Tackling the Augmentative Puzzle in Turkish

Utku Türk*

Abstract. This paper deals with the suffix *-CIk* in Turkish, and argues that the suffix *-CIk* has an augmentative meaning, as well as a diminutive meaning which has been mentioned in the literature extensively. The different meanings of the suffix *-CIk* are due to two different structures that I propose. While the structure of the diminutive is made of one single morpheme *-CIk*, that of the augmentative suffix is composed of two morphemes: *-CI* and *-k*. I offer phonotactic and morphological evidence that shows *-k* has its own life. I also show strict categorical selection properties for the diminutive and augmentative suffix: The augmentative one can only be attached to adjectival bases that express negative orderings. The diminutive *-CIk*, on the other hand, attaches to nouns. I capture this contrast using the nanosyntax model of spellout.

Keywords. adjectives; augmentative; scalar; gradability; nanosyntax

1. Introduction. I start with some background on the suffix *-CIk* in Turkish.[1] Consider the following data:

BASE	*-CIk* FORM		CONTRIBUTION	STRESS MOVE	K-DELETION
kü.ˈçük	ˈkü.çü -.cük	*'small'*	AUGMENTATIVE	✓	✓
kö.ˈpek	kö.pek-.ˈçik	*'dog'*	ENDEARMENT	*	*
ka.ˈpak	ka.pak-.ˈçık	*'lid'*	DIMINUTIVE	*	*
ge.ˈmi	ge.mi-.ˈcik	*'boat'*	DIMINUTIVE	*	N/A

Table 1: Different *-CIk* formation patterns.

The first thing to notice in the table is the different contributions of the suffix *-CIk*. Previous research only mentions the endearment and diminutive meanings of the suffix (Sebüktekin 1984, Sezer 1981, Taylan 2015, Zimmer 1970) apart from Göksel & Kerslake (2005) mentioning the emphatic nature of the suffix. This gap has led to an over-generalization that calls for ˈkü.çü.cük to be analyzed as the diminutive of kü.ˈçük. However, rather than expressing a slightly reduced degree of the base 'small', it amplifies the smallness meaning. The following data show that the suffix *-CIk* has this property with certain lexical items.

(1) a. al.ˈçak
      low
      'low'

b. yu.mu.ˈşak
   soft
   'soft'

c. yu.var.ˈlak
   round
   'round'

Another difference between the items in Table 1 lies in the different patterns of stress. Turkish stress is mostly word final. Following any regular suffixation, stress moves to the new suffix,

---

[1]*-CIk* has 4 phonological allomorphs: [-ʤik], [-ʤɯk], [-ʤuk], and [-ʤyk]. I use *-CIk* in this paper to refer all of the allomorphs since allomorphy due to the vowel harmony is not relevant for our purposes. I use Turkish orthography throughout the paper, some of which does not match with proper IPA symbols. The following is their IPA counterparts. ü: [y], ö: [ø], ı: [ɯ], ç: [ʧ], c: [ʤ], ş: [ʃ].

maintaining its final position. The previous examples show that when the contribution of the suffix is AUGMENTATIVE, stress falls on the non-*cIk* part according to the Sezer stress rule (Sezer 1981).

(2) a. ˈal.ça.-cık
      low-AUG
      'very low'

    b. yu.ˈmu.şa.-cık
      soft-AUG
      'very soft'

    c. yu.ˈvar.la.-cık
      round-AUG
      'very round'

The last difference between the items is the deletion of root-final k sound. The grouping with regards to k-deletion is again AUGMENTATIVE vs. ENDEARMENT/DIMINUTIVE. Previous research does not mention such a contrast. K-deletion is regarded as a phonological process (Sebüktekin 1984, Zimmer 1970). However, it is clear that k-deletion is not related to phonological processes when considereing the ENDEARMENT and DIMINUTIVE examples in Table 1. There is no phonological or acoustic reason for the k sound to be dropped, and Turkish allows for the adjacency of *k* and *c* when separated by a syllable boundary[2].

    All of these contrasts align with the word categories distinction. The AUGMENTATIVE contribution is only available with adjectival bases, whereas the DIMINUTIVE and ENDEARMENT contributions are available with nominal bases.

    Now consider the following data in Table 2, which shows all the possible AUGMENTATIVE -*CIk* constructions available in Turkish.

BASE	GLOSS	AUG FORM	GLOSS
alçak	*'low'*	alçacık	*'very low'*
az	*'few'*	azıcık	*'very few'*
dar	*'tight'*	daracık	*'very tight'*
minik	*'tiny'*	minicik	*'very tiny'*
minnak	*'wee'*	minnacık	*'weeny'*
sıcak	*'warm'*	sıcacık	*'very warm'*
toparlak	*'roundish'*	toparlacık	*'very roundish'*
ufak	*'little'*	ufacık	*'very little'*
yumuşak	*'soft'*	yumuşacık	*'very soft'*
yuvarlak	*'round'*	yuvarlacık	*'very round'*
genç	*'young'*	gencecik	*'very young'*
ince	*'thin'*	incecik	*'very thin'*
küçük	*'small'*	küçücük	*'very small'*
kısa	*'short'*	kısacık	*'very short'*

Table 2: The full list -*CIk* formations with a simplex adjectival base.

There are two types of adjectival bases in Table 2: k-ending and non k-ending. It seems that the root-final *k* in bold-faced items is deleted when the suffix -*CIk* is added. Considering that there is no obvious reason for k-deletion, I argue that *k* is not deleted; rather, it is preceded by

---

[2]An anonymous reviewer pointed out that there is actually no phonological word that contains adjacent *k* and *c* even when they are separated by a syllable boundary. This is indeed impossible in Standard Turkish. Due to voicing assimilation, the co-occurance of these sounds will always be pronounced as *k* and *ç*. Then, one can argue that -CIk suffix may not contain an archiphoneme /C/ and the deletion of *k* happens when we have a phonotactically impossible string. However, this would be the only case in Turkish where a phonotactic impossibility leads to a deletion, considering Turkish phonotactics would deal with this with an insertion of an emphatic vowel.

another suffix *-CI*. This hypothesis would mean that adjectives such as the ones above consist of two morphemes, and the augmentative suffix is not *-CIk* but *-CI*. I will support this hypothesis with another morpo-syntactic process, namely causativization, in which the the root-final *k* is detached from the adjectival base.

In this paper, I begin by exploring the semantics behind adjectives in Table 2 and explain the decomposing analysis of adjectives based on Wyngaerd et al. (2020). Before delving into the details of the analysis, I lay theoretical prerequisites for my analysis in Section 3. In section 4, I offer my analysis of AUGMENTATIVE formations with necessary lexical items and trees and show derivational steps.

2. Turkish Adjectives. Consider the following data:

BASE	GLOSS	AUG FORM
yüksek	*'high'*	*yüksecik/*yüksekcik
derin	*'deep'*	*derincik
büyük	*'big'*	*büyükcük/*büyücük
geniş	*'wide'*	*genişcik

Table 3: An exemplary set of adjectives that do not allow *-CIk* affixation.

All of the examples in Table 3 create an ungrammatical utterance when they are suffixed with *-CIk*. The main difference between them and the ones in Table 2 is the ordering they express. They are on the opposing sides of the same spectrum introduced by the dimension. Following Kennedy & McNally (2005), I will group these adjectives into two categories: POS and NEG. To differentiate between such adjectives, tests regarding markedness effects are commonly used. It is shown that POS ordering adjectives are the less marked ones and they are the base element when the degree of the dimension is asked about. Consider the following data:

(3) a. Bu bina   ne kadar geniş / uzun / yüksek?
         this building.NOM what much wide / tall   / high
         'How wide/tall/high is this building?'

     b. * Bu bina   ne kadar dar   / kısa / alçak?
         this building.NOM what much narrow / short / low
         Intended: 'How wide/tall/high is this building?'

When NEG orderings are used for asking the degree of a dimension as in Example (3b), the sentence presupposes that the building is already narrow, short, low, etc. Thus, the intended question turns into a different question which has a presupposition of NEG ordering.

Considering the requirement of an adjective, specifically a NEG ordering one, as a base, the structure of a full adjective has to encode the full scale, the direction of the scale, and the point specified by the adjective itself. For this reason, I assume a decomposed adjectival structure proposed in Wyngaerd et al. (2020) based on the previous work of De Clercq & Wyngaerd (2019), Kennedy & McNally (2005), Neeleman & Szendroi (2004). The proposed structure is shown in (4a). Every terminal node in this tree represents an element necessitated by the semantics of gradable adjectives. At the bottom of the tree, dimension (DIM) introduces the main theme of the adjective: be it height, length, color, temperature, etc. Direction (DIR) introduces the ordering that scalar adjectives need. This ordering is then reversed by the lower negative (NEG) (De Clercq & Wyngaerd 2019). On this reversed scale, a POINT is introduced.

As stated in Wyngaerd et al. (2020), there is no need to have a specified point all the time; context may also provide the necessary point as a contextual standard. The node POINT's job is to divide the scale into two so that the UP feature makes a selection between the parts.

(4) a.
```
        UPP
       /   \
      UP   POINTP
           /    \
       POINT    NEGP
               /    \
             NEG    DIRP
                   /    \
                 DIR    DIM
```

b.
```
        UPP
       /   \
      UP   POINTP
           /    \
       POINT    NEGP
               /    \
             NEG    DIRP
                   /    \
                 DIR    DIM
```
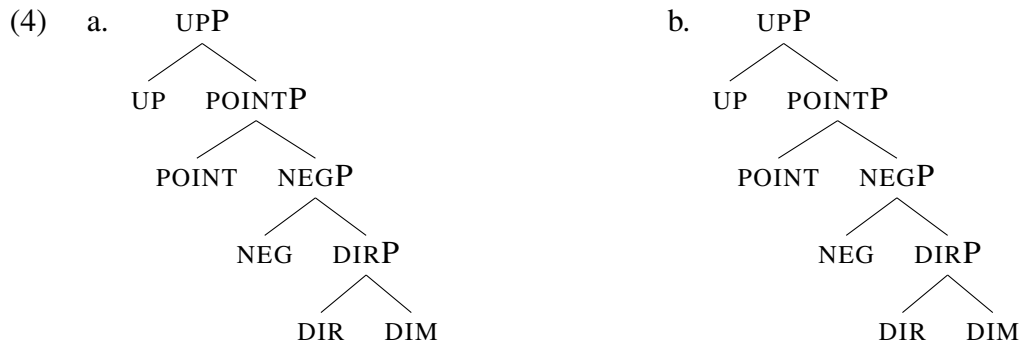
Some lexical items of adjectives in Turkish, like *az*, *genç* or *dar* (see the non-bold ones in Table 2), encompass the entire structure as in (4b). These types of roots are referred to as extra large (XL) roots. Other items, such as *küçük*, *yumuşak*, and *sıcak*, have small (S) roots; they only spellout a tree up to the NEGP, as shown in (5a). The rest of the tree is the final *-k* morpheme.

(5) a.
```
            UPP
           /   \
          UP   POINTP
               /    \
  -k      POINT     NEGP
               /    \
             NEG    DIRP
                   /    \
                 DIR    DIM
```
NEG **S-root:** *alça-, küçü-, yumuşa-*

b.
```
            UPP
           /   \
          UP   POINTP
               /    \
  -k      POINT     DIRP
                   /    \
                 DIR    DIM
```
POS **S-root:** *büyü-, yükse-*
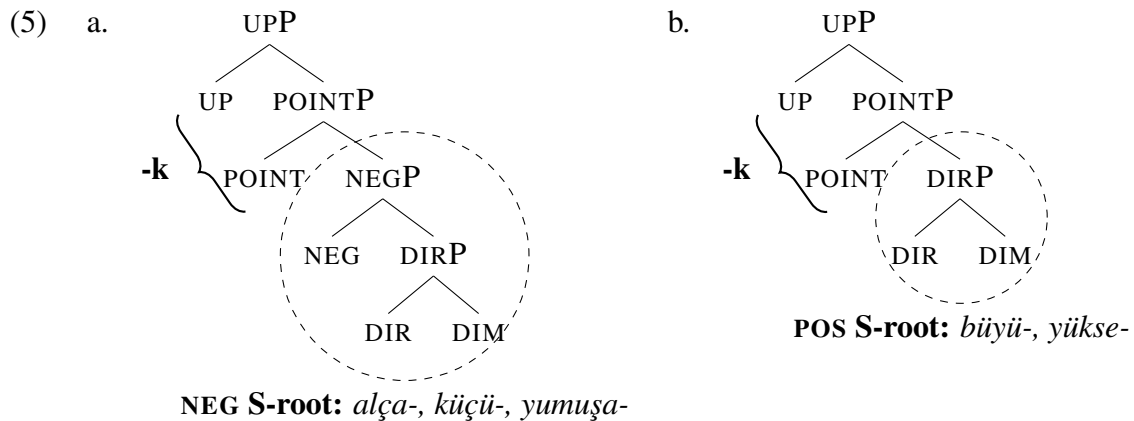
One morphological reflex that shows the difference between XL and S roots is the causativization process. Unlike English, Turkish has a suffixal causative morpheme. When S-roots are causativized in Turkish, the final *-k* is dropped and replaced with a causative morpheme. In the case of XL-roots, there is no modification on the base as shown in Table 4). The difference between XL-roots and S-roots is not only visible in NEG ordering adjectives, but also in POS ordering ones. Some S-root POS ordering adjectives, like *büyük* or *yüksek*, also undergo k-deletion in the causativization process.

	BASE	GLOSS	CAUS	GLOSS
S	alça-k	*'low'*	alça-lt	*'to lower'*
	yumuşa-k	*'soft'*	yumuşa-t	*'to soften'*
	yükse-k	*'high'*	yükse-lt	*'to ascend'*
XL	geniş	*'wide'*	geniş-le-t	*'to widen'*
	az	*'few'*	az-alt	*'to lessen'*

Table 4: Causative formations of some S and XL-root adjectives

This analysis of Turkish gradable adjectives captures the causativization data, as well as the fact that the augmentative suffix has strict selectional properties. Now, we are at least able to show a clear distinction between the adjectives that allow for the annexation of the augmentative *-CI* suffix and those which do not via the polarity of ordering. We achieve this clear picture which can be seen in Table 4 with independently justified syntactic structures and formal semantic applications.

3. Theoretical Ground. The analysis presented in this paper makes use of the nanosyntactic framework. Nanosyntax is a model of grammar that puts the syntax and the formation of constituents prior to the lexicon (Starke 2009). This view on the architecture of language puts heavy stress on how the lexicon is structured. This emphasis on the lexicon is due to how items come to be. After any syntactic tree formation, the spellout algorithm is applied to the prepared tree. Before delving into the details of AUGMENTATIVE structure, I will provide a few details regarding the framework and its machinery. Instead of giving a detailed list of every piece of machinery used by nanosyntax, I will focus on the ones that I use in this paper: phrasal spellout, the Superset Principle, and the spellout algorithm.

3.1 PHRASAL SPELLOUT. The syntactic model I use is founded on the idea that not only terminal nodes, but also phrasal constituents that contain multiple heads can spell out. Consider the tree in (4b). No terminal node has a specified utterance. Only after the whole tree is constructed and every head has made its contribution to the phrase, a pronunciation of a lexical entry is determined. This approach enables us to disentangle the feature bundles and represent hierarchies among features without positing any zero morphemes.

3.2 THE SUPERSET PRINCIPLE. The Superset Principle (Starke 2009) is mainly concerned with what is a candidate for spellout. A spellout may be a candidate for tree α sent by the syntax module only if it contains a subtree of α. Now consider the tree in (4b) again. Following the Superset Principle, the XL-roots under question may also be a spell out of any subtree of the tree dominated by UPP.

3.3 THE SPELLOUT ALGORITHM. The Nanosyntax Framework makes use of a spellout algorithm which is heavily dependent on derivationally building the structure (Baunaz & Lander 2018, Caha et al. 2019, Starke 2018). After every merge of a primitive semantic feature (F), the algorithm actively searches for a lexical match. If a match cannot be found, spellout-driven movements are applied. Consider the following structures. Structures in (6) and (7) show primitive merge when no item in the structure is complex. Structures (8) and (9) show a merge operation and two possible outcomes following the spellout algorithm.

(6) F_1 F_0

(7) F_0 F_1

(8) F_2 F_0 F_1

(9) F_0 F_1 F_2
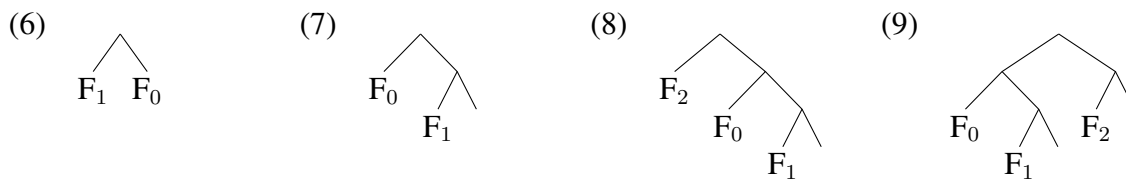
After every merge, lexicon tries to find a match for the structure, as in (6) and (8). If not found, the structure is sent back to syntax once more in order to move the whole complement out without any trace as in (9) and (7) (Caha 2019, Starke 2018). When there is a complex specifier in the structure (10), it is moved out first as shown in (11). Only if this movement does not locate a match in the lexicon, then the whole complement is moved out (12).

(10)
F_3
F_0 F_2
F_1

(11)
F_0 F_3
F_1
F_2

(12)
F_0 F_2
F_1
F_3
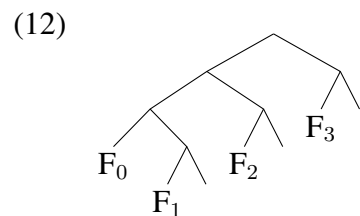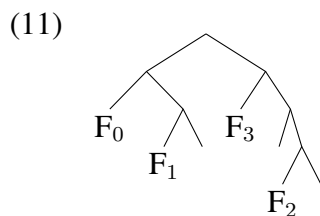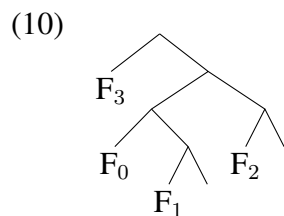
One other important point is when there is a new match for the whole syntactic structure, the spellout of the smaller parts are forgotten. That is, when we have found a match for the whole tree of (9), the spellout of (7), which is also a subtree of (9), becomes irrelevant.

4. Augmentative. I propose the following lexical entry for augmentative in Turkish in (13a). This proposed tree allows us to limit the augmentative suffix to NEG ordered adjective as the Turkish data suggests. NEGP part is a residue from the adjectival tree in (4a). I also clarify what spells out as -*k* in (13b).

(13) a. AUGP ≫ */-CI/*

AUG   NEGP
        |
       NEG

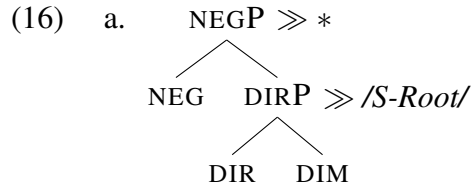b. UPP ≫ */-k/*

UP   POINTP
        |
      POINT
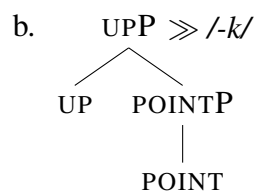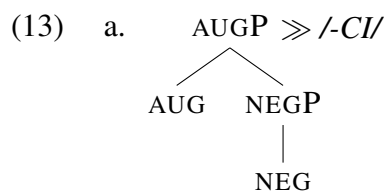
4.1 DERIVATION. I will now derive the type of *-CIk* formation that has AUGMENTATIVE contribution as in (2), that is /S-root + cI + k/. We start by merging DIR and DIM. Following the superset principle, they spellout as an S-root, since the full lexical tree contains the subtree in (14a). Then, we merge NEG semantic prime. The new structure overrides the previous one since it is a better match to the lexical item.

(14) a. DIRP ≫ */S-Root/*

DIR   DIM

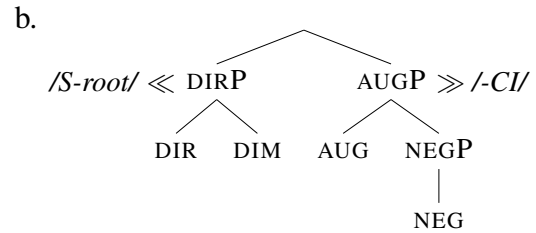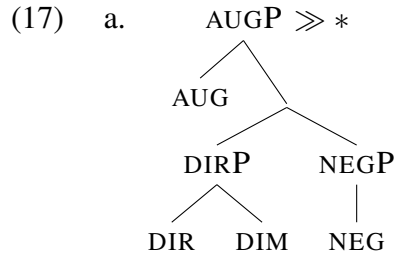b. NEGP ≫ */S-Root/*

NEG   DIRP

DIR   DIM
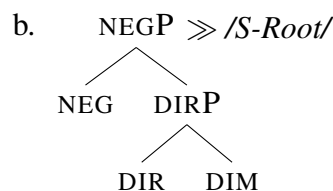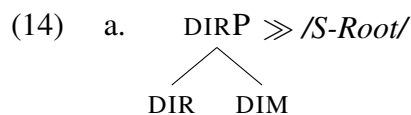
After this successive merge operations, we merge AUG feature to our established structure. Since there is no match for (15a), we move the complement out, which results in a structure as in (15b). Even though we have a match for NEGP, we do not have a match for AUGP. Thus, we backtrack our operations to the previous cycle (Wyngaerd et al. 2020).

(15) a. AUGP ≫ *

AUG   NEGP

NEG   DIRP

DIR   DIM

b.

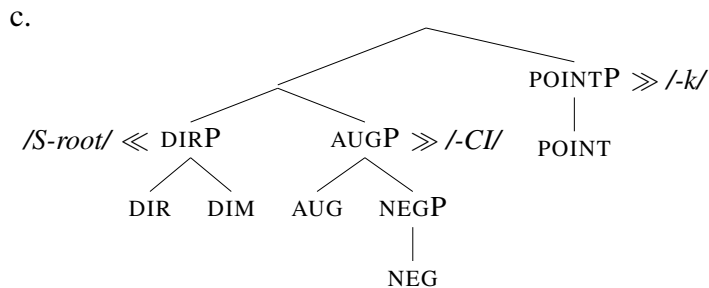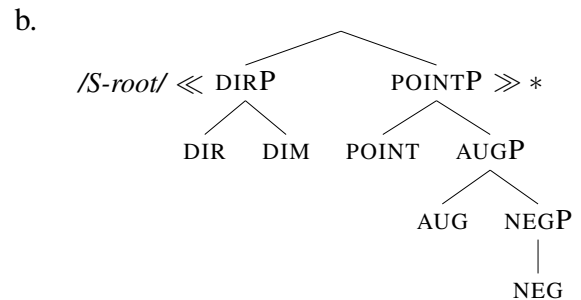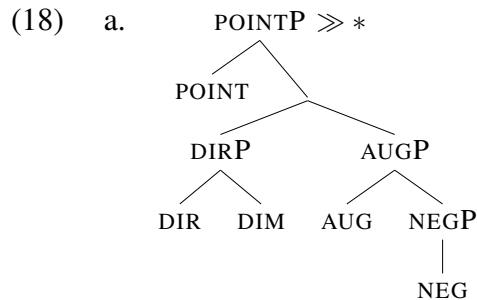*/S-root/* ≪ NEGP   AUGP ≫ *

NEG   DIRP   AUG

DIR   DIM
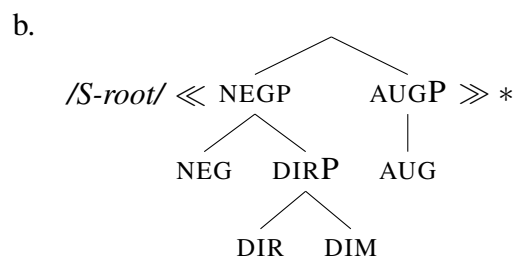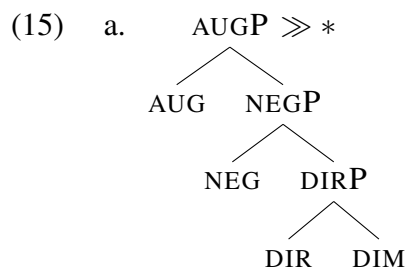
When backtracked, we have a lexical match for DIRP, but not for the whole tree as in (16a). Thus, we move the DIRP out, resulting in (16b). Utilizing the Superset Principle, we can say that both DIRP and NEGP are matched with lexical items, respectively an S-root and */-CI/*.

(16) a.   NEGP ≫ ∗
      　　　／＼
      　NEG　DIRP ≫ */S-Root/*
      　　　　／＼
      　　　DIR　DIM

b.
      　　　／＼
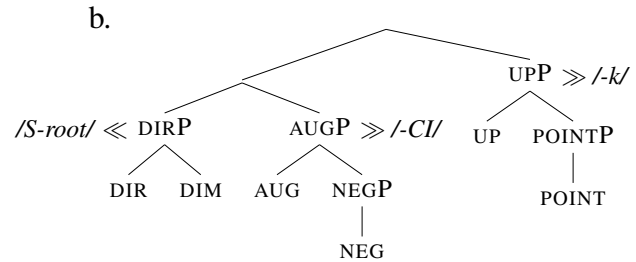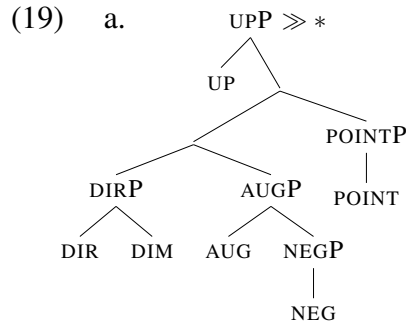   */S-root/* ≪ DIRP　　NEGP ≫ */-CI/*
      　　　／＼　　　　｜
      　　DIR　DIM　　NEG
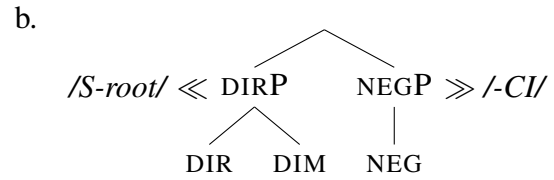
Now that we completed the spellout succesfully, we continue our derivation by merging AUG as in (17a). Again, we do not have any exact match for the whole tree. So, we move out the Spec DIRP first (17b). This spellout driven movement gives us /S-Root/ and /-CI/ as a match.

(17) a.   AUGP ≫ ∗
      　　　／＼
      　AUG
      　　　DIRP　NEGP
      　　　／＼　　｜
      　　DIR　DIM　NEG

b.
      　　　　／＼
      /S-root/ ≪ DIRP　　AUGP ≫ */-CI/*
      　　／＼　　　　／＼
      DIR　DIM　　AUG　NEGP
      　　　　　　　　　｜
      　　　　　　　　　NEG

We, then, merge POINT to our last tree from (17b). This structure in (18a) does not match with any lexical item. Due to having no match, we move the Spec first. Even though we have a spellout for DIRP now, we do not have a lexical match for the rest of the tree. The tree is sent back again to syntax so that spellout-driven movements can be applied. We try to move out the whole complement. The structure in (4.1) has parts that successfully spell out in the lexicon: DIRP as /S-root/, AUGP as *-CI*, and POINTP as *-k* as a virtue of the Superset Principle.

(18) a.   POINTP ≫ ∗
      　　　／＼
      　POINT
      　　　DIRP　　AUGP
      　　／＼　　　／＼
      DIR　DIM　AUG　NEGP
      　　　　　　　　｜
      　　　　　　　　NEG

b.
      　　　　／＼
      /S-root/ ≪ DIRP　　POINTP ≫ ∗
      　　／＼　　　　／＼
      DIR　DIM　POINT　AUGP
      　　　　　　　　／＼
      　　　　　　　AUG　NEGP
      　　　　　　　　　｜
      　　　　　　　　　NEG

c.
      　　　　　　　／＼
      　　　　　　　　　POINTP ≫ */-k/*
      　　／＼　　　　　　｜
      /S-root/ ≪ DIRP　AUGP ≫ */-CI/*　POINT
      　　／＼　　／＼
      DIR　DIM　AUG　NEGP
      　　　　　　　｜
      　　　　　　　NEG

Finally, we merge our UP to our final structure. Due to the fact that there is no match for the structure, we move the specifier of POINT as in (19b). After this movement, we spell out the whole augmented adjective succesfully. Remember that, if we did not have a NEG ordering adjective, we would not be able to spellout */-CI/* which needs [NEGP NEG] or [AUGP AUG[NEG NEGP]].

(19) a.

```
          UPP ≫ *
         /
       UP
      /    \
   DIRP    POINTP
   /  \    /
AUGP  POINT
 ...
```

a.
UPP ≫ *
UP
POINTP
DIRP   AUGP   POINT
DIR DIM   AUG NEGP
NEG

b.
UPP ≫ /-k/
/S-root/ ≪ DIRP   AUGP ≫ /-CI/   UP POINTP
DIR DIM   AUG NEGP   POINT
NEG

5. Conclusion. In this paper, I analyzed the suffix *-CIk* which shows different patterns according to its semantic contribution. Stemming from this difference, I argued that these are indeed different suffixes of different sizes. The AUGMENTATIVE suffix is used with a relatively small number of adjectives, all of which have NEG orderings. The causative patterns and the phonotactics of Turkish pointed towards a decomposition of the suffix: *-CI* and *-k*. This analysis provided a way out from the k-deletion and the productivity mysteries of the so-called *-CIk* suffix. I explained the derivational nature of the word-formation for words like *küçücük, alçacık, yumuşacık*, composed of three morphemes: /S-root/ + /-CI/ + /-k/.

References

Baunaz, Lena & Eric Lander. 2018. Deconstructing categories syncretic with the nominal complementizer. *Glossa* 3(1). 31.1–27.

Caha, Pavel. 2019. *Case competition in Nanosyntax. A study of numerals in Ossetic and Russian*. Ms. Masaryk University.

Caha, Pavel, Karen De Clercq & Guido Vanden Wyngaerd. 2019. The Fine Structure of the Comparative. *Studia Linguistica* 73(3). 470–521. http://dx.doi.org/10.1111/stul.12107.

De Clercq, Karen & Guido Vanden Wyngaerd. 2019. Negation and the Functional Sequence. *Natural Language & Linguistic Theory* 37(2). 425–460.

Göksel, Aslı & Celia J. Kerslake. 2005. *Turkish: A Comprehensive Grammar*. Routledge.

Kennedy, Christopher & Louise McNally. 2005. Scale structure, degree modification, and the semantics of gradable predicates. *Language* 81. 345–381.

Neeleman, Ad & Kriszta Szendroi. 2004. Superman Sentences. *Linguistic Inquiry* 35(1). 149–159.

Sebüktekin, Hikmet. 1984. Turkish word stress: Some observations. In *Proceedings of the Turkish Linguistics Conference*, 295–307.

Sezer, Engin. 1981. On non-final stress in Turkish. *Journal of Turkish studies* 5. 61–69.

Starke, Michal. 2009. Nanosyntax: A Short Primer to a New Approach to Language. *Nordlyd* 36. 1–6.

Starke, Michal. 2018. Complex Left Branches, Spellout, and Prefixes. In Lena Baunaz, Karen De Clercq, Liliane Haegeman & Eric Lander (eds.), *Exploring Nanosyntax*. 239–249. Oxford University Press.

Taylan, Eser E. 2015. *The Phonology and Morphology of Turkish*. Boğaziçi Üniversitesi.

Wyngaerd, Guido Vanden, Michal Starke, Karen De Clercq & Pavel Caha. 2020. How to be positive. *Glossa* 5(1). 23. http://dx.doi.org/10.5334/gjgl.1114.

Zimmer, Karl E. 1970. Some observations on non-final stress in Turkish. *Journal of the American Oriental Society.* 160–162.