THE STATUS OF ROOTS
IN EVENT COMPOSITION: LAZ

ÖMER DEMİROK

ABSTRACT: In this paper, based on the empirical data from the Pazar dialect of Laz (PL), the status of roots in event composition is revisited. The basic proposal is that the event structure building mechanism in Ramchand (2008) can be incorporated into the cyclic spell-out system of Pantcheva (2011) in the framework known as Nanosyntax (Starke 2009). The association model in Ramchand (2008) that does not recognize root as a syntactic node is shown to undergenerate. In particular, the aspectual system of PL, which appears to provide the lexical match for the event heads/features, creates a theory-internal problem with respect to the root whose potential matching nodes are already lexically occupied. If cyclic phrasal spell-out is adopted and the root is acknowledged as a syntactic node eligible for lexical matching, not only the theory-internal problem is solved but also an interesting case of root allomorphy is accounted for. The proposal appears to be in line with the empirical facts in that all attested patterns of allomorphy are allowed whereas unpredicted lexicalization patterns are outlawed.

KEYWORDS: cyclic spell-out, event structure, Laz, nanosyntax, root.

1. DESCRIPTION OF LAZ FACTS*

The Pazar dialect of Laz (henceforth PL) is an endangered South-Caucasian language spoken in Turkey. PL exhibits typologically rare and theoretically interesting morphosyntactic properties. In particular, the case alignment system and the aspectual marking properties of PL reveal a fine-grained cartography of the thematic/event structure domain.

In the following two subsections, the case alignment and aspectual marking system of PL will be discussed. It will be shown that the case system encodes the basic argument structure (i.e. external vs. internal split) while the aspectual marking appears to be the morphological manifestation of the fine-grained event structure.

* I owe my thanks to my Laz informant İsmail Avcı-Bucaklı and the audience at the WoRSE in Tromso. Abbreviations: IMPF=imperfective, PST=past, 1=first person, 3=third person, SG=singular, ERG=ergative, NOM=nominative, OPT=optative, COP=copula.
1.1 Case and argument structure

Laz has an active case alignment system, exhibiting overt differential marking of external and internal arguments (Dixon 1994). Active alignment is substantially different from accusative and ergative alignment in that it does not refer to transitivity. Thus, active alignment in PL does not encode grammatical functions but argument structure. The examples in (1) illustrate the active alignment of case marking in PL.

(1) a. Bere-k \textit{diʃk’a} \textit{t’ax-u}  
\begin{tabular}{ll}
child-ERG & wood.NOM \hline
\end{tabular}  \break
\begin{tabular}{ll}
& break-3SG.PST
\end{tabular}  
‘The child broke the wood.’

b. Bere \textit{ɣur-u}  
\begin{tabular}{ll}
child.NOM & die-3SG.PST
\end{tabular}  
‘The child died.’

c. Bere-k \textit{k’i-u}  
\begin{tabular}{ll}
child-ERG & scream-3SG.PST
\end{tabular}  
‘The child screamed.’

Notice that the external argument of the transitive event in (1a) and the unergative event in (1c) are overtly marked ERGative while the internal argument of the transitive event in (1a) and the unaccusative event in (1b) are (unmarked) NOMinative.\textsuperscript{1} It should be noted that the external vs. internal argument split in PL is consistent as evidenced by the differential case marking of sole arguments of intransitive events.

1.2 Aspectual marking and event structure

The active case alignment system of PL encodes the external and internal argument split, differentiating three basic event types, i.e. transitive, unergative, and unaccusative. However, the aspectual morphology in PL presents evidence that a finer event typology is necessary.

Öztürk & Taylan (2012) have argued that PL aspectual marking is the synchronic morphological manifestation of \textit{event structure}, which is also detectable by several syntactic tests (Rappaport-Hovav & Levin 2000).

PL has a set of aspectual formatives (i.e. \textit{-um, -ur, -am}) that are also known as thematic suffixes (Lacroix 2009; Öztürk 2011). The aspectual formatives immediately follow the verbal root and simultaneously encode the imperfective outer aspect and the structure of events.\textsuperscript{2} While all of

\textsuperscript{1} The case labels ERG and NOM are not crucial in that PL does not exhibit dependent case licensing constraints as in accusative or ergative systems.

\textsuperscript{2} There is an additional imperfective suffix \textit{-er}, which marks stative eventualities and is used
these suffixes invariably encode imperfective outer aspect, the selection of a specific suffix out of this set is contingent on the event structure.

The examples in (2) below illustrate the event structure dependent selection of the imperfective formatives. Notice the variation in the imperfective markers.

(2) a. Ma\(^3\) diʃk’a  p-t’ax-um / p-t’ax-i\(^3\)
   1.ERG wood.NOM  1-break-IMPF  1-break-PST
   ‘I am breaking / I broke the wood.’

b. Ma  b-yur-ur / b-yur-i
   1.NOM  1-die-IMPF  1-die-PST
   ‘I am dying/ I died.’

c. Ma  tajʃ’i  b-zd-am / b-zd-i
   1.ERG rope.NOM  1-pull-IMPF  1-pull-PST
   ‘I am pulling/ I pulled the rope.’

d. Ma  p-k’i-am / p-k’i-i
   1.ERG  1-scream-IMPF  1-scream-PST
   ‘I am screaming/ I screamed.’

As the set of data in (2) indicates, there is root-dependent allomorphic variation in the imperfective morphemes. This variation, on the surface, might be analyzed as diacritic information referring to arbitrary classes of verbs. Yet, the close investigation of these verb classes reveals that the variation is not trivial in that the sets of verbs that respectively require -um, -ur, or -am are in fact semantically coherent.

Öztürk & Taylan (2012) have identified that the class of verbs that selects for -um and -ur are change of state (CoS) events that entail a culmination/result to a dynamic event. While -um is required with transitive CoS events with overt initiators, -ur is selected by unaccusative CoS events with no overt initiator. They have claimed that dynamic events that do not entail a change of state (i.e. culmination/result) require -am, regardless of their transitivity, i.e. unergative or transitive non-CoS events.

The change of state events are complex in the sense that they semantically entail a result and contain the syntactic representation of a state.\(^5\)

with derived intransitives. I limit the discussion in this paper to the non-derived dynamic events.

\(^3\) Simplex first and second person pronominal forms in PL exhibit case syncretism, hence the absence of overt case marking.

\(^4\) The past forms are also provided so as to allow the reader to isolate the roots.

\(^5\) Note that CoS events are not inherently telic. A piece of evidence for this claim might be that even in the non-referential reading of the object in (2a), the verbal root t’ax ‘break’ still selects for the IMPF morpheme -um.
For instance, the event of dying in (2b) entails a change of state from being alive to dead while the event of breaking in (2a) entails a change of state from being not-broken to broken. However, the event of screaming in (2c) and the event of pulling in (2d) are dynamic Activity events in that they do not necessarily entail a result or change of state.

It should also be noted that the event structure marking is only seen with the imperfective outer aspect in PL. The past forms on the right in (2) are perfective and do not morphologically express the event structure. In the Imperfect construction with past tense, however, the imperfective morphemes show up again as seen in (3) below.

(3)  
   a. \textit{Nana-}k \textit{k’aptʃa} \textit{tf’-um-t’-u}  
       mother-ERG anchovy.NOM fry-IMPF-COP-3SG.PST  
       ‘The mother was frying anchovy.’
   b. \textit{Nana-}k \textit{k’aptʃa} \textit{tf’-u}  
       Mother-ERG anchovy.NOM fry-3SG.PST  
       ‘The mother fried anchovy.’

I conclude this section with a diagram that summarizes the classification of dynamic events in PL and the corresponding imperfective formatives each class requires. In the next section, I will introduce the theoretical framework.

\[
\begin{array}{c}
\begin{array}{c}
\text{[+dynamic]} \\
\text{[-CoS]} \\
\text{[-am]}
\end{array}
\end{array}
\begin{array}{c}
\begin{array}{c}
\text{[+CoS]} \\
\text{[+initiator]} \\
\text{[-initiator]}
\end{array}
\end{array}
\begin{array}{c}
\begin{array}{c}
\text{[-um]} \\
\text{[-ur]}
\end{array}
\end{array}
\]

\textsc{Diagram 1. Event classification in PL.}

2. THEORETICAL FRAMEWORK: NANOSYNTAX

The theoretical options available for matching lexical and syntactic information (i.e. basically forming form-meaning pairs) are not unlimited. The primary division is between early insertion and late insertion accounts. While some versions of the Minimalist Program (MP) propose that lexical information is present in syntax from the beginning (dubbed as early insertion), post-syntactic frameworks like Distributed Morphology (DM) (Halle & Marantz 1993) propose late insertion accounts (i.e. lexical matching follows syntactic composition). While the MP has the advantage of allowing the manipulation of the syntactic derivation by lexical features, it fails to provide a principled account of the word-internal linear morpheme order facts. DM has the su-
priority of proposing the theoretical tools that can deliver these facts; however, post-syntactic lexicon-blind operations like Fusion and Fission seem to impose extra acquisitional burden and overgeneration.

As an attempt to combine the advantages of these two systems and eliminate overgeneration and undergeneration as much as possible, Nanosyntax (Starke 2009) offers a novel perspective on linguistic architecture, in particular, the nanosyntactic proposal that the matching of lexical-syntactic information is cyclic and phrasal.

In Nanosyntax, syntax is assumed to have all the generative power while the lexicon is a “dumb repository”. As in Distributed Morphology, syntax operates on abstract linguistic features by MERGE and the insertion of lexical items is assumed to be post-syntactic. The MERGE order of linguistic features is subject to a universal functional sequence (\textit{fseq}). For instance, in no derivation can the merger of [past] precede the merger of [imperfective]. Starke (2009) argues that MERGE and \textit{fseq} are the only tools available for syntactic derivation. The idea of \textit{fseq} builds on the related work in the cartographic approach (Cinque 2002).

Nanosyntax disallows hierarchically unordered lists of features merged as one syntactic head. That is, no pre-syntactic merger of syntactic features (i.e. into feature bundles) is available. Each syntactic feature must be merged in syntax according to \textit{fseq}. Decomposing syntactic derivations into \textit{nano} bits has a profound consequence particularly for the shape of lexicon: syntactic terminals are mostly sub-morphemic, i.e. smaller than a morpheme. Therefore, the listed words are mostly phrasal entries in the lexicon and when they are matched with the syntactic trees, they can span more than one terminal node. In the next two subsections, I will provide an overview of the nanosyntactic composition and spell-out mechanism proposed in Pantcheva (2011) and event composition theory proposed in Ramchand (2008).

\textbf{2.1 Cyclic phrasal spell-out}

While there is more than one way of exercising spell-out at non-terminals within the Nanosyntactic framework, I will make use of a version of \textit{phrasal spell-out} articulated in Pantcheva (2011) where each instance of \textit{external merge} is followed by \textit{lexical access}. Any time a new feature is introduced into the derivation, syntax looks for a matching lexical entry in the lexicon. This model is referred to as \textit{cyclic lexicalization} (i.e. cyclic lexical matching of abstract syntactic features during the derivation). The derivation with cyclic lexical access proceeds as follows:
Let us now see the simplified cyclic derivation of English plurals, as illustrated in (4) below.

\[
\text{(4) } \qquad \begin{array}{c}
\text{MERGE: } \quad \text{[a] + ROOT} \\
\text{LEXICAL ACCESS:} \quad \text{match [aP]} \\
\text{MERGE:} \quad \text{[b] + [aP]} \\
\text{LEXICAL ACCESS:} \quad \text{match [bP]}
\end{array}
\]

The first attempt to lexicalize in (4) happens when the head \( n^o \), which projects an \( nP \), is merged. The phrase \( nP \) will get a match depending on the root (e.g. \( nP/^/\text{tʃaɪld}^/ \) or \( nP/^/\text{bʊk}^/ \)). Then follows the merger of the head/feature \( pl^o \), which triggers the second lexical access. Notice that it is not only the \( pl^o \) that can match. The lexical access searches for a full match for the \( plP \). If the lexicon has a full-match for the \( plP \) to be lexicalized, it will win and the structure will be lexicalized. This is exactly what happens when the root below is \( \text{CHILD} \). The whole \( plP \) will be matched with the form \( /\text{tʃɪldrən}/ \) which corresponds to the phrasal lexical entry \( [\text{pl } [n \text{ CHILD}]] \).

If it is assumed that the root below is \( \text{BOOK} \) and the lexicon has no pre-stored entry for \( /\text{bʊks}/ \), the structure will be lexicalized by two different entries, i.e. \( /\text{bʊk}/ \) and \( /s/ \). However, this matching cannot happen in situ. Let us assume that the entry for \( /s/ \) corresponds to a phrasal entry \( [\text{plP } [pl^o]] \) and not the terminal \( pl^o \). In such a situation, the entry \( /s/ \) cannot be inserted as it does not contain the \( nP \) in the structure. Thus, the matching of \( plP \) with \( /s/ \) will trigger an evacuation movement of \( nP \) (adjunction to the root node). The structure attained with this movement is given in (5). The spell-out triggered movement will leave a trace which is assumed to be ignored for lexicalization purposes. Thus, the entry \( /s/ \) can now fully match the \( plP \).

Notice that this movement also derives the word-internal morpheme order in \( /\text{bʊk-s}/ \). Although evacuation movements on the surface complicate the derivations, they are not arbitrary but lexically motivated. Also notice that if a frequent entry like \( /\text{bʊks}/ \) is stored in the lexicon, there will be no evacuation movement and the whole \( plP \) will find the lexical match \( /\text{bʊks}/ \).

\[
\text{(5) } \quad \begin{array}{c}
\text{nP} \\
/\text{bʊk}/
\end{array} \quad \text{plP } /\text{s}/ \\
\text{pl}^o \quad t_{nP}
\]
The last important technical tool assumed in Pantcheva (2011) is the **Superset Principle**. According to the Superset Principle, a phrasal entry can shrink downwards. Therefore, the phrasal entry that corresponds to \([a[b[c]]]\) can be inserted at the non-terminal nodes cP, bP, or aP. However, it is crucial that this shrinking is downwards. Thus, the entry that corresponds to \([a[b[c]]]\) can never lexicalize aP to the exclusion of bP or cP. It is also very important that the entry that corresponds to \([a[b[c]]]\) is not inserted to lexicalize bP if there is also an entry that can fully match \([b[c]]\). This constraint is named **Elsewhere Principle** (also minimize junk), which stipulates that the entry with the minimum number of unmatched features wins.

A simple example to illustrate these two principles is given in (6). Let us assume that the entry *fish* /fɪʃ/ corresponds to plP but not nP as in (6). In this scenario, the lexicalization of the structure nP (e.g. before the merger of pl\(^o\) during derivation) is only possible if the entry in (6) shrinks down to nP in accordance with Superset Principle.

\[
\begin{array}{c}
\text{plP} \\
\text{nP}
\end{array} \sim /fɪʃ/\quad \begin{array}{c}
\text{pl}^o \\
\text{n}^o
\end{array} \quad \text{FISH}
\]

In the alternative scenario, i.e. if the lexicon has two /fɪʃ/ entries that correspond to nP and plP, the Elsewhere Principle will ensure that the structure nP is matched by the /fɪʃ/ entry that corresponds to nP but not plP. As the /fɪʃ/ entry that corresponds to plP will have an extra feature (i.e. pl\(^o\)) that will remain unmatched, the /fɪʃ/ entry that corresponds to nP will win.

### 2.2 First Phase Syntax

Comparable to the related work of Hale & Keyser (1993) and Borer (2005), Ramchand (2008) proposes her theory of **First Phase Syntax**. In FPS, the event structure is compositionally built and linked to the event participants in syntax.

She argues that three functional heads are employed in the composition of dynamic events, namely `init^o > proc^o > res^o`. The head res\(^o\) indicates the result state of a dynamic event. Every CoS event incorporates res\(^o\) head in its composition. The functional head init\(^o\) denotes a causing/initiation event and is present in every dynamic event that has an overt initiator. The functional head proc\(^o\) is the core component of every eventive, as opposed to stative, eventuality. Therefore, every dynamic event obligatorily incorporates proc\(^o\).

\^6 \text{x>y: x is merged higher than y according to fseq.}
Although Ramchand employs a compositional approach for event structure, she does not assume that verbal roots are totally devoid of syntactic information (contra DM). Rather she proposes that the roots are stored with the tagging information, i.e. syntactic category features corresponding to the functional event heads. In (7) below is the tagging information for different event types.

(7) \([\text{init} \quad [\text{proc} \quad [\text{res}]]] \sim \text{Transitive CoS Events}\)
\([\text{proc} \quad [\text{res}]] \sim \text{Unaccusative CoS Events}\)
\([\text{init} \quad [\text{proc}]] \sim \text{Dynamic Non-CoS Events}\)

Ramchand (2008) assumes an association model that matches eventive features with the roots. Yet, she does not assume a separate root category that is merged in the derivation. Rather, she assumes that roots associate with the eventive heads as illustrated for the verb \(\text{push}\) in (8).

(8) \[
\begin{array}{c}
\text{initP} \\
\text{init}^o \\
/pʊʃ/ \\
\text{procP} \\
/\text{proc}^o \\
\end{array}
\]

Thus, the nature of the eventive specification appears to be a non-structured set of features associated with a lexical entry as in (9).

(9) \{/pʊʃ/, [init; proc], PUSH\}

In the next section, I will argue that the unification of event composition in Ramchand (2008) and the cyclic spell-out mechanism in Pantcheva (2011) is a necessary move to deliver the empirical facts in PL. In particular, I will propose that ROOT has a privileged status in syntax as a syntactic object that has no inner syntax yet still can be an independent node for lexical insertion.

3. PROPOSAL

In this section, I will argue for a cyclic compositional system that attempts to unify Ramchand (2008) and Pantcheva (2011). In particular, I will claim that Ramchandian event heads [res, proc, init] build on a ROOT within a cyclic compositional model proposed in Pantcheva (2011). Accordingly, the specifi-

7 I use capital letters to signify ‘conceptual information’.

8 This may be interpreted as an ambiguous head/phrase status.
ation of roots for eventive features will be phrasal as in (10) below as opposed to the unstructured association model depicted in (9) above.

(10)  /prɔʃ/  ~  initP

\[\text{init} \quad \text{procP} \quad \text{PUSH}\]

The motivation behind the unification proposal is empirical. Thus, it is necessary that we analyze the imperfective formatives in PL in Ramchand’s system, which is attempted in the following subsection.

### 3.1 Analysis of case and aspect marking in PL

In Ramchand (2008), in the specifier of the init head is introduced the external argument. This assumption trivially accounts for the inherent ERG case on the external argument in PL. However, there is not much evidence for the inherent or structural status of NOM in PL. I will simply assume that NOM is also an inherent case introduced in the specifier of the lower eventive head proc or res as it does not show any case alternation and is semantically invariant.

The more interesting empirical prediction of Ramchand’s event composition system appears to be relevant for the imperfective formatives. Remember that we have identified that the variation in the imperfective formatives is root-dependent. Yet, there is strong evidence that the allomorphic variation is not trivial in that the sets of roots that require a certain imperfective morpheme are semantically coherent. The potential analysis that treats the imperfective morphemes as diacritic information specified to each verbal root will certainly fail to account for the empirical generalizations. As discussed in Section 1.2, the imperfective morphemes seem to correspond to different event structures. The correspondences are repeated in (11) below.

(11)  -um  ~ transitive change of state events

-ur  ~ unaccusative change of state events

-am  ~ transitive/unergative non-change of state events (activity)

---

9 See Demirok (2013) for an investigation of the case system in PL.

10 This will depend on the whether the NOM argument is just an UNDERGOER or also a RESULTEE. See Ramchand (2008) for the details.
The preliminary structural representations of the imperfective morphemes in PL with the event heads in Ramchand (2008) are given in (12) below.

\[ (12) \quad \begin{array}{c}
\text{/-um/} \\
\text{impfP} \\
\text{impf} \\
\text{init} \\
\text{proc} \\
\text{res} \\
\end{array} \quad \begin{array}{c}
\text{/-ur/} \\
\text{impfP} \\
\text{impf} \\
\text{proc} \\
\text{res} \\
\end{array} \quad \begin{array}{c}
\text{/-am/} \\
\text{impfP} \\
\text{impf} \\
\text{init} \\
\text{proc} \\
\end{array} \]

Notice the tentative structures that imperfective morphemes can lexicalize contain an imperfective head/feature in addition to eventive heads. This can capture the fact that event structure marking is only available when the syntactic structure contains the imperfective head/feature. However, assuming that the eventive heads are lexicalized by the imperfective morphemes in PL creates a theory-internal problem which constitutes the basic argument for the proposal in this paper. The problem at hand will be discussed in the following section.

3.2 The problem

The assumption that roots are not devoid of eventive specification is the core aspect of the event composition system in Ramchand (2008). Rather, roots seem to overtly lexicalize the eventive heads and do not correspond to a separate position/node in syntax.

If the eventive heads can in fact be lexicalized by the imperfective morphemes in PL, a serious problem occurs: what does the root morpheme lexicalize? In Ramchand’s (2008) association model, there is nothing that can ensure that the root morpheme can be inserted in a syntactic node when all eventive heads are lexicalized by separate morphemes. In (13), I illustrate the problem that occurs when both the root morpheme and the imperfective morphemes are eligible to lexicalize the eventive heads. It is not clear how the root morpheme finds a matching syntactic node if it is assumed that the imperfective morphemes can and do lexicalize the eventive heads. Notice that there is no problem when the structure is not imperfective.

\[ (13) \quad \begin{array}{c}
\text{[past][proc][res]} \quad \rightarrow \quad /\text{ɣur}/ \\
\text{[proc][res]} \quad \rightarrow \quad /\text{ɣur}/ + /\text{i}/ = X \text{ died} \\
\text{[impf][proc][res]} \quad \rightarrow \quad /\text{ɣur}/ + /\text{ur}/ = X \text{ is dying} \\
\end{array} \]
In the following section, I will propose that the adaptation of Ramchand (2008) into the lexicalization mechanism in Pantcheva (2011) solves the problem at hand. Furthermore, I will present data that provides further evidence for the claim that eventive heads and root can be lexicalized separately, contra Ramchand (2008).

### 3.3 A solution by unification

Assuming the structure in (14) below for the derivation depicted in (13) is a potential move to solve the problem discussed in Section 3.2. But is there any empirical evidence for the proposal at hand? It will be shortly clear that the answer is affirmative.

(14) \[
\begin{align*}
\text{impfP} & \quad \text{pastP} \\
\text{impf} & \quad \text{past} \\
\text{procP} & \quad \text{procP} \\
\text{proc} & \quad \text{proc} \\
\text{resP} & \quad \text{resP} \\
\text{res} & \quad \text{res} \\
\text{ROOT} & \quad \text{ROOT} \\
/ɣur/ & \quad + \quad /ur/ & \quad /ɣur/ & \quad + \quad /i/ \\
[\text{proc[res ROOT]}] & \quad [\text{impf[proc[res]]}] & \quad [\text{proc[res ROOT]}] & \quad [\text{past}]
\end{align*}
\]

Remember that Pantcheva (2011) assumes a cyclic lexicalization mechanism in which every external merge is followed by lexical access. The derivational steps will be the same till procP, which is illustrated in (15) below. Notice that the matching lexical items can correspond to bigger structures than the structure to be lexicalized as ensured by the Superset Principle. This is what happens when the phrasal root entry that corresponds to \([\text{proc [res ROOT]}]\) is chosen to lexicalize the smaller structure resP. In the first lexical access, procP shrinks down to resP. In the second step, the whole procP structure is again matched with /ɣur/.

(15) \[
\begin{align*}
\text{MERGE: } & \quad \text{res + ROOT} \\
\text{L.A.: } & \quad \text{match } [\text{res + ROOT}] \text{ with } \{/ɣur/, [\text{proc [res ROOT]}]\} \\
\text{MERGE: } & \quad \text{proc + resP~/ɣur/} \\
\text{L.A.: } & \quad \text{match } [\text{proc[res ROOT]}] \text{ with } \{/ɣur/, [\text{proc [res ROOT]}]\}
\end{align*}
\]

\[11\] I use strikethrough notation to show that a lexical entry lexicalizes less structure than it can maximally lexicalize. In this case, it means that /ɣur/ is specified as [proc[res ROOT]]; however, it shrinks down to ROOT (i.e. lexicalizes only ROOT).
At this point in the derivation, every syntactic node has a lexical match. If the derivation proceeds with the merger of the [past] as illustrated on the right in (14), the past feature will be lexicalized by the phrasal entry /i/ that corresponds to pastP, triggering the evacuation movement of procP and thereby deriving the correct linear order /ɣur/ + /i/.

If the derivation proceeds with the merger of the impf head, then the lexical access to match the impfP will trigger the evacuation movement of the ROOT so that the phrasal entry that corresponds to /ur/ can be inserted into the structure [impf[proc[res]]]. Note that what triggers this movement is the absence of a separate [impf] morpheme in the lexicon. The phrasal entry /ɣur/ will shrink down to lexicalize only the ROOT, deriving again the correct linear order /ɣur/ + /ur/.

The final picture will look like in (16).

\[\text{(16)}\]

\[
\begin{aligned}
\text{ROOT} = /ɣur/ \\
\text{impfP} = /ur/ \\
\text{impf} \\
\text{procP} \\
\text{proc} \\
\text{resP} \\
\text{res} \\
\text{t}_{\text{ROOT}}
\end{aligned}
\]

3.4 Implications and predictions

The primary prediction of the proposal in this paper is that a lexical entry that corresponds to only ROOT is possible (i.e. as a category devoid of any structural/functional information). That is, ROOT has its own syntactic node/position that is eligible for lexical insertion. The prediction is borne out in that in PL there are root morphemes which are used iff the imperfective morpheme lexicalizes the eventive heads above the ROOT. Therefore, the analysis can also explain an interesting type of root allomorphy. Let us give an overview of the different allomorphic variations in lexical entries that contain minimally ROOT or more structure above it.

3.4.1 ImpfP entries

Due to the contiguity constraint on the lexicalization system, it is predicted that if a lexical entry that contains ROOT can lexicalize impfP, the eventive

---

12 The implementation of the minimize junk principle I have in mind is somewhat different from the standard. See Section 3.5 for the discussion on this issue.

13 See Section 3.4.3 for some discussion regarding the status of ROOTs in syntax.
heads in between are also lexicalized. We find such lexical impfP entries in PL. An example is given in (17) below. Notice that in the absence of an initP entry that contain ROOT, impfP will shrink down to initP and lexicalize the structure [init [proc [ROOT]]].

\[(17) \begin{align*}
\text{a. } & \text{ibgar} /\text{ibgar}/ \sim [\text{impf}[\text{init}[\text{proc}[\text{CRY}]])] \\
\text{cry.IMPF} & \text{‘You are crying.’} \\
\text{b. } & \text{ibgar-i} /\text{ibgar}/ \sim [\text{impf}[\text{init}[\text{proc}[\text{CRY}]])] + /i/ \sim [\text{past}] \\
\text{cry-PST} & \text{‘You cried.’}
\end{align*}\]

It is also possible that separate impfP and initP entries that contain the same ROOT are present in the lexicon. In this case, the impfP morpheme (i.e. [impf[init[proc[EAT]]]]) cannot win to lexicalize the structure [init[proc[EAT]]] due to the minimize junk principle. That is, if the impfP morpheme were inserted, the [impf] feature would remain unmatched, hence the ungrammaticality of (18c). Since both impfP and initP entries contain the same ROOT, there is never any chance for the imperfective morpheme /am/ to win. That is, the presence of an impfP morpheme that contains the same ROOT automatically blocks the impfP morphemes that do not contain the ROOT, hence the ungrammaticality of (18d) and (18e).

\[(18) \begin{align*}
\text{a. } & \text{imxor} /\text{imxor}/ \sim [\text{impf}[\text{init}[\text{proc}[\text{EAT}]])] \\
\text{eat.IMPF} & \text{‘You are eating’} \\
\text{b. } & \text{ʃk’om-i} /\text{ʃk’om}/ \sim [\text{init}[\text{proc}[\text{EAT}]]) + /i/ \sim [\text{past}] \\
\text{eat-PST} & \text{‘You ate.’} \\
\text{c. } & \ast \text{imxor-i} \\
\text{d. } & \ast \text{imxor-am} \\
\text{e. } & \ast \text{ʃk’om-am}
\end{align*}\]

3.4.2 initP and procP entries

It is also trivially predicted that there are initP and procP entries that contain ROOT. In (19) is the example that illustrates an initP entry that shrinks down to lexicalize only the ROOT when the imperfective feature is merged and is to be lexicalized.

\[(19) \begin{align*}
\text{a. } & \text{t’ax-i} /\text{t’ax}/ \sim [\text{init}[\text{proc}[\text{res}[\text{BREAK}]])] + /i/ \sim [\text{past}] \\
\text{break-PST} & \text{‘You broke.’}
\end{align*}\]
b. \( t'ax\)-\textit{um} \quad /t'ax/ \sim [\text{init}[\text{proc}[\text{res}]]] + \text{break-IMPF} \\
\quad /\text{um}/ \sim [\text{impf}[\text{init}[\text{proc}[\text{res}]]]] \\
\quad \text{‘You are breaking.’}

The example in (20) illustrates a procP entry that shrinks down to lexicalize only the ROOT when the imperfective feature is merged and is to be lexicalized.

(20) a. \( \text{ʤol}-i \) \quad /\text{ʤol}/ \sim [\text{proc}[\text{res}[\text{FALL}]]] + /i/ \sim [\text{past}] \\
\quad \text{fall-PST} \\
\quad \text{‘You fell.’}

b. \( \text{ʤol}-\text{ur} \) \quad /\text{ʤol}/ \sim [\text{proc}[\text{res}[\text{FALL}]]] + \\
\quad \text{fall-IMPF} \\
\quad /\text{ur}/ \sim [\text{impf}[\text{proc}[\text{res}]]] \\
\quad \text{‘You are falling.’}

3.4.3 ROOT entries

PL lexicon seems to have lexical entries that merely correspond to the ROOT. The evidence for this claim is that PL exhibits an interesting pattern\(^{14}\) of root suppletion, an example of which is in (21) below. As (21) shows, the suppletive roots are accompanied by regular morphology.

(21) a. \( \text{zit’}-\text{am} \) \quad /\text{zit’}/ \sim [\text{SAY}] + /\text{am}/ \sim [\text{impf}[\text{init}[\text{proc}]]] \\
\quad \text{say-IMPF} \\
\quad \text{‘You are saying (it).’}

b. \( \text{tkv}-\text{i} \) \quad /\text{tkv}/ \sim [\text{init}[\text{proc}[\text{SAY}]]] + /i/ \sim [\text{past}] \\
\quad \text{say-PST} \\
\quad \text{‘You said (it).’}

c. \( \text{tkv}-\text{a} \) \quad /\text{tkv}/ \sim [\text{init}[\text{proc}[\text{SAY}]]] + /\text{a}/ \sim [\text{optative}] \\
\quad \text{say-OPT} \\
\quad \text{‘May you say (it)!’}

d. *\( \text{tkv}-\text{am} \)

e. *\( \text{zit’}-\text{i} \)

f. *\( \text{zit’}-\text{a} \)

The lexical entry /\text{zit’}/ seems not to have any event specification as the ungrammaticality of (21e) and (21f) indicates. The eventive heads merged above the ROOT remain unlexicalized in (21e) and (21f). The existence of separate ROOT and initP entries that contain the same ROOT is what creates the root allomorphy in (21). The ROOT entries can only be used with

\(^{14}\) There are several examples of this pattern:

(i) \( \text{ik}-\text{um} \ ‘\text{do-IMPF}’; \text{xven}-\text{i} ‘\text{do-PST}’; *\text{ik}-\text{i} \text{ and } *\text{xven-um} \)

(ii) \( \text{mo}+\text{l}-\text{ur} \ ‘\text{PV+come-IMPF}’; \text{mo}+\text{x}’-\text{i} \ ‘\text{PV+come-PST}’; *\text{mo}+\text{l}-\text{i} \text{ and } *\text{mo}+\text{x}’-\text{ur} \)
the regular imperfective morphemes, i.e. /am/, /um/, /ur/. This is predicted as this is the only case where the event heads above the ROOT are not lexicalized by the initP or procP entries. The form in (21d), on the other hand, is blocked due to the minimize junk principle. As there is a more specific entry (i.e. the ROOT entry /zit'/), the bigger initP entry cannot be used. This pattern of root allomorphy that is accompanied by regular morphology may be considered to be robust evidence for the existence of ROOT node in syntax. An alternative analysis to argue against a ROOT node in syntax would be forced to list in the lexicon all the entries that show root allomorphy although there is nothing that is irregular regarding the morphology they require. It should also be noted that root suppletion is only induced by the imperfective aspect in PL. There is no such pattern available for past, optative, or future, which might indicate that root suppletion is not random at all.

The implication of the data above is that ROOT seems to correspond to a separate syntactic node/position that is eligible for lexical insertion (i.e. alone or with projections above it). As the reviewers pointed out, the status of the ROOT, however, has not been so far clarified in this study. The ROOT merged in syntax cannot correspond to a phonological matrix as it is assumed that there is a later matching between a ROOT and a lexical entry. A ROOT node is also very unlikely to be an empty placeholder. If it were, we would not be able to distinguish between different ROOTs and eventually cannot capture how the ROOT interacts with the heads above it in terms of lexicalization. The most promising idea which is also compatible with the proposal at hand seems to be the proposal that a ROOT is an index (Acquaviva 2009).15

Another important question with respect to a ROOT node that is independent of any syntactic head above it is whether there is independent evidence for its existence. Is it only PL that exhibits such an evidence? Typological facts inform us that the answer should be negative. There are languages which derive transitives from intransitives. Likewise, there are languages which derive intransitives from a transitive base. Yet, there is another type of languages which exhibit equipollent transitivity marking (Haspelmath 1993). Such languages have both derived transitives and derived intransitives (Schäfer 2009).16

15 According to Acquaviva (2009: 16), “[...](the) root DOG acts as an index [...] In the abstract syntactic representation [...], roots do not mean anything by themselves, but act as name-tags which define identity and difference. Their function is differential, not substantive, like that of the indices 1 and 2 in he 1 likes broccoli, but he 2 doesn’t. What is crucial before spellout [...] is that the root nodes DOG and CAT should be distinct”.

16 Turkish ya-k ‘burn (tr)’ and ya-n ‘burn (intr)’ is an example of this. /ya/- cannot stand alone, i.e. it gains an intransitive or transitive sense with the suffix that follows. Another
3.5 The Minimize Junk Problem

Under the proposal that eventive heads are built on a ROOT and the imperfective exponents in PL may lexicalize eventive heads, there is an important technical issue that needs to be addressed.\footnote{I owe my thanks to the anonymous reviewers for bringing up this important issue.} Observe the problem in (22) below. The predicted lexicalization for the syntactic structure in (22) is */t’ax+am/* while the attested lexicalization is /t’ax+um/* as illustrated in (23). This problem stems from the ban on the lexicalization with more junk when a lexicalization with less junk (i.e. unmatched node on the exponents) is available, i.e. the minimize junk principle. I will claim that there are at least two potential ways to understand and implement this principle. First let us see what the standard understanding yields. When the ROOT is lexicalized along with imperfective morpheme, the lexicalization with the least junk should be */t’ax+am/*. The lexicalization /t’ax+um/ leaves out [init[proc[res]]] nodes of /t’ax/ unmatched while the lexicalization */t’ax+am/ leaves out [init[proc]] nodes of /t’ax/ unmatched. Hence, the lexicalization with the less amount of total junk is */t’ax+am/*.

(22) lexical entries syntactic structure
| /um/   | [impf [init[proc[res]]]] | [impf [init[proc[res BREAK]]]] |
| /am/   | [impf [init[proc]]]      |
| /t’ax/ | [init[proc[res BREAK]]]  |

(23) a. prediction: [impf [init[proc[res BREAK]]]]

b. attested: [impf [init[proc[res BREAK]]]]

The standard implementation of the minimize junk principle that yields (23) is as follows: the selection of /am/ vs. /um/ (neither of which have any junk) is considered to be dependent on the junk of another lexical entry i.e. /t’ax/. That is, the amount of junk is computed globally, not locally. I will tentatively propose that there is an alternative (and perhaps simpler) way of implementing the minimize junk principle, which also solves the problem at hand. I will claim that the total override effect seen in child~children or go~went cases are also true for partial match cases. The item that wins is the one that maximally matches the structure downwards. The minimize junk principle is relevant for the selection of one and only one specific lex-

---

example of equipollent transitivity marking is Japanese atum-eru ‘gather (tr.)’ and atum-aru ‘gather (intr.)’ (Haspelmath 1993). Thus, it is not very inconceivable to map Turkish /ya-/ and Japanese /atum-/ to the ROOT node proposed in this study.
The status of roots in event composition: Laz

When selecting a lexical entry, the system does not consider the amount of junk found in another lexical entry. Hence, the system does not make a global computation of junk, rather junk computation is local. The proposal is illustrated in (24) below.

(24)

<table>
<thead>
<tr>
<th>Lexical Entries</th>
<th>Syntactic Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>/x/</td>
<td>[a[b[c]]]</td>
</tr>
<tr>
<td>/y/</td>
<td>[b[c[d]]]</td>
</tr>
<tr>
<td>/z/</td>
<td>[a[b]]</td>
</tr>
</tbody>
</table>

Lexicalization: Standard Version Proposed Version

/y/+z/ [total junk computed] /y/+x/ [local junk computed]

In (24) above, the merger of [a] triggers the lexical access for the best match. If there was an entry that had the specification [a[b[c[d]]]], it would have to win as it would be the full match for the whole structure. But since there is no such entry, the closest to full match is chosen, i.e. the one that lexicalizes the most. This is still a type of override effect, albeit partial. The minimize junk principle is not an issue here as there is only one candidate that can lexicalize [d]. Notice that if there was an entry [d], we would be forced to choose it due to minimize junk. In (24), the [b[c]] nodes of the entry /y/ happen to be junk. Yet, this is not an issue in the proposed implementation. As the junk is not globally computed, the amount of junk /y/ has is just the indirect outcome of what lexicalizes the higher structure. However, it is crucial for our purposes that maximal matching/override effect may lead to shrinking till ROOT node. The biggest empirical advantage of the analysis is that we can now capture the root suppletion cases with no recourse to unnecessary listing. The proposal, if on the right track, can explain the partial override effect in the selection of the imperfective exponent. If we need to go back to the original problem, the prediction of the proposed alternative of the minimize junk principle is now /t’ax+um/, not */t’ax+am/.

4. CONCLUSION

In this paper, I have attempted to unify two nanosyntactic models proposed in Ramchand (2008) and Pantcheva (2011) with the aim of accounting for the empirical facts regarding the interaction of the aspectual system and event structure composition in Pazar Laz. The basic idea defended in this work is that the roots in event composition should have a status to the ex-

---

18 See Section 3.4.3 for an example of this case, i.e. the ROOT entry.
tent that they occupy a syntactic position and can be target nodes for lexical
matching. What differentiates a syntactic feature/head from a root in a syn-
tactic derivation is that the latter is a privileged syntactic object that comes
with no inner syntax and thereby is neither a head nor a phrase. The cyclic
and phrasal lexicalization system combined with the assumption that roots
can occupy syntactic positions appears to be promising in that it presents the
adequate theoretical apparatus to be able to derive the attested allomorphic
variations involving roots and outlaws the unattested patterns.

REFERENCES

York: University of York.

University Press.

Cinque, G. (2002). Functional structure in DP and IP. The cartography of
syntactic structures, Volume 1, New York: Oxford University Press.

Boğaziçi: Boğaziçi University M.A. thesis.


290-306.

Hale, H. & S. J. Keyser (1993). On argument structure and the lexical expression
of syntactic relations. In K. Hale & S. J. Keyser (eds.), The view from Building

Halle, M. & A. Marantz (1993). Distributed Morphology and the pieces of
inflection. In K. Hale & S. J. Keyser (eds.), The view from Building 20, 111-

In A. Carnie & H. Harley (eds.), MITWPL 21: Papers on phonology and

alternations. In B. Comrie & M. Polinsky (eds.), Causatives and transitivity,

Lacroix, R. (2009). Description du dialecte laze d’Arhavi. Lyon: University of
Lumiere Lyon 2 Ph.D. dissertation.

“Workshop on Functional Categories”, Boğaziçi University, 6 October 2011.


Ms., Boğaziçi University.


**Ömer Demirok**

Boğaziçi University

Department of Western Languages and Literatures

34342 Bebek, Istanbul

Turkey

e-mail: omer.demirok@boun.edu.tr