4

Parsing with Minimalist Grammars and prosodic trees

Kristine M. Yu

4.1 Introduction

Advances in the syntactic parsing of written language have proceeded apace in the last few decades, but much less progress has been made in the syntactic parsing of *spoken* language. This chapter addresses one question important for such progress: *how can we bring prosody in to inform syntactic parsing of spoken language*? It has long been thought that prosody ought to be informative in syntactic parsing, for the simple reason that syntax determines certain aspects of prosody (Chomsky 1955: II-2fn). Yet how to bring prosody in to inform syntactic parsing has remained unclear. Why is this?

First, it is not enough to know that syntax conditions certain aspects of prosody; we must also understand how this conditioning works in a clear enough way to characterize and implement the interface between syntax and prosody. Here the computational parsing literature is almost silent: the norm is to refrain from stating how syntactic structure conditions prosody, e.g. "We avoid the pitfall . . . of specifying an explicit relationship between prosody and syntax; rather, the model infers the relationships from the data" (Dreyer and Shafran 2007). In contrast, the linguistic literature has enjoyed an abundance of theories. Over a decade ago, Harper et al. (2005: 58) wrote on parsing: "The lack of authoritative theories on syntax-phonology interface makes it difficult to develop a generative model for utilizing prosodic structure . . . ".

Proposals about the syntax–phonology interface have since continued to proliferate (see e.g. Elordieta 2008; Selkirk 2011; Féry 2017 for reviews). And while there are some theories that are treated as authorative, it is not obvious that the details of some proposal might be much closer to the truth than those of another. As such, a first step is to model general properties that a number of syntax–phonology interface proposals share as a class (see §4.1.2). One such general property is that linguists model prosodic structures with trees, while computational (syntactic) parsing models have not. If we admit prosodic trees as representations, then we must also consider how prosodic trees might be parsed. The question becomes not only how prosody can inform syntactic parsing, but also *how syntax can inform prosodic parsing*.

Another obstacle to the pursuit of prosodically informed syntactic parsing is that it is not just syntax that conditions prosody, of course—a multitude of interacting factors

Minimalist Parsing. First edition. Robert C. Berwick and Edward P. Stabler (eds.) This chapter © Kristine M. Yu 2019. First published 2019 by Oxford University Press.

conditions the appearance and realization of prosodic events in spoken language, beyond just syntax (see e.g. Ladd 2008; Yu 2014: 777). These factors range from linguistic ones like phonology (e.g. the rising pitch accent associated with predictable primary stress in Egyptian Arabic (Hellmuth 2009; 2006)) and pragmatics (e.g. the English contrastive topic rise–fall–rise contour (e.g Jackendoff 1972; Büring 2003; Constant 2014)); to socio-affective factors such as modulating interlocutor attention (Stern et al. 1982; Thorson and Morgan 2014); to individual differences in cognition (e.g. Clifton Jr. et al. (2002); Ferreira and Karimi 2015; Speer and Foltz 2015). The sensitivity of prosody to these diverse factors can obscure the informativity of prosody for syntactic analysis. As a first step around this, we can focus on modeling syntax–prosody interface phenomena in natural language where non-syntactic factors seem to play a minimal role, if any.

In computational work on parsing that has used prosodic information, the emphasis has been on prosodic edges coming in as (noisy) signals for sentences and syntactic constituent edges (see Shriberg et al. 2000; Kahn et al. 2005; Huang and Harper 2010, among many others). But relations between syntactic and prosodic domains are not the whole of the syntax–phonology interface. As pointed out by Selkirk (2011: 435), this is only one aspect of the interface; two "further core aspects" are "the phonological realization (spell-out) of the morphosyntactic feature bundles of morphemes and lexical items that form part of syntactic representation and the linearization of syntactic representation which produces the surface word order of the sentence as actually pronounced" (Selkirk 2011: 435). Prosodically informed computational parsing work has missed studying the distinct contributions of these other core aspects to conditioning prosody. And it may be precisely these other core aspects that provide interface phenomena where syntax plays (virtually) the only role in conditioning the prosody.

In summary, we've highlighted two[1] key challenges for understanding how to bring prosody in to inform syntactic parsing: (i) explicitly defining aspects of the syntax–prosody interface, including prosodic trees, and (ii) modeling distinct contributions of syntax to conditioning prosodic events, including aspects other than relations between

---

[1] There's another related and daunting challenge to introducing prosody into parsing that we'll abstract away from in this chapter. This is our poor understanding of the mapping from the speech signal to prosodic structure (see e.g. see Cole and Shattuck-Hufnagel 2016). Ultimately, to take advantage of prosodic information, we need to know how to recognize it. One aspect of this mapping that has received some attention in the computational parsing literature is the acoustics of prosodic concepts: "a challenge of using prosodic features is that multiple acoustic cues interact to signal prosodic structure, including pauses, duration lengthening, fundamental frequency modulation, and even spectral shape" (Tran et al. 2017). But another aspect of this mapping that has received scant attention is the definition of the range of the mapping: what are the atoms of prosodic structure, anyway; what should we be mapping from the speech signal to? The parsing literature has sometimes assumed that human expert-annotated intonational/break labels have ground truth status (e.g. Steedman 1991a; Kahn et al. 2005; Hale et al. 2006; Dreyer and Shafran 2007; Huang and Harper 2010). However, the particular intonational transcription system for English (MAE_ToBI) that has been used in the parsing literature, as well as intonational transcription more generally, has been under active scrutiny and development to the current day (see e.g. Gussenhoven 2016; Hualde and Prieto 2016, and other articles in a special issue of *Laboratory Phonology* on "Advancing Prosodic Transcription," D'Imperio et al. 2016). One interesting avenue for tackling the prosodic mapping has been pursued by Tran et al. (2017); this work used neural networks to learn word-level feature vectors from fundamental frequency and energy time-series, rather than using pre-determined hand-annotated prosodic category labels.

syntactic and prosodic domain edges. This chapter takes on these tasks. We present a proof-of-concept, detailed case study of parsing a single sentence in Samoan, a Polynesian language, and ask: *How might prosodic information come in to help in syntactic parsing? How might syntactic information help in prosodic parsing?* Our contribution to tackling the first challenge is to implement a model of the syntax–prosody interface following key properties of current theoretical proposals on the syntax–prosody interface. Unlike other work on prosodically informed syntactic parsing, we accordingly assume the existence of prosodic trees separate from syntactic trees, as well as an optimality-theoretic grammar with constraints penalizing misalignment of syntactic and prosodic constituents. Our implementation allows us to study the consequences of current theoretical proposals on the interface for parsing. Our contribution to grappling with the second challenge is to model both prosodic reflexes (spellout) of morphosyntactic structures and relations between prosodic and syntactic domains in the parser. In particular, we show how studying prosodically informed parsing in a language like Samoan—with clearly syntactically controlled prosodic events in addition to more variably occurring prosodic events—can be a fruitful first step to understanding the diverse ways that prosody (syntax) might inform syntactic (prosodic) parsing more generally. The spotlight on Samoan puts our work in contrast with previous work, which has been done almost entirely on English, with also a little work on German, French, and Mandarin. These are languages in which the relation between syntax and prosody is quite obscured by interactions with non-syntactic factors, and in which the different ways in which prosody can inform syntactic analysis are difficult to articulate and factor out.

The rest of this chapter is structured as follows; the remainder of this introductory section (§4.1) consists of background information on: previous work on prosodically informed syntactic parsing (§4.1.1), theoretical proposals about the syntax–prosody interface (§4.1.2), and Samoan syntax and prosody (§4.2). The introductory section is followed by a section that defines the syntactic grammar fragment (§4.3.2), a section that describes the generation of licit prosodic parses (§4.3.1), and a section that describes the implementation of the syntax–prosody interface (§§4.3.3,4.3.4). We close with a discussion and conclusion (§4.4).

4.1.1 Previous work on prosodically informed syntactic parsers

A representative selection of computational work on prosodically informed syntactic parsing is summarized in Table 4.1. This summary shows that much work has used broad-coverage probabilistic context-free grammars for syntactic analysis and been based on a single corpus of spoken language (the English Switchboard corpus). In contrast—and closer to the work presented here—the older case studies of Steedman (1991a), Blache (1995), and Batliner et al. (1996) targeted particular syntactic phenomena using hand-crafted Combinatory Categorial Grammar (CCG) and Head-driven Phrase Structure Grammar (HPSG) fragments. One challenge introduced

Table 4.1 Summary of a representative sample of relatively recent work on prosodically informed syntactic parsing

	Batliner et al. (1996)	**Gregory (2004)**	**Kahn et al. (2005)**	**Hale (2006)**	**Dreyer and Shafran (2007)**	**Huang and Harper (2010)**	**Pate (2013)**	**Tran et al. (2017)**
Language	German	English	English	English	English	English	English	English
Data/corpus	Verbmobil	Switchboard	Switchboard	Switchboard, Fisher	Switchboard	Switchboard, Fisher	WSJ, Brent, Switchboard	Switchboard-NXT
Special topic	V2 verbal traces	None	Disfluencies	Speech repair	None	None	Infant-directed speech	Disambiguation
Syntactic grammar/parser	HPSG hand-crafted BU parser	PCFG broad-coverage	PCFG N-best	PCFG CYK N-best, N = 1	PCFG N-best, N = 1	PCFG, PCFG-LA N-best	Unsupervised Bayesian dependency parser	Linearized PCFG trees represented as word embeddings, LSTM-RNNs
Acoustic features	Duration, f0 regression coeff	Energy, f0, f0' pause/phone duration	Mentioned but not given	F0, energy, duration	F0, energy, duration	Used for automatic break detection	Word duration	Word-level features. Pauses, duration; f0, energy input into CNN
Phonological features	None	Quantization of extreme values in acoustic distributions	Posterior probabilities of ToBI break indices	Decision trees for ToBI break indices	Decision trees for ToBI break indices	Previously detected ToBI break indices	Classification by word durations	None (output of CNN)
Syntactic features	presence of syntactic boundary	None	Non-local dependencies	"-UNF" unfinished tags daughter annotation	Category splits, EM	Category splits into latent tags	Word identity, POS tags, direction of dependency arc	Word embeddings
Interface	Acoustic features directly to syntactic boundaries	PCFG enriched with prosodic features	Weighted co-occurrence with syntactic features for re-ranking parses	Enriched PCFG	Prosodic breaks used in category refinement	PCFG enriched with break indices	Dependency learning conditioned on quantized duration	Prosodic and word embeddings in bag of features
Effect of prosody	Reduces runtime	Degrades performance	Improves rank order and F-score	Better accuracy in finding disfluencies	Better parse performance	Lack of performance gains unless further restrictions	Better constituency scores, dependency accuracy	Improves parse accuracy

by spoken language that has been the focus of some attention is parsing in the presence of disfluencies and speech repairs (e.g. Charniak and Johnson 2001; Spilker et al. 2001; Harper et al. 2005; Kahn et al. 2005; Hale et al. 2006; Lin and Lee 2009; Miller 2009; Wu et al. 2011; Zwarts and Johnson 2011); less work has focused on fluent speech. A number of parsers have included acoustic features such as fundamental frequency, energy, and duration measures to train classifiers for ToBI break index categories, especially a category for disfluencies. Tones and Break Indices (ToBI) is a widely used set of conventions for the intonational transcription of English (Pierrehumbert 1980; Beckman and Pierrehumbert 1986; Beckman and Elam 1997; Wightman et al. 1992). Also, prosody has generally entered into parsing either via a syntactic grammar enriched with prosodic tags, or in a bag of features used in (re-)ranking generated parses. A main result of the work has been to show that prosodic features such as changes in fundamental frequency (the acoustic correlate of pitch) and human-annotated prosodic break strength indices can sometimes be informative in the detection of points of disfluency, and introduce modest gains in parsing accuracy and efficiently in their presence.

4.1.2 The syntax–prosody interface

We've briefly reviewed the body of computational work on prosodically informed syntactic parsing and seen that prosody has entered into parsing either via a syntactic grammar enriched with prosodic tags or in a bag of features used in (re-)ranking generated parses. This work does not assume that prosody might have its own structure—independent from syntax—which itself may need to be parsed in the course of syntactic parsing. Yet this is exactly what has long been assumed about prosody in the linguistic literature (Selkirk 1978/1981; Pierrehumbert and Beckman 1988; Beckman 1996; Féry 2017), e.g. "Thinking now in terms of speech production and perception, we would hypothesize that the units for prosodic structure we have discussed here in linguistic terms are indeed the appropriate units in production and perception models, that the effect of syntactic phrasing in production, or access to that phrasing in perception, are crucially mediated by these units of the prosodic hierarchy" (Selkirk 1978/1981). In this chapter, we assume that this is one of the ways that prosody comes into parsing: via prosodic trees that are independent structures from syntactic trees.

As the idea of trees as data structures in (prosodic) phonology is quite alien to the parsing literature,[2] we spend some time introducing it here. Two common assumptions about prosodic structure have endured in linguistics since its early formulations in the mid-1970s: (i) prosodic structure is *hierarchical*, and (ii) while prosodic structure reflects syntactic structure in systematic ways, prosodic and syntactic structure are distinct and independent. Below, we first explicate what is meant by 'hierarchical'

---

[2] And strikingly, mathematical/computational descriptions of phonological patterns have revealed strong structural universals without referring to constituents at all; see Heinz (2018) for a overview and review.

structure (§4.1.2.1), and then we sketch the working model of the syntax–prosody interface that we assume in our implementation here (§4.1.2.2).

4.1.2.1 "Hierarchical" structure in prosody

One of the earliest invocations of the term "prosodic structure" in theories of the syntax–phonology interface came in Selkirk (1978/1981), which defined prosodic structure as "a suprasegmental, hierarchically arranged organization to the utterance," in contrast to the "simple linear arrangement of segments and boundaries" assumed as the phonological data structure in Chomsky and Halle (1968)'s *The Sound Pattern of English* (SPE). The assumption (or definition) that prosodic structure is "hierarchical" has persisted (e.g. Selkirk 1984; Nespor and Vogel 1986; Selkirk 1986; Pierrehumbert and Beckman 1988; Ladd 1996; Ito and Mester 2003; Jun 2005a; Ladd 2014; Selkirk and Lee 2015; Cheng and Downing 2016; Féry 2017. But what does "hierarchical" mean, and how does that property distinguish a data structure from the data structures in SPE? If "hierarchical" is an informal way of referring to recursive data structures—data structures which can be defined in terms of themselves—then the "linear" segment sequences of SPE, i.e. strings, are also hierarchical, since strings are composed of substrings. What is really meant by "hierarchical" in this context is that prosodic data structures are ordered, rooted trees rather than strings, following Liberman (1975a: 49–51).[3] *And unlike strings, trees are data structures that can pick out substring chunks—in particular, a set of nodes that are exhaustively dominated by a common node in the tree forms a constituent.*

The motivation for introducing trees in prosody has been the same as that in syntax: positing constituents has been one alternative amongst the arsenal of theoretical machinery that has helped phonologists capture generalizations in the observed patterns of natural language, (see e.g. Nespor and Vogel 1986: 58–9). Phonological analyses suggest that constellations of phonological processes (Selkirk 1978/1981; McCarthy and Prince 1986/1996; Nespor and Vogel 1986; Pierrehumbert and Beckman 1988; Hayes 1995; Jun 1996; 1998; Selkirk 2011; Myrberg and Riad 2015), as well as phonotactic restrictions (Flack 2007) and syntagmatic prominence relations (Liberman 1975a; Liberman and Prince 1977), consistently target or refer to particular chunks of phonological material; it is these chunks that have then been posited to be categories in the derivation of prosodic trees (Selkirk 1978/1981; Nespor and Vogel 1986).[4]

---

[3] Other phonological data structures besides trees have been considered to be "hierarchical" or "non-linear" in contrast to the strings of SPE (McCarthy 1982; Hayes 1988). Most notably, these include the tiered representations of autosegmental theory (Goldsmith 1976; 1990) and the grids and bracketed grids of metrical theory (Liberman 1975a; Liberman and Prince 1977; Hammond 1984; Halle and Vergnaud 1987). Here, we set aside the question of how these different structures relate to one another and whether different structures are relevant for different aspects of prosody (see e.g Liberman and Prince 1977; Selkirk 1984 §4.2; Pierrehumbert and Beckman 1988: ch. 6), and focus on trees—the kind of data structure most actively discussed in recent work on the syntax–prosody interface.

[4] Alternate analyses have also been proposed that do not assume prosodic constituents, (Kiparsky 1983; Kenstowicz 1995). See also a comparison of alternative analyses for Samoan word-level prosody in Zuraw et al. (2014).

While the motivation for introducing trees as data structures in prosody and syntax is shared, prosodic trees have been thought to be different from syntactic trees: "the reason for assuming the existence of an independent prosodic structure . . . is that the constituent structure with respect to which structure-sensitive phenomena of phonology and phonetics are defined may diverge in significant respects from the syntactic structure of the sentence" (Selkirk and Lee 2015: 5). A classic example of this divergence or (bracketing) "mismatch" introduced in Chomsky and Halle (1968: 372) is given in (1).[5] The prosodic tree has only one level of embedding—it's flat—while the syntactic tree exhibits five levels of embedding, as well as recursion §.

(1) Classic example of mismatch between syntactic and prosodic domains (Chomsky and Halle 1968: 372)
  a. Syntactic constituency: This is $[_{NP}$ the cat $[_{S'}$ that caught $[_{NP}$ the rat $[_{S'}$ that stole $[_{NP}$ the cheese$]]]]]$
  b. Prosodic constituency: (This is the cat) (that caught the rat) (that stole the cheese)

Much work on the syntax–phonology interface in the 1980s and 1990s centered on finding evidence that: (i) there are "mismatches" between syntactic and prosodic domains, and (ii) that there exist phonological processes that are defined—and can *only* be defined/understood—with respect to prosodic rather than syntactic domains (Nespor and Vogel 1986; Selkirk 1986; Hayes 1989; Price et al. 1991; Jun 1996; 1998; Shattuck-Hufnagel and Turk 1996; Truckenbrodt 1999). To fit with the accumulating evidence of mismatches, interface theories assumed that prosodic constituents were aligned or related to syntactic constituents only at *one* edge, see Féry (2017: §§4.2–4.4) for a brief review. At the same time, some work defended the claim that there were some phonological processes that could be better understood in terms of syntactic domains (e.g. Kaisse 1985; Odden 1987). And some work pointed out that so-called "mismatches" might not in fact be mismatches; rather, apparent mismatches are actually cases where the prosodic structure reveals alternate syntactic structures, and syntax is much more flexible than we have thought (e.g. Liberman 1975a; 1975b; Steedman 1991b; Taglicht 1994; 1998; Wagner 2005; 2010; Steedman 2014; Hirsch and Wagner 2015). A strong version of this hypothesis has been articulated by Wagner (2010: 231–2):

> The original motivation in favor of edge-alignment came from certain apparent bracketing mismatches. However, a closer look at the syntax in representative examples suggests the prosody does not mismatch syntax after all. If this conclusion is correct, then we can take prosodic evidence seriously as a source of syntactic evidence. In cases where syntactic and prosodic evidence seem in contradiction, we may have to rethink our syntactic analysis. (Wagner 2010: 231–2)

---

[5] But see Wagner (2010: 224–6) for a discussion of whether this is really an example of a mismatch.

A weaker version of this hypothesis has become a kind of consensus view in at least some communities of theoretical work on the syntax–prosody interface in recent years: the default is that there is a grammatical correspondence relation between syntactic and prosodic domains—not that there are mismatches. If mismatches occur, they are "the consequence of properly phonological pressures on the hierarchical structure of phonological representation," i.e. what phonologists call "phonological markedness" (Selkirk and Lee 2015: 4), see §4.1.2.4. Another aspect of the consensus view seems to be that constraint-based grammars are well suited to define this correspondence. In such grammars, a mismatch can occur, e.g. if a phonological markedness constraint is ranked higher than a constraint demanding the faithful correspondence of prosodic domains to syntactic domains. What has remained a continuing source of debate is which syntactic objects are relevant for the correspondence relation (or relevant for phonological processes, under theories where phonological processes directly refer to syntactic objects). While some work has defined these objects to be syntactic constituents, other work has defined them to be phases (e.g. Dobashi 2004; Kratzer and Selkirk 2007; Downing 2010; Cheng and Downing 2016; Ahn 2016; McPherson and Heath 2016). However, the various phase-based theories that have been proposed have quite disparate assumptions about what phases are, so we set them aside for this first case study, and assume syntactic constituents to be the relevant object for the correspondence relation for now.

4.1.2.2 A working model for the relation between prosodic and syntactic trees

The definition and implementation of the syntax–prosody interface in this chapter is based on work in the Match Theory framework (Selkirk 2009; 2011), which uses (violable) MATCH constraints to enforce correspondence between syntactic and prosodic constituents. In recent years, Match Theory has been widely used to analyze a range of interface phenomena across different languages; see e.g. the special issue of *Phonology* on constituency in sentence phonology (Selkirk and Lee 2015), Ito and Mester (2013) on Japanese, Myrberg (2013) on Swedish, Kandybowicz (2015) on Asante Twi, and Ito and Mester (2015) on Danish, as well as work on verb-initial languages like Samoan: Elfner (2012; 2015); Bennett et al. (2016) on Irish, Clemens (2014) on Niuean, Sabbagh (2014) on Tagalog.

The statement of syntax–prosody MATCH constraints is intended to encode the core assumption of Match Theory that there is "a strong tendency for phonological domains to mirror syntactic constituents" (Selkirk 2011). The other core assumption about the interface treated in Match Theory is that syntax–prosody mismatches are due to the satisfaction of phonological markedness constraints at the expense of violating MATCH constraints. Thus, while Match Theory is one among many theories of the syntax–prosody interface, it is commonly used by linguists and exhibits the properties we described in §4.1.2.1 as characteristic of a kind of consensus view of the interface at the present.[6] Match Theory is therefore a suitable choice for the definition of the

---

[6] For comparative overviews that describe other theories, too, see Elordieta (2008), Selkirk (2011), and Féry (2017: ch. 4).

interface in this first case study. In the remainder of this section, we introduce MATCH constraints in §4.1.2.3 and phonological markedness constraints relevant to our case study in §4.1.2.4.

4.1.2.3 Match Theory: interface constraints

Suppose we are given a finite, ordered set of categories in a prosodic grammar such as (2) (Selkirk, 2011: (1)), and that prosodic trees are derived using these categories.

(2) Enumeration of prosodic categories
  a. Intonational phrase (ι)
  b. Phonological phrase (ϕ)
  c. Prosodic word (ω)
  d. Foot (Ft)
  e. Syllable (σ)

Given such an enumeration from highest to lowest in a "prosodic hierarchy," Match Theory assumes the existence of syntactic and prosodic trees for an utterance and a set of optimality-theoretic faithfulness constraints (Prince and Smolensky 1993; 2004) that enforce "matching" between the constituents in these trees, as stated in (3), quoted from Bennett et al. (2016). The interface constraints are defined as relations over syntax–prosodic tree pairs, and each of them is defined in such a way that it is multiply violated if there are multiple loci in the prosodic tree that don't satisfy the constraint.

(3) Definition of syntax–prosody MATCH constraints (Bennett et al. 2016: 187, (34))
  a. MATCHWORD: Prosodic words correspond to the heads from which phrases are projected in the syntax (heads that will often have a complex internal structure determined by head movement).
  b. MATCHPHRASE: Phonological phrases correspond to maximal projections in the syntax.
  c. MATCHCLAUSE: Intonational phrases correspond to those clausal projections that have the potential to express illocutionary force (assertoric or interrogative force, for instance).

More precisely, a prosodic constituent is defined as "corresponding" to a syntactic constituent when both the left and right edges of the prosodic constituent are aligned to the left and right edges of the syntactic constituent, respectively (Selkirk 2011: §2.2.1). For example, Selkirk (2011: (20)) defines MATCHPHRASE as in (4). Another set of prosody–syntax MATCH constraints enforce the correspondence of syntactic constituents to prosodic constituents, under the same edge-based definition of correspondence. We abstract away from these prosody–syntax constraints in our case study; one could say we consider them too low-ranked to be active in filtering out prosodic parses.

(4) Definition of MATCHPHRASE (MATCH(XP, ϕ)): The left and right edges of a constituent of type XP in the input syntactic representation must correspond to the left and right edges of a constituent of type ϕ in the output phonological representation.

Elfner (2012) developed a revised definition of MatchPhrase motivated by generalizations she discovered in her analysis of the syntax–prosody interface in Irish. This is given in (5) and is the one we use in our implementation.

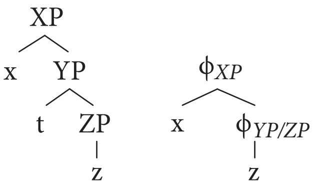
(5) Definition of MatchPhrase (Bennett et al. 2016: 188, (35); Elfner 2012: 28, (19)):
Given a maximal projection XP in a syntactic representation *S*, where XP dominates all and only the set of terminal elements $\{a, b, c, \dots, n\}$, there must be in the phonological representation *P* corresponding to *S* a ϕ-phrase that includes all and only the phonological exponents of $a, b, c, \dots, n$.

Assign one violation mark if there is no ϕ-phrase in the phonological representation that exhaustively dominates all and only the phonological exponents of the terminal nodes in $\{a, b, c, \dots, n\}$.

The effect of the statement about phonological exponents in (5) is to redefine MatchPhrase over syntactic–prosodic tree pairs where the syntactic tree has been "flattened." The result of "flattening" is just the tree structure that is visible to the phonological constraints. By definition, phonological exponents must be phonologically overt, and the prosodic tree has only phonologically overt terminals. The definition in (5) says that the MatchPhrase relation for a tree pair $\langle S, P\rangle$ is computed over a flattened syntactic tree S' transduced from *S*, where any phonologically empty terminals are deleted, and then any two syntactic nodes in *S* are merged if they both exhaustively dominate the same set of remaining terminals.

Consider the discussion in Elfner (2012: 32–3, (24)), revisited here: suppose we are evaluating MatchPhrase (5) for the syntactic–prosodic tree pair $\langle S, P\rangle$ given in (6), where t is a trace or unpronounced copy left by a movement operation.

(6) Syntactic–prosodic tree pair

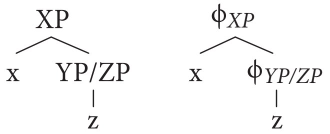


In this example, there are three maximal projections in the syntactic tree, which dominate all and only the terminals given in (7).

(7) Terminals dominated by maximal projections in the syntactic tree in (6)
  a. XP: $\{x, t, z\}$
  b. YP: $\{t, z\}$
  c. ZP: $\{z\}$

But since *t* is phonologically null, it does not enter into the MatchPhrase relation. Thus, we can think of MatchPhrase as operating on a flattened syntactic tree where *t* is deleted and YP and ZP are merged, as shown in (8). YP and ZP are merged because they exhaustively dominate the same set of phonological (non-null) exponents, $\{z\}$.

(8) Syntactic–prosodic tree pair, with flattened syntactic tree

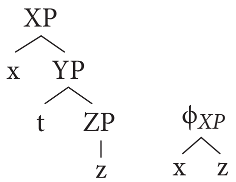


The syntactic–prosodic tree pair in (6,8) incurs no violation of MATCHPHRASE—either definition, (4) or (5). For each unique set of terminals exhaustively dominated by a node in the syntactic tree, there is a node in the prosodic tree that exhaustively dominates the same set, as shown in (9).

(9) Terminals dominated by maximal projections and phonological phrases in the tree pair in (8)
a. XP: $\{x, z\} \mapsto \phi_{XP}$: $\{x, z\}$
b. YP/ZP: $\{z\} \mapsto \phi_{YP/ZP}$: $\{z\}$

Now consider a different syntactic–prosodic tree pair, given in (10). This tree pair has the same syntactic tree as (6), and thus the same set of terminals dominated by each syntactic maximal projetion, (7).

(10) Syntactic–prosodic tree pair (Elfner 2012: 32, (24))



However, this tree pair incurs one violation of MATCHPHRASE as defined in (5), since no ϕ exhaustively dominates the set of terminals exhaustively dominated by YP/ZP: $\{z\}$. (Under the definition of MATCHPHRASE in (4), it incurs two violations, one for YP and one for ZP.)

4.1.2.4 Match Theory: constraints on prosodic trees

Besides the assumption of a default correspondence between syntactic and prosodic constituents, the other core assumption about the interface treated in Match Theory is that while prosodic structure reflects syntactic structure in systematic ways, prosodic and syntactic structure are distinct and independent. The way Match Theory encodes this assumption is by positing the existence of phonological "markedness" constraints that penalize "marked" properties of prosodic structure. If these outrank MATCH constraints, then an optimal syntactic–prosodic tree pair under the constraint-based interface grammar might be one that incurs violations of the MATCH constraints. Thus, the phonological markedness constraints constitute the part of the interface grammar that drives prosodic trees to diverge in structure from syntactic ones. Depending on the relative rankings of the prosodic markedness constraints and the MATCH constraints, from language to language, prosodic trees may not diverge from syntactic trees at all, or might diverge in many ways. There is no consensus on what the "right" set of phonological markedness constraints is, but there are some constraints that are widely used and we define these here.

One class of these constraints regulates dominance relations and is defined following Selkirk (1996: 189–90) in (11).[7]

(11) Definition of constraints on prosodic domination (Selkirk 1996: 189–90) (where C^n = some prosodic category)
   a. Layeredness: No C^i dominates a C^j, $j > i$, e.g., No σ dominates a Ft.
   b. Headedness: Any C^i must dominate a C^{i-1} (except if $C_i = \sigma$), e.g., A PWd must dominate a Ft.
   c. Exhaustivity: No C^i immediately dominates a constituent C^j, $j < i - 1$, e.g., No PWd immediately dominates a σ.
   d. Nonrecursivity: No C^i dominates C^j, $j = i$, e.g., No Ft dominates a Ft.

As summarized in Selkirk (1996: 190), a significant body of work has indicated cases where it appears that Exhaustivity and Nonrecursivity might be violated in prosodic forms, but there isn't a similar body of work pointing out cases where Layeredness and Headedness appear to be violated (although see Seidl 2000). In the work here, we will assume that Layeredness and Headedness are inviolable. (This same assumption seems to be made in Bellik et al. (2015)'s Syntax–Prosody for OTWorkplace (SPOT), software for "automatic candidate generation and violation assessement for investigations on the syntax-prosody interface". Their current web interface includes options to add Exhaustivity and Nonrecursivity to the constraint set, but not Layeredness or Headedness.)

Let us consider for a moment how the constraints on prosodic trees in (11) compare to constraints on syntactic trees. Examples of constraints similar to Layeredness, Headedness, and Exhaustivity in syntax include the restrictive rule schema of X-bar theory (Chomsky 1970: 210–11) and the universal, fixed order of categories proposed in cartographic approaches (Cinque 1999). However, it is difficult to think of a constraint on syntactic trees that has been proposed like Nonrecursivity. Indeed, Selkirk (2011) states that an important motivation for proposing Match constraints is to account for the preponderance of phonological analyses that conclude that there is recursivity in prosodic trees: the idea is that recursion in prosodic trees is a result of faithfulness to recursion in syntactic trees. §4.3.1 discusses further details about recursion in prosodic trees and its implementation in this work. §4.4 also considers prosodic hierarchies that have been proposed in the literature which do not admit recursivity (see e.g. Shattuck-Hufnagel and Turk 1996: 206, fig. 2).

Besides the constraints defined in (11), there are two more classes of prosodic markedness constraints that we implement in this work: binarity constraints (12)

---

[7] This set of constraints was developed from an older idea termed the "Strict Layer Hypothesis", following work suggesting that that hypothesis should be decomposed (Inkelas 1989; Ito and Mester 1992; 2003). The Strict Layer Hypothesis is: "a category of level i in the hierarchy immediately dominates a (sequence of) categories of level $i - 1$ (Selkirk 1981a). (Assuming *syllable* to be level 1, the others will be levels 2, . . . ,n.) We will call this the *strict layer hypothesis*" (Selkirk 1984: 26).

and constraints computed over sisters (13). A binarity constraint can be decomposed into two separate ones, as in Elfner (2012: 153, (4), and refs. within) (12): BIN-MIN enforces that a prosodic constituent be minimally binary, while BIN-MAX enforces that a prosodic constituent be maximally binary.

(12) Definition of constraints on binary branching (Elfner 2012: 153, (4), and references therein)
   a. BIN-MIN(κ): assign one violation mark for every prosodic constituent of type κ that immediately dominates less than two daughter constituents.
   b. BIN-MAX(κ): assign one violation mark for every prosodic constituent of type κ that immediately dominates more than two daughter constituents.

Given that a binary branching constraint is a widely held hypothesis for syntactic trees (Kayne 1984: ch. 7), MATCH constraints favor binary branching in prosodic trees, too. But a unary branch in a syntactic tree might not survive transduction to the prosodic tree if BIN-MIN constraints are ranked above MATCH constraints. In the work here, we'll assume a single BINARITY(κ) constraint that is a conjunction of BIN-MIN(κ) and BIN-MAX(κ).

A constraint computed over sisters[8] that we implement here is STRONGSTART (Selkirk 2011: 470, (38)), see (13).[9] The constraint is motivated by the typological tendency for "strengthening" to occur at the left edges of prosodic domains. Prosodically weak elements get strengthened at the left edge. Strong, stressed forms of function words occur here, e.g. pronouns that are otherwise appear as unstressed clitics appear in their strong, stressed form in initial position (Halpern and Zwicky 1996); weak pronouns are postposed rightward (Elfner 2012; Bennett et al. 2016).

(13) Definition of STRONGSTART (Elfner 2012: 157, (11)), also (Selkirk 2011; 470, (38))
   Assign one violation mark for every prosodic constituent whose leftmost daughter constituent is lower in the Prosodic Hierarchy than its sister constituent immediately to its right: $*(\kappa_n\ \kappa_{n+1}\ \ldots$.

The phonological pressure of STRONGSTART has been used to motivate "bracketing paradoxes," e.g. where a prosodically weak element cliticizes to the left (right), although it syntactically phrases to the right (left) (see e.g. Kiparsky 1983; Pesetsky 1985; Sproat 1985; Himmelmann 2014). For example, Bennett et al. (2016: 200, (62)) describes one repair for weak pronouns in contemporary Irish at the left edge of a ϕ-phrase as "Option B: Leave the pronoun in its syntactically expected position, but cliticize it to a preceding word or phrase, thereby removing it from the left edge of the ϕ-phrase

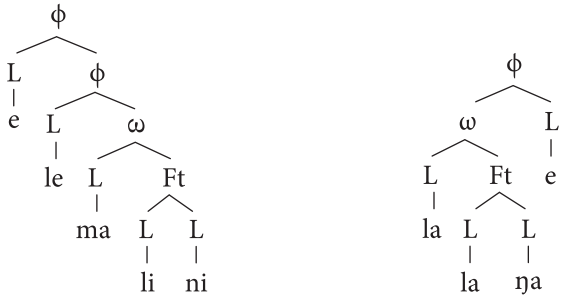
---

[8] Constraints computed over sisters are a fundamental part of syntactic theory via head–complement relations, e.g. between participles and auxiliaries.

[9] This definition makes the computation over sisters clear; computation over sisters is not needed by Bennett et al.'s (2016: 198, (55) definition (2016: 198, (55)): "Prosodic constituents above the level of the word should not have at their left edge an immediate subconstituent that is prosodically dependent. For our purposes here, a 'prosodically dependent' constituent is any prosodic unit smaller than the word", although they point out that STRONGSTART can be thought of as a special case of EQUALSISTERS (Myrberg 2013: 75, (4)): "Sister nodes in prosodic structure are instantiations of the same prosodic category."

and avoiding a violation of Strong Start." An instance of such a repair for Samoan is shown in (14) for the prosodic parse of the ergative case marker *e*. Here, the parse of *e le malini* 'ERG DET.SG marine' in (14a) violates StrongStart. The case marker *e*,which we assume is an unstressed light syllable (L), is phrased *rightward*. It is at the left edge of a ϕ-phrase and sister to a ϕ-phrase—a prosodic constituent higher than a syllable in the prosodic hierarchy (2). The repaired structure (14b) instead phrases the case marker *leftward* to be dominated by the preceding ϕ and sister to a ω (*lalaŋa* 'weave'). This results in a 'bracketing paradox' since *e* is syntactically phrased to the right, as the head of the DP_{erg} *e le malini* 'the marine'.

(14) A repair of a StrongStart violation by prosodically phrasing a weak element to the left



a. ϕ-initial ergative case marker *e*: StrongStart violated

b. ϕ-final ergative case marker *e*: StrongStart satisfied

Summing up, in this section, we have introduced the idea of prosodic trees and the particular theory of the interface between syntactic and prosodic trees that we will adopt for our implementation here: Match Theory. Defining the interface means defining a map from syntactic to prosodic phonological grammar, so of course the definition of the mapping depends on the definition of syntactic grammar and the definition of prosodic phonological grammar. Our choice of implementing Match Theory doesn't commit us to some particular syntactic theory; here we follow Elfner (2012), Bennett et al. (2016), Féry 2017) in assuming aspects of bare phrase structure (Chomsky 1955), implemented with Minimalist Grammar (MG) (Stabler 1997, 2011b). As Match Theory has been stated in terms of optimality-theoretic constraint-based grammar, here we also assume such a grammar for the phonological grammar, as well as the interface so that we can interleave phonological and interface constraints. We implement the constraint-based phonological grammar using the finite state tools of `xfst` (Beesley and Karttunen 2003; Karttunen 1998).

4.2 Samoan syntax, spellout, prosody, and interfaces: background

Having introduced the theory of the interface assumed, in this section we introduce the particular linguistic aspects of the syntax–prosody interface in Samoan to be formalized and implemented. §4.2.1 introduces aspects of Samoan syntax and spellout relevant for the case study, based on Yu and Stabler (2017); §4.2.2 discusses current, tentative empirical evidence for aspects of Samoan prosodic constituency. §4.2.3 defines the syntax–prosody interface to be modeled for the parsing problem. Throughout, we discuss consequences of the linguistic analyses at hand for parsing.

4.2.1 Samoan syntax and spellout

Samoan is a Polynesian language with an ergative/absolutive case system (see Deal 2015 for an overview of ergativity). The two sentences in (15) show how this case system manifests: the subject of a transitive clause, e.g. *le malini* 'the marine' in (15a), is marked with a distinct case—the "ergative." The subject of an intransitive clause, e.g. *le malini* in (15b), and the object of a transitive clause, e.g. *le mamanu* 'the design' in (15a), both appear unmarked and receive "absolutive" case (Chung 1978: 54–6; Ochs 1982: 649). Samoan primarily has VSO word order in transitive clauses, as exemplified in (15a), which also shows that the transitive subject is marked by the ergative case marker *e*. The intransitive clause (15b) demonstrates that the prepositional element [i] is a marker of oblique case.

(15) Ergative-absolutive patterns in transitive and intransitive clauses[10]

a. Transitive clause

na	lalaŋa	*(e)	le	malini	le	mamanu.
PAST	weave	ERG	DET.SG	marine	DET.SG	design

'The marine wove the design.'

b. Intransitive clause

na	ŋalue	le	malini	(i	le	mamanu).
PAST	work	DET.SG	marine	OBL	DET.SG	design

'The marine worked (on the design).'

Throughout this chapter, we use "absolutive" as an descriptive term. Under the analysis of Samoan syntax we assume—Collins (2016; 2015; 2014), following Legate (2008)—"absolutive" is in fact a default, syncretic marking of nominative and accusative case. While Massam (2001) and others have assumed that Samoan has absolutive case marking, Collins (2014) argues that Samoan is actually a language of the type Legate (2008) classifies as ABS=DEF, that is, a language where the marking that has been called "absolutive" is actually the default case marking for nominative and accusative.[11] While Collins and others originally assumed the default case marking in Samoan was null, Yu (2011; to appear) and Yu and Stabler (2017) showed that Samoan reliably presents a high edge tone (notated as H-) in these positions, immediately

---

[10] The following abbreviations are used in morphosyntactic glosses in this chapter: ABS absolutive; DET determiner; ERG ergative; GEN genitive; OBL oblique; SG singular; TOP topic marker.

[11] We follow Collins' analysis here because it is relatively well worked out and defended, but there are various alternative views about case in Samoan and related languages (e.g Chung 1978; Bittner and Hale 1996; Massam 2006; 2012; Koopman 2012; Tollan 2015). We leave consideration of how adopting these alternative perspectives might affect the implementation of the interface and parsing to future work.

preceding the absolutive argument, as shown in (16). The specific challenge we take on here is: *How might we simultaneously parse the syntactic and prosodic structure of (16a), and how might the two different parsing tasks inform each other?* (We assume that (16a) is uttered out-of-the-blue, so that no elements are under contrastive focus, and all elements are new information to the discourse.)

(16) Revision of (15): a high edge tone (H-) precedes absolutive arguments

a. Transitive clause
na lalaŋa *(e) le malini **H-** le mamanu.
PAST weave ERG DET.SG marine ABS DET.SG design
'The marine wove the design.'

b. Intransitive clause
na ŋalue **H-** le malini (i le mamanu).
PAST work ABS DET.SG marine OBL DET.SG design
'The marine worked (on the design).'

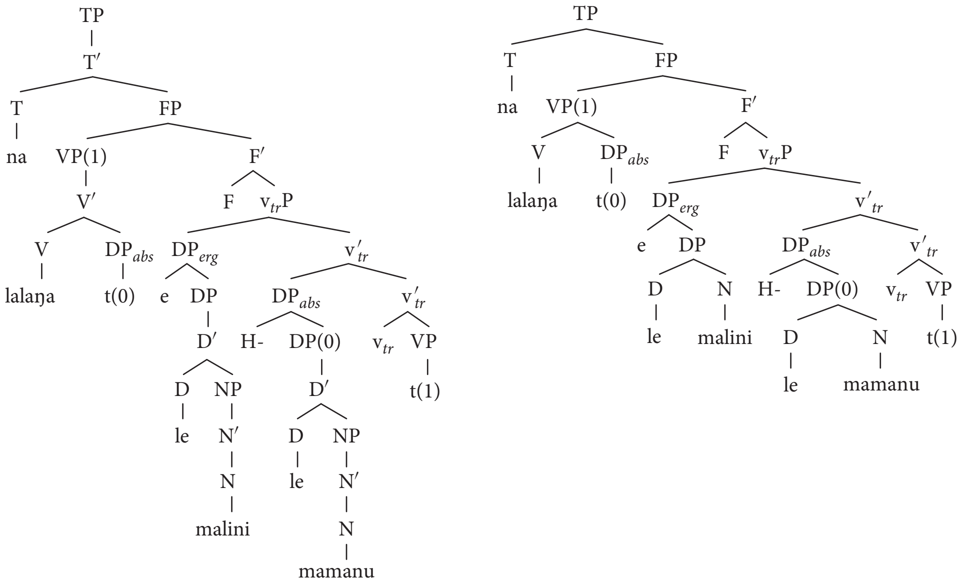
Yu and Stabler (2017) formalizes case marking in Samoan (whether ergative or absolutive) as a 'post-syntactic' operation (Marantz 1991; Bobaljik 2008). Under this proposal, ergative *e* and absolutive H- are inserted as pronounced reflexes of particular structural configurations. For example, DPs taken as an argument by a transitive verb are marked with ergative *e*.

We assume that (16a) has the derived post-syntactic structure (after case marking) given in (17). The tree in (17a) assumes X-bar theory (as in Yu and Stabler 2017: (11)). The tree in (17b) assumes aspects of Bare Phrase Structure (Chomsky 1995a), which is what we assume for implementation, as discussed in §4.1.2.2. We describe (17b) as "X-bar like" because it depicts aspects of both the derivational steps and the results of these derivational steps in one tree, like X-bar trees; cf. the derivational and derived trees depicted in (36), which separate the depiction of the derivational steps and the results of those steps. The structure given in (17) is almost the same structure as Yu and Stabler (2017: (11)); the difference is that the structure here abstracts away from head movement. In Yu and Stabler (2017: (11)) and Collins (2016, (66)), head movement moves T *na* to C. Here, we abstract away from head movement and focus on syntax–prosody interface issues, so the root of the tree is of category TP, not CP.[12] Following Collins (2016: (66)), verb-initial ordering is derived by fronting the VP to a functional head F below T after the arguments have been raised out of it. Phrasal movements are shown coindexed. The case markers inserted in spellout are shown as as adjoined to their arguments.

---

[12] We can handle head movement as well, as described in Yu and Stabler (2017: appendix B).

(17)


a. X-bar tree structure for (16a)

b. "X-bar like" bare phrase tree structure for (16a)

Yu and Stabler (2017: §7.1) shows that a range of syntactic structures in Samoan including those in (15)—prior to "post-syntactic" case marking—are finite-state definable on trees and can be computed with MGs. This result guarantees a range of provably correct and efficient parsing algorithms for these structures (Harkema 2000; 2001; Stabler 2013b; Fowlie and Koller 2017)—including the bottom-up MG parser we implement here. Furthermore, Yu and Stabler (2017: §7.2, appendix B) shows that the post-syntactic computation of case-marker insertion is also finite-state definable on trees and can be composed with the syntactic computations, following closure properties of MGs. Thus, the syntactic derivation and the spellout can be correctly and efficiently parsed.

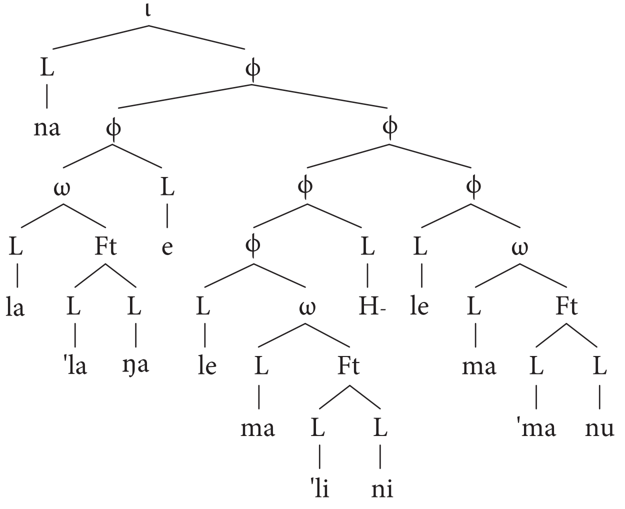
Because the case marking in spellout is composable with the syntactic derivation of (16a), we can define a MG grammar for (16a) that "folds in" the spellout with the syntax, see §4.3.2. So this is the first place that prosody comes in to inform parsing: the H- is the spellout of absolutive case. Accordingly, we treat the H- not as a local cue for a prosodic domain edge, but as a (post-syntactic) lexical item in the "folded in" syntactic/post-syntactic MG grammar. Yu and Stabler (2017: fn. 28) notes that whether case marking is treated as "post-syntactic" reflex of case marking or syntactic, e.g. as a (syntactic) lexical item, is not in fact important for the point that a range of syntactic structures in Samoan and case marking can be computed with MGs and therefore correctly and efficiently parsed. We'll see that the assumption that case marking happens in spellout *does* have important and interesting implications for the computation of MatchPhrase in this case study: see §4.2.3.

4.2.2 Samoan prosodic constituency

Besides the tonal spellout of case, the other phonological component of our implementation of the interface is a transducer that generates a candidate set of derived prosodic trees for the sentence in (16a) that then gets passed to the MG parser for the computation of the interface. We assume that the prosodic tree for a typical production of utterance (16a) is something like the structure shown in (18): this is the prosodic tree that we would like our implementation of the interface to compute as being optimal, given the syntactic/post-syntactic analysis (17). The node labels in (18) follow the naming conventions given in (2), with the exception of node labels at the syllable level. Samoan has quantity-sensitive stress (see §4.2.2.1), so we have a finer-grained set of categories at the syllable level than just σ, and make a distinction between light (L) and heavy (H) syllables.

Empirical evidence to support the claim that this particular prosodic tree is definitely the "right" one, i.e. the prosodic structure of a typical production of (16a), is unfortunately not yet available. Moreover, there might be multiple ways to prosodically phrase the sentence in (16a)—of which this is one—while the syntactic structure remains constant. What matters for us is simply the following: (i) the structure exemplifies properties that makes this case study on Samoan interesting and relevant for work at the syntax–prosody interface, and (ii) the prosodic structure is a reasonable one given what we currently know about Samoan prosody. In (19), we highlight properties of (16a) that make it interesting and relevant as a case study. In the sections following, §4.2.2.1 and §4.2.2.2, we defend the claim that the prosodic structure (18) fits with the currently available empirical evidence. In (18), primary stress is indicated with the IPA diacritic ˈ.

(18) A prosodic structure for (16a) on page 84 that fits the current empirical evidence



(19) Properties of interest in the prosodic tree (18) for our case study

a. The tonal case marker *H-* is treated as an element alongside segmental material in the input string. (See discussion of case marking and H- tones in §4.2.1).

b. The case markers, ergative *e* and absolutive *H-*, are phrased to the left. Syntactically, they are phrased to the right in (17). (See discussion of StrongStart and "bracketing paradoxes" in §4.1.2.4).[13]

c. Each of the case markers is a (light) syllable immediately dominated by a category two levels higher in the prosodic hierarchy, a ϕ-phrase. (See discussion of Exhaustivity in §4.1.2.4.)

d. *na*, the tense marker initiating the sentence, is a (light) syllable immediately dominated by a category much higher in the prosodic hierarchy, an intonational phrase (ι). (See discussion of Exhaustivity in §4.1.2.4.)

e. There is recursivity in the tree in §4.1.2.4): ϕ-phrases dominate other ϕ-phrases. (See discussion of NonRecursivity in §4.1.2.4.)

The inclusion of *H-* as material to be prosodically parsed allows us to model one way to factor out how diverse prosodic information is used for syntactic analysis. As a "post-syntactic" lexical item, H- provides information that guides the syntactic parse—it comes into the computation of MatchPhrase only through its role in the composed syntactic derivation and spellout. However, the prosodic parse (18) into ϕ-phrases, ω's, etc. is a candidate prosodic parse which enters in the computation of MatchPhrase, a computation that checks the alignment of a prosodic parse to the (post-)syntactic parse. The encliticization of the case markers, despite their being being syntactically phrased rightward, exemplifies a typologically common instantiation of "bracketing paradoxes". The violations of Exhaustivity and Nonrecursivity are also typical in current interface analyses, as described in §4.1.2.4. Thus, the assumed prosodic tree (18) clearly factors aspects of the syntax–prosody interface and exemplifies a number of properties that are typical in the analyses of a number of prosodic systems.

In the work here, we implement a phonological transducer that chunks an utterance into prosodic constituents using the categories of the prosodic hierarchy defined in (2). As a consequence—since stress assignment is defined over the prosodic domain of the prosodic word—the transducer also assigns stress to the individual content words, following Zuraw et al. (2014); Yu (2018) (primary stress is shown in (18)). Thus, where stress is observed gives us some clues about what the prosodic structure of an uttered sentence might be at the lower levels of the prosodic hierarchy. §4.2.2.1 describes how stress tells us about prosodic constituency up to the level of the prosodic word. In addition, we use preliminary evidence from studying phonological phrasing in a spoken corpus of Samoan fable retellings (Moyle 1981) to help support hypotheses about the prosodic structure at higher levels of the prosodic hierarchy in §4.2.2.2.

---

[13] The promotion of *lalaŋa* to a ϕ-phrase upon enclitization of the ergative *e* contravenes the Function Word Adjunction Principle hypothesized for Irish in Elfner (2012: 145). This states that when a function word (that isn't a prosodic word) is adjoined to a prosodic category of type β, the node at the adjunction has the same category β. But the current empirical evidence suggests that in Samoan, weak function words like *e* don't fall inside prosodic words (see §4.2.2.1). We leave further consideration of "prosodic adjuction" to future work.

4.2.2.1 Samoan word-level prosody: stress assignment

The basic primary stress pattern in Samoan is moraic trochees at the right edge, exemplified in (20), adapted from Zuraw et al. (2014: (4)). Footing is indicated with parentheses, and periods indicate syllable boundaries. If the final syllable is light (one mora), primary stress falls on the penult; if the final syllable is heavy (two moras), primary stress falls on the final syllable. So feet may be formed from a single heavy syllable (ˈH) or as two light syllables, with the first receiving primary stress: (ˈLL). In both cases, the foot is bimoraic, i.e., it has two moras.

(20) Basic primary stress pattern: moraic trochees at the right edge

...L(ˈH)#	la.(ˈvaː)	'energized'
...(ˈLL)#	(ˈma.nu)	'bird'
...L(ˈLL)#	i.(ˈŋo.a)	'name'

The phonological analysis we assume for stress assignment in Samoan is adapted from the optimality-theoretic analysis in Zuraw et al. (2014). For this case study, we leave aside some of the complexities treated there. Specifically, we abstract away from secondary stress assignment, effects of segmental features on stress assignment (like the complexities of vowel–vowel and vowel–vowel–vowel sequences, and epenthetic vowels), and morphologically complex words. However, we also tentatively expand beyond the phonological analysis in Zuraw et al. (2014) since it doesn't treat prosodic constituency above the level of the prosodic word: see the immediately following section, §4.2.2.2.

The constraint set for stress assignment in this chapter is a a subset of the constraints used in Yu's (2018) implementation of Zuraw et al. (2014)'s analysis, up to an additional inviolable constraint, Headedness. Besides removing constraints that are only relevant for segments, morphologically complex words and multiple prosodic words as in Yu (2018: §2.4), we also remove constraints relevant only for secondary stress assignment. The remaining constraints are given in (21); see Yu (2018) for more details on some slight differences between the definitions of these constraints in (21) vs. the definitions in Zuraw et al. (2014). For the purposes of our case study, since we consider only primary stress in monomorphs, we can treat all these constraints as "undominated" (no constraints are ranked above any of these constraints) and thus inviolable. This means that surviving prosodic parse candidates must satisfy each one of these constraints. We include the additional inviolable Headedness constraint instead of dominated Parse-σ for simplicity of implementation. This way, all phonological constraints ranked above the interface constraint MatchParse are inviolable.[14]

---

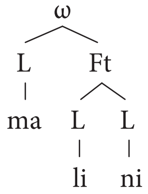
[14] For simplicity, we keep all (purely) phonological constraints except StrongStart undominated in the work here, since the focus is on the interaction of the interface constraint MatchPhrase with the purely phonological constraints. What's of interest here is simply whether a phonological constraint is ranked above or below MatchPhrase—not whether it is violable. We can, however, straightforwardly handle a series of ranked constraints, using an operation called "lenient composition" (Karttunen 1998), as illustrated in Yu (2018).

(21) Constraints for primary stress assignment, all inviolable
- FootBinarity (FootBin) A foot must contain exactly two moras.
- RhythmType=Trochee (RhType=Trochee) A foot must have stress on its initial mora, and its initial mora only.
- Align(PWd,R; ˈFt,R) (Edgemost-R) The end of the prosodic word must coincide with the end of a primary-stressed foot.
- Align(PWd;L,Ft,L) The beginning of the prosodic word must coincide with the beginning of a foot.
- Headedness(Ft) Any prosodic word must dominate a foot.

The definition of the constraints for stress assignment in (21) make it clear how stress assignment is dependent on prosodic constituency: every constraint definition refers to the foot, or the prosodic word, or both. Constraints on the position of stress are defined in terms of the foot and the prosodic word (RhythmType=Trochee and Align(PWd,R; ˈFt,R)). The other constraints regulate how an utterance is chunked into prosodic words and feet.

What does stress assignment—which is observable in elicitations with language consultants[15]—tell us about prosodic constituency in the sentence to be parsed (16a)? In elicitations with language consultants, the content words in the sentence *na lalaŋa e le malini H- le mamanu* all receive penultimate stress when uttered in isolation or when uttered as part of the sentence. Under the inviolable constraints in (21), we can therefore infer the prosodic subtree for [ma.ˈli.ni] 'marine' and the other content words ([la.ˈla.ŋa] 'weave', [ma.ˈma.nu] 'design') to be that shown in (22).

(22) Prosodic constituency for [ma.ˈli.ni] and other content words in (16a)



What about stress assignment for the functional elements in the sentence—the case markers *e*, the determiner *le*, and for that matter, also the H-? First, we'll consider *e* and *le*. These segmental function words do not show evidence of being stressed when they are uttered in (16a): they never receive pitch accents, as far as we can tell.[16] This means they cannot form a prosodic word alone. Moreover, there's no positive evidence that they get incorporated into a prosodic word, if they are adjacent to one: having these monomoraic functional elements uttered alongside the content

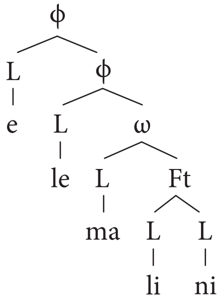
---

[15] See Zuraw et al. (2014) for a discussion of the phonetic realization of stress in Samoan.

[16] Perhaps the specific determiner *le* could receive a pitch accent under contrastive focus, with an alternative set of non-specific determiners; we haven't tested this. Additionally, Yu and Stabler (2017: fig. 9, §6) shows examples where case markers immediately after a pause are uttered with a "pitch reset." It could be that these are instances when monomoraic functional elements are promoted to full prosodic words and receive stress. In any case, *e* and *le* do not appear after pauses in the utterance parsed in this chapter, and so aren't in a context where they might receive stress.

words does not affect stress assignment in the content words. Contrast this with the rightward primary stress shift that occurs with suffixation of the nominalizing, monomoraic -*ŋa*: [ŋa.ˈlu.e] ‘work (V)’ + -[ŋa] → [ˌŋa.lu.ˈe-ŋa] ‘work (N)’ (Zuraw et al. 2014: 309, (57)). We observe no such stress shift with [la.ˈla.ŋa] ‘weave’ + [e] ‘erg’ → [la.ˈla.ŋa e], *[ˈˈla.la.ˈŋa.e]. Similarly, in a monomorphemic sequence of five light syllables, the observed stress pattern has an initial dactyl—with secondary stress on the first syllable: ((ˌte.mo)$_{Ft}$ka(ˈla.si)$_{Ft}$)$_{\omega}$‘democracy’ (Zuraw et al. 2014: 281, (8)). But we’ve never observed *[ˌe.le.ma.ˈli.ni], which is also a string of five lights, nor do we observe *[ˌla.la.ŋa.ˈe.le].[17] In sum, we tentatively conclude that the determiner and segmental case markers in Samoan are neither individual prosodic words nor are they chunked together in a prosodic word with neighboring content words in utterances of (16a). (An alternative would be to assume that the function words form recursive prosodic words with neighboring content words, and say that the only the deepest prosodic word is the domain of stress.) Rather, a reasonable structure for *e le malini* in the utterance of (16a) might be something like the right-branching prosodic subtree in (23), with *e* and *le* outside the ω. This leads us into prosodic constituency above the level of the prosodic word, where we’ll also consider the prosodic phrasing of the H-.

(23) A possible prosodic subtree for *e le malini* in the utterance of (16a)



4.2.2.2 Samoan prosodic constituency above the prosodic word

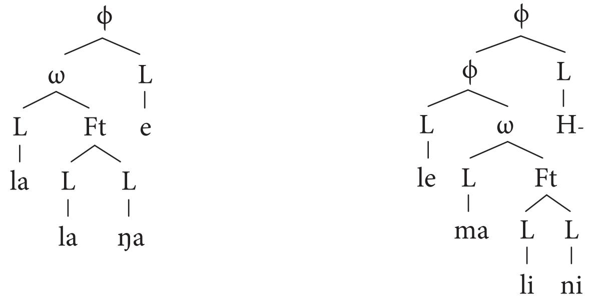
There is some evidence that suggests that (23) is not, in fact, the most plausible prosodic structure for *e le malini*. Rather, there is reason to support the hypothesis that both ergative *e* and absolutive H- are prosodically phrased to the left in a bracketing paradox, as in the StrongStart repair in (14); see e.g. see prosodic structures below in (24). We’ll refer to case markers phrased like this descriptively as being “encliticized.” In (24), ergative *e* is encliticized to preceding VP *lalaŋa* and absolutive H- is encliticized to preceding agent DP *le malini*. These prosodic trees satisfy StrongStart, but violate MatchPhrase.

---

[17] Zuraw et al. (2014) shows that five-mora loanwords can also surface with the stress pattern LˌLLˈLL in the presence of epenthetic vowels, but we don’t observe that stress pattern either.

(24) Prosodic structures with case markers phrased to the left

a. Encliticized ϕ-initial ergative *e* b. Encliticized ϕ-initial absolutive H-



While the idea of an encliticized high tone may seem bizarre, we'll build up the case for it by first turning to evidence for encliticization of segmental monomoraic function words like ergative *e*. Unfortunately, we haven't yet found phonological processes besides stress shift that can potentially reveal prosodic constituency. But there has been work that has made connections between speech planning and prosodic chunking (e.g. Wagner 2012; Himmelmann 2014; Krivokapić 2014). In particular, in studies of corpora of unscripted English, German, and Tagalog retellings of a silent film called the *Pear Story* (Chafe 1980), Himmelmann (2014) found that articles and prepositions in English and German and Tagalog "phrase markers" were "frequently separated from their lexical hosts by disfluencies." That is, pauses and other disfluencies often occur between a function word phrased to the left and its following lexical (morphosyntactic) host. For example, *And that's the end of the ... story* (Himmelmann 2014: 935, (5)). Moreover, there's an asymmetry in the distribution of disfluencies: they tended to occur after function words breaking them off from their following host, rather than occur before function words. Himmelmann (2014) places this asymmetry at the center of a speech planning/production account for the strong typological asymmetry that syntactic proclitics are often prosodically enclitic, while syntactic enclitics are rarely prosodically proclitic.

Following the methods in Himmelmann (2014), we've found some very preliminary evidence for encliticization of function words from studying phonological phrasing in a spoken corpus of unscripted Samoan fable retellings (Moyle 1981) (ongoing, joint work with Matthew Frelinger). While we haven't yet completed a quantitative study, we've certainly found a number of cases where case markers and even determiners are followed by disfluencies (e.g. in (25) below, recording available at http://www.fagogo.auckland.ac.nz/content.html?id=1). In (25), there are hesitations (which we indicate informally with ...) between the determiner *le* and its following host noun, and between the topic marker *o* and its following proper name.

(25) Example of disfluencies after function words in unscripted Samoan fable retellings collected in Moyle (1981)

a.

ʔo	le...	uluga:liʔi.	ʔo	le	iŋoa	o	le	tama
TOP	DET.SG	couple	TOP	DET.SG	name	GEN	DET.SG	boy

ʔo...	ʔulafala-Manoŋi-Sasala-ʔi-Tausala
TOP	Ulafala-Manogi-Sasala-Tausala

'There was once a couple; the young man's name was Ulafa-Manogi-Sasala-Tausala.'

What about the case for the encliticization of the absolutive H-? A puzzling fact about the phonetic realization of the H- is that the associated peak in fundamental frequency (the acoustic correlate of pitch) occurs on the phonological material *preceding* the absolutive argument, not on the absolutive argument (Yu to appear). As described in Yu and Özyıldız (2016) and Yu (to appear), one explanation for this tonal "enclitization" is that diachronically, the absolutive H- may be the remnant of a segmental absolutive particle *ia*. The particle *ia* is bimoraic and also seems to be stressed, while the other case markers and many other function words are monomoraic and unstressed. In contemporary Samoan, the usage of *ia* seems very much in flux and deprecated, but speakers do have systematic intuitions about its distribution and produce it sometimes reduced to monomoraic [jɛ]. If, like other function words (including case markers), this (possibly reduced) *ia* was also encliticized, then as the segmental material may have been reduced and elided, the associated pitch accent may have been left behind and also realized leftward.

In sum, there is some (admittedly speculative, preliminary) evidence for the encliticization of case markers. We therefore tentatively assume that the optimal prosodic trees computed in this case study should have prosodic structures like (24) that satisfy StrongStart but result in violations of MatchPhrase. Accordingly, we include StrongStart as a prosodic markedness constraint at the level above the prosodic word. Following the mainstream trend in interface work, we also assume Headedness (Any C^i must dominate a C^{i-1} (except if $C_i = \sigma$), e.g., a PWd must dominate a Ft) and Layeredness (No C^i dominates a C^j, $j > i$, e.g., no σ dominates a Ft) to be inviolable (see 11). We also assume Binarity at all levels of the prosodic hierarchy (12). As we'll see in §4.2.3, a high-ranking Binarity constraint can prune out prosodic parses with unary branches early on. While it may be the case that Binarity constraints don't (all) need to be inviolable for the desired prosodic parse to be computed as optimal, we implemented them as such to prune the candidate set and reduce the processing load (otherwise, `xfst` ran out of memory). Finally, we restrict the prosodic parses to ones where a single intonational phrase exhaustively dominates the entire input. Sentences like (16a) can be uttered as multiple intonational phrases, but not usually in typical conversational speech.

4.2.3 The Samoan syntax–prosody interface

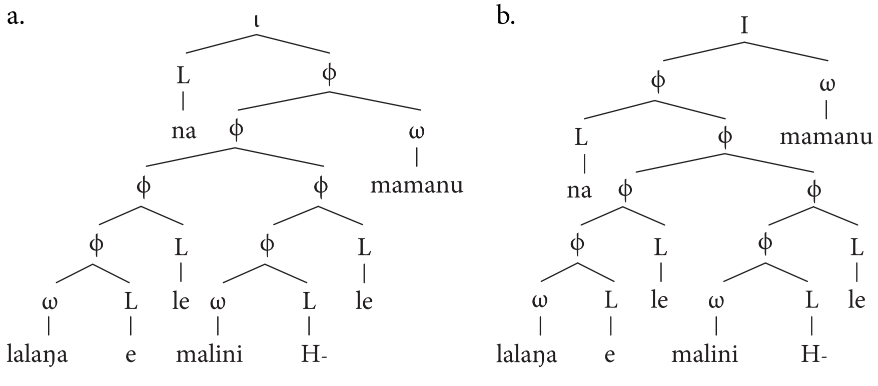
So far, we've enumerated phonological constraints for stress assignment and prosodic chunking up to the level of the prosodic word (21), as well as phonological markedness constraints active at higher levels in the prosodic hierarchy (§4.2.2.2). We've also introduced MatchPhrase as an interface constraint. Here, we'll discuss how we're ranking the interface constraint MatchPhrase relative to the various phonological constraints (§4.2.3.1). We'll also explain how we'll apply MatchPhrase to the composed syntactic/spellout tree (§4.2.3.2).

4.2.3.1 Constraint ranking: an interface constraint interleaved between phonological constraints

In constraint-based analyses of the interface in the literature, interface constraints often are ranked to be interleaved between higher-ranked and lower-ranked (purely) phonological constraints, so this is a property that we'd like to model in our case study. We already mentioned in §4.2.2.1 that all the phonological constraints relevant for stress assignment and prosodic parsing up to the level of the prosodic word are undominated. In §4.2.2.2, we also explained why HEADEDNESS and BINARITY constraints above the level of the prosodic word are treated as undominated. However, we rank STRONGSTART below MATCHPHRASE. This means that all phonological constraints except STRONGSTART are used to winnow down the candidate set of prosodic parses to be submitted to the MG parser. The MG parser is then used to compute MATCHPHRASE, given the syntactic analysis in (17b). Finally, the remaining prosodic candidates—those with the fewest violations of MATCHPHRASE—are passed to STRONGSTART for the final winnowing step to compute the optimal candidate.

The two prosodic parse candidates in (26) help provide an argument for ranking STRONGSTART below MATCHPHRASE. These two candidates are submitted to the MG parser because they satisfy all the inviolable phonological constraints. Moreover, each candidate has only a single violation of STRONGSTART, due to the stray syllable [na] at the left edge of the intonational phrase. However, as we'll show in the next section, §4.2.3.2, the optimal prosodic parse Candidate 9 (18) has 3 STRONGSTART violations. We'll also see in §4.2.3.2, that Candidate 9 incurs no MATCHPHRASE violations, while the prosodic trees in (26) each incur 3 (under our extension of the definition of MATCHPHRASE to post-syntactic material in (27)). Therefore, under our constraint set, STRONGSTART must be ranked below MATCHPHRASE to keep Candidate 9 from being pruned out before being submitted to the MG parser.

(26) Prosodic candidates with only 1 STRONGSTART violation submitted to MG parser



4.2.3.2 Extending the definition of MATCHPHRASE to spellout

The definition of MATCHPHRASE given in (5) and explicated in §4.1.2.3 doesn't extend to post-syntactic material; but the insertion of case markers *e* and H- and derivation steps involving the phrases they project, DP_{erg} and DP_{abs}, are post-syntactic and composed with the syntactic lexical item insertion and derivation in this case study. Therefore, we need to extend the definition of MatchPhrase. We extend it as stated in (27). The motivation for defining the extension this way will become clear by the end of this section.

(27) Extension of MatchPhrase to spellout

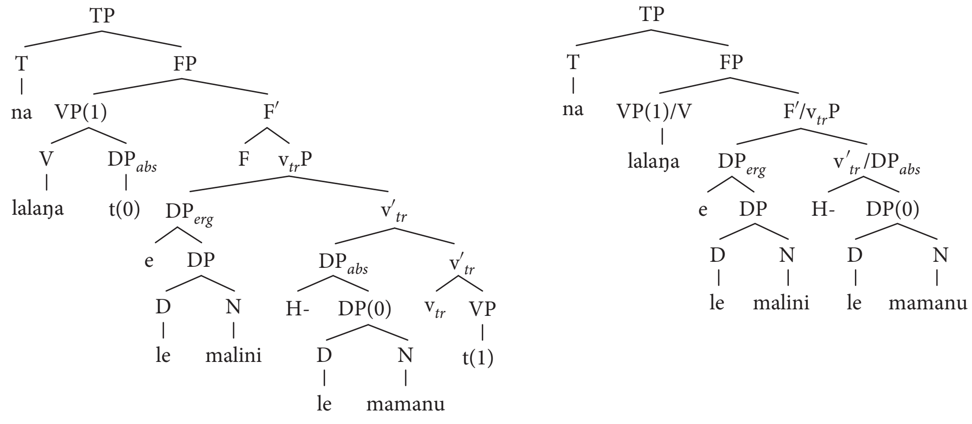
a. Maximal projections derived post-syntactically, e.g. DP_{erg} and DP_{abs}, are not in the domain of MatchPhrase. Only maximal projections derived *syntactically* are. Thus, MatchPhrase is not assessed for either DP_{erg} or DP_{abs}.

b. Phonological material inserted post-syntactically, e.g. ergative *e* and absolutive H-, is visible to prosodic parsing, but not to MatchPhrase. That is, the case markers are parsed prosodically like any other morpheme, and the resulting prosodic parse is entirely visible to MatchPhrase, but the case markers aren't visible as phonological exponents for MatchPhrase.

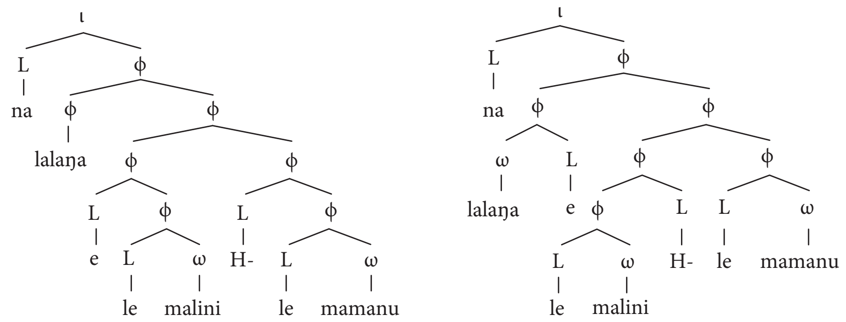
For instance, since [lalaŋa] 'weave' projects to a VP in (17b), it incurs a MatchPhrase violation unless it also projects to a ϕ-phrase. But the prosodic subtree (24a) does not incur a MatchPhrase violation even though [lalaŋa] projects to a prosodic word, because the encliticization of *e* to the prosodic word forms a ϕ-phrase. Even though the ϕ-phrase dominates both [lalaŋa] and *e*, since the phonological exponent *e* is invisible to MatchPhrase, there is a ϕ-phrase that dominates only [lalaŋa].

With the syntactic analysis in (28a) (repeated from (17b) above), the constraint ranking defined in §4.2.3.1, and the extension of the definition of MatchPhrase given in (27), we compute (28d) to be an optimal prosodic parse (repeated from (18) above), and filter out the prosodic parse (28c), which is isomorphic to the flattened bare syntactic tree (28b). The prosodic tree (28d) abstracts over prosodic structure below the word level that is shown in (18), e.g. it doesn't show the footing of [malini]. This abstraction makes the comparison to the (post-)syntactic trees more transparent. We call this prosodic tree Candidate 9 in our implementation.

(28)


a. bare syntactic tree for (16a)

b. flattened bare syntactic tree for (16a)

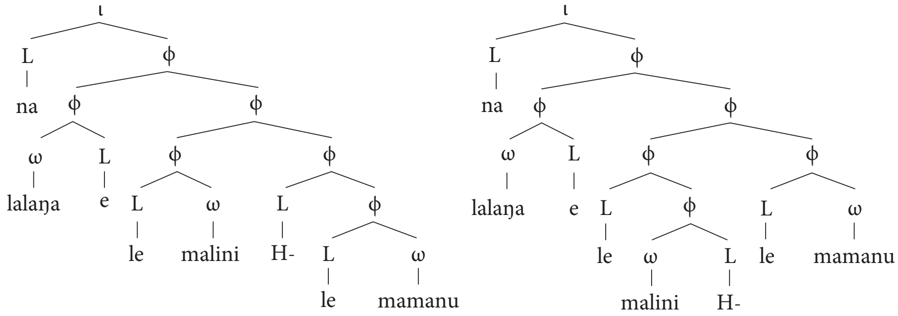

c. prosodic tree isomorphic to (b)

d. “actual” prosodic tree (Candidate 9)

Both (28c) and (28d) incur no violations of MatchPhrase. (28c) is isomorphic to the flattened bare tree (28b), and (28d) incurs no violations because of the definition (27) of the extension of MatchPhrase to post-syntactic material: the encliticization of post-syntactic *e* to [lalaŋa] forms a φ-phrase that corresponds to the VP dominating [lalaŋa], and the encliticization of the case markers results in no MatchPhrase violations for DP_{erg} or DP_{abs} because those are post-syntactic maximal projections—not syntactic ones. Both these prosodic parses violate phonological constraints, though. Candidate (28c) has 5 StrongStart violations, due to the prosodification of the light syllables [na], [e], [H-], [le], and [le]; candidate (28d) has only 3 StrongStart violations, due to the prosodification of [na], [le], and [le]. In addition, (28c) incurs a violation of Binarity because of the unary branch to [lalaŋa]. Therefore, (28c) is actually never even passed to the MG parser at the interface, since Binarity is ranked above MatchPhrase and undominated.

There are two other prosodic parse candidates passed to the MG parser besides Candidate 9 (28d) that incur no violations of our extended definition of MatchPhrase: Candidate 10 (29a) and Candidate 7 (29b). Candidate 10 has 4 StrongStart violations due to the prosodification of [na], [H-], [le], and [le], while Candidate 7 has only 3—the same number as Candidate 9—due to the prosodification of [na], [le], and [le]. Thus, Candidate 7 is also computed to be an optimal candidate, alongside Candidate 9, while Candidate 10 is filtered out by StrongStart when the remaining parses are passed back from the MG parser. The survival of an additional candidate to Candidate 9 was a surprise to us during the implementation, demonstrating the importance of implementation to check our analyses. Generally, a constraint-based grammar is expected to compute a single optimal candidate among the input candidate set—if this isn’t the case, then it’s assumed that the analyst has missed some crucial constraint(s). The difference between the two winning candidates is whether the absolutive H- is encliticized to the prosodic word [malini] (Candidate 7), or if it is enclitized to the φ-phrase [le malini] (Candidate 9). In this case, we don’t find the difference between Candidate 7 and 9 to be one that points clearly to some missed constraint, or one that might be empirically observable (at least given our current understanding of the Samoan syntax–prosody interface).

(29) Other prosodic parses with no MatchPhrase violations



a. Candidate 10

b. Candidate 7: also a winning candidate

Now we are ready to motivate the extension of MatchPhrase to post-syntactic material (27). Suppose, *contra* (27a), that we included maximal projections derived post-syntactically in the domain of MatchPhrase. If we did, then the encliticization of the ergative case marker *e* and the absolutive case marker H- would result in a violation of MatchPhrase for DP_{erg} and DP_{abs}, respectively. Then, since our desired winning candidates, Candidates 7 (29b) and 9 (28d), enclitize both case markers, each would incur 2 violations of MatchPhrase—one for DP_{erg}, and one for DP_{abs}. However, Candidate 10 (29a), which encliticizes only the H-, would incur just 1 MatchPhrase violation. The desired winning candidates would therefore both be filtered out by this alternate extension of MatchPhrase.

Turning to the second part of the definition (27b), suppose, *contra* (27b), that phonological elements inserted post-syntactically, e.g. ergativ *e* and absolutive H-, were visible as phonological exponents for MatchPhrase. Then we could not repair the MatchPhrase violation incurred by mapping a VP to a prosodic word for *lalaŋa* by the encliticization of ergative *e*, as in (24a). The VP would dominate a singleton set of terminals *{lalaŋa}*, while the ϕ-phrase would dominate the set of terminals *{lalaŋa, e}*, still resulting in a MatchPhrase violation. We also would incur other MatchPhrase violations due to encliticization. For example, the FP in (28b) would dominate the terminals {*e*, *le*, *malini*, *H-*, *le*, *mamanu*}, but its correspondent ϕ-phrase in (28d) would dominate only {*le*, *malini*, *H-*, *le*, *mamanu*}. We'll discuss the extension of MatchPhrase to post-syntactic material further in §4.4.

This concludes the presentation of background and overview of the implementation of prosodically informed parsing of an utterance of (16a). The next section, §4.3, builds on this section to explicate the full implementation.

4.3 Samoan syntax, spellout, prosody, and interfaces: implementation

We present the implementation of the computation of the interface and parsing in three sections. First, in §4.3.1, we describe the implementation of the phonological transducer in `xfst` that generates and filters prosodic parses to arrive at a final candidate set to submit to the MG parser. Then, in §4.3.2, we define the MG syntactic/post-syntactic lexicon that derives the assumed (post)-syntactic tree (17b). Finally, in §4.3.3, we describe the implementation of the MG parser to simultaneously syntactically parse (16a) and compute MatchPhrase for the submitted prosodic parses, returning only the prosodic parses with the fewest violations of MatchPhrase to be evaluated by StrongStart. The final StrongStart transduction is described in §4.3.4. The code for the implementation is available at https://github.com/krismyu/smo-prosodic-parsing.

4.3.1 Generation of candidate prosodic parses with an `xfst` transducer

We generate the set of candidate prosodic parses to be submitted to the MG parser by defining a phonological transducer in `xfst` that adds prosodic markup to the input, an utterance of (16a), *na lalaŋa e le malini H- le mamanu*. It is based on the finite state optimality-theoretic foot-based grammar fragment for Samoan word-level prosody defined in Yu (2018: §2.4), but is simplified to abstract away from secondary stress, and also extended to parse the input into prosodic constituents higher than the prosodic word, as described in §4.2.2. The transduction results in a set of 36 prosodic parses to be passed to the MG parser.

We represent an utterance of *na lalaŋa e le malini hi le mamanu* as an input string of syllables marked for weight (light (L) or heavy (H)) and morphosyntactic word boundaries (+), as in (4.3.1). The [ŋ] is transcribed as 'g', following Samoan orthography, and the *H-* as 'hi'.

(30) Input to the `xfst` transducer that adds markup indicating prosodic constituency
`+[L,na]+[L,la] [L,la] [L,ga]+[L,e]+[L,le]+[L,ma]`
`[L,li] [L,ni]+[L,hi]+[L,le]+[L,ma] [L,ma] [L,nu]+`

For instance, `+[L,e]+` denotes that the syllable *e* is its own morphosyntactic word (as it is immediately preceded and succeeded by +), and that it is a light syllable (as indicated by `L`). The generated parses contain markup indicating left and right edges of following prosodic constituents, enumerated from highest to lowest level, see (31), cf. the prosodic hierarchy given in (2).

(31) Markup indicating left and right edges of prosodic constituents enumerated in (2)
a. Intonational phrase (IP) `-..._`
b. (Maximal) Phonological phrase (PhP_{max}) `Z...z`
c. (Non-minimal) Phonological phrase (PhP_{nmin}) `Y...y`
d. (Functional) Phonological phrase (PhP_{fxn}) `X...x`
e. (Minimal) Phonological phrase (PhP_{min}) `<...>`
f. Prosodic word (PWd) `{...}`
g. Foot (Ft) `(...)`
h. Syllable `[L,...]` (Light), `[H,...]` (Heavy)

The definition of prosodic structure we gave in §4.1.2.1 and §4.1.2.2 allows for recursivity in prosodic trees, as exhibited in the recursive ϕ-phrases in (18). But our implementation is not truly recursive: *we allow recursion only up to finite bounds*. Different levels of embedding in ϕ-phrases—four in total—are indicated by different prosodic markup given in (31). Imposing finite bounds is a methodological abstraction, not a theoretical stance. Maintaining distinct markup for each level of ϕ-phrase allows us to have a simple implementation of MATCHPHRASE for this first case study. Namely, we can assess whether MATCHPHRASE is violated using just the prosodically marked-up string, as detailed in §4.3.3.2. If we allowed recursion to arbitrary depth, we'd have to compare non-isomorphic prosodic and syntactic trees. We discuss the implications of prosodic recursion to arbitrary depth for parsing further in §4.4.

A sketch of the algorithm defined for the `xfst` transduction is given in §4.3.1. The output from each step is the input to the following step. The basic structure of the algorithm is to loop over alternating steps of overgeneration of parses and pruning at each level in the prosodic hierarchy. HEADEDNESS is not used to prune parses for ϕ-phrases above the level of non-minimal ϕ-phrases in Step (i) in (32). As long as at least one ϕ-phrase has been introduced, i.e. at the level of PhP_{min}, then an intonational phrase will dominate a ϕ-phrase. Each step in the algorithm is implemented as a transducer, and the entire transduction is a cascade of these individual transduction composed together.

(32) Sketch of `xfst` transduction to generate prosodic parses
   a. **define** input string (30)
   b. **overgenerate** candidates for stress assignment to input string
   c. **overgenerate** stress-marked, foot-parsed candidates
   d. **prune** using foot-based constraints in (21)
   e. **overgenerate** prosodic-word parsed candidates
   f. **prune** prosodic word parses using prosodic-word-based constraints in (21)[18]
   g. **overgenerate** candidates parsed into minimal phonological phrases (PhP_{min})
   h. **prune** with HEADEDNESS(PhP_{min}) and BINARITY
   i. **for** phonological phrases PhP in (PhP_{fxn}, PhP_{nmin}, PhP_{max}):
      i. **overgenerate** candidates parsed into PhP
      ii. **prune** with BINARITY
   j. **overgenerate** candidates parsed into intonational phrases (IPs)
   k. **prune** with HEADEDNESS(IP), BINARITY, and for input to be exhaustively parsed into a single IP

---

[18] In addition to those constraints, we also prune using NOFUNCTIONWDPWD: Don't parse function words as prosodic words. This is purely an implementational detail, to reduce the number of candidates remaining at this stage from 28 to 2. Whether it is included or not does not affect the resulting set of candidates passed to the MG parser. It is included only to reduce the amount of processing needed in the transduction.

4.3.2 Syntactic/post-syntactic grammar fragment: MG lexicon and derivation

Once the prosodic parses have been filtered by the phonological constraints ranked above MatchPhrase, they are submitted to the MG parser. For our case study, we assume that the sentence to be parsed, (16a), is not ambiguous. Therefore, we define the MG grammar to admit only the parse shown in the derived structure (17b). We will not give a detailed explication of MGs here. For a leisurely introduction to minimalist grammars, including a comparison to Minimalism, see Graf (2013: chs. 1 and 2). Graf (p. 8) informally summarizes the fundamental characteristics of MGs with five points, given in (33).

(33) Graf (2013: 8)'s informal description of the feature calculus of MGs

a. Lexical items (LIs) consist of a (possibly null) phonetic exponent and one or more features.
b. Features come in two polarities, and both Merge and Move delete exactly two features that differ only in their polarity.
c. Each LI's features are linearly ordered and must be checked in that order.
d. Besides checking features, Merge and Move also build phrase structure trees in the familiar way.
e. The Shortest Move Constraint blocks every configuration where more than one phrase can be moved in order to check a given feature.

Given that MGs are feature-driven as described in (33), an MG is specified by its lexicon, where each lexical item specifies some phonological content (here we use IPA transcription, except for the high edge tone) and a sequence of features. Below we define the lexicon we use to derive and parse (16a). It suffices for the derivation here, and is inspired by—but much simplified from—Collins (2016; 2014). The lexicon has an unusual property that makes it different from typical minimalist lexicons (e.g. the first lexicon given in appendix B in Yu and Stabler (2017), which is similar to the one here). It is a syntactic *and post-syntactic* lexicon rather than a purely syntactic lexicon, and the case-marking rules in spellout are folded into the transduction rules, as well. The lexicon thus includes the case markers *e* and H- (here, notated as *hi*), as case-marked DPs, too. As discussed and justified in §4.2.3 and §4.2.2.2, the (tonal) H- enters the MG no differently than segmental lexical items, i.e. the same way as the ergative [e].

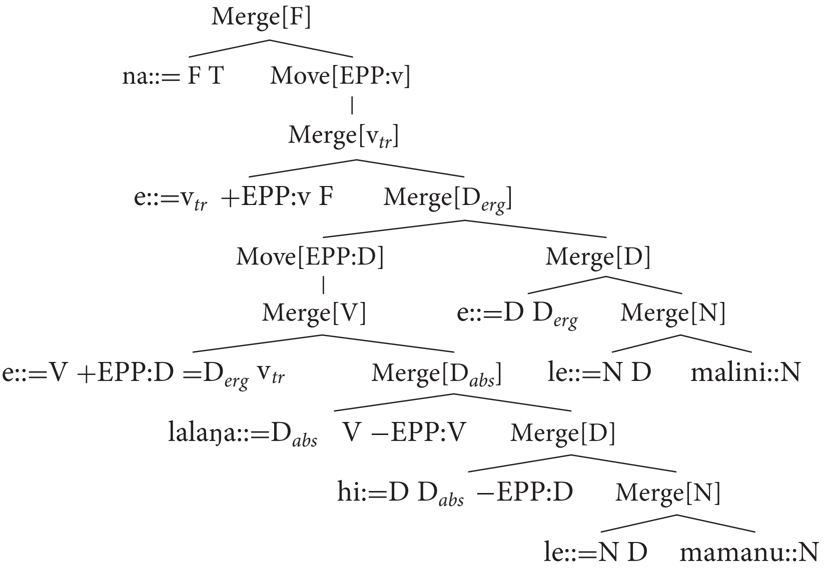
The definition of the lexicon below follows standard MG notation. For instance, ε indicates an empty functional head, '::' indicates that the item is lexical and not derived, and the feature '=D' indicates that a DP is to be selected. The feature '+EPP:D' triggers the movement of a '-EPP:D' phrase to a specifier; and the feature +EPP:V triggers the movement of a '-EPP:V' phrase to a specifier.

(34) MG lexicon for the syntactic derivation and spellout of (16a)

na::=F T	'past', selects FP to form T
ε::=v_{tr} +EPP:V F	selects v_{tr}p, moves -EPP:V, to form F
lalaŋa::=D_{abs} V -EPP:V	'weave', selects DP_{abs} to form VP, then moves to EPP
hi::=D D_{abs} -EPP:D	selects D to form DP_{abs}, then moves to EPP
e::=D D_{erg}	selects D to form DP_{erg}
le::=N D	selects NP to form DP
malini::N	'marine', noun
mamanu::N	'design', noun
ε::=V +EPP:D =D_{erg} v_{tr}	selects VP, moves EPP:D, selects DP_{erg} to form v

Given the lexicon in (34), (16a) can be derived using standard MG Merge and Move rules. This derivation—that is, the actual step-by-step process, not the result of the derivation—is depicted in the "augmented derivation tree" (see Graf 2013: 12–15) in (35). The tree displays the lexical items as leaves at the point in which they enter the derivation, and internal nodes are labeled with the operation taking place at that point and the feature being checked.[19]

(35) An augmented derivation tree showing the composed derivation and spellout of (16a) given the MG lexicon in (34)



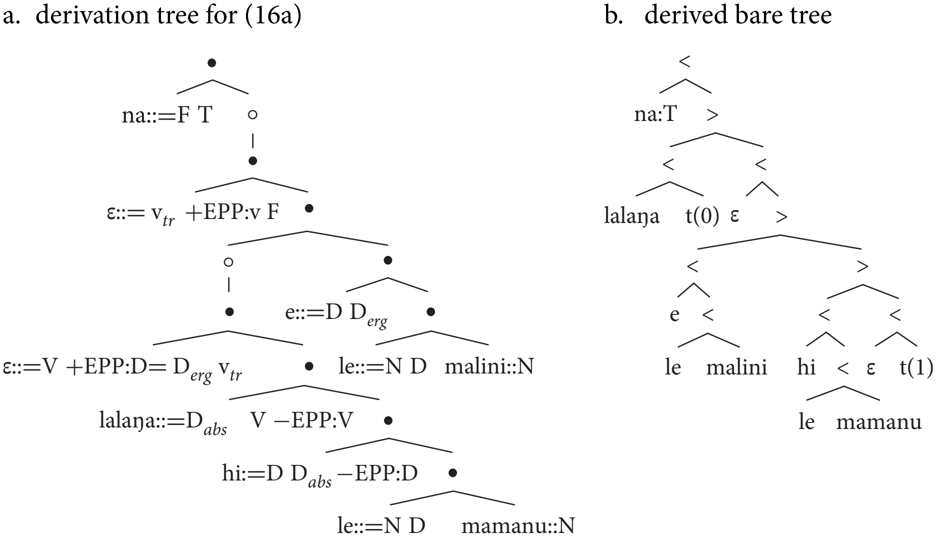
A simplified derivation tree is given in (36a) below. This is the same depiction as the augmented derivation tree (35), but with node labels for Merge and Move operations abbreviated to reduce notational clutter. Following standard practice for MG derivation trees, • indicates a Merge step, while ∘ indicates a Move step. We can

---

[19] As pointed out in Graf (2013: 13), an augmented derivation tree can be described as a "strictly binary branching multi-dominance tree", since a subtree could be involved in both Merge and Move operations. For completeness, we could indicate movement dependencies in (35) with lines between the dependents in the pairs (Move[EPP:v], lalaŋa::=V -EPP:V) and (Move[EPP:D], hi::=D D_{abs} -EPP:D), but doing so is redundant since the dependency is already specified in the lexicon.

also straightforwardly transduce the derivation tree (35) into the derived bare tree (36b), similar to the Bare Phrase Structure trees in Chomsky (1955), as well as the X-bar-like bare tree (36c). The derived tree (36b) shows the result of the derivation (36a) at each step. The lexical items are stripped of their feature stacks because all their features have been consumed by the point they are shown in the tree. The interior nodes are labeled with arrows < and > that point to the projecting head after each derivational step—as noted in Graf (2013: 23), derived trees generated by MGs, e.g. (36b), are *ordered* trees. MGs are defined this way because ordered trees, compared to unordered ones, are well-behaved mathematically. The ordering is the same as that obtained by the c-command linearization algorithm of Kayne (1994).[20]

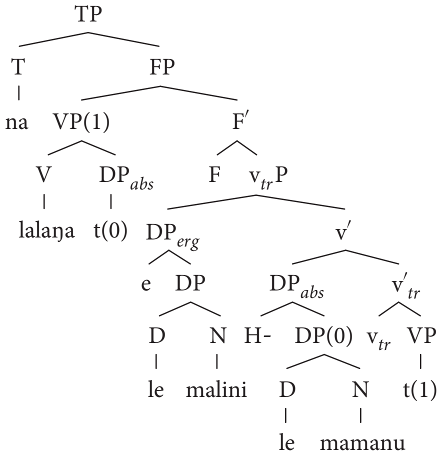
(36)



a. derivation tree for (16a)

b. derived bare tree

c. "X-bar-like" bare tree



---

[20] Note that the property that MG derivations result in ordered trees sets aside issues about linearization that are of interest to linguists, including the role of prosodic information in linearization considered in Richards (2010) and Bennett et al. (2016). But MGs can also be used to model different linearization procedures by adding additional assumptions to the machinery of MGs (e.g. Graf 2012)—we leave these issues to further work.

Further details about the definitions of Merge and Move operations in MGs become important in the implementation of MatchPhrase. We introduce those in §4.3.3.

4.3.3 Simultaneous (post-)syntactic parsing and MatchPhrase computation

Harkema (2000; 2001); Stabler (2013b); and Fowlie and Koller (2017) have already defined a range of provably correct and efficient parsing algorithms for MG structures like the ones we've defined in the previous section. We implement one of these, a bottom-up parser for MGs without head movement (Harkema 2001: ch. 4), adapted from code in Stabler (2013a: 110–22). We use it to simultaneously (post-)syntactically parse (16a) and compute MatchPhrase. We first explain the implementation of the bottom-up parser for (post-)syntactic analysis in §4.3.3.1. Then we explain how MatchPhrase is computed during the (post-)syntactic parse in §4.3.3.2.

4.3.3.1 Bottom-up (post)-syntactic parsing of MGs

We repeat the derivation tree for (16a) in (38), first introduced in §4.3.2. It is annotated at each node with the alphabetic label of the individual derivation step being depicted, where the labels come from the 11-step derivation given in (40). The derivation steps in (40) are given in a reformulated, tuple notation (Harkema 2001: §4.1–4.3) that is used by the parser. The Merge and Move functions can also be reformulated in this notation into several different functions and are exemplified in (41) using steps from (40). We'll discuss these functions in more detail when we describe the implementation of MatchParse in §4.3.3.2. The numbers in parentheses in the reformulated notation define the span of an expression generated by the grammar and refer to the string yield of (16a), represented as a finite state acceptor in (39). For instance, (6,8) in Step $\boxed{\text{A}}$ in (40b) can be read off of (39) as *le mamanu*. The reformulation also replaces each tree by a tuple of categorized strings where commas are used to separate moving phrases, as can be seen by comparing the examples in (41) with their reformulated versions in (40b). These reformulated rules are further discussed in the context of the computation of MatchPhrase in §4.3.3.2.

The algorithm for the bottom-up parser is similar to the CKY algorithm for context-free languages (see e.g. Younger 1967). The initial steps are schematized in the matrix in Table 4.2, where each cell (i, j) corresponds to the span (i, j) in (39). The feature stacks of lexical items and empties are inserted in the matrix in the appropriate cell. For instance, feature stacks from empties ε go in all cells (i, j) where $i = j$, since empties have an empty span. Each inserted item is added to an agenda stack. After initialization, the parser scans down the stack and checks to see if any of the different Merge and Move functions can be applied. If a successful derivational step is taken, it is recorded in the matrix. So each step given in (40) would be recorded in cells (i, j) according to the spans of the resulting expressions, as illustrated in (37).

(37) Recording the results of the derivational steps shown in (41) in the chart shown in Table 4.2

a. Merge-1, $\boxed{\texttt{A}}$ in (40b)
   - The feature D is recorded in cell (6,8), i.e., the cell in row 6, column 8.

b. Merge-2, $\boxed{\texttt{H}}$ in (40b)
   - The expression v_{tr}, (1,2) −EPP:V is recorded in cell (2,8). The comma in the expression separates out the moving phrase from the feature v_{tr}.

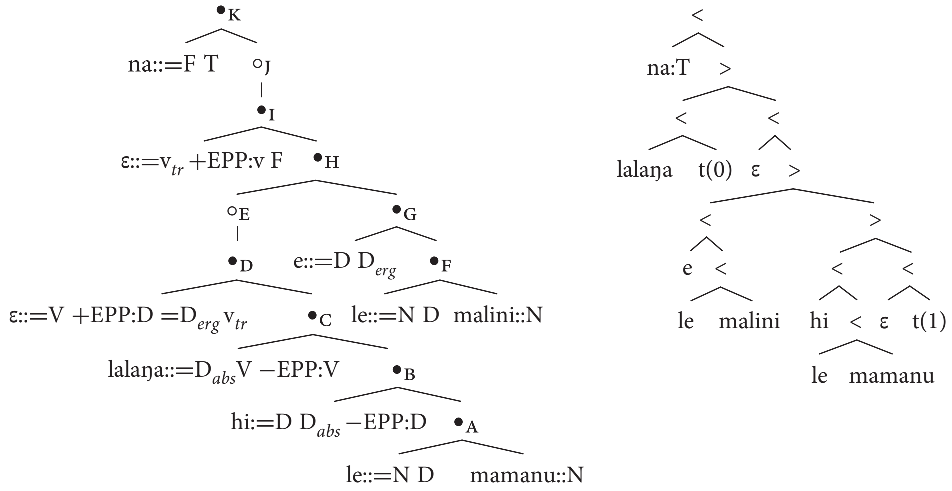
c. Merge-3, $\boxed{\texttt{C}}$ in (40b)
   - The expression V −EPP:V, (5,8) −EPP:D is recorded in cell (1,2).

d. Move-1, $\boxed{\texttt{E}}$ in (40b)
   - The expression $=D_{erg}$ v_{tr}, (1,2) −EPP:V is recorded in cell (5,8).

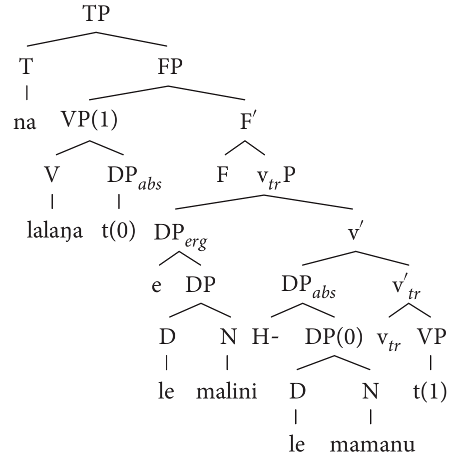
(38)

a. derivation tree for (16a)

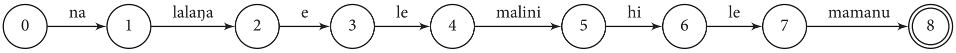
b. derived bare tree



c. ‘X-bar like’ bare tree



(39) Finite state acceptor for the string yield of (16a)



(40) Derivation of sentence in tuple notation

a. Lexicon:

$\boxed{1}$ na::=F T

$\boxed{2}$ lalaŋa::=D_{abs} V -EPP:V

$\boxed{3}$ ε::=v_{tr} +EPP:V F

$\boxed{4}$ e::=D D_{erg}

$\boxed{5}$ le::=N D

malini::N $\boxed{6}$

hi::=D D_{abs} -EPP:D $\boxed{7}$

mamanu::N $\boxed{8}$

ε::=V +EPP:D =D_{erg} v_{tr} $\boxed{9}$

b. Derivation, 11 steps:

Merge-1($\boxed{5}$,$\boxed{8}$) = (6,8):D $\boxed{A}$

Merge-1($\boxed{7}$,$\boxed{A}$) = (5,8):D_{abs} −EPP:D $\boxed{B}$

Merge-3($\boxed{2}$,$\boxed{B}$) = (1,2):V −EPP:V, (5,8): −EPP:D $\boxed{C}$

Merge-3($\boxed{9}$,$\boxed{C}$) = (8,8):+EPP:D=D_{erg} v_{tr}, (1,2):V −EPP:V, (5,8): −EPP:D $\boxed{D}$

Move-1($\boxed{D}$) = (5,8):=D_{erg} v_{tr}, (1,2):−EPP:V $\boxed{E}$

Merge-1($\boxed{5}$,$\boxed{6}$) = (3,5):D $\boxed{F}$

Merge-1($\boxed{4}$,$\boxed{F}$) = (2,5):D_{erg} $\boxed{G}$

Merge-2($\boxed{E}$,$\boxed{G}$) = (2,8):v_{tr}, (1,2):−EPP:V $\boxed{H}$

Merge-1($\boxed{3}$,$\boxed{H}$) = (2,8):+EPP:V F, (1,2):−EPP:V $\boxed{I}$

Move-1($\boxed{I}$) = (1,8):F $\boxed{J}$

Merge-1($\boxed{1}$,$\boxed{I}$) = (0,8):T $\boxed{K}$

(41) Examples of reformulated Merge and Move operations from the derivation (40), annotated with spans and illustrated with derived trees

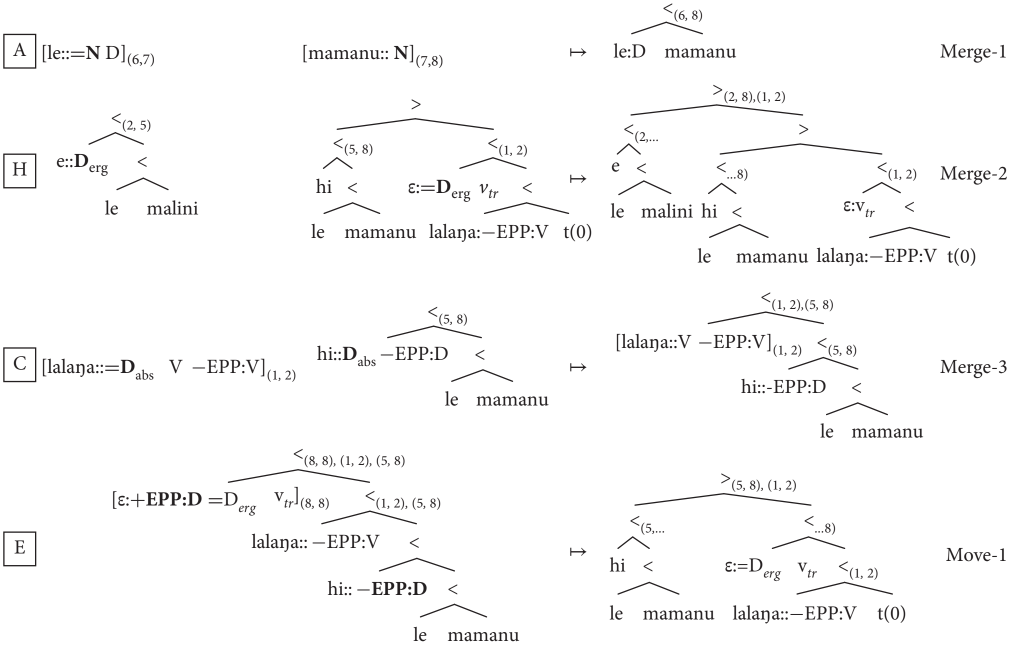


Table 4.2 Matrix after initialization of empties and lexical items. The features for empty categories down the diagonal are abbreviated for space as ε_F ($\boxed{9}$) for ::=v_{tr} +EPP:V F and $\varepsilon_{v_{tr}}$ ($\boxed{3}$) for ::=V +EPP:D =D_{erg} v_{tr}.

	0	1	2	3	4	5	6	7	8
0	ε_F $\varepsilon_{v_{tr}}$	$\boxed{1}$ :: =F T							
1		ε_F $\varepsilon_{v_{tr}}$	$\boxed{2}$:: = D_{abs} V −EPP:V						
2			ε_F $\varepsilon_{v_{tr}}$	$\boxed{4}$:: = D D_{erg}					
3				ε_F $\varepsilon_{v_{tr}}$	$\boxed{5}$:: = N D				
4					ε_F $\varepsilon_{v_{tr}}$	$\boxed{6}$::N			
5						ε_F $\varepsilon_{v_{tr}}$	$\boxed{7}$:: = D D_{abs} −EPP:D		
6							ε_F $\varepsilon_{v_{tr}}$	$\boxed{5}$:: = N D	
7								ε_F $\varepsilon_{v_{tr}}$	$\boxed{8}$:: N
8									ε_F $\varepsilon_{v_{tr}}$

4.3.3.2 Computing MatchPrase

In the description of the MG parsing algorithm so far, we left out the modifications needed for the computation of MatchPhrase. To also compute MatchPhrase, we must parse 36 candidates with the same phonetic content but different prosodic markup. For efficiency, we can compress the representation of these candidates into a prefix tree acceptor, rather than represent each prosodically marked-up candidate as a separate acceptor like (16a). Then we don't need to compute derivational steps shared among candidates multiple times.

For each of the 36 parses, we keep a running count of how many MatchPhrase violations there are. Any time we compute MatchPhrase, this count might increase. At each step in the parsing algorithm where a syntactic constituent could be formed—any derivational step where two expressions land or merge to be adjacent to one another—we check if the result of the Merge or Move step is a syntactic maximal projection. (If the result is DP_{erg} or DP_{abs}, it is not a syntactic maximal projection, but a post-syntactic one.) If the result is a syntactic maximal projection, then we compute MatchPhrase and increment the violation count if there is a violation.

Each derivational step illustrated in (41) is one where two expressions land or merge to be adjacent to one another. In Merge-1, a lexical head selects a complement. For example, in $\boxed{\text{A}}$, head *le* selects complement *mamanu* to form the syntactic DP *le mamanu*. The feature D is then recorded in the chart (Table 4.2) in cell (6,8) (see (37)), and a MatchPhrase computation is triggered. In Merge-2, a head merges with a specifier, e.g. in $\boxed{\text{H}}$, a D_{erg} head merges with specifier DP_{erg} *e le malini* to form a v_{tr}P, triggering a MatchPhrase computation. (For details of what is recorded in the chart in this and the following examples, see (37).) Merge-3 "launches" a moving element and thus doesn't place two expressions next to each other, but can result in the formation of a temporary merged complex that is a syntactic XP in the derived tree. For instance, in $\boxed{\texttt{C}}$, *lalaŋa* forms a VP with complement DP_{abs} *hi le mamanu*, before the DP_{abs} later moves out to leave a trace; this triggers a MatchPhrase computation. In Move-1, a constituent lands next to something else, which could result in the formation of an XP. In $\boxed{\texttt{E}}$, the EPP:D feature is checked as an empty ($\boxed{\texttt{9}}$) lands next to *hi le mamanu*. In this case, no constituent is formed because the result is not an XP.

Even if the MatchPhrase algorithm is called, the computation doesn't proceed in two cases: (i) if it is passed an empty span, or (ii) if it is passed a span from an initial to final state. In the first case, the expression is an empty category with no phonetic exponence. In the latter, the expression is the entire string, and thus maps to an intonational phrase, not a ϕ-phrase, so it is in the domain of MatchClause (3), not the domain of MatchPhrase. If the computation proceeds, then MatchPhrase, as defined in (5) and extended to post-syntactic material in (27), assesses violations only for ϕ-phrases. Thus, in our implementation, the only phonological constituents relevant for computation—as indicated by particular open–close bracket pairs— are listed in (42):

(42) Phonological phrase open–close bracket pairs
  a. Phonological phrase (PhP_{max}) Z . . . z
  b. Phonological phrase (PhP_{nmin}) Y . . . y
  c. Phonological phrase (PhP_{fxn}) X . . . x
  d. Phonological phrase (PhP_{min}) < . . . >

Abstracting away from dealing with post-syntactic material, we can state MatchPhrase as in (43):

(43) Definition of MatchPhrase in implementation, abstracting away from post-syntactic material
  a. If XP is a syntactic phrase that is not unary branching down to a terminal element, and if the yield of XP is exactly enclosed in one of the following open–close bracket pairs:
    i. < . . . > (PhP_{min})
    ii. X . . . x (PhP_{fxn})
    iii. Y . . . y (PhP_{nmin})
    iv. Z . . . z (PhP_{max})
  then assess no violations.
  b. If the yield of XP is not exactly enclosed in one of those bracket pairs, assess a violation.

This case-by-case implementation of MatchPhrase is possible because we place a finite bound on recursion in prosodic trees and label each level of recursion in ϕ-phrases with distinct open–close bracket pairs.

The complication added by treating post-syntactic material according to (27b) is that it is not enough to consider the yield of XP contained in span (i, j). On the one hand, we have to extend the span to be prefixed by the predecessor state of i and suffixed by the successor state of j, and then consider the yield of this extended span. If the two phonetic exponents at the edges in (i, j) are not post-syntactic, i.e. a case marker, but a case marker is adjacent to an edge, then MatchPhrase could be satisfied if we also include the case marker in the span. For instance, the enclitization of *e* to prosodic word *lalaŋa* forms a ϕ-phrase to repair a potential violation of MatchPhrase, see (24a).

On the other hand, we also have to shrink the span to range from the successor state of i to the predecessor state of j, and then consider the yield of this shrunken span. This is because the extension of MatchPhrase to post-syntactic material in (27) states that phonetic exponents that are post-syntactic are invisible to MatchPhrase. Thus, if a phonetic exponent at the edges of (i, j) is a case marker, then the open–close bracket pairs in the domain of the MatchPhrase computation enclose the span shrunken to exclude the case marker. For example, step $\boxed{\texttt{H}}$ in (40) results in the formation of a v_{tr}P with the span (2,8), spanning *e le malini hi le mamanu*. This is a syntactic XP, even though it is initiated by the the case marker *e*, so it triggers a MatchPhrase computation. Consider the prosodic parse Candidate 9, (28d). If we checked the alignment of the v_{tr}P edges to the open–close bracket pair around *e le malini hi le mamanu*, then there would be a MatchPhrase violation. The case marker *e* has been encliticized to *lalaŋa* to the left. However, if we shrink the span to be included in the open–close bracket pair to only *le malini hi le mamanu*, then we have a v_{tr}P mapping to a ϕ-phrase, and no MatchPhrase violation is assessed.

After computing the number of MatchPhrase violations for each of the 36 prosodic parses, we keep just the parses with the least number of violations—in this case, none, leaving just three candidates.

4.3.4 Final phonological transduction after MG parse: StrongStart

The remaining three prosodic parses, Candidate 7 (29b), Candidate 9 (28d), and Candidate 10 (29a)—all with no MatchPhrase violations—are passed back for a final `xfst` transduction implementing StrongStart. Unlike the other (purely) phonological constraints implemented (taking MatchPhrase to be an interface constraint), StrongStart is a multiply violable constraint and dominated by other constraints. It must be implemented in such a way that we can count the total number of violations in a candidate. The other phonological constraints are inviolable, and so it is not necessary to assess how many times they are each violated (for constraints that can be multiply violated): as long there is at least one violation in a candidate, that candidate will be filtered out. Following Karttunen (1998)'s implementation of multiply violated Parse-Syllable (see also Yu 2018), we define StrongStart as a "leniently composed" family of StrongStart$n$ constraints, each of which allow candidates only up to n violations. Note that this finite-state implementation can only handle counting up to a finite number of violations—or, put another way, it can only make a finite number of distinctions in well-formedness. Here we only need to count up to four, since the winning candidates have three violations and the losing candidate has four.

4.4 Discussion and conclusion

In this chapter, we demonstrated a proof-of-concept implementation of simultaneous (post-)syntactic and prosodic parsing, with syntactic parsing informed by prosody, and prosodic parsing informed by syntax. We showed how syntactic and post-syntactic structures could be derived by a composed Minimalist Grammar and parsed simultaneously bottom-up for a string of Samoan. We modeled the inclusion of a high-tone absolutive case marker as a post-syntactic item in the MG lexicon that entered the (post)-syntactic derivation like the segmental ergative case marker, alongside lexical items. In this way, a prosodic reflex of syntactic structure informed the syntactic parser. We also incorporated the computation of a syntax–phonology interface constraint into the (post-)syntactic parsing. The bottom-up MG parser computed the number of violations of MatchPhrase, a constraint penalizing the misalignment of prosodic and syntactic constituent edges for a set of candidate prosodic parses of the string. Thus, the (post-)syntactic analysis of the string was used to rank and filter the prosodic parses. For this first case study, we abstracted away from ambiguity in syntactic parsing, and thus did not model how prosodic parses could help disambiguate between a set of syntactic parses. However, the present model provides the scaffolding for such an extension—we leave this to future work.

For computational models of syntactic parsing, our work contributes to showing how prosodic information can be brought into syntactic parsing in other ways than as a local, noisy signal of syntactic domain edges. We showed how prosodic information could come in from spellout and via prosodic trees, and hope to inspire more work along these lines and more connections between the computational parsing and linguistic literature. We hope that consideration of how syntax might inform prosodic parsing can lead to a fresh and fruitful perspective on how prosody might inform syntactic parsing. If Wagner (2005; 2010), Steedman (2014), Hirsch and Wagner (2015), and others are (even only partly) right about the flexibility of syntax and the re-analysis of syntax–prosody mismatches as only apparent mismatches, then prosodic structure could be very informative indeed.

For linguistic theories of the syntax–phonology interface, our case study shows that constraint-based grammars of the interface defined on prosodic trees—and possibly with interface constraints interleaved between phonological constraints—can, in principle, be implemented. However, there's a caveat—the imposition of finite bounds. We were able to compose spellout and syntax in the MG grammar because all syntactic and post-syntactic structures were finite state definable over trees. Additionally, we assumed a finite depth of recursion in prosodic trees, and we assumed a finite bound on the number of violations of a constraint. The cap on the number of violations allowed us to define the constraint-based grammar using a finite-state implementation of optimality theory (Karttunen 1998). Without that cap, it's not clear how we would have been able to implement the grammar. The question of whether phonology needs unbounded recursion and whether optimality-theoretic phonology needs unbounded counting has been extensively discussed in the literature (see e.g. Frank and Satta 1998;

Karttunen 1998; Wagner 2005; 2010; Hao 2017). We leave to future work the question of how the finite bounds might be relaxed.

Carefully exploring the consequences of recursion in prosodic trees for computing the interface and parsing could also inform phonological perspectives on the role of recursion in prosodic structure. In §4.1.2.4, we set up a perspective where prosodic trees are recursive, following Selkirk (2011), among many others (e.g. Ladd 1986; Ito and Mester 1992; 2003; Féry and Truckenbrodt 2005; Wagner 2005; Krivokapić and Byrd 2012; Elfner 2012; Ito and Mester 2013; Kentner and Féry 2013; Myrberg 2013; Elfner 2015; Kügler 2015; Truckenbrodt and Féry 2015). However, there are other phonological perspectives where it is assumed that prosodic trees are not recursive—rather, they have a small, finite number of categories (e.g. Pierrehumbert 1980; Beckman and Pierrehumbert 1986; Nespor and Vogel 1986; Hayes 1989; Beckman 1996; Jun 2005ab; 2014). While the debate on recursion in prosodic structure has been informed by fieldwork, experiments, and theoretical analyses in a wide range of languages, mathematical and computational perspectives have not yet come into play.

More generally, extensions of the work here could help characterize the complexity of computing different kinds of syntax–phonology interface relations and how these relations might accordingly be restricted. Formalizing different proposals precisely enough to implement them would allow us to analyze them in ways that could be prohibitive with pen-and-paper analyses (Karttunen 2006b; 2006a; Bellik et al. (2015); Hulden (2017)). Further work could continue to probe the interaction of factored components of the syntax–phonology interface—not only spellout and relations between prosodic and syntactic trees, but also linearization, e.g. building on work by Richards (2010), Bennett et al. (2016), and Kusmer (2019). Our proposed extension of MATCHPHRASE to spellout in (27) could be examined further. The case study has shown how special treatment of spellout in the interface computation is necessary (under the defined constraint set) to compute optimal prosodic parses with bracketing mismatches like the enclitized case markers here. We could explore how the special treatment of post-syntactically derived and inserted material defined here would fare in characterizing other interface phenomenon defined with post-syntactic operations.

Finally, we can build on the work here to connect with empirical work on real-time prosodic/syntactic parsing (e.g. Beckman 1996; Brown et al. 2012). While the chart parser used here is not psychologically compatible, we can extend our model to incremental parsing models. And we can also eventually extend the model to include the computation of the map from the speech signal to the phonological parse.