance ($\triangleleft^+$) is defined as the transitive closure of immediate dominance ($\triangleleft$).

2. Tree model for input tree in Figure 2
   Domain $D = \{\varepsilon,0,1,00,01,10,11\}$
   Unary relations $L \subset R$:

Figure 3: Illustration of the tree model for the input tree in Figure 2



- $\mathsf{a}(x)=\{\varepsilon,0,01,12\}$
- $\mathsf{b}(x)=\{1,00\}$
- $\mathsf{c}(x)=\{11\}$

Binary relations in R:

- $\triangleleft(x,y)=\{\langle\epsilon,0\rangle,\langle\epsilon,1\rangle,\langle 0,00\rangle,\langle 0,01\rangle,\langle 1,10\rangle,\langle 1,11\rangle\}$
- $\prec(x,y)=\{\langle 0,1\rangle,\langle 00,01\rangle,\langle 10,11\rangle\}$

In order to transform input trees into output trees, MSO logical transductions define a *copy set* C of some fixed size k. The k members of the copy set act as indexes for copies of the input. If the output structure needs less than or equal nodes as the input, then a copy set of size 1 is sufficient: $|C|=1$. If the output has a larger number of nodes than the input, then a larger copy set is needed.

Output functions define segments in the output copies in terms of the input segments. The apostrophe marks output elements. We mark these functions using ϕ`this font`. For example, to change a root node *a* to *b*, we need a transduction with a copy set of size 1, since the output tree has the same number of nodes as the input tree. In order to make the transduction easier to read, we define the root *a* segment with the predicate in (1) as a shorthand, using **this font**. Crucially, every pair of segments has the same dominance relation in the output as in the input (2). Nodes in the output are labeled *a* if they are labeled *a* in the input and they are not the root (3). The label *b* is generated for all underlying *b*'s and for underlying root *a*'s (4). Nodes labeled *c* in the input stay *c* in the output (5). We visualize an example of this transduction in Figure 2.

$$\mathbf{root\_a}(x) \overset{\text{def}}{=} \mathsf{a}\wedge\neg\exists y[\triangleleft(y,x)] \quad (1)$$

$$\triangleleft(x',y') \overset{\text{def}}{=} \triangleleft(x,y) \quad (2)$$

$$\phi\mathsf{a}(x') \overset{\text{def}}{=} \mathsf{a}(x)\wedge\neg\mathbf{root\_a}(x) \quad (3)$$

$$\phi\mathsf{b}(x') \overset{\text{def}}{=} \mathsf{b}(x)\vee\mathbf{root\_a}(x) \quad (4)$$

$$\phi\mathsf{c}(x') \overset{\text{def}}{=} \mathsf{c}(x) \quad (5)$$

For representational ease, in what follows we use simple integers like $\{1,2,3,...\}$ as numeral shorthands for Gorn addresses.

4 Formalizing core syntactic information

In ditransitives, prosodic phrasing is sensitive to some but not all aspects of the syntactic structure (Nespor and Vogel, 1986; Selkirk, 1986, 2011; Inkelas and Zec, pages; Truckenbrodt, 1995, 1999, 2007; Elfner, 2015; Bennett and Elfner, 2019). These aspects are overtness, headedness, tree geometry, arguments, and linearity. It ignores category labels.

In this section, we define predicates that pick out these aspects of syntactic structure. These predicates will be later used to define the logical mappings from syntax to prosody.

Note that existent prosodic mapping studies have not directly addressed adjunction, namely the nature of the prosodic mapping when an unbounded number of adjoining phrases are added to the sentence. Additionally, unbounded adjunction introduces non-locality between a head and its argument. Because of the lack of data and these non-trivial open issues related to adjunction, we set it aside in our preliminary formalization.

4.1 Overt material

Prosody works over overt or pronounced terminal items. Predicate $\mathbf{Trm}(x)$ defines terminal syntactic items (N, V, v). $\mathbf{oTrm}(x)$ defines the overt items (thus excluding the trace of the verb once it moves to v, assuming V-to-v movement in all cases).

$$\mathbf{Trm}(x) \overset{\text{def}}{=} \mathsf{N}(x)\vee\mathsf{V}(x)\vee\mathsf{v}(x) \quad (6)$$

$$\mathbf{oTrm}(x) \overset{\text{def}}{=} \mathsf{N}(x)\vee\mathsf{v}(x) \quad (7)$$

4.2 Headedness

For headedness, we assume that we can reconstruct which terminal node x is the head of a maximal projection y based on the local geometry of the tree (hence, on their indexes).[3]

$$\mathbf{mxPrj}(x) \overset{\text{def}}{=} \mathsf{NP}(x)\vee\mathsf{VP}(x)\vee \mathsf{vP}(x) \quad (8)$$

$$\mathbf{hdOf}(x,y) \;\; is \;\; \text{TRUE if } (x,y)\in \{(1,8),(2,5),(3,7),(4,6)\} \quad (9)$$

[3]Though it is possible to define a predicate $\mathbf{hdOf}$(x,y) with MSO logic, such definition requires an explicit list of the syntactic features on each lexical item, which is outside the scope of this paper. In lay terms, terminal node x is the head of the phrase represented by node y, if y is the result of the Merge operation that checks off the last selector feature on x during the derivation.

A maximal projection is then headed if it contains an overt head.

$$\mathbf{hdedPhr}(x) \overset{\text{def}}{=} \mathbf{mxPrj}(x) \wedge \exists y\, [\mathbf{hdOf}(y,x) \wedge \mathbf{oTrm}(y)] \quad (10)$$

$$\mathbf{unhdedPhr}(x) \overset{\text{def}}{=} \mathbf{mxPrj}(x) \wedge \exists y\, [\mathbf{hdOf}(y,x) \wedge \neg \mathbf{oTrm}(y)] \quad (11)$$

4.3 Tree geometry

For tree geometry, phrasing is sensitive to whether a pair of nodes x,y are structurally sisters, and arguably to c-command.

$$\mathbf{sisOf}(x,y) \overset{\text{def}}{=} x \neq y \wedge \forall z\, [z \triangleleft x \leftrightarrow z \triangleleft y] \quad (12)$$

$$\mathbf{ccom}(x,y) \overset{\text{def}}{=} x \neq y \wedge \forall z\, [\triangleleft^{+}(z,x) \rightarrow \triangleleft^{+}(z,y)] \quad (13)$$

4.4 Argument structure and head movement

For argument structure, we distinguish two types of configurations: with and without head-movement. Without head-movement, a maximal projection XP has at most two arguments: a complement and a specifier. Thus the VP_7 has the two noun phrases NP_5 and NP_6 as arguments. The head X of XP (the covert V_3) can then claim the arguments of its maximal projection.

$$\mathbf{cmpOf}(x,y) \overset{\text{def}}{=} \mathbf{mxPrj}(x) \wedge \mathbf{mxPrj}(y) \wedge \exists z [\mathbf{hdOf}(z,y) \wedge \mathbf{sisOf}(x,z)] \quad (14)$$

$$\mathbf{spcOf}(x,y) \overset{\text{def}}{=} \mathbf{mxPrj}(x) \wedge \mathbf{mxPrj}(y) \wedge y \triangleleft x \quad (15)$$

$$\mathbf{argOf}(x,y) \overset{\text{def}}{=} \exists z [(\mathbf{cmpOf}(x,z) \vee \mathbf{spcOf}(x,z)) \wedge \mathbf{hdOf}(y,z)] \quad (16)$$

The above predicates capture the fact that the covert V_3 has two arguments. However, this V is covert because its lexical item *gave* underwent head movement to v_1. Based on observations made in the prosodic literature Kalivoda (2018), we make the (syntactically anomalous) assumption that when some item undergoes head-movement, its final landing slot inherits the arguments of its base position. Thus the verb '*gave*' as v_1 inherits the arguments of the covert V_3.

For simplicity, we assume that the movement path of head movement is defined a priori in terms of indexes or Gorn addresses. V_3 is the base position, while v_1 is the target or landing position. This is not a problem given that the head-movement relations observed in the typology work we rely on are always local, but we will come back to this point in Section 6.

$$\mathbf{mvPth}(x,y) \;\; is \;\; \text{TRUE if } (x,y) = (1,3) \quad (17)$$

$$\mathbf{mvBase}(x) \overset{\text{def}}{=} \neg \mathbf{oTrm}(x) \quad (18)$$

$$\mathbf{mvLand}(x) \overset{\text{def}}{=} \exists (y) [\mathbf{mvBase}(y) \wedge \mathbf{mvPth}(x,y)] \quad (19)$$

Thus, the argument x of some terminal node y is either a) the direct argument of y, if y did not move, or b) the argument that y inherited via head-movement from a node z moved into y from its base position.

$$\mathbf{genArg}(x,y) \overset{\text{def}}{=} \mathbf{argOf}(x,y) \vee [\mathbf{mvLand}(y) \wedge \exists z\, (\mathbf{mvPth}(y,z) \wedge \mathbf{mvBase}(z) \wedge \mathbf{argOf}(x,z))] \quad (20)$$

4.5 Linearity

The final syntactic property that prosody is sensitive to is linearity. In a ditransitive phrase, the verb can be phrased with its closest argument. We define 'closeness' in terms of c-command. We assume that if a node underwent head movement, then it c-commands all its arguments from its landing position.[4] Using c-command, we can define the first and second argument of a ditransitive verb.

$$\mathbf{arg1}(x,y) \overset{\text{def}}{=} \mathbf{genArg}(x,y) \wedge \mathbf{ccom}(y,x) \wedge \neg \exists z\, [\mathbf{ccom}(y,z) \wedge \mathbf{ccom}(z,x) \wedge \mathbf{genArg}(z,y)] \quad (21)$$

$$\mathbf{arg2}(x,y) \overset{\text{def}}{=} \mathbf{genArg}(x,y) \wedge \neg \mathbf{arg1}(x,y) \quad (22)$$

4.6 Avoiding category labels

As observed during our earlier discussion of the prosodic typology of ditransitives, in the SVO languages under analysis, vPs and NPs behave differently with respect to what kind of nodes they are mapped into in the output prosodic trees. However, syntax-prosody mappings are generally taken to be blind to category labels (except for CP). Thus, the prosody should not be able to distinguish between vPs and

[4] We define the first argument of a head as the the one that follows the head after linearization. That is, the first argument of the verb head is the direct object, not the subject.

NPs based on the labels of their heads, but possibly only in terms of argument structure and linearity.

While from a modern syntactic perspective it is debatable that the verbal and nominal domain actually differ in terms of the geometry of their argument structure, the examples reported by Kalivoda (2018) are of NPs without arguments. We thus do not know how more complex NPs (e.g. NPs with a complement prepositional phrase) would be mapped into prosodic constituents. Given the preliminary nature of our formalization attempt and our reliance on existing work on prosodic parsing, in what follows we define predicates that pick out headed phrases that have arguments (the vP) and headed phrases that lack arguments (NPs).

$$\mathbf{hasArg}(x) \overset{\text{def}}{=} \exists y[\mathbf{genArg}(y,x)] \quad (23)$$

$$\mathbf{hdedWArg}(x) \overset{\text{def}}{=} \mathbf{hdedPhr}(x) \wedge \mathbf{hasArg}(x) \quad (24)$$

$$\mathbf{hdedWoArg}(x) \overset{\text{def}}{=} \mathbf{hdedPhr}(x) \wedge \neg\mathbf{hasArg}(x) \quad (25)$$

5 Logical transductions for the syntax-to-prosofy typology

With all the preliminary predicates in place, in this section we define tree-to-tree logical transductions for each type of prosodic mapping laid out in Section 2 As discussed before, for each case there are multiple possible choices for the exact node-to-node maps. For reasons of space, here we only showcase predicates for one option per language, and focus on highlighting the necessary formal mechanisms that arise due to differences in the typology of the mappings.

5.1 Commonalities

Some node-to-node relations are common across all the typological examples. In particular, the iP node is mapped from the CP node at index 9.

$$\phi\texttt{iP}(x') \overset{\text{def}}{=} \mathsf{CP}(x) \quad (26)$$

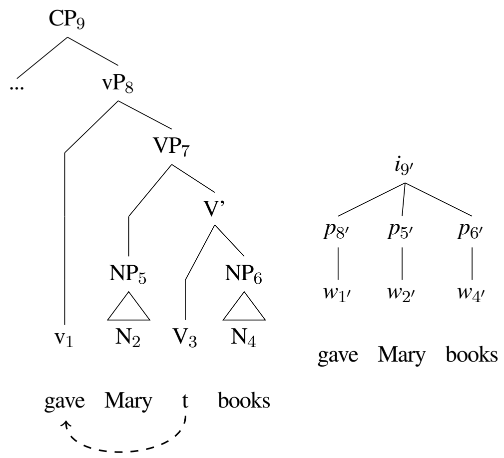
Additionally, all the overt terminal items (N and V) map to prosodic words (PW).

$$\phi\texttt{PW}(x') \overset{\text{def}}{=} \mathbf{oTrm}(x) \quad (27)$$

5.2 Ewe: (V)(N)(N)

For Ewe-type languages, the NPs each map to a prosodic phrase. The V is also part of a separate prosodic phrase. Let us assume that the V is phrased in a prosodic phrase $\text{PPh}_{8'}$, mapped from the $v\text{P}_8$.

Figure 4: Structure of Ewe: (V)(N)(N)



Thus, each overtly headed phrase (vP and NP) is mapped to a prosodic phrase.

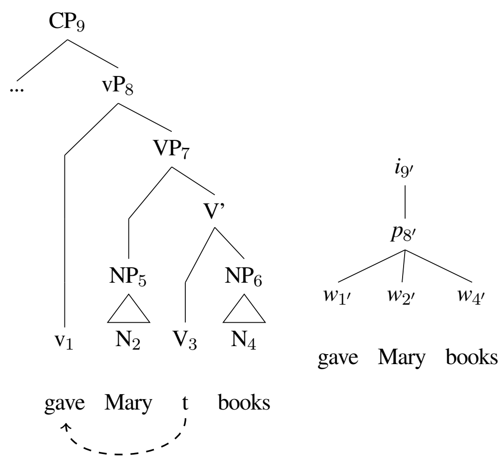
$$\phi\texttt{PPh}(x') \overset{\text{def}}{=} \mathbf{hdedPhr}(x) \quad (28)$$

In terms of dominance relations, each PPh (p, mapped from an overt headed phrase) dominates its overt head (mapped into a w). The iP then dominates every p.

$$\phi\lhd(x',y') \overset{\text{def}}{=} [\phi\texttt{PW}(x')\wedge\lhd(y,x)]\vee [\phi\texttt{PPh}(x')\wedge\mathbf{hdOf}(y,x)]\vee [\phi\texttt{iP}(x')\wedge\phi\texttt{PPh}(y')] \quad (29)$$

5.3 Zulu: (VNN)

Figure 5: Structure of Zulu (VNN)



For Zulu-type languages, only one prosodic phrase is created. Assume this phrase is mapped from the vP at index 8. The vP is the only headed phrase that has arguments. Only this XP gets its own PPh.

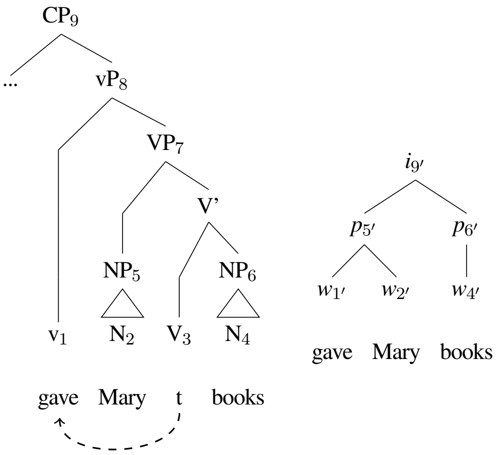
$$\phi\texttt{PPh}(x') \overset{\text{def}}{=} \mathbf{hdedWArg}(x) \quad (30)$$

In terms of dominance, the sole PPh dominates every PWord.

$$\phi\lhd(x',y') \stackrel{\text{def}}{=} [\phi\texttt{PW}(x')\wedge\lhd(y,x)]\vee [\phi\texttt{PPh}(x')\wedge\phi\texttt{PW}(y')]\vee [\phi\texttt{iP}(x')\wedge\phi\texttt{PPh}(y')] \quad (31)$$

5.4 Chimwiini: (VN)(N)

Figure 6: Structure of Chimwiini (VN) (N)



For the Chimwiini system, there is an ambiguity in the syntactic origins of the first PPh. This PPh can map either from the vP, the first NP, or be epenthetic. To make it easier to contrast this system with the one for Kimatuumbi (in the following section), we here only illustrate how this PPh can be mapped from the NP.

In this system, the two PPhrases originate from NPs, thus from XPs that have overt heads but no arguments.

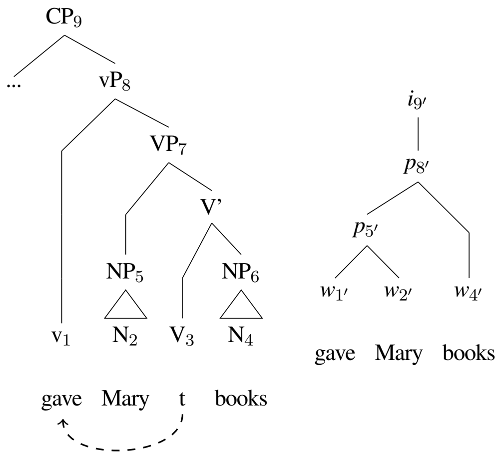
$$\phi\texttt{PPh}(x') \stackrel{\text{def}}{=} \mathbf{hdedWoArg}(x) \quad (32)$$

In terms of prosodic dominance: PPhrases dominate the PWords that are the heads of the PPhrase's XP (second disjunct). Additionally (third disjunct), the PPhrase of the first NP (the first argument) dominates the PW of the vP (the argument-taking XP).

$$\phi\lhd(x',y') \stackrel{\text{def}}{=} [\phi\texttt{PW}(x')\wedge\lhd(y,x)]\vee [\phi\texttt{PPh}(x')\wedge\mathbf{hdOf}(y,x)]\vee \exists z[\mathbf{hdedWArg}(z)\wedge \mathbf{hdOf}(y,z)\wedge\mathbf{arg1}(x,z)]\vee [\phi\texttt{iP}(x')\wedge\phi\texttt{PPh}(y')] \quad (33)$$

5.5 Kimatuumbi: ((VN)N)

Figure 7: Structure of Kimatuumbi ((VN)N)



In the Kimatuumbi system, we need to allow for prosodic recursion. The highest PPhrase is mapped from the vP. The bottom PPhrase must be mapped either from the first NP or be epenthetic. We assume it is mapped from the first NP: the first argument of the headed phrase.

$$\phi\texttt{PPh}(x') \stackrel{\text{def}}{=} \mathbf{hdedWArg}(x)\vee\exists y [\mathbf{hdedWArg}(y)\wedge\mathbf{arg1}(x,y)] \quad (34)$$

Even in this bounded context, the use of recursion requires more convoluted contexts for prosodic dominance. The bottom PPhrase is mapped from the NP, the PPhrase dominates the head of the vP (second disjunct) and the head of the first NP (third disjunct). The top PPh is mapped from vP: it dominates the lower PPhrase and the head of the second argument (fourth disjunct).

$$\phi\lhd(x',y') \stackrel{\text{def}}{=} [\phi\texttt{PW}(x')\wedge\lhd(y,x)]\vee [\exists z[\mathbf{hdedWArg}(z)\wedge \mathbf{hdOf}(y,z)]\vee [\exists z[\mathbf{hdedWArg}(z)\wedge \mathbf{arg1}(x,z)\wedge\mathbf{hdOf}(y,x)]\vee [\exists z[\mathbf{hdedWArg}(x)\wedge \mathbf{arg2}(z,x)\wedge\mathbf{hdOf}(y,z)]\vee [\phi\texttt{iP}(x')\wedge\phi\texttt{PPh}(y')] \quad (35)$$

The logical formulation of prosodic dominance relations in this system would likely be more straightforward if we defined *both* of the two surface prosodic phrases as mapped from the same vP. This would require one-to-many associations for prosodic mappings, such that an input XP can correspond to two output PPhrases — however, such one-to-many associations are usually avoided in prosodic theory (Ito and Mester, 2019).

6 Discussion

In this paper, we used logical tree transductions to characterize mappings between syntactic and prosodic structure in ditransitive constructions. Based on the cross-linguistic typology of prosodic mappings reported in Kalivoda (2018), we showed that logical transductions seem appropriate to derive the alignment mismatches between syntactic and prosodic constituents. In doing so, we highlighted how details of prosodic and syntactic structures often left unspecified in the linguistic literature become fundamental in deciding the linguistic naturalness of such mappings. These results then provide a baseline for future, extensive formalization of syntax-prosody mismatches and open the way for a vast array of computationally informed questions and computationally-driven empirical predictions.

6.1 Head-movement and locality

In this paper we relied on Gorn addresses (node indexes) to handle the discontinuity created by head movement of V into v. While seemingly ad-hoc, this move was justified by the assumption that the observed head-movement dependency is — in the examples provided in the prosodic literature — always bounded within the vP domain. Hence, the information relevant to that a particular syntax-prosodic relation could be deterministically inferred from the geometry of the trees, and Gorn addresses were just a convenient shorthand. Theoretically, if we adopt a fully explicit syntactic formalism (e.g. Minimalist Grammars, Stabler, 1996), then it should be possible to extend our predicates to account for unbounded head-movement paths explicitly, for example by relying on feature chains (Kobele et al., 2007; Graf, 2012).

However, the open linguistic question is whether we can find cases where unbounded head-movement of the verb is relevant for prosodic structure, and what exactly would the resulting prosodic constituents be. Similarly, it is unclear whether the approach we adopted for the "recursive" structure in Kimatuumbi would work as straightforwardly for additional levels of embedding. Potential issues related to unbounded prosodic recursion that are not tied to local contexts have been pointed out by other work on prosodic transductions (Yu, 2021; Dolatian et al., 2021a).

6.2 Category Blindness

Throughout the paper, we had to make assumptions about properties of the syntactic/prosodic representations based on what had been observed/assumed in the existing literature on prosodic constituency. Among these, a non-trivial issue was the hypothesis that prosody is blind to category information — and thus, that mappings can only rely on tree geometry. For instance, based on this hypothesis we defined mappings that differentiated vPs from NPs based on the number of arguments they have in the trees. This allowed us to be faithful to the observation that, in the examples studied by Kalivoda (2018), vPs and NPs behaved strikingly differently with respect to prosodic mappings. Crucially though, such examples only reported bare NPs without complements nor specifiers — and it is thus possible that what we are observing is a prosodic sensitivity to syntactic phrases with and without complements.

Additionally, modern linguistic theory tends to assume that the verbal and nominal domain are similar in terms of domain-internal syntactic relations, and we would not predict a difference in behavior with respect to systems that are blind to category information. We can thus ask whether "*category-blindness*" is actually a real property of prosodic mappings, or whether it is just an epiphenomenon arising from the particular type of observations collected in the literature. If category blindness is indeed a core property, and if syntax-prosody mappings are tied to tree geometry, we would predict that complex nominal domains (e.g. NPs with prepositional complements) should be parsed the same way as vP.

6.3 Broad complexity considerations

From a formal perspective, this paper looks at the computational requirements of prosodic transductions via logical transductions (cf. logical formalizations in Dolatian, 2020). Following a rich tradition in model-theoretic syntax and phonology, we started out with the intent of using MSO to express the syntax-prosody relations. However, if we go back and look at the predicates we defined, we will note that we only make use of quantification to scope over individual variables. Thus, our mappings are essentially just first-order logic predicates. In this respect, recent work on phonological transformations has shown that they can be handled with Quantifier-Free string transductions (Chandlee and Lindell, in in review; Strother-Garcia, 2019; Chandlee and Jardine, 2019b), and in the future it would be interesting to see if our mappings could be further refined to work in terms of Quantifier-Free tree transductions (Ikawa et al., 2020; Dolatian, 2020).

Similarly, it is important to note that while logical transductions allow us to focus on the global properties of the representations we cast our mappings onto, existing computational work on prosody has made use of tree transducers (in particular, multi-bottom up tree transducers, Dolatian et al., 2021a; Yu, 2022). Multi-bottom up tree transducers have been shown to be relevant to syntactic processes (specifically involving copying, Kobele et al., 2007) and their computational properties are relatively well-understood. Moreover, tree transducers can be incorporated within a variety of parsing algorithms, and therefore offer a way to more deeply integrate prosodic and syntactic parsing (Yu and Stabler, 2017; Graf and De Santo, 2019; Yu, 2019). On the other side, the specification of tree transducers is more focused on the procedural requirements of the transformations and might, for instance, put stricter constraints on the relation between constituent rewriting and unboundedness (Yu, 2021).

7 Conclusion

This paper offers a contribution to the scarce existing literature on the formal characterization of prosodic processes, and their relation to syntactic representations. While much work remains to be done, our results further show how careful mathematical formalization can help up refine long-standing theoretical questions, suggest the need for more and different types of data, and make us more critical of theoretical assumptions about linguistic representations across subdomains.

References

Joseph Aoun and Yen-hui Audrey Li. 1989. Scope and constituency. *Linguistic inquiry*, 20(2):141–172.

Ryan Bennett and Emily Elfner. 2019. The syntax-prosody interface. *Annual Review of Linguistics*, 5:151–171.

Jane Chandlee. 2014. *Strictly Local Phonological Processes*. Ph.D. thesis, University of Delaware, Newark, DE.

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

Jane Chandlee and Adam Jardine. 2019b. Quantifier-free least fixed point functions for phonology. In *Proceedings of the 16th Meeting on the Mathematics of Language*, pages 50–62, Toronto, Canada. Association for Computational Linguistics.

Jane Chandlee and Steven Lindell. in in review. A logical characterization of input strictly local functions. In Hossep Dolatian, Jeffrey Heinz, and Kristina Strother-Garcia, editors, *Doing Computational Phonology*. Oxford University Press, Oxford.

Bruno Courcelle. 1994. Monadic second-order definable graph transductions: A survey. *Theoretical Computer Science*, 126(1):53–75.

Bruno Courcelle and Joost Engelfriet. 2012. *Graph Structure and Monadic Second-Order Logic, a Language Theoretic Approach*. Cambridge University Press, Cambridge.

Yoshihito Dobashi. 2003. *Phonological phrasing and syntactic derivation*. Ph.D. thesis, Cornell University.

Hossep Dolatian. 2020. *Computational locality of cyclic phonology in Armenian*. Ph.D. thesis, Stony Brook University.

Hossep Dolatian, Aniello De Santo, and Thomas Graf. 2021a. Recursive prosody is not finite-state. In *Proceedings of the 18th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 11–22, Online. Association for Computational Linguistics.

Hossep Dolatian, Jonathan Rawski, and Jeffrey Heinz. 2021b. Strong generative capacity of morphological processes. *Proceedings of the Society for Computation in Linguistics*, 4(1):228–243.

Emily Elfner. 2015. Recursion in prosodic phrasing: Evidence from Connemara Irish. *Natural Language & Linguistic Theory*, 33(4):1169–1208.

Emmanuel Filiot. 2015. Logic-automata connections for transformations. In *Logic and Its Applications*, pages 30–57, Berlin, Heidelberg. Springer Berlin Heidelberg.

Thomas Graf. 2012. Movement-generalized minimalist grammars. In *International Conference on Logical Aspects of Computational Linguistics*, pages 58–73. Springer.

Thomas Graf and Aniello De Santo. 2019. Sensing tree automata as a model of syntactic dependencies. In *Proceedings of the 16th Meeting on the Mathematics of Language*, pages 12–26, Toronto, Canada. Association for Computational Linguistics.

Peter Ara Guekguezian. 2017. *Prosodic recursion and syntactic cyclicity inside the word*. Ph.D. thesis, University of Southern California.

Peter Ara Guekguezian. 2021. Morphosyntax–phonology mismatches in Muskogee. *Phonology*, 38(2):277–316.

Heidi Harley. 2002. Possession and the double object construction. *Linguistic variation yearbook*, 2(1):31–70.

Jeffrey Heinz. 2018. The computational nature of phonological generalizations. In Larry Hyman and Frans Plank, editors, *Phonological Typology*, Phonetics and Phonology, chapter 5, pages 126–195. Mouton de Gruyter, Berlin.

Mans Hulden. 2009. *Finite-state machine construction methods and algorithms for phonology and morphology*. Ph.D. thesis, University of Arizona.

Shiori Ikawa, Akane Ohtaka, and Adam Jardine. 2020. Quantifier-free tree transductions. In *Proceedings of the Society for Computation in Linguistics*, volume 3, pages 145–153.

Sharon Inkelas and Draga Zec. pages. The phonology-syntax interface. In John Goldsmith, editor, *The Handbook of Phonological Theory*, 1 edition, pages 535–549. Blackwell Publishers, Cambridge, MA.

Junko Ito and Armin Mester. 2019. Match as syntax-prosody MAX/DEP: Prosodic enclisis in English. *English linguistics*, 36(1).

Adam Jardine. 2016. *Locality and non-linear representations in tonal phonology*. Ph.D. thesis, University of Delaware, Newark, DE.

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

Ronald M Kaplan and Martin Kay. 1994. Regular models of phonological rule systems. *Computational linguistics*, 20(3):331–378.

Gregory M. Kobele, Christian Retoré, and Sylvain Salvati. 2007. An automata-theoretic approach to minimalism. *Model theoretic syntax at 10*, pages 71–80.

Richard K Larson. 1988. On the double object construction. *Linguistic inquiry*, 19(3):335–391.

Marina Nespor and Irene Vogel. 1986. *Prosodic phonology*. Foris, Dordrecht.

Janet Breckenridge Pierrehumbert. 1980. *The phonology and phonetics of English intonation*. Ph.D. thesis, Massachusetts Institute of Technology.

Christopher Potts and Geoffrey K Pullum. 2002. Model theory and the content of OT constraints. *Phonology*, 19(3):361–393.

Emmanuel Roche and Yves Schabes, editors. 1997. *Finite-state language processing*. MIT press, Cambridge.

Elisabeth Selkirk. 1982. *The syntax of words*. Number 7 in Linguistic Inquiry Monographs. MIT Press, Cambridge, Mass.

Elisabeth Selkirk. 1986. On derived domains in sentence phonology. *Phonology Yearbook*, 3(1):371–405.

Elisabeth Selkirk. 2000. The interaction of constraints on prosodic phrasing. In Merle Horne, editor, *Prosody: Theory and experiment*, pages 231–261. Springer.

Elisabeth Selkirk. 2011. The syntax-phonology interface. In John Goldsmith, Jason Riggle, and Alan C. L. Yu, editors, *The Handbook of Phonological Theory*, 2 edition, pages 435–483. Blackwell, Oxford.

Edward Stabler. 1996. Derivational minimalism. In *International Conference on Logical Aspects of Computational Linguistics*, pages 68–95. Springer.

Kristina Strother-Garcia. 2018. Imdlawn Tashlhiyt Berber syllabification is quantifier-free. In *Proceedings of the Society for Computation in Linguistics*, volume 1, pages 145–153.

Kristina Strother-Garcia. 2019. *Using model theory in phonology: a novel characterization of syllable structure and syllabification*. Ph.D. thesis, University of Delaware.

Hubert Truckenbrodt. 1995. *Phonological phrases–their relation to syntax, focus, and prominance*. Ph.D. thesis, Massachusetts Institute of Technology.

Hubert Truckenbrodt. 1999. On the relation between syntactic phrases and phonological phrases. *Linguistic Inquiry*, 30(2):219–255.

Hubert Truckenbrodt. 2007. The syntax-phonology interface. In Paul de Lacy, editor, *The Cambridge Handbook of Phonology*, page 435–456. Cambridge University Press, Cambridge.

Kristine M Yu. 2017. Advantages of constituency: Computational perspectives on Samoan word prosody. In *International Conference on Formal Grammar 2017*, pages 105–124, Berlin. Spring.

Kristine M Yu. 2019. Parsing with minimalist grammars and prosodic trees. In Robert C. Berwick and Edward P. Stabler, editors, *Minimalist Parsing*, pages 69–109. Oxford University Press, London.

Kristine M Yu. 2021. Computational perspectives on phonological constituency and recursion. *Catalan journal of linguistics*, 20:77–114.

Kristine M Yu. 2022. Representing multiple dependencies in prosodic structures. *Proceedings of the Society for Computation in Linguistics*, 5(1):171–183.

Kristine M Yu and Edward P Stabler. 2017. (in) variability in the Samoan syntax/prosody interface and consequences for syntactic parsing. *Laboratory Phonology: Journal of the Association for Laboratory Phonology*, 8(1):1–44.