Nanosyntactic Analysis of Turkish Case System

Utku Türk & Pavel Caha*

Abstract. This paper takes two challenging characteristics of the Turkish case system and shows that a nanosyntactic analysis can cover both. The first puzzle is that some cases, namely ACC and GEN, in Turkish show alternations between specific and non-specific forms, while other cases like DAT and INS do not. The second puzzle concerns containment relations in morphology. Caha (2009) proposes that cases stand in a containment relation. In some languages like Estonian, Tocharian, and Vlax Romani, the ACC form serves as the foundation of the oblique cases. The puzzle is that in Turkish, the morphological containment holds only for ACC and GEN, but not for ACC and the other obliques. The comparison leads us to expect that the INS in Turkish could be **adam-ı-la*, with the ACC marker to the left of *-la*. Interestingly, this expectation fails precisely in those cases which do not distinguish specific and non-specific forms. We propose a solution to both of these puzzles within the Nanosyntactic framework. The main idea is that Turkish nouns and cases can be composed of smaller, sub-morphemic features. These features allow specificity information to be encapsulated within the noun itself, rather than the case as previously suggested by Öztürk (2005).

Keywords. Turkish; case; nanosyntax; morphology; specificity; containment

1. Introduction. At the first glance, the Turkish case system looks quite simple. Consider, for instance, the paradigm of the common noun *adam* '*man*,' shown in Table 1.[1, 2] The paradigm is agglutinative, with case markers clearly separable from the root/stem.

CASE	man, sg.	man, pl.
NOM	adam	adam-lar
ACC	adam-ı	adam-lar-ı
GEN	adam-ın	adam-lar-ın
DAT	adam-a	adam-lar-a
LOC	adam-da	adam-lar-da
ABL	adam-dan	adam-lar-dan
INS	adam-la	adam-lar-la

Table 1: Turkish case paradigm of a common noun.

However, the picture gets more complicated when we consider some additional facts. In particular, in a proper subset of these cases (namely in ACC and GEN), an overt case marker is used

*We would like to thank Furkan Atmaca, Ömer Demirok, Balkız Öztürk, and Michael Starke for their comments and contribution in the early form of this work. We would also like to thank the anonymous reviewers on Tu+6. Authors: Utku Türk, Boğaziçi University (utku.turk@boun.edu.tr) & Pavel Caha, Masaryk University (pavel.caha@phil.muni.cz).

[1] Abbreviations: 1 = first person, 2 = second person, 3 = third person, ABL = ablative, ACC = accusative, ADJ = adjective, ALL = allative, CAUS = causative, COM = comitative, DAT = dative, EVD = evidential, GEN = genitive, INF = infinitive, INS = instrumental, LOC = locative, NEG = negative, NMLZ = nominalizer, NOM = nominative, PL = plural, POSS = possessive, PROG = progressive, PST = past, SG = singular, WHEN = when.

[2] We use Turkish orthography throughout the paper, some of which do not match with proper IPA symbols. The following are their IPA counterparts. ü: [y], ö: [ø], ı: [ɯ], ç: [ʧ], c: [ʤ], ş: [ʃ].

only when the noun is specific (Enç 1991, Öztürk 2005). When the noun is non-specific, the case marker is omitted. The example (1) shows this for ACC, (2) for GEN. The relevant nouns are in bold.

(1) a. **Kitap** oku-mak kolay-dı. (non-specific)
book read-INF easy-PST.3SG
'Reading a book was easy.'

b. **Kitab-ı** oku-mak kolay-dı. (specific)
book-ACC read-INF easy-PST.3SG
'Reading the book was easy.'

(2) a. **Adam** gel-me-sin-i bekle-m[e]-iyor-du-m. (non-specific)
man come-NMLZ-POSS-ACC expect-NEG-PROG-PST-1SG
'I was not expecting a man to come.'

b. **Adam-ın** gel-me-sin-i bekle-m[e]-iyor-du-m. (specific)
man-GEN come-NMLZ-POSS-ACC expect-NEG-PROG-PST-1SG
'I was not expecting that man to come.'

However, such alternation is impossible for other cases. This statement amounts to two facts: (i) cases other than ACC/GEN cannot be dropped, and (ii) nouns marked by other cases than ACC/GEN may be in principle ambiguous between a specific and a non-specific reading (controlling for orthogonal factors that may disambiguate the readings in one way or another).

Property (ii) is illustrated in (3a). The example features a dative noun *okul-a* 'school, DAT.' As other case-marked nouns, it can be interpreted as specific. However, it also has a non-specific reading. To see that, let's imagine that (3a) is uttered by a politician. Given such a context, hearers will interpret the dative marked *okul-a* as 'to schools throughout the country,' instead of 'to the school.' Analogous observations hold for the overt ablative case (3b).

(3) a. Korona bit-ince öğrenci-ler **okul-a** dön-dü. (ambiguous)
Corona end-WHEN student-PL school-DAT return-PST.3PL
Specific: 'After the Covid-19 passed, the students returned to the university.'
Non-specific: 'After the Covid-19 passed, the students returned to the university (throughout the country).'

b. Çocuk-ken ben-i **palyaço-dan** çok kork-ut-tu-lar. (ambiguous)
kid-when 1SG-ACC clown-ABL a.lot fear-CAUS-PST-3PL
Specific: 'When I was a kid, they scared me from that clown a lot.'
Non-specific: 'When I was a kid, they made me fear clowns a lot.'

Let us now turn to the property (i), namely the ungrammaticality of case-marker omission with other cases than the accusative or the genitive. (4a) and (4b) illustrate this. Their ungrammaticality stems from the fact that neither the word *okul* or *palyaço* is marked with a case.

(4) a. * Korona bit-ince öğrenci-ler **okul** dön-dü.
Corona end-WHEN student-PL school return-PST.3PL
Intended: 'After the Covid-19 passed, the students returned to the school.'

b. * Çocuk-ken ben-i **palyaço** çok kork-ut-tu-lar.
kid-when 1SG-ACC clown a.lot fear-CAUS-PST-3PL
Intended: 'When I was a kid, they made me fear clowns a lot.'

One cannot simply argue that these differences are due to the argument structure since the same effects can be found with either an adjunct, a direct object, or an indirect object.[3]

In order to depict the fact that the marking of ACC/GEN is dependent on specificity, we split each Turkish paradigm (such as those given in Table 2) into two sub-paradigms. The first sub-paradigm (Paradigm$_{\emptyset}$) surfaces when the nouns are non-specific, and the second one (Paradigm$_{overt}$) when the nouns are specific.[4]

	Paradigm$_{\emptyset}$	Paradigm$_{overt}$
NOM	-Ø	-Ø
ACC	Ø	-i
GEN	Ø	-in
DAT	-a	-a
LOC	-da	-da
ABL	-dan	-dan
INS	-la	-la

Table 2: Our proposed sub-paradigms for Turkish cases.

The shading in the table brings out the main puzzle we set out to explore, which is that the two sub-paradigms differ only in ACC and GEN (shaded). In other cases, we find forms that are ambiguous between the two paradigms. In this paper, we shall treat this as syncretism (identity of form for two different grammatical meanings). Our goal is to provide an account of this syncretism (and non-syncretism in ACC/GEN) in the framework of Nanosyntax (Starke 2018). In sum, the issue that we want to answer in this paper is why and how the specific/non-specific dichotomy is only relevant for two cases and not the rest.

The paper is organised as follows. In Section 2, we formulate a generalisation that relates our main puzzle (the absence of specificity distinctions in the oblique cases) to the morphology of the case markers. In Section 3, we sketch our assumptions about the underlying structure of specific and non-specific nouns, as well as our assumptions about case features. We also present here a detailed analysis of the specific/non-specific distinction and its correlation with the morphology of case. Section 4 concludes.

2. Decomposing the genitive. Looking at the paradigms in Table 2, an interesting generalization emerges. The generalization is that the two cases which make the specific/non-specific

[3]There is a small number of special verbs where the dative case can also be deleted as in (5). We leave such examples for future work.

(5) a. Dışarı(-ya) çık-tı-m.
outside(-DAT) exit-PST-1SG
'I went outside.'

b. Kimya(-ya) çalış-tı-m.
chemistry(-DAT) study-PST-1SG
'I studied chemistry.'

[4]We note that even though specificity is an important factor, there are other semantic features important for the dichotomy between marked/unmarked ACC and GEN. In this paper, we investigate how the split works for specificity, and we leave it for future research how the additional properties can be fitted into the proposal to be developed.

distinction both begin with an *-i* (i.e., when they have an overt marker). If this *-i* was analysed as a separate morpheme, we could unify the two cases by saying that only cases that contain this morpheme have a non-specific version. We depict the proposed segmentation in Table 3.

	Paradigm$_{\emptyset}$	Paradigm$_{overt}$
NOM	-Ø	-Ø
ACC	Ø	-i
GEN	Ø	-i-**n**
DAT	-a	-a
LOC	-da	-da
ABL	-da-**n**	-da-**n**
INS	-la	-la

Table 3: Our proposed sub-paradigms for Turkish cases as genitive split.

Abstracting away from the generalisation that only *-i* initial case markers show specificity alternations, the segmentation in Table 3 has one independent advantage and one disadvantage. We discuss the advantage first.

The advantage relates to the analysis of the ablative. Specifically, once *–n* is identified as a separate morpheme, it becomes possible to decompose also the ablative *dan* into two morphemes, which is something that we show in the relevant cell of the table. The decomposition of the ablative into a locative marker (*-da*) and a genitive marker (*-n*) is a cross-linguistically common pattern of marking (Noonan & Mihas 2007). We illustrate this pattern in (6) for Meithei (a Tibeto-Burman language, Chelliah 1997).

(6) a. mə-si əy-**gi** yum -ni.
this I-GEN house is
'This is my house'

b. yum-**də** mí məri ləy.
house-LOC men four live
'Four people live in the house.'

c. mánə Nyurk-**tə-gi** Jaipur-də čətkhí.
he New York-LOC-GEN Jaipur-LOC went
'He went from New York to Jaipur.'

The point of the data in (6) is to show that the decomposition of the ablative *-dan* into a locative *-da* and a genitive *-n* is independently justified, which supports the initial decision to treat *-n* as an independent genitive marker.

Let us now proceed to a problem with the decomposition. We find it in the domain of vowel-final stems. Previously, we have used the stem *adam* 'man,' which ends with a consonant. When we have a V-final stem such as *kapı* 'door,' the containment relationship between the ACC and GEN does not hold. Specifically, the ACC has an 'epenthetic *y*,' yielding *kapı-**yI*** 'door-ACC.' On the other hand, the GEN surfaces with an 'epenthetic *n*,' yielding *kapı-**nI**-n* 'door-GEN,' rather than the predicted **kapı-**yI**-n*.[5]

[5]We are aware that we use the word epenthetic liberally here. We note that while the sounds *s* and *n* only appear with certain lexical items, *y* is a elsewhere epenthetic consonant.

Before we address this issue, it is worth noting that this problem does not arise in all dialects of Turkish. For example, in the Çorum dialect, containment is observed also after V-final stems, since GEN has the expected *-yI-n*.[6]

(7) Çocuk sokak-ta-ki **araba-yın** bir-in-e çarp-mış.
kid street-LOC-ADJ car-GEN one-POSS-DAT hit-EVD
'The kid hit one of the cars in the street.'

Similarly, in both Uzbek and Azerbaijani, Turkish's closest relatives, containment between ACC and GEN is observed regardless of the stem. Table 4 shows the declension of the V-final root 'father' in Turkish, Azerbaijani, and Uzbek respectively. We see that in Uzbek and Azerbaijani, both endings (i.e., ACC and GEN) have the same consonants.

CASE	father, SG, TR	father, SG, AZ	father, SG, UZ
NOM	baba	ata	ota
ACC	**baba-yı**	**ata-nı**	**ota-ni**
GEN	**baba-nı-n**	**ata-nı-n**	**ota-ni-ng**
DAT (ALL)	baba-ya	ata-ya	ota-ga

Table 4: V-final stems and case endings in Turkish, Azerbaijani, and Uzbek.

Finally, note that also in standard Turkish, there are some V-final roots where GEN surfaces as *-yın*. See the examples in (8), where *-yın* marks GEN on *ne* 'what' and *su* 'water.'

(8) a. **Ne-yin** reng-in-i beğen-di-n?
what-GEN color-POSS-ACC like-PST-2SG
Lit: 'What did you like the color of?'

b. **Su-yun** gel-diğ-i yön-e doğru git-ti-k.
water-GEN come-NMLZ-POSS direction towards go-PST-1PL
'We went towards the direction that water is flowing.'

From these points, we conclude that even though standard Turkish GEN cannot be analyzed in strictly decompositional terms, the idea that the genitive contains the accusative has some independent plausibility. The way we shall approach this is perhaps best illustrated by drawing an analogy to English pronouns. In this domain, it seems clear that the possessive *its* can be decomposed into *it* and *s*. The same decomposition – even if tempting – is impossible in the case of *hi(-)s*, because of the irregular difference in vowel length. At the same time, it seems that the segmentation *hi-s* is both historically plausible and morphosyntactically sensible, it's just that in synchronic terms, the form is irregular and stored as non-decomposable. We indicate this 'intermediate' status of a morphologically complex form rendered opaque through grammaticalisation by placing the form in a box: hi-s. A similar approach will be adopted here with respect to the standard Turkish genitive -i-n.

What this means in concrete terms is that we will first formulate our analysis as if *-i-n* contained two separate pieces, because this allows us to make a better sense of the morphosyntax. Specifically, the decomposition will allow us to explain why it is exactly ACC and GEN

[6]We are aware that *-yI-n* is also used for the portmanteau morpheme for 2SG+POSS+GEN in constructions like *senin arabayın kapısı* meaning 'the door of your car.' Our informant notified us that the word *arabayın* is ambiguous 2SG+POSS+GEN and GEN for him. The specific example provided in (7) forces the GEN reading.

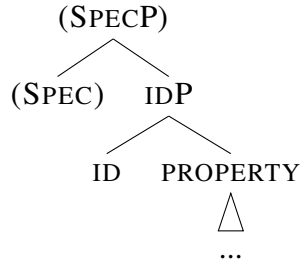
that show the double declension (it is because they share the *-i*). Once this analysis is in place, we shall augment it by proposing that the two underlying pieces (*-i* and *-n*) are stored inside a single lexical item (as a sort of an idiom that comprises independent pieces). This explains why the containment is strictly speaking incorrect for Standard Turkish.

3. The analysis. We will present our analysis in three steps. In Section 3.1, we introduce the features that we assume. In doing so, we follow the Cartographic and Nanosyntactic approach (Cinque & Rizzi 2010, Starke 2010), according to which each feature is an independent head in the syntactic tree. In Section 3.2, we provide a basic overview of our analysis. In Section 3.3, we introduce the technical definitions of the Nanosyntax theory of spellout (Starke 2018). The core idea of this approach is that individual lexical items (both roots and affixes) spell out phrasal constituents containing multiple features. In Section 3.4 and 3.5, our goal is to show how the principles of phrasal spellout – combined with Turkish specific lexical entries – produce the forms of the non-specific and specific paradigm. We show how the Turkish specific paradigm is generated in an algorithmic fashion, including, crucially, the syncretism of specific and non-specific nouns in the oblique cases.

3.1 FEATURES. In the analysis of the Turkish declension, we will need two types of features. The first type of features will be related to case. The second type of features will be related to the nominal denotation, including specificity.

Starting from the nominal features, we will assume that each noun can be decomposed into a PROPERTY part and an identity feature ID. This proposal reflects the idea that having a referential identity index is a core property of nouns (Baker 2003). The feature REF similar to our ID is also used in Harley & Ritter (2002) as the essential feature common to all referring expression, pronouns and nouns in particular. We choose to use ID to not create confusion in the existing literature about non-referentiality. As the tree (9) shows, the feature ID applies to a property and turns it into a referring expression (a noun, IDP per our proposal).

(9)



Specificity will be encoded by the optional feature SPEC that comes on top of IDP. Its optionality is indicated by placing it in brackets. When the feature is present, the noun is interpreted as specific and non-specific when it is absent. Let us stress at this point that there is no expectation that the nominal part will contain three morphemes. Recall that we will adopt the idea that nominal roots may spell out a full phrase.We return to this in Section 3.2.

Let us now turn to case features. As a general point, we will be adding case features on top of the nominal constituent in (9). When case features will be added on top of IDP, this will yield the non-specific declension. When case features will be added on top of SPECP, this will yield the specific declension.

We will represent cases using the features proposed in Caha (2009) (cf. Caha 2013). What Caha proposes is that cases stand in a containment relation. Specifically, the NOM case is the

least marked case, characterized by just one feature K1. The nominative is contained in the accusative (K1+K2), which is in turn contained in the genitive (K1+K2+K3). The whole sequence proposed by Caha (2009) is in (10). The xNP at the bottom of the tree is a variable for different sizes of the nominal structure in (9). The hierarchy proposed in Caha (2009) is motivated by cross-linguistic constraints on case syncretism (only adjacent forms may be syncretic) and by case containment (more marked cases contain less marked cases).

(10) [$_{COM}$ K6 [$_{INS}$ K5 [$_{DAT}$ K4 [$_{GEN}$ K3 [$_{ACC}$ K2 [$_{NOM}$ K1 [xNP]]]]]]]

Some of the facts that led Caha to propose the feature hierarchy (10) are illustrated in the first three columns of Table 5, representing Estonian, Tocharian, and Vlax Romani. Of particular relevance is that in all these paradigms, the ACC form serves as the foundation of all the oblique cases. The oblique cases in some languages perform multiple functions as indicated by the brackets on the left. We can also see syncretism between the ACC and GEN (Estonian), GEN and DAT (Tocharian) as well as INS and COM (Turkish).

CASE	church, SG, EST	horse, SG, TOCH	boy, SG, ROM	man, SG, TR
NOM	kirik	yakw-i	čhav-ó	adam
ACC	kirik-**u**	yakw-**em**	čhav-**és**	adam-**ı**
GEN	kirik-**u**	yakw-**em**-ts	čhav-**és**-k(or)o	adam-**ı**-n
DAT (ALL)	kirik-**u**-le	yakw-**em**-ts	čhav-**és**-ke	adam -a
INS (COM)	kirik-**u**-l	yakw-**em**-mpa	čhav-**és**-ar	adam -la

Table 5: Syncetism & containment in case morphology (Caha 2010)

Interestingly, in Turkish, containment is only observed for ACC and GEN, but not for ACC and the other obliques. The comparison with the other three languages leads us to expect that the INS in Turkish could be **adam-ı-la*, with the ACC marker to the left of *la*. Clearly, this expectation fails; in order to depict it graphically in the table, we have left a blank space in the Turkish paradigm where an *ı* could be expected, but it is in fact not there.

Interestingly, the blank space appears precisely in those cases that do not distinguish specific and non-specific forms, as we have discussed above. This makes us think that the failed expectation is an interesting morphosyntactic problem to address, rather than an artifact of a particular (possibly spurious) analysis.

The reason why we think the problem is independent of the decomposition is the fact that there are several ways in which we could expect *-ı*'s presence in the empty spaces in Table 5. For example, based on the fact that the presence/absence of *-ı* is what distinguishes specific and non-specific interpretation of the direct object, we could analyze *-i* as a specificity marker. Then we would likewise expect the specific comitative form **adam-ı-la* 'with a specific man' (just like we get *adam-ı-n* 'of a specific man'). In other words, the issue of failed morphological containment and a failed semantic distinction coincide.

Therefore, in our discussion of the details of morphological realisation, the central puzzle will be linked to the oblique cases DAT and INS, which are doubly mysterious: from the morphological perspective, the puzzle is that they are simplex at the level of form, even though Caha's proposal in (10) shows that they are morphologically complex. The second puzzle concerns the interpretation: unlike ACC and GEN (which have the *-ı*), the oblique cases are ambiguous between a specific and a non-specific interpretation. In the next section, we will introduce the spellout procedure of Nanosyntax and present our solution to the two puzzles.

3.2 A SKETCH OF THE ANALYSIS. With the features in place, we will now show how the specific and non-specific paradigms can be generated using the Nanosyntax framework. Nanosyntax is a late-insertion model of morphology, which entails that the syntactic structure is built first, and it is then mapped on surface forms. This mapping relies on the interaction between a language invariant spellout procedure (also called the lexicalization procedure) and language-particular lexical entries.

Nanosyntax crucially relies on the idea that a single lexical item can spell out multiple features, which is modeled by means of phrasal spellout. This idea also applies to roots, which too may lexicalise several features (including grammatical features). This is most clearly visible in cases of root suppletion like *bad–worse* (Caha et al. 2019), but the idea applies to roots quite generally (Vanden Wyngaerd et al. 2020, Caha 2021). As a result of this idea, the features of a particular form may be completely spelled out by the root (e.g., *worse* = [Adj+CMPR]), or, if the root is not lexically specified as spelling out all the features of a form, an affix may be needed to spell out grammatical features (e.g., *long-er*).

In order to show our basic idea about how the features of particular forms in Turkish are realised by the individual exponents, we include here the Table 6. This is a so-called lexicalisation table that shows the 'division of labour' between roots and suffixes in realizing all the relevant features of a particular form. The top row of the table gives the full set of features that we have introduced in Section 3.1.

	SURFACE FORM	IDP	SPEC	NOM (K1)	ACC (K2)	GEN (K3)	DAT (K4)
NON-SPEC	adam (NOM)	[adam]		Ø			
	adam (ACC)	[adam]			Ø		
	adam (GEN)	[adam]				Ø	
	adama (DAT)	[adam]					-a
SPEC	adam (NOM)		[adam]				
	adamı (ACC)		[adam]		-ı		
	adamın (GEN)		[adam]		-ı	-n	
	adama (DAT)		[adam]				-a

Table 6: Lexicalisation table that shows phrasal spellouts for the root and the features.

The first four rows of the table depict the feature hierarchy of non-specific nouns. These nouns (recall) lack the SPEC feature. Its absence is indicated by the black shading in the SPEC column. The lower part of the table shows the spellout of the specific nouns.

Our basic idea is that nominal roots in Turkish are lexically specified as spelling out not only the IDP, but also other grammatical features. In particular, our idea is that they can also spell out the specificity feature SPEC and the feature K1 (NOM). For example, we can see that in the bottom part of the table (representing the specific paradigm), the noun *adam* 'man' always spells out the feature SPEC. In NOM, ACC and GEN cells, the root also spells out the nominative feature K1. The remaining features are realized by the suffixes; *-ı* spells out just K2, *-n* spells out K3. We note that this is the place where an adjustment may be needed for standard Turkish: we will need to postulate a portmanteau *-(n)in* for these two cells.

An interesting situation arises in the dative. In the dative, we need to spell out the case feature K4. Our hypothesis is that K4 (the dative feature) is only contained in the ending *-a*, which is, however, also specified as spelling out the features K1, K2, and K3. As a result of

this, the lexicalisation algorithm employed in Nanosyntax requires that *-a* must be inserted starting at its lowest feature K1, which means that the root will have to spell out only the residual features IDP and SPEC. The process through which the root retracts in the dative and leaves the K1 feature to be spelled out by the ending is referred to as 'Backtracking' in Nanosyntax, and we will discuss this feature that delivers the syncretism in detail in Section 3.5. We add that we assume that also INS is spelled out in a way analogous to DAT, just adding an extra K5 feature.

The main idea for the non-specific paradigm we convey with Table 6 is that the lack of the feature SPEC will influence the way spellout works. Specifically, the absence of this feature will prevent the root *adam* from lexicalizing the entirety of its possible maximal tree (unlike what happens in the specific declension.) As a result, in the non-specific declension, the feature K1 will always need to be spelled out by an ending. This is different from the specific paradigm, where K1 is generally spelled out by the root – except DAT and other obliques. The different amount of case features to be spelled out by the endings in the specific vs. non-specific forms is what drives the difference between the endings in the two paradigms in our account. Note, though, that in DAT the amount of case features spelled out by the endings is the same, hence no difference in inflection.

As the table indicates, we assume the existence of a zero ending in the non-specific paradigm, which spells out the case features and which is syncretic among NOM, ACC and GEN. Due to the existence of this zero ending, there is therefore a difference between the specific and non-specific paradigm, because the endings spell out a different number of case features in the respective paradigms. Specifically, the zero ending is not used in the specific declension because it is lexically specified as spelling out K1, but there is no need to spell out K1 in the specific declension.

In sum, the gist of the idea is that the presence/absence of the SPEC feature influences how many features a root spells out (specifically whether it is allowed to spell out K1). Since the root spells out different features in the specific and non-specific declensions, this leaves a different number of case features for realization by the case endings, and indirectly leads to there being two declensions. This approach is unlike the one in Öztürk (2005), where case endings are linked to specificity directly.

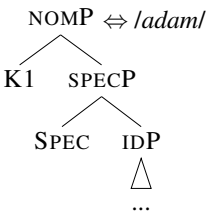
We think that the indirect relationship between case endings and specificity is a good feature of the analysis. If we proposed, for instance, that *-ı* spells out the specificity feature, it would be challenging to explain why we do not find the specific/non-specific distinction in the dative and instrumental, recall the impossible hypothetical form **adam-ı-la* 'with a specific man.' Instead, we propose that SPEC is always spelled out by the root, and the case endings reflect this indirectly via the spell out algorithm proposed in Nanosyntax. We introduce this algorithm in the next subsection.

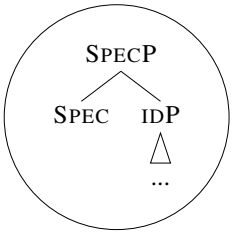
3.3 THE SPELLOUT ALGORITHM. Before diving into the details of specific case spell-outs, let's provide some background concerning the machinery we use in this paper. We assume here a post-syntactic spellout model (Nanosyntax) where morphosyntactic structures are first built by syntax, and only after they have been assembled, they are (cyclically) mapped onto pronunciation following a language-invariant spellout procedure. The procedure has access to a language-specific post-syntactic lexicon, which stores links between syntactic structures (S) and phonology (P). Matching between syntax and the lexicon is based on the so-called Super-

set Principle, see (11).

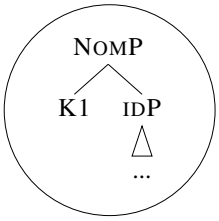
(11) *The Superset Principle* (Starke 2010)
A lexically stored tree L matches a syntactic node S iff L contains the syntactic tree dominated by S as a subtree.

This principle is the core feature of the framework that allows for syncretism. Suppose, for example, that we store a lexical entry as in (12a). This lexical item can be used to spell out any constituent it contains: an example is shown in (12b). We need this matching principle to be able to spell out different amount of features given in Table 6.

(12) a. 

b. 

adam
'man'

c. 

**adam*
'man'

Note, however, that there are also important restrictions on matching and syncretism. Specifically, constituent matching has the effect that the lexical item *adam* can spell out the K1 feature only when the structure contains also the SPEC feature. This is because when K1 is merged directly on top of IDP, as in (12c), such a constituent is not contained in the lexical entry (12a). Therefore, matching fails in (12c), which is indicated there by the asterisk before *adam*; the noun cannot spell out this structure. Thus, when the SPEC feature is absent, NOM feature will need to be spelled out by another lexical item, namely the zero case suffix, as apparent from Table 6.

With the principle for matching in place, let us now turn to the so-called Spellout algorithm, as proposed by Starke (2018). We give it in (13) and (14). Simplifying slightly, we can think of the spellout algorithm as a procedure which specifies how exactly we should proceed when we are trying to match syntactic structure against lexical items stored in the lexicon. We would like to note that movements made in the steps (13b) and (13c) does not leave a trace.

(13) Spellout Algorithm (Starke 2018)
   a. Merge F and spell out.
   b. If (a) fails, try spec-to-spec (roll-up) movement of the node inserted at the previous cycle, and spell out.
   c. If (b) fails, move the complement of F, and spell out.

(14) *Backtracking*
When spellout fails, go back to the previous cycle, and try the next option for that cycle.

Let us now describe in words what kind of procedure the formulation in (13) gives rise to. The first thing to notice is that the formulation 'Merge F and spell out FP' implements the idea of a cyclic spellout, where after every step of external merge, spellout takes place. Spellout

means that a lexical entry matching the newly created FP must be found in the lexicon. Note that matching must succeed, else the derivation crashes at the interface.[7]

The following is important. In a cyclic model, when a lexical item matches a syntactic tree, the tree is lexicalised by some phonology associated to that tree via the lexical entry. However, 'lexicalised' does not mean 'pronounced.' The derivation goes on, and the initial lexicalisation may be replaced by a different one at a later stage (this is called Cyclic Override in Nanosyntax). Only the final lexicalisation is pronounced.

Let us now discuss what happens when matching at FP fails. In such case, the spellout procedure triggers various types of movements in order to avoid crash. First, the movement of the specifier of the complement is tried, see (13b). If that does not help, complement movement takes place, see (13c). We will go through example derivations in the next section.

If all movement operations fail, Backtracking (14) applies. The purpose of backtracking is to go back through the derivation with the possibility to change some of the previous derivational steps (since later on, these steps led to a crash). See Vanden Wyngaerd et al. (2020) for a more thorough exposition of the same procedure.

3.4 DERIVATION OF THE NON-SPECIFIC PARADIGM. Let us now show how the non-specific paradigm is derived step by step. In order to do so, we need to know the specific lexical items. We give them in (15). They are based on Table 6. We have already discussed the lexical entry for the nominal root, recall (12a).

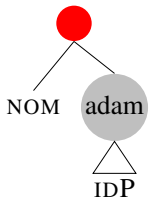
(15) a. adam ⇔ [NOM (K1) [SPEC [IDP]]]
b. Ø ⇔ [GEN (K3) [ACC (K2) [NOM (K1)]]]
c. -a ⇔ [DAT (K4) [GEN (K3) [ACC (K2) [NOM (K1)]]]]

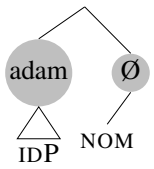
We also know that in the non-specific paradigm, the feature K1 cannot be spelled out by the root, recall (12c). The ending we postulate to spell out K1 is the zero ending in (15b). This ending has K1 as its lowest feature. In addition to the nominative feature K1, the ending can also spell out the features of the accusative (K2) and the genitive case (K3), so all three of these are realised by this zero morpheme, yielding syncretism between the non-specific NOM/ACC/GEN. In the dative, we will need to use the ending *-a*, given in (15c).

Let us now describe how the non-specific paradigm is derived. In the trees below, the grey circle in the non-terminal nodes signals a successful spell-out, and the red circle signals the failure of spellout. We use the noun *adam* as an example for our spell-out algorithm.

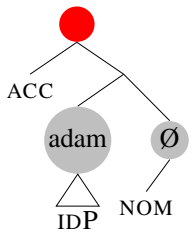
We start by assembling IDP, which is spelled out by *adam* 'man.' When we add the nominative feature K1 on top of IDP, we get the structure in (16a). We know that *adam* does not match (recall (12c)), hence this structure crashes at the interface. The spellout algorithm tells us to try to move the spec of the K1's complement, but the complement has no Spec. Thus, we will have to move the complement of K1, yielding the stucture (16b). Note that we are leaving out traces of moved items in Nanosyntax. Spellout succeeds here, and we insert the zero ending with the lexical entry as in (15b).

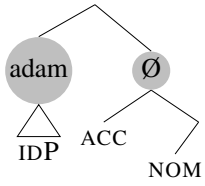
[7]What we refer as 'interface' is the place where syntactic representations which are made of features like SPEC are mapped onto phonological representations. The phonological representations may then be subject to further computation, leading to the ultimate output at PF.

(16) a. 

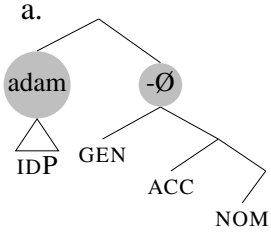
b. 

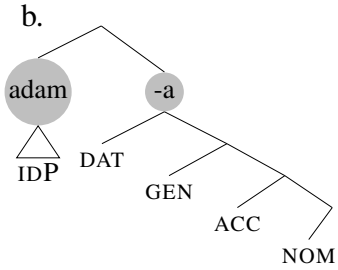
When ACC is merged on top of (16b), we get the structure (17a). Again, we are not able to spell it out. We try to move the Spec of the complement, which gives the structure (17b). Spellout at the phrasal level succeeds.

(17) a. 

b. 

When we add more case features, we keep doing Spec movements, ultimately producing the GEN as in (18a) and DAT as in (18b).

(18) a. 

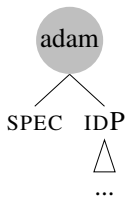
b. 

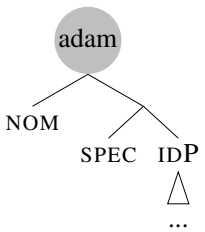
These derivations show that in the non-specific paradigm, the endings simply spell out all the case features. The situation is slightly different in the specific paradigm to which we turn now.

3.5 DERIVATION OF THE OVERT PARADIGM. Let us now turn to the derivation of the specific paradigm. For the specific paradigm, we shall need additional endings, namely the accusative *-i* and the genitive *-n*. We give them in (19).

(19) a. -ı ⇔ [ACC (K2)] b. -n ⇔ [GEN (K3)]

Let us now go through the derivation. We start out by simply merging IDP and SPEC and try to spell it out. Our lexical item in (15a) is able to spell out the tree in (20a). The same lexical item is also able to spell out the K1 feature for NOM case as in (20b).

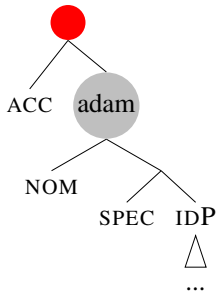
(20) a. 

b. 

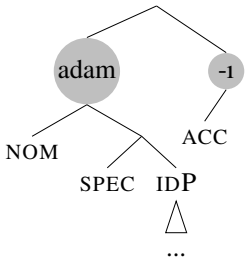
Since bare nouns cannot be used as specific objects, we say that the root cannot spell out ACC. Our lexical items reflect this limitation. In (15a), we do not have the K2 feature within the lex-

ical item. Thus, when we merge ACC (K2), we fail to have a lexical match as in (21a). Since there is no Spec to move, a roll-up movement takes place (recall 13), yielding (21b). Here, the whole subtree containing NOM, SPEC, and IDP moved out. As a result, ACC is spelled out in its own phrase, see (21b).

(21) a.

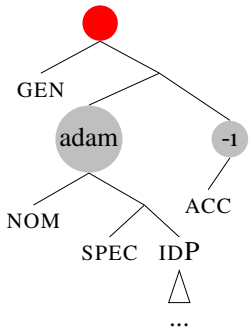


b.

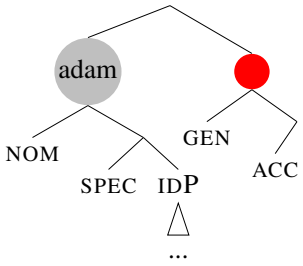


When GEN is merged, direct spellout fails (22a). Following the spellout algorithm (13), we first try to do cyclic movement and try to spell it out, but it fails again (22b). (This is so under the version of the account where *-i* and *-n* are separate morphemes. If *-(n)in* was analysed as a non-decomposable portmanteau, spellout would succeed.)

(22) a.

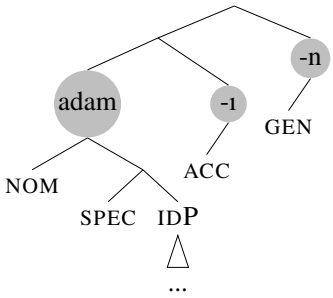


b.



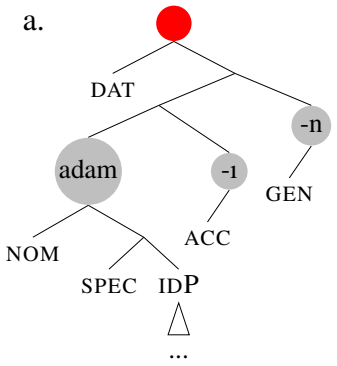
Since Spec movement fails, it is undone and we try complement movement instead, yielding (23). The tree is constructed by moving the complement of GEN in (22a).

(23)

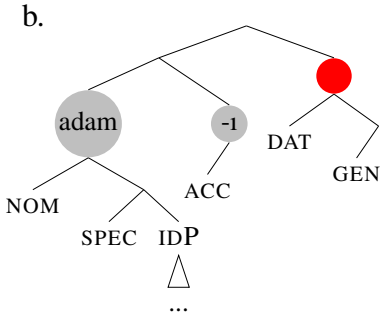


When DAT is merged on top of (23) as in (24a), direct spellout fails. We try cyclic movement according to our spellout algorith, but it also fails as in (24b).

(24) a.

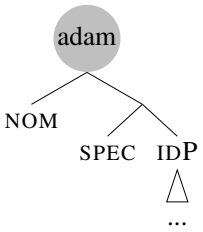


b.

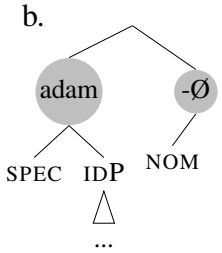


When we try roll-up movement it will fail too since we do not have any lexical item that has DAT as a most embedded feature. This activates backtracking procedure specified in (14). We try out every possible side route that could have been taken and try to spell out DAT in these scenarios. Every spell out fails until we go back to the spell-out of NOM and instead of spelling it out within the root as in (25a), we spell it out using roll-up movement and the "non-specific" ending surfaces as in (25b).

(25) a.

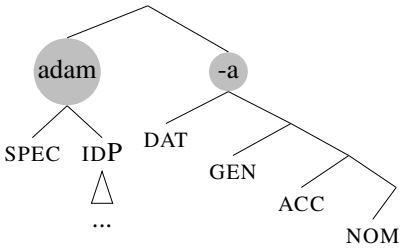


b.



Through several merges and cyclic movements, we will be able to spell the DAT structure out as follows.

(26)



4. Conclusion. In this paper, we have proposed a nanosyntactic analysis for the Turkish case system to model two challenging characteristics: (i) only a subset of Turkish cases show alternation between specific and non-specific forms and (ii) the containment relation propose by Caha (2009) does not hold for the non-alternating cases. We argued that this alignment of characteristics is non-arbitrary. Encapsulating the specificity information within the noun itself and proposing two sub-paradigms enabled us to solve both of these puzzles. After laying out the observations of the Turkish case system and the specific machinery we are using, we presented our step by step analysis for both the specific and non-specific sub-paradigms.

References

Baker, Mark. 2003. *Lexical Categories*. Cambridge: Cambridge University Press. http://dx.doi.org/10.1017/CBO9780511615047.

Caha, Pavel. 2009. *The nanosyntax of case*: Universitetet i Tromsø dissertation. https://hdl.handle.net/10037/2203.

Caha, Pavel. 2010. The parameters of case marking and spell-out driven movement. In Jeroen Van Craenenbroeck (ed.), *Linguistic variation yearbook* 10. John Benjamins. http://dx.doi.org/10.1075/livy.10.02cah.

Caha, Pavel. 2013. Explaining the structure of case paradigms by the mechanisms of Nanosyntax. *Natural Language & Linguistic Theory* 31. 1015–1066. http://dx.doi.org/10.1007/s11049-013-9206-8.

Caha, Pavel. 2021. Modeling declensions without declension features. The case of Russian. *Acta Linguistica Academica* 68. 45–95.

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.

Chelliah, Shobhana Lakshmi. 1997. *A Grammar of Meithei*. Berlin, New York: Mouton de Gruyter. http://dx.doi.org/10.1515/9783110801118.

Cinque, Guglielmo & Luigi Rizzi. 2010. The Cartography of Syntactic Structures. In Bernd Heine & Heiko Narrog (eds.), *The Oxford Handbook of Linguistic Analysis*. 51–65. Oxford: Oxford University Press. http://dx.doi.org/10.1093/oxfordhb/9780199544004.013.0003.

Enç, Mürvet. 1991. The Semantics of Specificity. *Linguistic Inquiry* 22(1). 1–27. https://www.jstor.org/stable/4178706.

Harley, Heidi & Elizabeth Ritter. 2002. Person and Number in Pronouns: A Feature-Geometric Analysis. *Language* 78(3). 482–526. http://dx.doi.org/10.1353/lan.2002.0158.

Noonan, Michael & Elena Mihas. 2007. Areal Dimensions in Case Syncretism: Ablatives and Genitives. http://dx.doi.org/10.11588/xarep.00000209. Ms. University of Wisconsin-Milwaukee.

Öztürk, Balkız. 2005. *Case, Referentiality, and Phrase Structure*. J. Benjamins Publishing Company. http://dx.doi.org/10.1075/la.77.

Starke, Michal. 2010. Nanosyntax: A short primer to a new approach to language. *Nordlyd* 36(1). 1–6.

Starke, Michal. 2018. Complex Left Branches, Spellout, and Prefixes. *Exploring Nanosyntax* 239–249. http://dx.doi.org/10.1093/oso/9780190876746.001.0001.

Vanden Wyngaerd, Guido, Michal Starke, Karen De Clercq & Pavel Caha. 2020. How to be positive. *Glossa* 5(1). 23. 1–34. http://dx.doi.org/10.5334/gjgl.1114.