A Synchronized Processing Model for the Syntax and Prosody of Swahili*

Leston C. Buell
bulbul@ucla.edu

April 19, 2002

Simultaneous processing. This paper will present a model for simultaneously processing syntax and prosody of a human language, exemplified by a subset of Swahili. The term 'simultaneous' implies two things:

a. The syntax and the prosody are two separate modules.

b. A sentence need not be (and perhaps cannot be) completely parsed by one of the two modules before being parsed by the other. The modules do not parse the sentence in serial fashion, but at the same time.

Let us first consider why we would want to assume that the syntax and prosody constitute two separate modules.

Syntactic/prosodic correspondences. It is often pointed out that certain prosodic boundaries appear to regularly correspond to a particular syntactic boundary. A clear example of this occurs in Zulu, in which the right edge of a verb phrase coincides with the right edge of an intonational phrase. The effects of this boundary are clearly seen in a phenomenon called 'prepausal lengthening':

(1) Prepausal lengthening in Zulu

a. Ku- zo- funda uSi:pho.$_{VP}$] (subject focus, expletive subject)
17.subj- fut- read Sipho
'SIPHO will read.'

b. U- zo- fu:nda $_{VP}$] uSipho.
1.subj- fut- read Sipho
'Sipho will read.'

[0]I am most grateful to Ed Stabler, without whose guidance and support this paper would not have been possible. Thanks also go to Greg Kobele for his valuable input.

c. U- zo- funda incwa:di $_{VP}$] uSipho.
1.subj- fut- read book Sipho
'Sipho will read a book.'

d. USipho u- zo- funda incwa:di $_{VP}$]
Sipho 1.subj- fut- read book
'Sipho will read a book.'

e. Incwadi i- zo- fun:dwa $_{VP}$]
book 1.subj- fut- read.passive
'The book will be read.'

These sentences show that the words *uSipho*, *fund(w)a*, and *incwadi* undergo lengthening only when they occur to the immediate left of the VP boundary. (The utterance-final word is also lengthened, but this is not relevant here and has not been indicated.) Thus we have a syntactic boundary (Verb Phrase) which 'corresponds' to a prosodic boundary (Intonational Phrase).

It is because of such regular points of correspondence between syntactic and prosodic structures that some linguists have come to believe that the prosody of a language is read directly off the syntactic tree. However, this view faces some very real problems, one of which is the phenomenon of bracketing paradoxes, such in (2):

(2) a. the cat that chased the rat that stole the cheese

b. syntactic bracketing:
[I saw [$_{DP}$ the cat that chased [$_{DP}$ the rat that stole the cheese]]]

c. prosodic bracketing:
[[I saw the cat] [that chased the rat] [that stole the cheese]]

In (2), the prosodic brackets contradicts the syntactic brackets, cutting across complex noun phrases.

One way of handling an inconsistency between prosodic and syntactic bracketing of the type found in (2) is to force the syntax to conform to the observed prosodic bracketing. But this approach can lead to far-fetched syntactic structures. As an example, it has been proposed that pitch accent falls on the most deeply constituent in the syntactic structure (cite?). Although this can be made to work for many types of sentences, there are cases (discussed in Gussenhoven (cite?)) where the pitch accent falls on an element which is quite implausibly the most deeply embedded constituent. Consider the following example:

(3) a. Why did you call the fire brigade? The house isn't ON fire.

b. A: Why doesn't she do some work?
B: She doesn't have any work TO do.

In these two examples, termed 'non-counterassertive' by Gussenhoven, to maintain that the pitch accent falls on the most deeply embedded constituent, one would have to claim that 'fire' and 'do' are right-dislocated, which seems entirely implausible.

Separate modules, shared indices. Because of the bracketing paradoxes inherent in reading the prosody directly off the syntactic tree, we will here take the view that the syntax and the prosody are autonomous modules, operating simultaneously, in tandem. However, we must not forget the regular prosodic/syntactic correspondences of the type shown for prepausal lengthening in Zulu. Whatever the precise mechanism by which such correspondences come about, a realistic processing model must provide a way to express these correspondences. In the model presented in this paper, this will be achieved with shared indices which situate both prosodic and syntactic constituents in the input string.

Simultaneity. Given the assumption that the syntax and the prosody are separate modules, the idea that these two modules must operate in non-serial fashion is easy to illustrate. Consider in the sentence in (4):

(4) * The will have dancing the jig, the minuet, and the gavotte.

In this example, it is clear that the listener does not need to complete the prosodic parse of the sentence (and thus wait until the end of the sentence) to realize that there is no possible syntactic derivation for the sentence.

Similarly, consider the sentence in (5), which employs the reduced form *'em* for the pronoun *them*:

(5) * It's 'em who will be dancing the jig, the minuet, and the gavotte.

In this example, it is clear that we do not need a complete syntactic parse (and thus again waiting until the end of the sentence) to realise that the sentence is prosodically ill-formed. Since (4) shows that the prosody cannot happen before the syntax and since (5) shows that the opposite also cannot be true, we conclude that the syntax and prosody must be processed simultaneously.

Formal properties of the language faculty. The notion of simultaneous processing of separate linguistics modules assumed here poses certain questions about the formal properties of the language faculty. The computational literature models linguistic modules as various kinds of machines, such as Pushdown Automata (which accept Context Free Grammars (CFGs)) for phonology or Minimalist Grammar (ML) acceptors for syntax. If the linguistic system in its entirety is the intersection of these various machines, it is not clear what the properties of this intersection machine would be, for beyond the realm of finite state automata (FSAs), most types of machines (such as Pushdown Automata) are not closed under intersection. This is problematic because there is no known polynomial-time strategy for recognizing Context Sensitive Languages (CSLs). However, as explored in (Boullier 1999), parsing a string separately with multiple machines does not result in a parse time greater than the order of the highest-order machine. This formal situation provides some further support for the linguist's intuition that syntactic analysis and prosodic analysis are separate 'modules' of linguistic competence, and it is this idea that will be utilized in the implementation of the model presented in this paper. Syntax and prosody will be represented as distinct machines sharing a lexicon. The two machines will run simultaneously, and a derivation will crash upon the failure of either of the two machines.

Modeling Swahili. This paper will describe an implementation of a model in which the prosody and syntax modules parse sentences in parallel, rather than in serial fashion, and in which the resulting syntactic and prosodic derivations are checked against each other to ensure that they correspond. I will call the process of checking the parses against each other 'synchronization'. The language to be modeled is a subset of Swahili–simple root clauses with transitive verbs. Let the sentence (6) serve as an example:

(6) Ni- li- ku- la kitunguu.
1s.subj- past- epenth- eat onion
'I ate an onion.'

While superficially simple, a sentence such as (6) is not trivial from a parsing perspective. The prosodic structure of the verb form is complex. Furthermore, the Swahili language has two distinct morphemes *li*,[1] one an auxiliary (as in this sentence) and another an object marker. These two prefixes are prosodically distinct with respect to the distribution of the epenthetic prefix *ku*, which appears in certain environments on monosyllabic verbs. Thus, combining the object marker *li* with the monosyllabic root *la* 'eat', it is not possible to insert epenthetic *ku*, as shown in (7a), while when combining the past tense auxiliary *li* with the same root *la*, such epenthesis is obligatory, as shown in (7b):

(7) a. Object marker *li*:

ni- na- (*ku-) li- (*ku-) la
1s.subj- pres- epenth- 5.obj- epen- eat

'I'm eating it'

b. Past auxiliary (tense marker) *li*:

ni- li- *(ku-) la
1.s-subj- past- epenth- eat

'I ate'

Given this ambiguity, in parsing sentence (6), the prosody must know to form a word using the auxiliary morpheme rather than the object marker morpheme.

Original analyses of Swahili syntax and prosody will be employed in this paper. The syntactic analysis uses remnant movement and no head movement and will be formulated as a Minimalist Grammar (MG). The prosody employs a variant of the Prosodic Hierarchy (Selkirk 1984), which will be formulated in a grammar equivalent in power to a Regular Grammar. The model is implemented in Prolog, as reviewed in Appendix B in this paper.

To sum up, the three most important properties of the model to be presented are these:

a. The prosody and the syntax are separate modules which process the input string simultaneously.

[1] There are actually more than two, but we will consider just two of them.

b. A process of synchronization will ensure that a syntactic parse corresponds to a prosodic parse.

c. Indices which situate the prosodic and syntactic constituents in the input string will be shared between the prosody and the syntax, making it possible express correspondences between prosodic and syntactic boundaries.

The paper will be organized as follows:

1. Analysis and formalization of Swahili prosody,
   which also incorporates some elements of morphophonology.
2. Analysis and formalization of the Swahili syntax.
3. Integrating the prosody and the syntax.
4. Discussion and conclusion.

Throughout the paper the input string will be assumed to be preparceled into morphemes in such a way that a string such as *ninakisoma kitabu* is fed to both the prosody and the syntax as the morpheme list [ni,na,ki,soma,kitabu]. Each of the morphemes in this list will be referred to as a'word of input'.

1 Swahili Prosody

1.1 Prosodic Analysis

The prosodic module of our model will consider only that aspect of the prosody which groups strings of morphemes into larger units such as words and phrases. Intonation will not be considered. As such, the function of the prosodic module is to correctly group a string of morphemes into words and phrases. This task is not trivial. For instance, function words have prosodic properties different from those of content words with respect to the formation of Clitic Phrases, and affixes must be made sure to attach to the correct side. The framework to be adopted here is that of Lexical Phonology (Selkirk 1984). Within this framework, at least five prosodic domains seem justifiable in Swahili: Prosodic Root (Rt), Prosodic Stem (Stm), Prosodic Word (Wrd), Clitic Phrase (Clt), and Prosodic Sentence (Snt).[2] For the rest of this paper, these will be referred to as prosodic categories. Abstracting away from conditions involving feature values and from the phenomenon of epenthetic prefixes, the general structure of the prosodic grammar is as follows:

(8) Rt → LexItem
Stm → LexItem* Rt
Wrd → Stm LexItem*
Clt → Wrd*
Snt → Clt*

Although the grammar has both left- and right-branching rules, the language it generates is actually regular because of the strict hierarchy of the categories–no right-hand side of a rule contains a category of an order higher than the left-hand side.

[2]For extensive argumentation for the sub-word prosodic domains in Nguni languages, see (Downing 1992).

For ease in formulation, the rules will be written as at most binary branching context free rules rather than with Kleene stars as in (8). Thus, a rule like Stm → LexItem* Rt will be rewritten as the two rules Stm → Rt, Stm → LexItem Stm. This does not imply, however, that the intermediate structure which the second notation imposes is in any sense linguistically real. For this example, then, we are not claiming that a Stm can actually be embedded within another Stm, like this: [Stm [Stm [Stm]]]. Such structures are to be interpreted as if they had been generated by the rules in (8), with no embedding of same-order constituents: [Stm Stm Stm].

The prosody is presented as a Context Free Grammar, and context free rewrite rules are given for each domain. A word must be said about how the lexicon is to be represented and how the rewrite rules are to be interpreted. From the standpoint of the prosody,[3] lexical items have a pronunciation, the prosodic domain (prosodic category) *LexItem*, and a set of features:

(9) $< V, \text{LexItem}, [F_1 \ldots F_n] >$

The rewrite rules will omit the pronunciation and are written in the following form:

(10) $Cat_1[F_1 \ldots F_i] \rightarrow Cat_2[G_1 \ldots G_2]\ Cat_3[H_1 \ldots H_k]$

Concatenation of the pronunciations is implied, such that the pronunciation of the expression on the left-hand side of the rule is the concatenation of the pronunciation of all of the expressions on the right-hand side in the order in which they occur:

(11) $< V_1 + V_2, Cat_1, [F_1 \ldots F_i] > \rightarrow < V_1, Cat_2, [G_1 \ldots G_2] >< V_2, Cat_3, [H_1 \ldots H_k] >$

Now that these points have been addressed, we are ready to examine each prosodic domain in detail.

Prosodic Root (Rt). The justification for the Prosodic Root domain is weaker in Swahili than in some other Bantu languages, such as the Nguni languages. The evidence for this tier lies in the fact that the passive form of a monosyllabic verb root is augmented by a vowel (in both Swahili and the Nguni languages). This is taken to mean that a phonological constraint of minimality applies to the passive form on the Prosodic Root tier, which can be argued to be distinct from the next level up in the prosodic hierarchy, the Prosodic Stem, because the augmentation strategy for these passive forms is distinct from that used to satisfy another minimality constraint operating on the Prosodic Stem tier (Downing 1992).

This model will not concern itself with any phonological phenomena associated with the Prosodic Root tier, but the tier will be used to identify a lexical item which can serve as the nucleus of a Prosodic Stem. The Prosodic Root being the lowest tier in our hierarchy, it is composed solely of lexical items and is derived by the following rule, where X, Y are the values + or −:[4]

[3] The lexicons needed for the prosody will first be characterized as independent. Their integration into a single shared lexicon will be discussed in the section on synchronization.

[4] While rule (12) works for our somewhat simplified model of Swahili morphology and for the limited types of data to be parsed, a more realistic model would need to combine multiple lexical items (such as a verb stem, the passive suffix, and a final mood vowel) to form a Prosodic Root.

(12) Rule *rt*:

Rt[α sylls, γfunc] $\rightarrow$ LexItem[α sylls, γfunc, +root]

This rule identifies as a Prosodic Root lexical items which have been identified as possible Prosodic Roots by virtue of being endowed with the feature [+root]. All content morphemes have the feature [+root], as do certain auxiliaries. Morphemes which are [−root] are the remaining auxiliaries, all function words, clitics, agreement markers, and suffixes. A [+root] lexical item is also marked for the feature [±func], which designates it as a functional ([+func]) or content ([−func]) morpheme. All [−root] lexical items are [−func].[5] This feature percolates up to the Prosodic word domain, since the Clitic Phrase domain is sensitive to the content/functional status of its components. Lexical Items which are [+root, +func] in our tiny lexicon are a subset of the tense markers (auxiliaries) and the word *na* 'and, with'.

Note that the number of syllables in the lexical item is percolated to the Root. This syllable count will utilized in a condition on the application of the rule forming Prosodic Stems, the next higher prosodic tier.

Prosodic Stem (Stm). The Prosodic Stem is the tier on which a different process of augmentation takes place. On this tier, a prefix *ku-* is added if needed to make this prosodic constituent disyllabic, if it is verbal.[6] In (13), the Prosodic Stem is already disyllabic and hence no *ku-* is added.

(13) a- la, *akula
3s- eat
'he eats'
/a + la/ $\rightarrow$ [$_{Stm}$ a [$_{Rt}$ la]]

But in (14), the second Prosodic Stem is monosyllabic, and a *ku-* must be added to obey the constraint on minimality.

(14) a- li- ku- la, *alila
3s- past- *ku-* eat
'he ate'
/a + li + la/ $\rightarrow$ [$_{Stm}$ a [$_{Rt}$ li]] [$_{Stm}$ ku [$_{Rt}$ la]]

In our prosodic grammar, Prosodic Stems are derived by the following rules, where γ takes the values $+, -$ and α is a number:[7]

(15) a. Rule *stm*1:
Stm[α sylls, γfunc] $\rightarrow$ Rt[α sylls, γfunc]
Condition: $\alpha \geq 2$

[5] This seems unintuitive, but formulating func in this way will simplify implementation. In the implementation, binary features will be expressed as privitive features. The func feature is only ever evaluated on [+root] items, so inclusion of a func feature on [-root] items would be redundant.

[6] There are also augmentation processes on nouns, all of which involve prefixation and which vary according to noun class. It is not obvious what prosodic tier these processes take place.

[7] Formulating the rules as a Context-Free Grammar creates unwanted structure. This fact is discussed below.

b. Rule *stm*2:
Stm[$\alpha + \beta$ sylls, γfunc] $\rightarrow$
LexItem[α sylls, +prefix, -root] Stm[β sylls, γfunc]
Condition: $\alpha + \beta \geq 2$

This rule attaches any number of prefixes which are themselves unsuitable as Prosodic Root nuclei to the left of a Prosodic Root to form a Prosodic Stem. As described in the previous domain (Prosodic Root), the [αfunc] feature percolates up for use in deriving Clitic Phrases. The [αroot] feature will not be used above this tier and can be discarded.

As described in the justification for the Prosodic Stem domain in the previous section, we must ensure that each verbal Prosodic Stem is at least disyllabic, and that we accept an epenthetic *ku* if it isn't:[8]

(16) a. $[_{Stm}$ a $[_{Rt}$ na]] $[_{Stm}$ *ku* $[_{Rt}$ la]]
b. * $[_{Stm}$ a $[_{Rt}$ na]] $[_{Stm}$ $[_{Rt}$ la]]
(Bad because the second Stm is not at least disyllabic.)

(17) a. $[_{Stm}$ a ka $[_{Rt}$ la]]
b. * $[_{Stm}$ a ka *ku* $[_{Rt}$ la]]
(Bad because the Stm would have been disyllabic even without inserting *ku*.)

Epenthetic *ku* will be treated exactly like a lexical item, with the features [+prefix, -root]. In the syntax, this item will be accomodated by allowing an item with no syntactic features to be incorporated into the tree. *Ku*, then, is a sort of mirror image of a syntactic empty category. While the latter has syntactic features and no phonetic features, *ku* has phonetic features but no syntactic features.

(18) Rule *stm*3:
Stm[$\alpha + \beta$ sylls, γfunc] $\rightarrow$
LexItem[α sylls, +prefix, -root] Rt[β sylls, γfunc]
Condition: $\alpha + \beta \geq 2$

This rule introduces an undesired ambiguity into the system, as non-monosyllabic Roots can now be incorporated into a Stem either by merging it directly with a lexical item using the rule in (1.1), or by first promoting the Root into a Stem using rule (15a) and then combining it with a lexical item using rule (15b). We will not concern ourselves with this ambiguity here. The latter derivation is presumably blocked by a constraint *STRUC (CITE).[9] The rule also overgenerates, since it will generate stems with epenthetic prefixes when no such epenthesis is needed. Such forms are assumed to be blocked by a constraint on epenthesis *EPEN.

[8] To account for a full range of data, the constraint would actually need to be a bit more finessed than this, since a non-syllabic nasal object prefix seems to satisfy minimality: *nili[kupa]* 'I gave', *nili[mpa]*, **nili[kumpa]* 'I gave him'.

[9] Of course, the syllabic minimality condition on the Stem rewrite rules is also best thought of as a constraint.

Prosodic Word (Wrd). The Prosodic word corresponds to Myers's (1987) 'syntactic word'. This is the tier on which all suffixes are attached to the Prosodic Stem and is necessarily distinct from that of the Prosodic Stem because such suffixes are unable to satisfy the minimality constraint on Prosodic Stems:

(19) A- na- *(ku-) la- je?
1.subj- pres- epenth- eat- how
'How does he eat?'

In our grammar, Prosodic Words are derived by the following rules, where $\gamma \in \{+, -\}$:

(20) a. Rule *wrd*1:
Wrd[γfunc] $\rightarrow$ Stm[γfunc]

b. Rule *wrd*2:
Wrd[γfunc] $\rightarrow$
Wrd[γfunc] LexItem[+suffix, -root]

This rule attaches any number (including zero) suffixes to the Prosodic Stem.[10] This is the highest tier to which the [αfunc] feature needs to percolate. The [αsuffix] and [αroot] features of the attaching suffixes are no longer needed and can be discarded. Examples (21), (22), and (23) are examples of Prosodic Words.

(21) shule
9.school
'a school'
/shule/ $\rightarrow$ [$_{Wrd}$ [$_{Stm}$ [$_{Rt}$ shule]]]

(22) shule- ni
9.school- loc
'at school'
/shule + -ni/ $\rightarrow$ [$_{Wrd}$ [$_{Stm}$ [$_{Rt}$ shule]] ni]

(23) a- soma- ye
3s.subj- read- 1.rel
'one who reads'
/a + soma + ye/ $\rightarrow$ [$_{Wrd}$ [$_{Stm}$ a [$_{Rt}$ soma]] ye]

The syllable count information is not needed on the Prosodic Word tier or on any higher tier. So, it can be discarded.

[10] In a more sophisticated model of Swahili morphology, a distinction would need to be made between derivational suffixes, which probably attach at the Prosodic Root level, and those which attach at the Prosodic Word level discussed here.

Clitic Phrase (Clt). The Clitic Phrase, which corresponds to Myers's 'phonological word' or 'Dokean word', is the domain of primary stress, which attaches any number of functional Prosodic Words to a non-functional Prosodic Word on their right. The Clitic Phrase is derived by the following rule:

(24) a. Rule *clt*1:
Clt → Wrd[−func]

b. Rule *clt*2:
Clt → Wrd[+func] Clt

The Clitic Phrase is the tier at which the func feature, which has percolated up from the Prosodic Root tier, is utilized. This feature is no longer needed and can be discarded.[11] Example (25) gives a somewhat complex example demonstrating how two functional prosodic Words, one whose root is the conjunction *na* 'and' and the other by the auxiliary *li*, are attached to a non-functional verbal Prosodic Word to their right:

(25) na ni- li- ki- soma
and 1s.subj- past- 7.subj- read
'and I read it'
/na + ni + li + ki + soma/ →
$[_{Clt}$ $[_{Wrd}$ $[_{Stm}$ $[_{Rt}$ na]]] $[_{Wrd}$ $[_{Stm}$ ni $[_{Rt}$ li]]] $[_{Clt}$ $[_{Wrd}$ $[_{Stm}$ ki $[_{Rt}$ soma]]]]

Prosodic Sentence (Snt). The Prosodic Sentence is the highest domain in this model, consisting of any number of Clitic Phrases. The domain is derived by the following rule:

(26) a. Rule *snt*1:
Snt → Clt

b. Rule *snt*2:
Snt → Snt Clt

An example of a Prosodic Sentence is given in (27), ignoring domains lower than the Clitic Phrase:

(27) Juma a- li- soma kitabu shule- ni.
3s.subj- past- read 7.book 9.school- loc
'Juma read a book at school.'

$[_{Snt}$ $[_{Clt}$ Juma] $[_{Clt}$ alisoma] $[_{Clt}$ kitabu] $[_{Clt}$ shuleni]]

[11]The model as described must attach word-initial functional morphemes such as *na* 'and, with' either as Prosodic Words on the Clitic Phrase tier, or as prefixes on the Prosodic Stem tier. Both of these approaches go against the intuition that they are instrinsically stressless (i.e. [−root]) function words. However, there are no obvious empirical differences between these three possibilities, and we will therefore not consider this third possibility in our simple model.

Prosodic Grammar Recap The structure-building rewrite rules of our prosodic grammar are listed here for reference:

(28) Prosodic Grammar

a. Root: Rule *rt*:
Rt[α sylls, γfunc] $\rightarrow$ LexItem[α sylls, γfunc, +root]

b. Stem:

i. Rule *stm*1:
Stm[α sylls, γfunc] $\rightarrow$ Rt[α sylls, γfunc]
Condition: $\alpha \geq 2$

ii. Rule *stm*2:
Stm[$\alpha + \beta$ sylls, γfunc] $\rightarrow$
LexItem[α sylls, +prefix, −root] Stm[β sylls, γfunc]
Condition: $\alpha + \beta \geq 2$

iii. Rule *stm*3:
Stm[$\alpha + \beta$ sylls, γfunc] $\rightarrow$
LexItem[α sylls, +prefix, −root] Rt[β sylls, γfunc]
Condition: $\alpha + \beta \geq 2$

c. Prosodic Word:

i. Rule *wrd*1:
Wrd[γfunc] $\rightarrow$ Stm[γfunc]

ii. Rule *wrd*2:
Wrd[γfunc] $\rightarrow$ Wrd[γfunc] LexItem[+suffix, −root]

d. Clitic Phrase:

i. Rule *clt*1:
Clt $\rightarrow$ Wrd[$-$func]

ii. Rule *clt*2:
Clt $\rightarrow$ Wrd[+func] Clt

e. Sentence:

i. Rule *snt*1:
Snt $\rightarrow$ Clt

ii. Rule *snt*2:
Snt $\rightarrow$ Snt Clt

1.2 Fused Morphemes

The phonological facts and anlysis just presented will be enriched to handle cases of unpredictable fusion of IP-region heads, specifically, fusions of the subject marker prefixes with the present tense marker *a*: /ni + a/ $\rightarrow$ [na] (first person singular), /a + a/ $\rightarrow$ [a] (third person singular), etc.

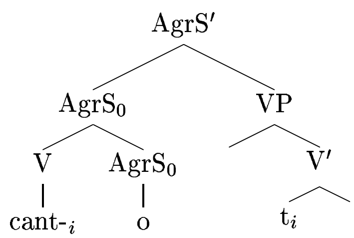
(29) na- imba, * ni- a- imba
1s.subj=pres- sing 1s.subj- pres- sing
'I sing'
/ni + a/ → [na]

For the purposes of the formalization, fusions such as this will be treated exactly like lexical items in the phonology, thus, no new rules will be required to accomodate these items. In the syntax, these fusions will associated with multiple tuples of syntactic features, one for each of the morphemes fused.

2 Syntax

Assumptions. The syntactic analysis employed assumes Kayne's Linear Correspondence Axiom (Kayne 1994) which correlates linear precedence with c-command. One of the consequences of the LCA is that all head movement attaches the moved head to the left of the head to which it adjoins. Hence, since subject agreement is standardly assumed to be higher than the verb, for a verb and AgrS head to comprise a single complex head, the LCA says that the verb must precede the AgrS head, as in Spanish:

(30) cant- o
sing- 1s
'I sing'



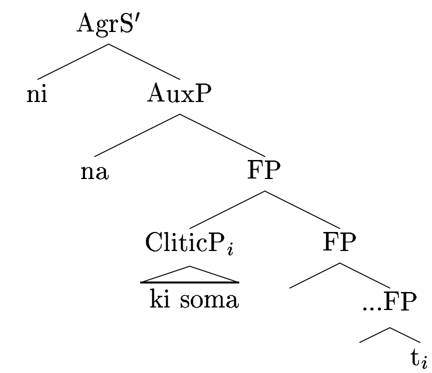
However, Swahili exhibits the opposite order, with subject agreement preceding the verb:

(31) ni- na- soma
1s.subj- pres- read
'I read'

Assuming the LCA thus precludes analysing Swahili verb forms as complex heads, because the morphemes in these forms would occur in an order which contradict what is widely accepted to be the hierarchy of functional projections in the IP region. Swahili tense and aspect morphemes would also be predicted to follow the verb stem, contrary to the attested order in Swahili.

Thus, in order to maintain the LCA, an alternative analysis is adopted which does not employ head movement of the Swahili verb. Instead, the heads of the IP region (AgrS, Aux), are in situ, while the AgrO and V heads come to immediately follow the IP region by virtue of moving in a VP remnant to a position just under AuxP.

(32)



There are reasons other than compatibility with the LCA which motivate such an analysis, which are discussed at length in (Buell 2002). These include the fact that the tense markers have some of the properties of auxiliary verbs, that at the subword level the IP-region heads form a prosodic domain distinct from that of the VP-region heads, and that native speakers sometimes write the IP-region heads together as a separate word.[12] (Using the latter two facts as evidence for syntactic constituency assumes a close correspondence between prosody and syntax.)

Analysis. To illustrate the analysis, let us consider the following simple matrix clause:

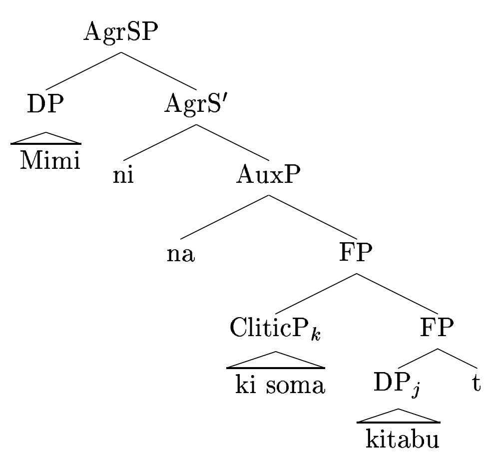
(33)

Mimi	ni-	na-	ki-	soma	kitabu.
me	1s.subj-	pres-	7.obj-	read	7.book

'I'm reading the book.'

Several conservative assumptions are made in the analysis of this sentence. The preverbal subject *mimi* is assumed to be generated within VP and to move to its surface position (Koopman and Sportiche 1988). The agreement observed between the object marker (*ki*) and the object is assumed to be triggered by a spec/head relationship. The final tree for this sentence is shown in (34):

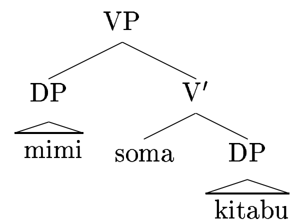
(34)



[12]This happens not so much with simple verbs but with synthetic relatives, which are morphologically identical to verbs, but for one additional infixed morpheme.

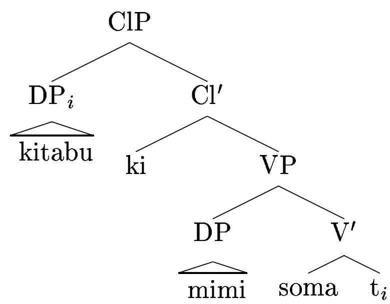
To illustrate how this tree is derived, we begin with the following structure for the Verb Phrase:

(35)



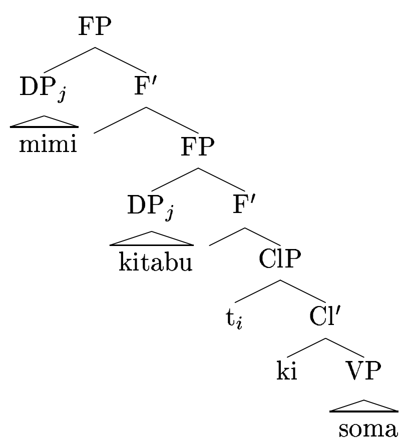
Agreement between the object and the preverbal object marker is established in a spec/head relationship in a Clitic Phrase (ClP) headed by the object marker:

(36)



Since *kisoma* will need to move, leaving the object *kitabu* behind,[13] and since non-maximal projections are assumed to be unmovable, the object must be moved out of its specifier to a functional projection. The subject will move to a functional projection on top of that one:

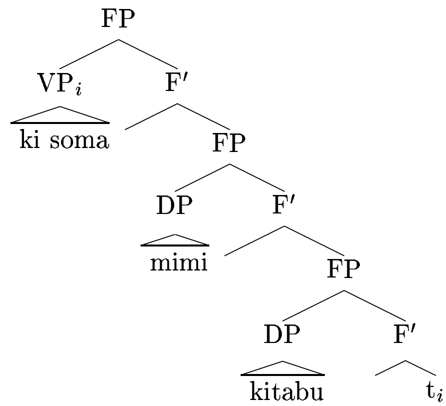
(37)



Now the ClP remnant moves to a functional projection, giving the illusion that the verb has head-moved:

[13] For argumentation on why agreeing objects in Bantu languages are assumed to move out of VP, see (Woolford 2000).

(38)



Merging the Aux and AgrS heads and moving the subject to spec-AgrS yields the final tree given above in (34).

Formalizing the syntax. This analysis can be formalized as a Minimalist Grammar without head movement. The following definition of a Minimalist Grammar is taken from (Stabler and Keenan 2000):

(39) **Definition of Minimalist Grammar** (Stabler and Keenan 2000)
A Minimalist Grammar is a generalized grammar $MG = \langle V, Cat, Lex, \mathcal{F} \rangle$, where

a. V is a finite set, the *pronounced elements*,

b. Cat is the disjoint union of the following sets:

- *base*, a non-empty finite set.[14]
- $selector = \{\texttt{=}b | b \in base\}$
- $licensee = \{\texttt{-}b | b \in base\}$
- $licensor = \{\texttt{+}b | b \in base\}$

c. Lex, a finite set of lexical items, is a subset of

$$V^* :: (selector + (selector + licensor)^*)^? base\ licensee^*$$

d. $\mathcal{F} = \{merge, move\}$ is a set of partial functions from tuples of expressions to expressions.[15] Expressions $E =_{def} V^*\{:, ::\}Cat^*$. Each expression has a type $\cdot \in \{:, ::\}$ that indicates whether it is *simple* (::) or *complex* (:).[16]

[14] A *base* is the feature selected by a selector and in traditional terms can be thought of as a lexical item's categorial feature, such as 'noun', 'verb', or 'preposition'.

[15] More informally, the merge and move functions can be described as follows:

*merge*1	–	Merges a head.
*merge*2	–	Merges an XP which will not move.
*merge*3	–	Merges an XP which will move.
*move*1	–	Moves an XP which will not move again.
*move*2	–	Moves an XP which will move again.

[16] The *simple* expressions correspond to simple heads, which here means all heads, since there is no head movement. The *complex* expressions are everything else.

- *merge* is the union of the following three functions:
  for $\sigma, \tau \in V^*$, $\gamma \in Cat^*$, $\delta \in Cat^+$, $\cdot \in \{:, ::\}$,

$$\frac{\sigma :: \texttt{=}x\gamma, \phi_1, \ldots, \phi_n \quad \tau \cdot x, \psi_1, \ldots, \psi_m}{\sigma\tau : \gamma, \phi_1, \ldots, \phi_n, \psi_1, \ldots, \psi_m}\ merge1$$

$$\frac{\sigma : \texttt{=}x\gamma, \phi_1, \ldots, \phi_n \quad \tau \cdot x, \psi_1, \ldots, \psi_m}{\tau\sigma : \gamma, \phi_1, \ldots, \phi_n, \psi_1, \ldots, \psi_m}\ merge2$$

$$\frac{\sigma \cdot \texttt{=}x\gamma, \phi_1, \ldots, \phi_n \quad \tau \cdot x\delta, \psi_1, \ldots, \psi_m}{\sigma : \gamma, \phi_1, \ldots, \phi_n, \tau : \delta, \psi_1, \ldots, \psi_m}\ merge3$$

- *move* is the union of the following two functions:
  for $\sigma, \tau \in V^*$, $\gamma \in Cat^*$,$\delta \in Cat^+$, and every ϕ_i has a different first feature,[17]

$$\frac{\sigma : \texttt{+}x\gamma, \phi_1, \ldots, \phi_{i-1}, \tau : \texttt{-}x, \phi_{i+1}, \ldots, \phi_n}{\tau\sigma : \gamma, \phi_1, \ldots, \phi_{i-1}, \phi_{i+1}, \ldots, \phi_n}\ move1$$

$$\frac{\sigma : \texttt{+}x\gamma, \phi_1, \ldots, \phi_{i-1}, \tau : \texttt{-}x\delta, \phi_{i+1}, \ldots, \phi_n}{\sigma : \gamma, \phi_1, \ldots, \phi_{i-1}, \tau : \delta, \phi_{i+1}, \ldots, \phi_n}\ move2$$

The language generated by a Minimalist Grammar G, $L(G)$, is the closure of the lexicon under the functions *merge* and *move*. For any $b \in base$, the string language of G at b is $\{\sigma | \sigma \cdot b \in L(G)\}$, and is denoted $L_b(G)$. If the first argument to *merge* is a simple expression (one of type ::), the second argument to *merge* is a complement. Otherwise, the second argument is a specifier. Note that though an expression can have at most one complement, there is no restriction on the number of specifiers it may have (other than finitude).[18]

2.0.1 A Minimalist Example

The MG $G = \langle V_G, Cat_G, Lex_G, \mathcal{F}_G \rangle$ below generates the sentence *mimi ninasoma* 'I am singing' of type *agrs*, by the derivation shown in (40).

(40) $Lex_G =$

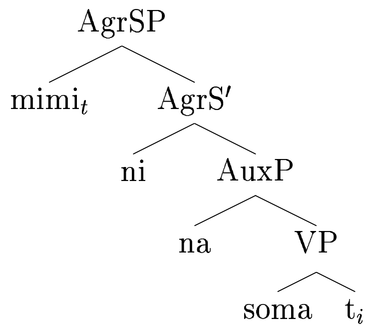
E_0 = *soma* :: =d v
E_1 = *mimi* :: d -subj
E_2 = *na* :: =v aux
E_3 = *ni* :: =aux +subj agrs

1. merge3(E_0, E_1) = E_4 = soma : v, mimi : -subj
2. merge1(E_4, E_2) = E_5 = na soma : aux, mimi : -subj
3. merge1(E_3, E_5) = E_6 = ni na soma : +subj agrs, mimi : -subj
4. move1(E_6) = E_7 = mimi ni na soma : agrs

This derivation corresponds to the following traditional syntactic phrase marker:

[17] This is the Shortest Move Constraint (SMC). The SMC does not come into play in any of the derivations in this paper.
[18] Multiple specificers are not used in the analysis of Swahili presented here.

(41)



3 Integrating the Prosody and the Syntax

As described in the introduction, the goal here is not to intersect the prosodic and syntactic machines, for it is not clear what the properties of this machine would be. Rather, we will restructure the lexicon in a way that it can be shared by the two machines. Then we will explore how the positional indices used by the two parsers can be shared. Sharing the lexicon will allow synchronization by providing a way to identify syntactic and prosodic parse pairs which use the same lexical items, while sharing the indices provides a way to express correspondences between syntactic and prosodic constituents. As will be seen, different types of lexical entries, such as empty categories and epenthetic material require special consideration.

Further complications arise in chart-based parsing, which is used in the implementation. These involve index-sharing between modules and interdependencies between certain types of axioms.

3.1 Sharing the Lexicon

The synchronization process to be described will require the lexicon to be shared between the prosody and the syntax. We have said that the lexicon used by the prosody contains lexical items consisting of a pronunciation, the prosodic category *LexItem* and a set of features, while the syntactic lexicon consists of a pronunciation, the :: ('simple expression') symbol, and a list of *Cat* symbols, as described in (39). In this section we will describe how these module-specific lexical entries can derived from a shared lexicon for the four types of lexical items needed for the subset of Swahili grammar we are considering here.

Canonical lexical entries. We will use the term 'canonical' in this context to refer to pronounced lexical items which have both prosodic features and a single list of syntactic features. We want the prosody and the syntax to use the same pronunciation for a given item. Furthermore, the *LexItem* and :: symbols are both entirely predictable. Calling the prosodic feature set *PFeatures* and list of syntactic features *SFeatures*, a pronounced, monomorphemic item in our shared lexicon can be written as a tuple of the following form:[19]

[19]Note that this format restricts the lexicon to items whose pronunciation consists of at most one 'pronounced element', a restriction absent in the definition of a Minimalist Grammar.

(42) Shared lexical entry
$< V, PFeatures, SFeatures >$

From the shared lexical entry, both prosodic and syntactic lexical items can be derived:

(43) Canonical case

a. Shared lexical entry:
$< V, PFeatures, SFeatures >$

b. Prosodic lexical entry:
< V, LexItem, $PFeatures$ >

c. Syntactic lexical entry:
$V :: SFeatures$

(44) Canonical case example

a. Shared lexical entry:
< *soma*, [2 sylls, +root], [=dobj, =d, v] >

b. Prosodic lexical entry:
< *soma*, LexItem, [2 sylls, +root] >>

c. Syntactic lexical entry:
soma :: =dobj =d v

Empty categories. To account for 'empty categories', which are lexical items without a pronunciation or prosodic features, we can allow the V to be the empty string, and the PFeatures to be an empty list. Such a lexical item is subject to a single rewrite rule which generates the syntactic lexical item:

(45) Empty category

a. Shared lexical entry:
$< \epsilon, PFeatures, SFeatures >$

b. Prosodic lexical entry:
None.

c. Syntactic lexical entry:
$:: SFeatures$

(46) Empty category example

a. Shared lexical entry:
< ϵ, [ ], [d,−subj, −fpssubj] >

b. Prosodic lexical entry:
None.

c. Syntactic lexical entry:
:: d −subj −fpssubj

Fused morpheme. Another special case which we need to consider is that of a fused morpheme. This is an item having a pronunciation and a non-empty set of prosodic features, but which is associated with two lists of syntactic features. We can include this type of item in the shared lexicon by using a list of syntactic feature lists instead of a single feature list. A fused morpheme corresponds to a single prosidic lexical item, but to a list of syntactic feature lists. The first item in this list has a pronunciation, while the subsequent items are silent: For example, if two syntactic morphemes are fused phonologically, the item can be listed in the shared lexicon as follows:[20]

(47) Fused morpheme

a. Shared lexical entry:
$< V, PFeatures, [SFeatures_1 ... SFeatures_n] >$

b. Prosodic lexical entry:
< *V*, LexItem, *PFeatures* >

c. Syntactic lexical entry list:
$[V :: SFeatures_1,$
...,
$:: SFeatures_n]$

(48) Fused morpheme example

a. Shared lexical entry:
< *na*, [1, prefix], [[=aux, +fpssubj, agrs], [=z, aux]] >

b. Prosodic lexical entry:
< *na*, LexItem, [1, prefix] >>

c. Syntactic lexical entry list:
[*na* :: =aux +fpssubj agrs, :: =z aux]

Epenthetic item. An epenthetic item is has an empty syntactic feature list. It corresponds to a single prosodic lexical entry, but has no corresponding syntactic entry:

(49) Epenthetic prefix

a. Shared lexical entry:
$< V, PFeatures, [\,] >$

b. Prosodic lexical entry:
< *V*, LexItem, *PFeatures* >

c. Syntactic lexical entries:
None.

(50) Epenthetic prefix example

[20] A list is used rather than a set on the assumption that fusions consist of morphemes occurring in a fixed order. There is an empirical question as to whether fused morphemes must actually be expressed as distinct syntactic heads: Can traces occur betweens fused heads?

a. Shared lexical entry:
$< ku, [1, \text{prefix}], [\ ] >$

b. Prosodic lexical entry:
$< ku$, LexItem, $[1, \text{prefix}] >$

c. Syntactic lexical entries:
None.

3.2 Situating Constituents in the Input String

This section will describe how words of input correspond to shared, prosodic, and syntactic items, and how the position of prosodic and syntactic constituents found can be expressed in such a way that they can be shared between the prosody and the syntax.

3.3 Positional indices and simple pronounced items.

Positional indices allow the parser to situate a constituent in a string. We will call the beginning point of a string, the position before the first word, 0, and the position after the first word, 1; the position after the second word, 2, and so on.[21] Then we will say that the first word has the span of $< 0, 1 >$, or less clumsily, that it is 'situated' at $< 0, 1 >$. Consider the input string [ni, na, soma]:

(51) 0 ni 1 na 2 soma 3

Both the prosody and the syntax will find a lexical item with the pronunciation *ni* at $< 0, 1 >$, a lexical item *na* at $< 1, 2 >$, and an item *soma* at $< 2, 3 >$. Other consituents found by the prosody will include a Stem pronounced *na* at $< 2, 3 >$, a Stem *nina* at $< 0, 2 >$, and a Clitic Phrase *ninasoma* at $< 0, 3 >$. Similarly, among other things, the syntax will find a Verb Phrase at $< 2, 3 >$ and an AgrS Phrase at $< 0, 3 >$. So far, then, using the same indices for both the prosody and the syntax seems unproblematic.[22]

A shared lexical item or a prosodic or syntactic constituent (either a lexical item or a derived constituent) positioned in the string will be expressed as a tuple consisting of the span, the name of the rule by which the axiom was derived, and either a shared lexical entry or the prosodic or syntactic expression. These tuples will be referred to as 'axioms'. Axioms referring to shared lexical items or to prosodic or syntactic item directly derived from one will be referred to as 'simple axioms'. Other axioms, that is, those derived by prosodic or syntactic rules, will be referred to as 'complex'.

3.4 Shared Axioms: Pronounced and Silent Lexical Items

When relating the input string to shared lexical items, there are only two distinct types of items which need be considered: items which have a pronunciation and silent items.

[21] This numbering scheme will be modified somewhat in what follows.

[22] Were it unnecessary to refer to correspondences between prosodic and syntactic constituents, disjoint indices could be used for the prosody and the syntax.

When a word of input is read, a shared axiom is found with the relevant span for each item in the lexicon that has the pronunciation of the word read. We will call the rule that creates this axiom *lex*. As described in the previous section, the shared axiom is a tuple of the item's span, the rule from which the axiom was derived (here *lex*), the prosodic features, and the syntactic features.

(52) The rule *lex*
When the current index is X and a word W is read from the input string, if there exists a lexical item of the form $< W, PFeatures, SFeatures >$, then an axiom is inferred of the form $<< X, X+1 >$,lex, $PFeatures, SFeatures >$.

It will be noted that the name of the rule is included in the axiom, and in all axioms to follow. This information will be used in collecting the derivation.

A silent item is different in that it does not correspond to a word of input. We shall call the rule which derives axioms from silent lexical items *lexempty*:

(53) The rule *lexempty*
For each shared index X, if there exists a lexical item of the form $< \epsilon, PFeatures, SFeatures >$, then an axiom is inferred of the form $<< X, X >$, lexempty, $PFeatures, SFeatures >$.

Thus, given the single input string [*kitabu*],

(54) 0 kitabu 1

given the following lexicon, containing only one canonical item and one empty category,

(55) $< kitabu, [3, \text{+root}], [\text{n}] >$
$< \epsilon, [\ ], [\text{=n, d}] >$

the following three shared axioms are inferred:

(56) $<< 0,0 >$, lexempty, $< \epsilon, [\ ], [\text{=n, d}] >>$
$<< 0,1 >$, lex, $< kitabu, [3, \text{+root}], [\text{n}] >>$
$<< 1,1 >$, lexempty, $< \epsilon, [\ ], [\text{=n, d}] >>$

3.5 Deriving Simple Module-Specific Axioms

This section will describe how simple module-specific axioms are derived from the shared axioms just described.

Canonical case. In the canonical case, one word of input yields one shared simple axiom, which in turn yields a single prosodic simple axiom and a single syntactic simple axiom:

(57) Canonical case (*canonical*)

a. Shared simple axiom:
$<< X, X+1 >$, lex, $< V, [PF_1 ... PF_m], [SF_1 ... SF_n] >>$

b. Prosodic simple axiom inferred:
$<< X, X+1 >$, canon, $< V$, LexItem, $[PR_1 ... PF_m] >>$

c. Syntactic simple axiom inferred:
$<< X, X+1 >$, canon, $V :: SF_1 ... SF_n >$

(58) Canonical case example

a. Shared simple axiom:
$<< 0, 1 >$, lex, $< kitabu$, [3 sylls, +root], [n] $>>$

b. Prosodic simple axiom inferred:
$<< 0, 1 >$, canon, $< kitabu$, LexItem, [3 sylls, +root] $>>$

c. Syntactic simple inferred:
$<< 0, 1 >$, canon, *kitabu* ::n$>$

Empty categories. The syntax will find other items in addition to these overt elements – empty categories – which are not seen by the prosody. For example, a functional head needs to be merged between *na* (at $< 1, 2 >$) and *soma* (at $< 2, 3 >$) through whose specifier the (covert) subject is moved. These empty syntactic categories do not pose a problem. They are not seen by the prosody and can hence be positioned between the visible input words using identical indices for the beginning and end points of a constituent. That is, they have the span of zero. For example, in this case, the functional head is situated at $< 2, 2 >$. Because empty categories are not directly dependent on words of the input, they are 'found' anywhere they can be used.

Here is the rule for deriving the module-specific axiom for an empty category:

(59) Empty category (*empty*)

a. Shared simple axiom:
$<< X, X >$, lexempty, $< \epsilon, [\ ], [SF_1 ... SF_n] >, lex >$

b. Prosodic simple axiom inferred:
None.

c. Syntactic simple axiom inferred:
$<< X, X >$, empty, $:: SF_1 ... SF_n >$

(60) Empty category example

a. Shared simple axiom:
$<< 0, 0 >$, lexempty, $< \epsilon, [\ ]$, [=n, d] $>>$

b. Prosodic simple axiom inferred:
None.

c. Syntactic simple axiom inferred:
$<< 0, 0 >$, empty, :: =n d $>$

In the case of canonical items and empty categories, then, it is straightforward to use the same indices to refer to constituents found by either the prosody or the syntax. Let's call indices with this property 'shared indices' because they are shared between the two modules. Shared indices can be used to express correspondences between the the prosody and the syntax. However, two cases will somewhat complicate this picture: epenthetic items and fused morphemes. These two types of lexical items will require indices which are visible to only one of the two modules. We will call such indices 'module-specific indices'.

Fused morphemes. In the case of a fused morpheme, consider the Swahili morpheme *wa*, which is the fusion of a second person singular subject agreement marker and a present tense morpheme.[23] We will call the two syntactic items $agrs$ and aux, respectively. Suppose that the word *wa* is found at $<1,2>$ in a string. This word corresponds to one prosodic lexical item but to two syntactic lexical items. At first glance, an easy way to accomodate this case seems to be to position the first of the two fused morphemes (here $agrs$) in the span of the prosodic morpheme, $<1,2>$, and to situate the second morpheme (aux) at $<2,2>$. The problem with this scheme is that we want a parse that uses this aux to be dependent upon using this $agrs$. The way around this is to use a syntax-internal index a (disjoint from the general indices) to delineate the two syntactic morphemes. Thus, $agrs$ will be situated at $<1,a>$ and aux at $<a,2>$.[24]

Here is the rule for inferring the module-specific axioms from a shared axiom for a fused morpheme:

(61) Fused morpheme ($fused$)

a. Shared simple axiom:
$<<X,X+1>$, lex, $<V,[PF_1...PF_m],[[SF1_1...SF1_n],[SF2_1...SF2_o]]>>$

b. Prosodic simple axiom inferred:
$<<X,X+1>$, fused, $<V$, LexItem, $[PR_1...PF_m]>>$

c. Syntactic simple axiom inferred:
$<<X,a>$, fused, $V :: SF1_1...SF1_n>$
$<<a,X+1>$, fused, $:: SF2_1...SF2_o>$

(62) Fused morpheme example

a. Shared simple axiom:
$<<1,2>$, lex, $<$ *na*, [1 sylls, +prefix], [[=aux, +fpssubj, agrs], [=z, aux]] $>>$

b. Prosodic simple axiom inferred:
$<<1,2>$, fused, $<$ *na*, LexItem, [1 sylls, +prefix] $>>$

[23] There are two distinct 'present tenses' in Swahili, indicated by two distinct morphemes–*a* and *na*. The relevant one here is *a*.

[24] More rigorously, we should ensure that the syntax-internal indices should be different for each fusion encountered. Using chart-base parsing, not doing so would allow an item found at $<1,a>$ using one fusion item to be used conjunction with an item at $<a,2>$ from another fusion item. There are no cases in our subset of Swahili where this problem might arise.

c. Syntactic simple axiom inferred:
$<< 1, a >$, fused, V :: =aux +fpssubj agrs>
$<< a, 2 >$, fused, :: =z aux>

Epenthetic words. The last type of lexical entry is an epenthetic item. To illustrate this case, assume that in the input string epenthetic *ku* is the first word and the monosyllabic verb stem *la* 'eat' is the second word. The epenthetic *ku* should be visible to the prosody, but invisible to the syntax. In a sense, an epenthetic item seems to be the mirror image of a syntactic empty category. While the latter has syntactic features but lacks prosodic features, the former has prosodic features but lacks syntactic features. This makes it tempting to use the $< X, X >$ indexing convention described for empty categories. The problem in doing so is that, using a chart-based parser, a parse could fail to employ the epenthetic word. This is undesirable, because it would allow epenthetic syllables in the input string to be ignored by the prosody. A solution which does not have this problem is to simply hide the 1 index from the syntax, using the span $< 0, 2 >$ for an syntactic axiom referring only to *la*. The prosody get two axioms, one for *ku* with span $< 0, 1 >$ and one for *la* with span $< 1, 2 >$. Here is the relevant rule of inference:

(63) Epenthetic prefix (*epenth*), followed by a canonical item

a. Shared simple axiom:
$<< X, X+1 >$, lex, $< V1, [PF1_1...PF1_m], [\,] >>$
$<< X+1, X+2 >$, lex, $< V2, [PF2_1...PF2_n], [SF2_1...SF2_o] >>$

b. Prosodic simple axiom inferred:
$<< X, X+1 >$, epenth, $< V1$, LexItem, $[PR1_1...PF1_m] >>$
$<< X+1, X+2 >$, epenth, $< V2$, LexItem, $[PR2_1...PF2_n] >>$

c. Syntactic simple axiom inferred:
$<< X, X+2 >$, epenth, $V2 :: SF_1...SF_n >$

(64) Epenthetic prefix, followed by a canonical item, example

a. Shared simple axioms:
$<< 0, 1 >$, lex, $< ku$, [1 sylls, +prefix], [ ] >
$<< 1, 2 >$, lex, $< la$, [1 sylls, +root], [=dobj, =d, v] >

b. Prosodic simple axioms inferred:
$<< 0, 1 >$, epenth, $< ku$, LexItem, [1 sylls, +prefix] >>
$<< 1, 2 >$, epenth, $< la$, LexItem, [1 sylls, −root] >>

c. Syntactic simple axiom inferred:
$<< 0, 2 >$, epenth, *la* :: =dobj =d v >

There are no cases in Swahili (as defined in our grammar) where an epenthetic word is followed by a fused mopreheme, but it is easy to see that a rule can be devised analogous to (63) which expands the list of syntactic features lists in the shared lexical item into multiple syntactic axioms.

There is an interesting issue in the morphological conceptualization concerning the coexistence of empty categories and fused morphemes. We have said that epenthetic morphemes are inferred at all spans $< X, X >$. However, it is not clear that this is actually desirable in cases where X is a syntactic (module-internal) index. There is an empirical question whether fused morphemes can span across an intervening empty category.

3.6 Finding Simple and Complex Items in the Input String

Given that we have stated the ways in which a word of input can correspond to prosodic and syntactic lexical items, mediated by shared lexical items, it is now possible to traverse the input string from beginning to end finding all of the positioned lexical items. We will consider the input string [na,soma], corresponding to the Swahili *nasoma* 'I read':

(65) 0 na 1 soma 3

We proceed in a left-to-right fashion. This first word read is *na*. The word is ambiguous in Swahili between a present tense auxiliary and a fused morpheme. Here are the two items as listed in the shared lexicon:

(66) a. Present tense marker *na*
< *na*, [1 sylls, +root, +func], [=z, aux] >

b. Fused morpheme *na*: AgrS head (subject marker) and auxiliary
< *na*, [1 sylls, +prefix], [[=aux, +fpssubj, agrs], [=z, aux]] >

Because this is the first word in the string, the first index of the span of this word is 0. Using the expansions described in (57) and (61) for simple and fused lexical items, the following simple axioms are found:

(67) << 0, 1 >, lex, < *na*, [1 sylls, +root, +func], [=z aux] >>
<< 0, 1 >, canon, < *na*, LexItem, [1 sylls, +root, +func] >>
<< 0, 1 >, canon, *na* :: =z aux>
<< 0, 1 >, lex, < *na*, [1 sylls, +prefix], [[+aux +fpssubj agrs], [=z aux]] >>
<< 0, 1 >, fused, < *na*, LexItem, [1 sylls, +prefix] >>
<< 0, *a* >, fused, *na* :: +aux +fpssubj agrs>
<< *a*, 1 >, fused, :: =z aux >

We have exhausted the simple axioms which can be derived from the input word *na*. Now we can proceed to *soma*, whose start index is 1. which is unambiguously a verb meaning 'read' in our grammar:

(68) < *soma*, [2 sylls, +root], [=d, v] >

Since this word has the span $< 1, 2 >$, the following axioms are inferred and put in the chart:

(69) << 1, 2 >, lex, < *soma*, [2 sylls, +root, +func], [=d, v] >>
<< 1, 2 >, canon, < *soma*, LexItem, [2 sylls, +root, +func] >>
<< 1, 2 >, canon, *soma* :: =d v >

This exhausts the input string and all of the simple axioms which can be derived from it. Using chart-based parsing, after all of the simple axioms are inferred, all complex axioms are found by closing the axioms under the rules given by our prosodic and syntactic grammars. Since any syntactic parse to derive agrs involves movement and (in our analysis) several empty categories, it is far easier to demonstrate how a derivation is collected using a prosodic parse. So, for the moment, let us concern ourselves only with the prosodic parse. After the entire string has been heard, the chart will contain the following shared and prosodic axioms:

(70) $<< 0,1 >$, lex, $<$ *na*, [1 sylls, +root, +func], [=z, aux] $>>$
$<< 0,1 >$, canon, $<$ *na*, LexItem, [1 sylls, +root, +func] $>>$
$<< 0,1 >$, lex, $<$ *na*, [1 sylls, +prefix], [[=aux, +subj, agrs], [=z, aux]] $>>$
$<< 0,1 >$, fused, LexItem, $<$ *na*, [1 sylls, +prefix] $>>$
$<< 1,2 >$, lex, $<$ *soma*, [2 sylls, +root], [=obj, =d, v] $>>$
$<< 1,2 >$, canon, LexItem, $<$ *soma*, [2 sylls, +root] $>>$
$<< 1,2 >$, stm1, Stm, $<$ *na soma*, [2 sylls] $>>$
$<< 0,2 >$, stm1, Stm, $<$ *na soma*, [3 sylls] $>>$
$<< 0,2 >$, wrd1, Wrd, $<$ *na soma*, [3 sylls] $>>$
$<< 0,2 >$, clt1, Clt, $<$ *na soma*, [3 sylls] $>>$
$<< 0,2 >$, snt1, Snt, $<$ *na soma*, [3 sylls] $>>$

We are now ready to consider how a derivation and a yield are collected from the chart.

3.7 Collecting the Derivation

A successful syntactic parse is indicated by an axiom of the highest category (*agrs* for the syntax, *Snt* for the prosody) which covers the entire span of the input string. All derivations are found for all successful parses. Recall that each complex axiom contains enough information to find the axioms used to derive it. Thus, to find the derivation, a recursive process is applied to find all of the axioms employed to derive the top-level axiom. To give an idea about how the derivation is collected, we will consider a few of the axioms found, listed in (70), when in the input string [*na*, *soma*] was parsed.

We start with the axiom of the highest category, Snt:

(71) $<< 0,2 >$, snt1, Snt, $<$ *na soma*, [3 sylls] $>>$

We note that this axiom was inferred using rule *snt*1, repeated here for convenience:

(72) Rule *snt*1:
Snt $\rightarrow$ Clt

We note that to infer a Snt (the left-hand side of the rule) using this rule, we must have had a Clt (the right-hand side). Furthermore, since the span of the Snt is $< 0,2 >$, the Clt must have also had the span $< 0,2 >$. We return to the chart in (70) and note that we do, in fact, have a Clt with the span $< 0,2 >$:

(73) $<< 0,2 >$, clt1, Clt, $<$ *na soma*, [3 sylls] $>>$

We note that this Clt was inferred by rule *clt*1, and go through this same process finding each axiom which must have served as input to the inference at hand. In the end, we will have picked out from the chart all of the axioms which were actually utilized to make the top-level inference we started with. In this case, the axioms used were these:

(74) << 0, 2 >, snt1, Snt, < *na soma*, [3 sylls] >>
<< 0, 2 >, clt1, Clt, < *na soma*, [3 sylls] >>
<< 0, 2 >, wrd1, Wrd, < *na soma*, [3 sylls] >>
<< 0, 2 >, stm1, Stm, < *na soma*, [3 sylls] >>
<< 1, 2 >, stm1, Stm, < *na soma*, [2 sylls] >>
<< 1, 2 >, canon, LexItem, < *soma*, [2 sylls, +root] >>
<< 1, 2 >, lex, < *soma*, [2 sylls, +root], [=obj, =d, v] >>
<< 0, 1 >, fused, LexItem, < *na*, [1 sylls, +prefix] >>
<< 0, 1 >, lex, < *na*, [1 sylls, +prefix], [[=aux, +subj, agrs], [=z, aux]] >>

While multiple derivations are sometimes available, in this case only one derivation can be found. Note that this means that some axioms in the chart may be useless–not employed in any derivation. This is the case with the axioms using the present tense marker *na*, which, due to its [+root], monosyllabic properties, cannot form a licit word in the simple string *nasoma*. The unused axioms are these:

(75) << 0, 1 >, lex, < *na*, [1 sylls, +root, +func], [=z, aux] >>
<< 0, 1 >, canon, < *na*, LexItem, [1 sylls, +root, +func] >>

3.8 Synchronization

Why synchronization is necessary. The synchronization process checks that a syntactic parse corresponds to a prosodic parse, which merely requires checking that the same lexical items were used in both parses. To understand why this is necessary, it is perhaps best to see how two parses might not correspond. Consider the following ungrammatical Swahili form:

(76) * ni- li- la
1s.subj- past- eat
'I ate, I eat it'

There is both a licit syntactic parse of this string and a licit prosodic parse of this string in the Swahili grammar I have specified. The ungrammaticality of the string, then, stems from the fact that there is no corresponding pair of syntactic and prosodic parses. That is, no syntactic parse of the string uses the same lexical items as any prosodic parse. The problem stems from the presence to two different morphemes pronounced *li*, shown first in (7), repeated here in (77) with the syntactic and prosodic features as found in the implementation:

(77) a. Object marker *li*:

ni- na- li- (*ku-) la
1s.subj- pres- 5.obj- epenth- eat

‘I’m eating it’

Lexical entry: < *li*, [1 sylls, +prefix], [=v, +n5obj, cl, -cl] >

b. Past auxiliary (tense marker) *li*:

ni-	li-	*(ku-)	la
1.s-subj-	past-	2s.obj-	eat

‘I ate’

Lexical entry: < *li*, [1 sylls, +root, +func], [=z, aux] >

The syntactic parse of the input string $[ni, li, la]$ uses the first item, the auxiliary *li*, giving the meaning ‘I ate’. But using this lexical item does not yield a syntactic parse, because the structure would contain a monosyllabic Prosodic Stem:

$*[_{Wrd}\ [_{Stm}\ \text{ni}\ [_{Rt}\ \text{li}\ ]\ ]\ [_{Stm}\ \text{la}\ ]\ ].$

The attested surface form of this syntactic structure has epenthetic *ku*:

$*[_{Wrd}\ [_{Stm}\ \text{ni}\ [_{Rt}\ \text{li}\ ]\ ]\ [_{Stm}\ \text{ku}\ [_{Rt}\ \text{la}\ ]\ ]\ ].$

The syntactic parse of the same input string [ni,li,la] uses the object marker *li*. But using this item does not yield a syntactic parse; the form lacks a needed auxiliary.

Thus, to say that a parse has been found for a given input string, it is not sufficient to have found a syntactic parse and a prosodic parse. The two parses must use the same lexical items.

Implementing synchronization. Since all syntactic and prosodic axioms are ultimately derived from shared axioms, synchronization entails taking a prosodic parse and a syntactic parse and making sure that they use the same shared axioms. Axioms which are seen by only one of the modules are ignored in this comparison. These axioms are empty categories, which are invisible to the prosody, and epenthetic material, which is invisible to the syntax. Each prosodic derivation found is compared in this way with each syntactic derivation. A successful parse, then, is actually a pair of derivations, one prosodic and one syntactic, which correspond in the lexical items (shared axioms) they use.

4 Discussion and Conclusion

Here you will address the question as to how this model relates to how humans actually process language.

You will describe how the model could be extended to synchronze to check prosodic domain boundaries against syntactic boundaries.

Since this creature is still evolving, i’m not quite sure what will go here.

References

Boullier, P. (1999). A cubic time extension of context-free grammars. In *Research Report Number 3611* at `http://www.inria.fr.RRRT/RR-3611.html`, INRIA-Rocquencourt, France, January 1999, 28 pages.

Buell, L. C. (2002). Swahili *Amba*-less relatives without head movement. *UCLA Working Papers in Linguistics ?*(7), ?

Downing, L. J. (1992). Prosodic stem $\neq$ prosodic word in Bantu. In T. A. Hall and U. Kleinhenz (Eds.), *Studies on the Phonological Word*, pp. 73–98. Amsterdam: John Benjamins Publishing Company.

Kayne, R. S. (1994). *The Antisymmetry of Syntax.* Cambridge, Massachussetts: MIT Press.

Koopman, H. and D. Sportiche (1988). Subjects. Ms, University of California, Los Angeles.

Selkirk, E. O. (1984). *Phonology and Syntax: The Relation Between Sound and Structure.* Cambridge, Mass.: MIT Press.

Stabler, E. P. and E. L. Keenan (2000). Structural similarity. In T. R. A. Nijholt, G. Scollo and D. Heylem (Eds.), *Algebraic Methods in Language Processing: AMiLP 2000.* University of Iowa. Revised version forthcoming in *Theoretical Computer Science.*

Woolford, E. (2000). Agreement in disguise. In V. Carstens and F. Parkinson (Eds.), *Advances in African Linguistics*, Volume 4 of *Trends in African Lingusitics*, pp. 103–117. Trenton, New Jersey: Africa World Press.

A Interactions of Note Between the Modules

Write something here that explains what this section is.

a. [na] ← /na/ *vs.* /ni + a/

The string *na* is ambiguous between the fusion of the first person subject marker *ni-* and a present tense marker *a-*, as in (78), and an unfused present tense marker, as in (79):

(78) na- soma
1.subj=pres- read
'I read'

(79) ni- na- soma
1s.subj- pres- read
'I read, I am reading'

The case of fusion must be resolved in the morphology. Obviously a syntactic derivation will converge only if the phonetic string is analyzed as the correct morpheme string.

The following sentences show the ambiguity between the present tense marker *na* and the fused first person singular (*ni*) and general present tense marker *a*.

(80) Fused subject marker and present tense *na-*:

Na- nunua kitabu.
1s.subj=pres- buy 7.book

'I buy a book.' `ni a nunua kitabu`

(81) Tense marker *na-*

Ni- na- nunua kitabu.
1s.subj- pres- buy 7.book

'I am buying a book' `ni na nunua kitabu`

b. Epenthetic versus lexical *ku-*

As seen earlier (?), the epenthetic syllable *ku* is inserted in a Prosodic Stem which would otherwise violates a constraint which says that a Prosodic Stem must be at least disyllabic. However, a genuine morpheme *ku-* also exists,[25] namely the second person singular object marker, as in (82):

[25] Various other entries for *ku* would be necessary for noun class 17 agreement and for infinitive verbs, but we will not deal with these in this simple grammar.

(82) ni- na- ku- ona
1s.subj- pres- 2s.obj- see
'I see you'

This involves an interaction between the phonology and the syntax, in that the syntax must know whether this syllable could have been inserted by the phonology.

A legitimate ambiguous string in this respect is *ninakupa*, which can mean variously 'I give' or 'I give (to) you', depending on whether *ku* is epenthetic or whether it is object marker:

(83) ni- na- ku- pa
1s.subj- pres- 2s.obj- give
'I give you'

(84) ni- na- ku- pa
1s.subj- pres- epenth- give
'I give'

The following illustrate the addition or omission of epenthetic *ku*. The syntax will never see this '*ku*', but rather, the grammaticality of these forms will be determined by the prosody. (All three sentences are syntactically grammatical.) An ambiguity exists between this epenthetic *ku* and the second person singular object marker (as well as a few other uses), but we have not included these alternative meaning of *ku* in the lexicon.

(85) With auxiliaries which form Prosodic Roots:

i. Epenthetic *ku*-, correctly inserted. The Prosodic Stem following thhe tense marker *li* must be at least disyllabic. This condition is not met here with adding *ku*.

   U- li- ku- la kitunguu.
   2s.subj- past- *ku*- eat 7.onion

   'You ate an onion.' `u li la kitunguu`

ii. Epenthetic *ku*-, incorrectly omitted. Again, the Prosodic Stem following *li* must be at least disyllabic.

   * U- li- la kitunguu.
   2.subj- past- eat 7.onioon

   'You ate an onion' `u li la kitunguu`

iii. *Ku*- correctly omitted. The Prosodic Stem following the tense marker *li* is already disyllabic.

U- li- ki- la kitunguu.
2s.subj- past- 7.obj- eat 7.onion

'You ate the onion.' `u li ki la kitunguu`

iv. Epenthetic *ku-*, superfluous. The Prosodic Stem *kila* following the tense marker *li* is already disyllabic without *ku.*

* U- li- ku- ki- la kitunguu.
2s.subj- past- *ku-* 7.obj- eat 7.onion

'You ate the onion.' `u li ki la kitunguu`

v. Epenthetic *ku-*, superfluous:

* U- li- ku- soma kitabu.
2s.subj- past- *ku-* read 7.book

'You read a book.' `u li soma kitabu`

(86) With auxiliaries which don't form Prosodic Roots:

i. Epenthetic *ku-*, correctly omitted. The *ka* cannnot serve as the nucleus of a Prosodic Stem, so the single prosodic stem *ukala* is already more than disyllabic.

U- ka- la kitunguu.
2s.subj- consec- eat 7.onion

'So then you ate an onion.' `u ka la kitunguu`

ii. Epenthetic *ku*, incorrect because the Prosodic Stem is already more than disyllabic without it.

U- ka- ku- la kitunguu.
2s.subj- consec- *ku-* eat 7.onion

'So then you ate an onion.' `u ka la kitunguu`

The following sentence is syntactically ungrammatical, but prosodically good. Its purpose is to show how a derivation will crash in the syntax.

(87) * Ni- li- ki- soma gazeti.
1s.subj- past- 7.obj- read 5.newspaper
'I read the newspaper.' `ni li ki soma gazeti`

c. Prefix *ni-* versus suffix *-ni*

The string *ni* is ambiguous between the prefix *ni-* which is the first person singular subject or object marker, as in (88) and (89), respectively, and the suffix *-ni* which is a locative suffix, as in (90):

(88) ni- na- ku- ona
1s.subj- pres- 2s.obj- see
'I see you'

(89) u- na- ni- ona
2s.subj- pres- 1s.obj- see
'you see me'

(90) ni- li- imba shule- ni
1s.subj- past- sing 7.school- loc
'I sang at school'

> The subject and object markers as Prosodic Stem attaching prefixes, while the locative suffix is a Prosodic Word attaching suffix.

The string *ni* can be parsed either as the first person singular subject marker, the first person singular object marker, or as a postnominal locative suffix.

(91) Subject marker *ni-*:

Ni- li- soma kitabu.
1s.subj- past- read 7.book

'I read a book.' `ni li soma kitabu`

(92) Object marker *ni-*:

U- li- ni- ona.
2s.subj- past- 1s.obj- see

'You saw me.' `u li ni ona`

(93) Postnominal locative suffix *-ni*:

U- li- soma kitabu shule- ni.
2s.subj- past- read 7.book 9.school- loc

'You read a book at school.' `u li soma kitabu shule ni`

B Implementation

The lexicon is loaded into memory. The lexicon contains items in the following four formats:

a. Items with both prosodic and syntactic features. Format:

```
lex( PRONUNCIATION, SYNTACTIC_FEATURES, PROSODIC_FEATURES ).
```

Examples:

```
lex( 'la', [ =dobj, =d, v ], [ 1, root ]  ).
lex( 'la', [ =dobj, =d, =adv, v ], [ 1, root ] ).
lex( 'kitunguu', [ n7 ], [ 4, root ] ).
```

b. Unpronounced lexical items with syntactic features ('empty categories'). Format:

```
lexempty(  SYNTACTIC_FEATURES, PROSODIC_FEATURES ).
```

The presence of the PROSODIC$_F$EATURES argument is a historical artifact. For *lexempty* items, this argument is always [0], indicating that the item has zero syllables and no other prosodic features.

```
lexempty( [ d, -subj, -fpssubj ], [ 0 ] ).
```

c. Pronounced fused morpheme. This is an item which is treated as a unit in the prosody, but which must be treated as two separate syntactic heads. The format has a tuple of tuples of syntactic features, one for each of the syntactic heads fused in the morpheme. Format:

```
fusion( PRONUNCIATION, PROSODIC_FEATURES, SYNTACTIC_FEATURE_TUPLES ).
```

Example:

```
fusion( 'na', [1,prefix], [ [=aux, +fpssubj, agrs],  [=z, aux] ] ).
```

d. Epenthetic material. This has prosodic features but no syntactic features. Format:

```
epen( PRONUNCIATION, PROSODIC_FEATURES ).
```

Example:

```
epen( 'ku', [ 1, prefix ] ).
```

Input words. To perform a parse, the parser is given an input string divided into morpheme chunks, and the desired category of the derived structure. The top-level syntactic category to be used in the sentences parsed here will be *agrs*. Thus, to parse the Swahili input string *ninakisoma kitabu* 'I am reading the book', the following query is entered:

```
parses( [ni, na, ki, soma, kitabu], agrs ).
```

The string is then parsed by both the syntactic parser and the prosodic parser. The syntactic parser will find all parses for the string of which the final category is *agrs*. The prosodic parser will find all parses of the string, keeping the parses of the highest category (e.g., Sentence). The yield of each syntactic parse are then compared with the yield of each prosodic parse to see that they use the same lexical items. The result is a list of synchronized parse pairs, each of which consists of one syntactic parse and one prosodic parse which use the same lexical items.

Simple and complex axioms. Simple axioms are those which relate words of input to lexical items (including unpronounced items, epenthetic elements, and fused morphemes). Complex axioms are axioms derived by the rules of inference.

As the words of input are read, prosodic and syntactic axioms are entered into the chart. The axiom indicates the location of the word in the input string using indices. To illustrate the use of indices, consider the input string $[ni, na, ki, soma, kitabu]$. In this input string *ni* is located between indices 0 and 1, and *na* is located between indices 1 and 2. The prosodic parser will eventually parse a Clitic Phrase *ninakisoma* between indices 0 and 4, while the syntactic parser will find a constituent *kisomakitabu* between indices 2 and 5.

The syntactic and prosodic parsers are designed in such a way that they always use the same indices for each item in the input string. For fused morphemes, in the syntax additional indices with a different format will be used to indicate distinct positions between prosodically adjacent positions. For example, the fused morpheme *na* fuses two syntactic morphemes. If *na* is found between indices 3 and 4, the two simple syntactic axioms added to the chart are for items located between 3 and $[3, 1]$ and between $[3, 1]$ and 4, respectively.

Each axiom, simple or complex, contains in addition to the features of the category found or inferred, the location indices and an indication of how the axiom was found or inferred. In the case of a simple axiom, this information will indicate whether the item is a *lex*, *lexempty*, $fusion$, or *epen* element, while a complex axiom will indicate which rule of inference was used to infer it. Here are the two simple axioms found for the word *kitabu* 'book', and the lexical entry from which they are inferred:

(94) `put examples here`

The information on how the rule was inferred are used to find build the derivation tree after the parses are found. This is necessary, because many axioms will be found which are not used in any particular successful parse.

Simple syntactic and prosodic axioms begin with s : and *sp* :, respectively. Complex syntactic and prosodic axioms begin with c : and *cp* :, respectively.

B.1 Implementation of the Prosody

Inferences After the simple prosodic axioms are found, they are closed under the prosodic rules of inference which correspond to the rewrite rules of the prosodic grammar. The inferences have the following format:

(95) `inference( RULE_NAME, EXISTING_AXIOMS, NEW_AXIOMS, CONDITIONS ).`

As an example of one of the rewrite rules converted into a Prolog inference, here is one of the Stem rewrite rules:

(96) a. Stm[$\alpha + \beta$ sylls, γfunc] $\rightarrow$

LexItem[α sylls, +prefix, $-$root] Stm[β sylls, γfunc]

Condition: $\alpha + \beta \geq 2$

b.
```
inference(stem2/2,
  [ sp:[[X,Y,_]:[LexNumSylls|LexFs]],
    cp:[[Y,Z,stem,_]:[OldStemNumSylls|OldStemFs]] ],
    cp:[[X,Z,stem,stem2/(Y,[LexNumSylls|LexFs],[OldStemNumSylls|OldStemFs])]:
  [ percolate(OldStemFs,[func],[$ $],NewStemFs),
    \+member( root, LexFs ),
    member( prefix, LexFs ),
    NewStemNumSylls is LexNumSylls + OldStemNumSylls,
    NewStemNumSylls >= 2 ] ).
```

This inference says that if a simple axiom (i.e. a lexical item) is found between indices X and Y and a complex axiom which is a stem is found between indices Y and Z, then we can infer that there is a stem between indices X and Z with the combined number of syllables and with any [func] feature from the stem at $[Y, Z]$ percolated, provided the following conditions are satisfied: that the total number of syllable is greater than or equal to 2, that the Lexical Item at $[X, Y]$ does not have the feature root and and the it does have the feature [+prefix].

B.2 Presenting the Syntax Implementation

Just as in the prosody, after the simple syntactic axioms are found, they are closed under rules of inference which correspond to the rules of *merge* and *move* listed in the definition of Minimalist Grammar. The Prolog implemention of the Minimalist Grammar is the same as that discussed in CITE.

B.3 Synchronization

The synchronization process checks that a syntactic parse corresponds to a syntactic parse, which merely requires checking that the same lexical items were used in both parses. To understand why that this is necessary, it is perhaps best to see how two parses might not correspond.

Consider the following ungrammatical Swahili form:

(97) * ni- li- la
1s.subj- past- eat
'I ate, I eat it'

I will describe here that there is both a licit syntactic parse of this string and a licit prosodic parse of this string in the Swahili grammar I have specified. The ungrammaticality of the string, then, stems merely from the fact that the is no corresponding pair of syntactic and prosodic parses. That is, no syntactic parse of the string uses the same lexical items as a prosodic parse. The problem stems from the presence to two different morphemes pronounced *li*:

(98) a. Auxiliary *li*:

```
lex( 'li', [=z, aux], [ 1, root, func ] ).
```

b. Object marker *li*:

```
lex( 'li', [ =v, +n5obj, cl, -cl ], [ 1, prefix ] ).
```

The syntactic parse of the input string $[ni, li, la]$ uses the first item, the auxiliary *li*, giving the meaning 'I ate'. But using this lexical item does not yield a syntactic parse, because the structure would contain a monosyllabic Prosodic Stem: *[$_{Wrd}$ [$_{Stm}$ ni [$_{Rt}$ li]] [$_{Stm}$ la]]. The attested surface form of this syntactic structure has epenthetic *ku*: *[$_{Wrd}$ [$_{Stm}$ ni [$_{Rt}$ li]] [$_{Stm}$ ku [$_{Rt}$ la]]].

The syntactic parse of the same input string $[ni, li, la]$ uses the object marker *li*. But using this item does not yield a syntactic parse; the form lacks a needed auxiliary.

Thus, to say that a parse has been found for a given input string, it is not sufficient to have found a syntactic parse and a prosodic parse. The two parses must use the same lexical item. In the implementation, this correspondence is checked by comparing the yields of the two parses. The yields are collected as the derivations are collected. In the example just illustrated, the prosodic yield will thus contain the an item pronounced *li* with the (prosodic) features $[1, prefix]$. The syntactic yield will contain an item with the (syntactic) features $[= z, aux]$. The two parses will fail to synchronize because there is no item in the lexicon of the following form, which would combine these syntactic and prosodic features:

```
lex( 'li', [=z, aux], [ 1, prefix ] ).
```