A local analysis of an apparent non-local allomorphy in Tamil: a perspective from Rutul

Pavel Caha (Masaryk University)

1. Introduction

One of the goals of a linguistic theory is to restrict the set of possible hypotheses that a child may entertain when acquiring a language.[1] In morphology, one of the most prominent restrictions of this kind is the Adjacency Condition on allomorphy (Siegel 1978, Embick 2010, Bobaljik 2012). The adjacency condition says that in a string of morphemes such as (1), morpheme α cannot influence γ across β (and vice versa).

(1) α ... β ... γ

A particularly challenging example for this condition has been brought up in Moskal and Smith (2016). The example comes from Tamil pronominal paradigm, where case morphemes seem to influence the phonological realisation of a pronominal root across an intervening plural marker. The facts are schematically depicted in (2).

(2) a. root_1 ... plural ... case_1

b. root_2 ... plural ... case_2

The subscripts indicate that specific case suffixes (case_1 and case_2) have the power to influence the shape of the pronominal root across an intervening plural, giving rise to a different form of the root (root_1 and root_2 respectively).

[1] The work on this chapter has been supported by the grant 25-15601S from the Czech Science Foundation (GA ČR).

This chapter argues that this long-distance interaction is only apparent, and it can be understood as the result of an interaction between two processes, namely strictly local allomorphy and syncretism. The main idea of the analysis is inspired by patterns of stem allomorphy in Rutul (Kibrik 2003), where we find a pattern of allomorphy that can be schematically depicted in (3).

(3) a. $root_1$ ... $plural_1$ ... $case_1$

b. $root_2$ ... $plural_2$ ... $case_2$

In (3), there are not only two different shapes of the root, but covarying with the shape of the root, also the allomorph of the plural changes. This is conveyed by the addition of the subscript on the plural marker. The interest of looking closer at this pattern is that here -- just like in (2) -- a particular case suffix obligatorily combines with a particular form of the root, yet the selection appears to take place under adjacency, mediated by the plural marker in the middle.

This chapter provides a particular account of the pattern in (3), building on the Nanosyntax lexicalisation algorithm proposed in Starke (2018, 2022). The main idea of the analysis is that the shape of the root in (3) is not influenced directly by case, but by the adjacent plural marker, which is in turn conditioned by case. More specifically, the analysis says that due to the presence of an oblique case, the plural marker in (3b) lexicalises different features than the one in (3a), and this in turn influences the lexicalisation of the root.

With an account of the pattern (3) in place, I turn back to Tamil and point out that it is possible to assume exactly the same type of interactions in Tamil as in Rutul, with the consequence that the plural markers in (2a) and (2b) are only superficially identical, but in reality they each realise a slightly different set of features, and, crucially, interact with the base in the same way as the different plural suffixes in Rutul.

Adopting the existence of two underlyingly distinct plural markers in Tamil, I argue that the only difference between the "local" Rutul system and the "non-local" Tamil system rests in the morphological realisation of the plural markers. In Rutul, their underlying difference is faithfully reflected by a morphological distinction. In Tamil, they are realised the same, i.e., they are syncretic. I depict this idea in (4), where the rectangle around the two different plural markers indicates syncretism.

(4) a. root_1 ... plural_1 ... case_1

b. root_2 ... plural_2 ... case_2

Summarising, the main idea is that it is possible to break down an apparent long-distance interaction in Tamil into a series of strictly local interactions (using Rutul as a proof of existence of such interactions). What obscures the local nature of the process in Tamil is syncretism, which fails to make a morphological distinction between the two distinct plural markers. The analysis allows us to maintain a restrictive theory of grammar, where the set of hypotheses available to the learner rules out non-local morpheme interactions.

2. Tamil

This section introduces the Tamil facts to be investigated. The table in (5) presents a paradigm fragment of the 1st person singular pronoun 'I'.[2]

[2] The forms are as given in Moskal and Smith (2016), including the segmentation. There is one exception, namely, an *l* is added to the plural markers in the nominative and in the oblique.

(5) Tamil 1st person pronoun (Moskal and Smith 2016, 306)

	I, SG	We, PL
NOM	naan	naan-gal
OBL	en-Ø	en-gal-Ø
DAT	en-akku	en-gal-ukku

We can see that in the nominative singular and plural, the root has the form *naan* (light shading). In all other cases, the root is *en* (dark shading). Clearly, the distribution of the root is governed by case, not by number. This is so even when case is separated from the root by the plural *gal*. To encode the allomorphy, Moskal and Smith (2016) posit the rules in (6).

(6) a. [1.sg] ↔ *naan*

b. [1.sg] / _] K ↔ *en*

The rules say that the 1st singular pronoun is generally realised as *naan*, see (6a), but when it is found in the context of a following case marker, the form is *en*, see (6b). This condition is satisfied in both the dative and in the so-called oblique case (roughly genitive), which Moskal and Smith analyse as featuring a zero affix. Crucially, the rule (6b) is not restricted by adjacency,

I did this to achieve consistent rendering across different works on Tamil, such as Lehmann (1989) or McFadden (2018), who include the final *l* in the plural marker. See also Newell (2023) for a more fine-grained decomposition of the forms, which, however, does not change the non-local nature of the allomorphy.

so in the plural, the root of the pronoun is allowed to inspect a larger context and see if there are any case suffixes even after the immediately adjacent plural morpheme.

By abandoning adjacency as a relevant condition, Smith and Moskal (2016) are clearly able to capture the facts. However, by doing so, they increase the range of possible hypotheses that the child has to consider when acquiring the morphemes of a particular language. This chapter argues that despite appearances, a strictly local account of the Tamil facts is possible using the Nanosyntax framework, relying on the new lexicalisation algorithm discussed in Section 6 of Chapter 1.

3. Rutul

The goal of this paper is to show that it is possible to analyse the apparent long-distance sensitivity of the root to the case marker (across plural) as a series of local relations that crucially involve the plural as the middleman. This is depicted in (7), repeated from (3).

(7) a. root_1 ... plural_1 ... case_1

b. root_2 ... plural_2 ... case_2

The goal of this section is to introduce the relevant facts that provide evidence for the existence of the pattern in (7). The table in (8) shows two paradigm fragments from Kina Rutul, a North-East Caucassian language (Kibrik 2003, 63, Filatov 2020).[3]

[3] The oblique case in Kina Rutul has a function similar to the oblique in Tamil, i.e., they are both used as attributes of the noun (standardly called the genitive). When the Rutul oblique modifies a noun, it is obligatorily followed by an additional suffix, -*d(ɨ)*. However, -*d(ɨ)* is not a genitive, but an all-purpose linker: any nominal modifier must be suffixed with it, including

(8) Kina Rutul (Filatov 2020:18-9)

	nose, SG	nose, PL	road, SG	road, PL
NOM	xex	xex-bɨr	raˤq	raˤq-bɨr
OBL	xexi-Ø	xexi-mɨ-Ø	rɨˤɢɨˤ-Ø	rɨˤɢɨˤ-mɨ-Ø
DAT	xexi-s	xexi-mɨ-s	rɨˤɢɨˤ-s	rɨˤɢɨˤ- mɨ-s

The table shows that the plural has two allomorphs, namely the nominative *-bɨr*, and the oblique *-mɨ*. We can further see that the nominative allomorph *-bɨr* attaches to the nominative singular stem (*xex, raˤq*), while the oblique plural *-mɨ* attaches to the oblique stem (*xexi, rɨˤɢɨˤ*). The identity of the stems between the singular and the plural form is highlighted by shading. This paradigm thus represents an instance of the abstract pattern in (7).

This inflectional pattern has been explicitly noted in Kibrik (2003:63), who depicts the system in the scheme (9).

(9)

$$\begin{tikzcd} \text{NOM.SG} \arrow[r, phantom, "="] & \text{ROOT} \arrow[r] \arrow[d] & \text{NOM.PL} \\ \text{OBL SG.} & \text{OBL} \arrow[l] \arrow[r] & \text{OBL.PL} \end{tikzcd}$$

adjectives, adverbs, and case-marked nouns in other cases, e.g., in the locative. Thus, I interpret the oblique form as a genitive, which, when it modifies a noun, must be suffixed by -*d(i)* like other modifiers (see Filatov 2020, 4-5). Similarly to Rutul, the Tamil oblique may also be followed by a variety of "euphonic" markers when it modifies a noun, sometimes obligatorily so (Lehmann 1989, 22).

In this scheme, the equal sign between the root and NOM.SG tells us that the root of the noun is used as the NOM.SG form. The arrow going to the right tells us that the very same root also functions as the base for NOM.PL. The arrow going down shows that there is an oblique base, derived by the addition of *-i*, which then functions as the base for the singular and plural oblique cases.[4]

The main point is that the Rutul scheme as described by Kibrik can also be used to describe Tamil, where the ROOT in the scheme (9) would be *naan* 'I', which is a form equal to NOM.SG, and also serving as the base for NOM.PL. (*naan-gal* 'we'). Then there is the oblique form, *en*, used as the base for oblique singular (*en-akku* 'to me') and oblique plural cases (*en-gal-ukku* 'to us'). Therefore, it appears that the two patterns are related, and I will be seeking a unified explanation for them. What kind of explanations are available?

When we consider the Rutul pattern in (8), it can be explained in two different ways. The first possible analysis is a trivial extension of Moskal and Smith's (2016) analysis of Tamil. This extension would say that the base in Rutul is sensitive to the case markers via a long-distance allomorphic relation (this is just like Tamil). In addition, the plural marker too shows allomorphy conditioned by case. This is depicted in (10), where the arrow indicates the "is conditioned by" relation: both the base and the plural are conditioned by case.

(10) a. base$_1$ ... plural$_1$ ... case$_1$

[4] Once we decide to segment the oblique base *xexi* into two morphemes (i.e., *xex-i*), we can no longer say that the oblique case suffixes in Rutul require a special form of the root, as stated in (8), but a special form of the stem. However, this does not change the main point, which is about the interaction between the case marker and the base (whether this is a root or a stem).

b. base_2 ... plural_2 ... case_2

In this chapter, I opt for a different interpretation, where the conditioning between case and the base can be broken down into a chain of two successive local-allomorphy relations, see (11).

(11) a. base_1 ... plural_1 ... case_1

b. base_2 ... plural_2 ... case_2

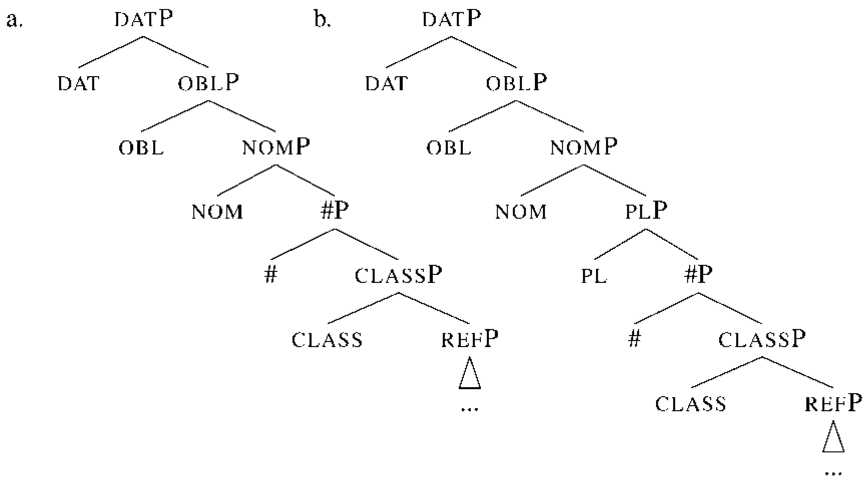
On both accounts, the plural is conditioned by case. The difference is that on the account in (11), the form of the base is not conditioned directly by case, but by the intermediate plural marker, yielding an "allomorphic chain."

At first, the analysis in (10) seems easier to carry over to the singular: since base_2 is directly conditioned by case_2, the presence or absence of plural does not matter. However, the price to pay is non-local allomorphy and the increased hypothesis space for the learner. On the analysis in (11), the learner only has one choice, but the challenge is to fine-tune the analysis in a way that we get base_2 even in the absence of plural_2 (i.e., when the base_2 directly precedes case_2). The next section presents such an analysis within the Nanosyntax framework.

4. Modelling the allomorphic chain in Rutul: an informal account

This section presents an informal account of the allomorphy chain in Rutul. The first ingredient of the analysis are the morphosyntactic structures underlying the relevant forms. I show the structures I assume for the singular and plural in (12a,b) respectively.

(12)



At the bottom of each projection, there is a REFP, which introduces a referential index (Harley and Ritter 2002). I assume that REFP is internally complex. For instance, in pronouns, this part of the structure contains person features, see Vanden Wyngaerd (2018). However, for the purpose if this article, its internal structure is irrelevant, and it is therefore simplified.

Above REFP, there are features corresponding to gender, number, and case, in that order going bottom up; see Lamontagne and Travis (1987), Picallo (1991), for relevant discussion.

More specifically, the feature CLASS, located above REFP, codes noun class, also called gender (when it is reflected by agreement). In this article, I abstract away from class distinctions and use just a single CLASS node.

The feature # codes the singular number, the plural has the PL head in addition. In the singular, the PL feature is missing, see (12a). When the PL feature is present, as in (12b), we are looking at the representation of the plural.

Above number we find case. Case is decomposed into three layers. The nominative has just a single feature, NOM. The oblique case adds the feature OBL on top of the nominative, the dative has one extra feature, namely DAT. See Caha (2009, 2013), McFadden (2018), Zompì (2019), Smith et al. (2019) for the reasons for this decomposition.

Structures like the ones in (12) are mapped onto pronunciation relying on two components: (A) the language-specific postsyntactic lexicon, containing stored links between well-formed syntactic representations and phonology, and (B) the language-invariant lexicalisation procedure, which specifies how exactly lexicalisation proceeds.

This chapter relies on a specific lexicalisation algorithm proposed in Starke (2022), which contains so-called subextractions (see Section 6 of Chapter 1, and also Wiland 2019). A detailed walk-through to some of the derivations is provided in Section 6 of the current chapter. The present section introduces the analysis informally, using so-called "lexicalisation tables." As explained in Chapter 1, lexicalisation tables depict the result of the derivation, showing which morpheme lexicalises which feature of the structures in (12). They will allow us to get an informal understanding of the analysis, without going into much of the technical detail. However, lexicalisation tables do not specify exactly how this lexicalisation was reached; this is something that I am going to discuss in Section 6.

To be able to read the tables, recall from Chapter 1 that lexical items in Nanosyntax generally lexicalise phrasal constituents containing several features. Therefore, in the lexicalisation table, each exponent may lexicalise multiple features. The amount of features lexicalised by a morpheme is dependent on its entry: specifically, morphemes may only lexicalise features for which they are specified. For instance, if the root is lexically specified for REF and CLASS, it cannot lexicalise # because # is not specified in the root.

Matching between roots and structures is governed by the Superset Principle, see (13).[5]

(13) Superset Principle (Starke 2009): A lexically stored tree matches a syntactic node iff the lexically stored tree contains the syntactic node.

[5] In Chapter 1, the Superset Principle is called the Matching Condition, and its wording is based on identity. I use the traditional formulation for clarity, though the content is the same.

Recall further from Chapter 1 that the lexicalisation algorithm is setup in such a way that all features must be lexicalised. When the root is not specified for a particular feature, e.g., #, affixes are introduced to lexicalise these features.

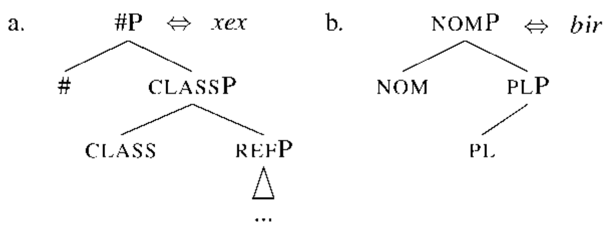
With the basics clarified, let me now turn to the Rutul plural paradigm of *xex* 'nose.' The analysis of the plural is depicted in the lexicalisation table (14). The top row gives the features, the rows below show which exponents lexicalise which features.

(14)

		REFP	CLASS	#	PL	NOM	OBL	DAT
NOM	*xex-bɨr*	*xex*			*bɨr*			
ACC	*xex-i-mɨ*	*xex*	*i*	*mɨ*				
OBL	*xex-i-mɨ-l*	*xex*	*i*	*mɨ*				*l*

The table shows that in the nominative, the root *xex* lexicalises all the features up to #. Therefore, the simplest lexical entry we could assume would be as in (15a). Here we have the #P constituent that contains all the three features that the root lexicalises in (14). It turns out that we will need to update this entry, but it is going to be instructive to see the reasons for it, so at this point, the root is listed as (15a).

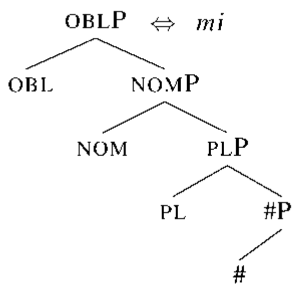
(15)



In (15b), the ending *-bɨr* is given. It is associated with a constituent (NOMP) that contains the two features that *-bɨr* lexicalises in (14). Readers familiar with Nanosyntax will have no trouble seeing how the two entries in (15) lead to the lexicalisation in (14): #P moves out of NOMP, creating the remnant constituent in (15b), which is lexicalised as *-bɨr.* I shall introduce these details in Section 6. The final remark concerning the nominative is that the dark cells at the end of the nominative row in (14) represent the information that the features OBL and DAT are missing in the nominative.

Let me now turn to the oblique case, containing the additional feature OBL. None of the items in (15) is capable of lexicalising OBL, and we therefore need a new item. The hypothesis depicted on the oblique row in (15) is that the only way to lexicalise OBL is to use the morpheme -*mɨ*, specified as in (16).

(16)



At this point, it is important to know that lexicalisation in Nanosyntax only targets phrasal nodes. Therefore, to lexicalise the OBL feature, lexicalisation must target a phrasal constituent containing OBL, rather than the feature itself. However, note that the OBLP in the entry contains not only the OBL feature itself, but also the features NOM, PL and #. Such an ending is thus incapable of lexicalising OBL alone: to lexicalise this feature, it must lexicalise the whole constituent, thereby lexicalising all the features. This is as depicted in Table (14), repeated below in (17) for convenience.

(17)

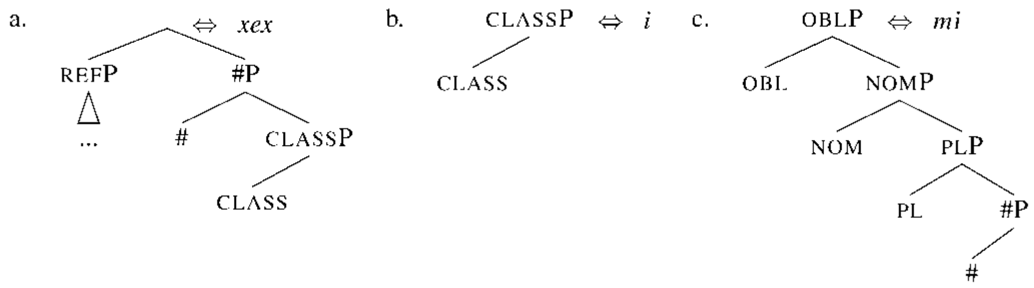
		REFP	CLASS	#	PL	NOM	OBL	DAT
NOM	*xex-bɨr*	*xex*			*bɨr*			
ACC	*xex-i-mɨ*	*xex*	*i*	*mɨ*				
OBL	*xex-i-mɨ-l*	*xex*	*i*	*mɨ*				*l*

Notice that by virtue of lexicalising #, the ending *-mɨ* makes it impossible for the root to lexicalise this feature, which is unlike what happens in the nominative. So, crucially, the ending *-mɨ* affects the number of features lexicalised by the base: while the base corresponds to #P in the singular, it corresponds to the CLASSP in the plural. This is how the plural marker controls the allomorphy of the root -- under adjacency.

However, nothing else said, we currently expect that the root *xex* will lexicalise the CLASSP in the oblique case, since CLASSP is contained in the entry of *xex* that we started with, recall (15a). But this is the wrong result: the oblique form is not **xex-mɨ*, but *xex-**ɨ**-mɨ*.

The correct result emerges once we update the entry of the root as in (18a), and add the additional marker *-ɨ*, see (18b). The entry for *-mɨ* is repeated for convenience in (18c).

(18)



Let us first focus on the entry of the root, given in (18a). The entry still contains REFP, CLASS and #. However, the entry has been changed by moving the REFP from the complement position of CLASS to the specifier of #. The idea behind this update is that the REFP moves to the specifier of #P in syntax, and it is only in this dislocated position where it can be lexicalised along with # and CLASS. Entries like this are possible in Nanosyntax, since lexical entries in general link any well-formed syntactic representation to its phonological realisation; and since well-formed syntactic trees may contain movement, entries like (18a) are a possibility offered by the system (see Blix 2022).

The updated entry still contains a phrasal constituent that contains all of REFP, CLASS and #: this is the top node. So, the entry can still lexicalise all these features in the nominative.

However, the lexical entry is incapable of realising only REFP and CLASS, since REFP and CLASS do not form a constituent to the exclusion of #. This leads to the result that the base of the oblique plural -- which corresponds precisely to CLASSP -- cannot be realised by the bare root.

Since there is no single entry that contains REFP + CLASS, we get two markers. The root lexicalises REFP, since it contains REFP. In addition, we need *-i* to lexicalise CLASS, see (18b). (The entry will be updated.)

Summarising, using the entries in (18), and assuming also the existence of the nominative plural marker *-bir* in (15b), we correctly derive the stem alternation between the nominative base *xex* and the oblique base *xex-i*. The last thing we need is the assumption that the dative *-l* realises the dative feature DAT, as shown in the dative row of the lexicalisation table (17). I am not showing the lexical entry for reasons of space.

The important thing in this explanation is the causal chain that leads from the case feature OBL via the plural marker *-mi* to the specific form of the base. The relevant steps in the reasoning are the following: (i) The need to lexicalise OBL leads to the emergence of *-mi*, because this is the only lexical item that can spell out OBL. (ii) The morpheme *-mi* forces the base to lexicalise only CLASSP, which is different from the nominative, where the base lexicalises #P. The difference is then reflected by a different form of the base, *xex* vs. *xex-i*.

What remains to be shown is how to capture the distribution of the bases *xex* and *xex-i* in the singular. A somewhat mechanical way to do this would be to assume that on analogy with the plural, there are silent singular markers, one for the nominative singular (see SG1 in (19)), and another one for the oblique singular cases (see SG2 in (19)). The two singular markers restrict the base to the same cells of the lexicalisation table as the plural markers, hence the same distribution arises.

(19)

		REFP	CLASS	#	NOM	OBL	DAT
NOM	*xex-bɨr*	*xex*			SG1		
ACC	*xex-i-mɨ*	*xex*	*i*	SG2			
OBL	*xex-i-mɨ-l*	*xex*	*i*	SG2			*l*

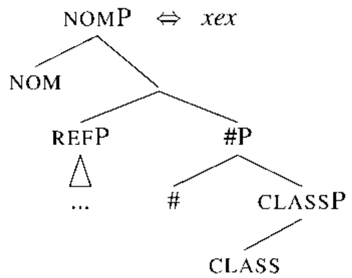
While this solution works, the need to postulate two different silent singular markers is suboptimal. Moreover, there is a way to analyse this paradigm without such markers, see (20).

(20)

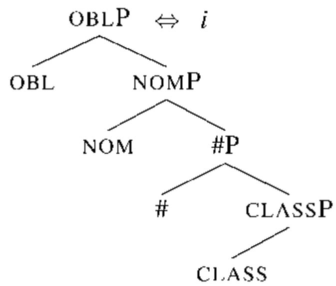
		REFP	CLASS	#	NOM	OBL	DAT
NOM	*xex-bɨr*	*xex*					
ACC	*xex-i-mɨ*	*xex*	*i*				
OBL	*xex-i-mɨ-l*	*xex*	*i*				*l*

In Table (20), *xex* actually lexicalises all the features of the NOM.SG, see the first row in (20). Similarly, the marker -*i* lexicalises the oblique feature in the singular, see the remaining rows. In this account, the appearance of the oblique stem -*i* is thus directly conditioned by the oblique feature, since -*i*, according to (20), lexicalises it directly. The lexical entries that generate this table are provided in (21), updating the entries (18a) and (18b). (This is the final analysis.)

(21) a.



b.



In both cases, additional features were added on top of the original specification assumed in (18), namely the feature NOM to the original entry of *xex*, and three additional features on top of the CLASS feature originally assumed for -*i*.

The analysis of the plural is unaffected by this, since *xex* cannot realise the nominative feature in the plural (it is not specified for PL). The same holds for -*i*, which cannot be used to lexicalise plural either. Therefore, we still need the plural markers *-bɨr* and *-mɨ*, which require a base of a certain size (#P and CLASSP respectively), and the lexicalisations remain the same.

Summarising, this section provided the gist of the analysis of the Rutul "allomorphy chain" in the plural, where the OBL feature conditions the appearance of a particular plural marker, and this plural marker in turn forces a particular base. In the singular, the oblique feature once again forces the appearance of the oblique stem marker *-i*.

5. Adapting the allomorphic chain to Tamil

This section turns to Tamil. Taking inspiration from the analysis of Rutul, the section suggests a solution to the conundrum identified in Moskal and Smith (2016).

Let us first apply to the Tamil data the exact same analysis as just proposed for Rutul. The lexicalisations proposed for Rutul are summarised in the top part of Table (22). To repeat, these lexicalisation tables are not the primitive of the theory, but a convenient device to depict the lexicalisation derived from the underlying structure, using the proposed lexical entries and the lexicalisation algorithm of Starke (2022). Notice that the PL feature is always missing in the singular.

(22)

<table>
<tr><td></td><td></td><td>REFP</td><td>CLASS</td><td>#</td><td></td><td>NOM</td><td>OBL</td><td>DAT</td></tr>
<tr><td>NOM</td><td>xex-bɨr</td><td colspan="5">xex</td><td></td><td></td></tr>
<tr><td>ACC</td><td>xex-i-mɨ</td><td>xex</td><td colspan="5">i</td><td></td></tr>
<tr><td>OBL</td><td>xex-i-mɨ-l</td><td>xex</td><td colspan="5">i</td><td>l</td></tr>
<tr><td></td><td></td><td>REFP</td><td>CLASS</td><td>#</td><td>PL</td><td>NOM</td><td>OBL</td><td>DAT</td></tr>
<tr><td>NOM</td><td>xex-bɨr</td><td colspan="3">xex</td><td colspan="2">bɨr</td><td></td><td></td></tr>
<tr><td>ACC</td><td>xex-i-mɨ</td><td>xex</td><td>i</td><td colspan="4">mɨ</td><td></td></tr>
<tr><td>OBL</td><td>xex-i-mɨ-l</td><td>xex</td><td>i</td><td colspan="4">mɨ</td><td>l</td></tr>
<tr><td></td><td></td><td>REFP</td><td>CLASS</td><td>#</td><td></td><td>NOM</td><td>OBL</td><td>DAT</td></tr>
<tr><td>NOM</td><td>naan</td><td colspan="5">naan</td><td></td><td></td></tr>
<tr><td>ACC</td><td>en</td><td>en</td><td colspan="5">Ø</td><td></td></tr>
<tr><td>OBL</td><td>en-akku</td><td>en</td><td colspan="5">Ø</td><td>akku</td></tr>
<tr><td></td><td></td><td>REFP</td><td>CLASS</td><td>#</td><td>PL</td><td>NOM</td><td>OBL</td><td>DAT</td></tr>
<tr><td>NOM</td><td>naan-gal</td><td colspan="3">naan</td><td colspan="2">gal$_1$</td><td></td><td></td></tr>
<tr><td>ACC</td><td>en-gal</td><td>en</td><td>Ø</td><td colspan="4">gal$_2$</td><td></td></tr>
<tr><td>OBL</td><td>en-gal-ukku</td><td>en</td><td>Ø</td><td colspan="4">gal$_2$</td><td>ukku</td></tr>
</table>

The lower part of the table (below the double line) then shows how this solution can be mechanically adopted for Tamil. The identity of the pattern is revealed by the identical colouring scheme, which is the same across the two languages. On this analysis, the only difference between Tamil and Rutul is purely in the phonological representations associated to an identical set of lexical entries.

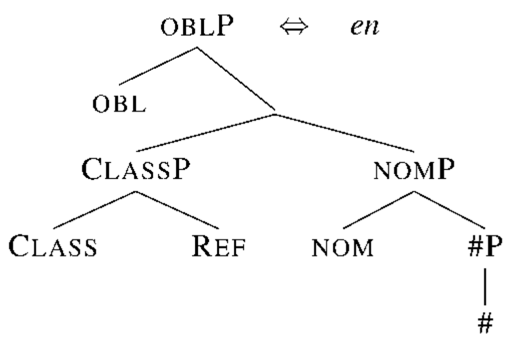
At the same time, there are two major differences between Rutul and Tamil. These suggest that the analysis of Tamil should be slightly adjusted. The first noteworthy difference is that the root *xex* in Rutul is invariant, while the Tamil 1.SG pronoun has two suppletive roots, *naan* and *en*. Interestingly, the distribution of the two allomorphs perfectly mirrors the presence/absence of the Rutul oblique marker *-i.* Wherever Rutul has just the bare *xex* 'nose' (without the oblique *-i*), Tamil has *naan* 'I.' Wherever we find *xex-i* 'nose-OBL,' Tamil shows *en* 'I, OBL.' I shall thus follow the idea that *en* lexicalises single handedly the exact same set of projections that Rutul lexicalises by the two pieces (i.e., *xex-i*). The final proposal for Tamil is thus as depicted in (23), which is an update on (22) and changed only in that *en*, a single marker, stretches across the cells where Rutul has *xex-i*. This analysis further eliminates the inelegant zero morpheme of (22).

(23)

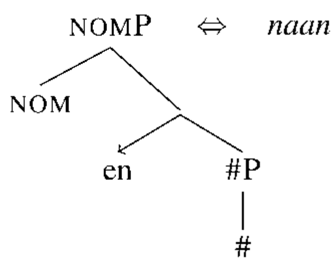
		REFP	CLASS	#		NOM	OBL	DAT
NOM	*naan*	*naan*						
ACC	*en*	*en*						
OBL	*en-akku*	*en*						*akku*
		REFP	CLASS	#	PL	NOM	OBL	DAT
NOM	*naan-gal*	*naan*			*gal₁*			
ACC	*en-gal*	*en*		*gal₂*				
OBL	*en-gal-ukku*	*en*		*gal₂*				*ukku*

The lexical items for the roots *naan* and *en* (both 'I') are given in (24).

(24) a.



b.



The entry for *en* 'I' in (24a) contains a complex left branch comprising the features CLASS and REF. These are relevant in the oblique singular, where *en* is found as the lexicalisation of these features.

Recall that since lexical entries may link any well-formed structure to a phonological representation, it is possible to also consider entries which lexicalise structures that contain movement (Blix, 2022). Specifically, the idea is that in (24a), CLASSP reaches the position in between NOM and OBL by movement. The details of this derivation are explored in Section 6.

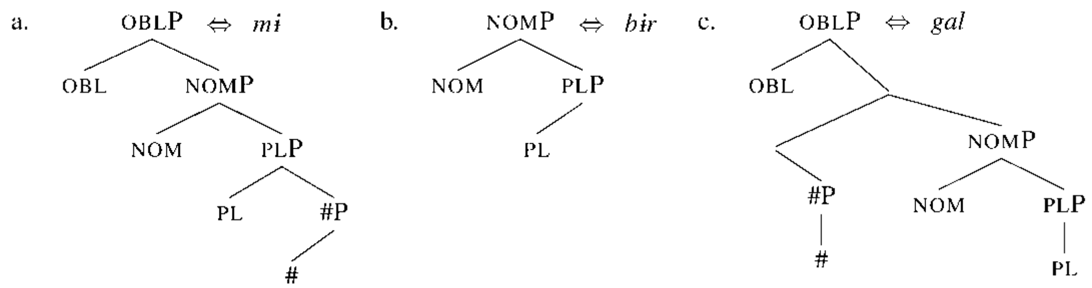
Notice that as an effect of the constituent structure of the entry, the form *en* cannot lexicalise #P, since the features CLASS, REF and # do not form a constituent. #P in the nominative plural is thus lexicalised as *naan*, with the entry in (24b). Note that the entry contains the so-called pointer (see, e.g., Vanden Wyngaerd et al. 2021). Entries with pointer are inserted when the branch pointed to is lexicalised by a specific entry, here *en*, and the rest of the entry is identical to the relevant structure that undergoes lexicalisation.

As said, this entry lexicalises #P in the plural, and it also lexicalises the whole nominative structure in the singular. The oblique singular is again lexicalised by *en*. The derivations will be discussed in greater detail in Section 6.

It is worth noting that the distribution of *en* and *naan* represents an ABA pattern, where a small structure (CLASSP) and a large structure (OBLP in the singular) are lexicalised by *en*, while the intermediate structures (#P and NOMP) are lexicalised by *naan*. This type of derivation is inspired by Kasenov's contribution to this volume.

With the lexical entries for the pronominal roots in place, let me turn to the second noteworthy fact, which is that the two different plural markers of Rutul (*-bɨr* and *-mɨ*) are associated to the same phonology in Tamil, namely *gal*. In the analysis of Rutul, each of these markers is associated to a different constituent. Therefore, adopting that analysis for Tamil would lead to an unwanted accidental homophony. The obvious question to ask is if it is possible to unify these entries. In the following paragraphs, I suggest that the unification is possible.

To begin with, note that in (25), the two *gal*'s lexicalise overlapping feature sets. This raises the hopes that an attempt at their unification may succeed. Using the Superset Principle (18), we could specify *gal* for the features #, PL, NOM and OBL, which, in the ideal case, would lead to the fact that *gal* would also be able to lexicalise just PL and NOM. However, there is a catch. To see it, consider again the lexical entries for the Rutul -*mɨ* and -*bɨr*, see (25a,b).

(25)


The issue is that even though the features PL and NOM are contained in the entry for the oblique plural -*mɨ* in (25a), the features do not form a subconstituent in it. Therefore, we cannot adopt the entry (25a) as the entry of the unified *gal* because it cannot be used to lexicalise the nominative plural constituent associated to -*bɨr* in (25b).

To resolve this issue, we need to adopt a different entry for the unified *gal*, given in (25c). In this entry, the features of *gal*$_1$ (PL and NOM) do form a constituent. This is the case also for the features of *gal*$_2$ (these are all the features contained in the top node). The property of the entry that gives us this result is that instead of placing the # feature in the base position below PL (as in (26a)), # is in a displaced position in between OBL and NOM. The idea underlying this entry is similar to the one in (24a), where the REFP was displaced by movement. On analogy, the tree in (25c) is a lexicalisation of a structure that arises as the product of a series of movements, where # is first pied-piped by the CLASSP across NOM, and then stranded there when CLASSP moves higher up. Given that in Nanosyntax, any well-formed syntactic structure may be stored in the lexicon and linked to phonological information, this is a perfectly legitimate entry within this approach.

I will discuss these derivations in detail in the next section; what is important for now is that the entry (25c) provides a unified entry for the two different *gal*'s. Note that at the morphosyntactic level (i.e., in terms of the features they lexicalise), *gal₁* and *gal₂* remain two different markers, realising two different feature sets, and each require a different form of the base to their left. However, at the surface level, they are realised the same, because the lexical entry (25c) matches both contexts.

Summarising this section, I first proposed a mechanical way of extending the analysis of Rutul to Tamil. This analysis was adjusted to allow for the fact that the root of the pronoun alternates between *naan* and *en*, and we unified the two *gal*'s. The final analysis I propose here is thus as given in (26).

(26)

<table>
<tr><th></th><th></th><th>REFP</th><th>CLASS</th><th>#</th><th></th><th>NOM</th><th>OBL</th><th>DAT</th></tr>
<tr><td>NOM</td><td>naan</td><td colspan="5">naan</td><td></td><td></td></tr>
<tr><td>ACC</td><td>en</td><td colspan="6">en</td><td></td></tr>
<tr><td>OBL</td><td>en-akku</td><td colspan="6">en</td><td>akku</td></tr>
<tr><th></th><th></th><th>REFP</th><th>CLASS</th><th>#</th><th>PL</th><th>NOM</th><th>OBL</th><th>DAT</th></tr>
<tr><td>NOM</td><td>naan-gal</td><td colspan="3">naan</td><td colspan="2">gal</td><td></td><td></td></tr>
<tr><td>ACC</td><td>en-gal</td><td colspan="2">en</td><td colspan="4">gal</td><td></td></tr>
<tr><td>OBL</td><td>en-gal-ukku</td><td colspan="2">en</td><td colspan="4">gal</td><td>ukku</td></tr>
</table>

In the next section, I show how this lexicalisation is derived using the lexicalisation algorithm and the set of entries proposed in the current section. Before I show this, it is worth mentioning that the analysis in (26) achieves the main goal of this paper, namely it shows that there exists an analysis according to which there is no non-local allomorphy involved in Tamil. Instead, there are two morphosyntactically distinct plural markers, where each of them locally conditions a different base. The only confusing thing is that the two different plurals are realised the same.

6. A complete derivation of the Tamil paradigm

This section shows how the lexicalisation depicted in Table (26) is derived by the lexical entries proposed and the lexicalisation algorithm (27).[6]

(27) a. Merge F and lexicalise FP

b. If fail, evacuate the closest labelled non-remnant constituent and lexicalise FP

c. If fail, evacuate the immediately dominating node and lexicalise FP (recursive)

d. If fail, go to the previous cycle and try the next option for that cycle

The algorithm in (27) implements a generative procedure where individual morphosyntactic features are introduced in the derivation one by one. Whenever a new feature is introduced, the phrase thus created must be lexicalised, see (27a). The word "lexicalise" in (27a) does not mean that the phrase is directly pronounced, however, it means that a matching lexical item must be found for the phrase that has been created by Merge F, where "matching" is evaluated on the basis of the Superset Principle, recall (13).

The algorithm (27) further says that when the phrase marker cannot be lexicalised directly, syntax performs evacuation movements (27b,c). When an evacuation movement is performed, it alters the phrase marker, and lexicalisation is tried again, until it ultimately succeeds. When it succeeds, a new feature may be introduced, and the derivation continues.

[6] The ideas embodied in this algorithm are due to Starke (2022). The phrasing is the same as in the Introduction to this volume (see also Caha and Taraldsen Medová's chapter in this volume).

Let me now introduce the movements, starting with (27b). This clause says that when a direct lexicalisation fails, we must evacuate from FP (which we are trying to lexicalise) the closest constituent with the right properties. The definition of "the right properties" stipulates, e.g., that we cannot move remnant constituents. The constituent also has to be labelled. I introduce the labelling conventions as we go, their essence is the following: any node created by external merge is labelled by the newly introduced feature. Lexicalisation movements give rise to unlabelled nodes.

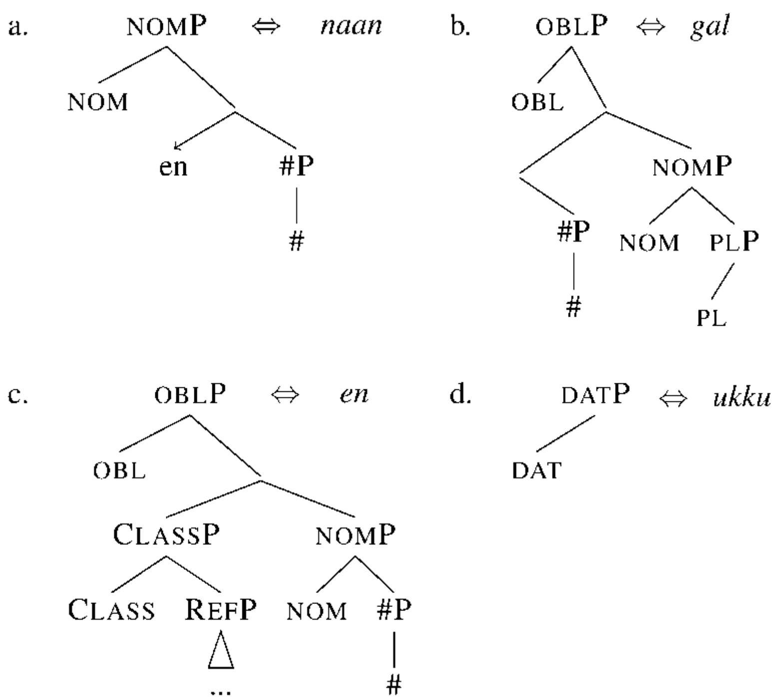
If the closest constituent has been evacuated out of FP, and we still cannot lexicalise FP, we place the moved constituent back, and move to the step (27c). This clause tells us that we should try pied-piping the node that immediately dominates the constituent that we moved on the previous step (regardless of whether the dominating node is labelled ot not). This step applies potentially recursively, pied-piping larger and larger nodes, stopping with this strategy only when pied-piping reaches FP itself.

If all such movements have been tried, and lexicalisation still does not succeed, backtracking is triggered, see (27d). Backtracking removes the feature F from the derivation and tries to change the phrase marker at the preceding stage. When backtracking goes back to the previous cycle of the derivation, it always tries the next option in the algorithm. If, for instance, we get back to a cycle where the derivation was lexicalised without any movement at first, then, when the algorithm returns to this stage, it tries the next option, i.e., the Evacuate Closest step (27b). If the phrase marker is changed, then when F is added again, it is added to a different configuration, and this creates conditions under which the matching of FP (initially unsuccessful) may succeed.

This completes the introduction of this algorithm. It is worth adding that the movements triggered by the algorithm (27) are exactly the same movements as those used in Cinque (2005). Specifically, Cinque observes that out of the 24 logically possible orders of DEM(ONSTRATIVES), NUM(ERALS), A(djectives) and N(ouns), only 14 are attested. To account for this, Cinque assumes that the nominal modifiers are base-generated in a universal hierarchy DEM>NUM>A>N, which is mapped on a left-to-right linear order. Any deviation from this ordering is the result of a constrained set of movements, which fulfil two crucial requirements. One, movement only proceeds to the left, and two, all movements must include the nominal head (which boils down to the requirement that remnant movement is disallowed). These are the exact same properties that are satisfied by the movements triggered by (27), as we shall see. The point is that the types of movements that are triggered by the algorithm are not invented *ad hoc* for the purpose of lexicalisation but represent movement types that have been independently proposed.

With the general assumptions in place, let me repeat for convenience all the lexical entries that we need in the derivation of the Tamil paradigms. They are given in (28).

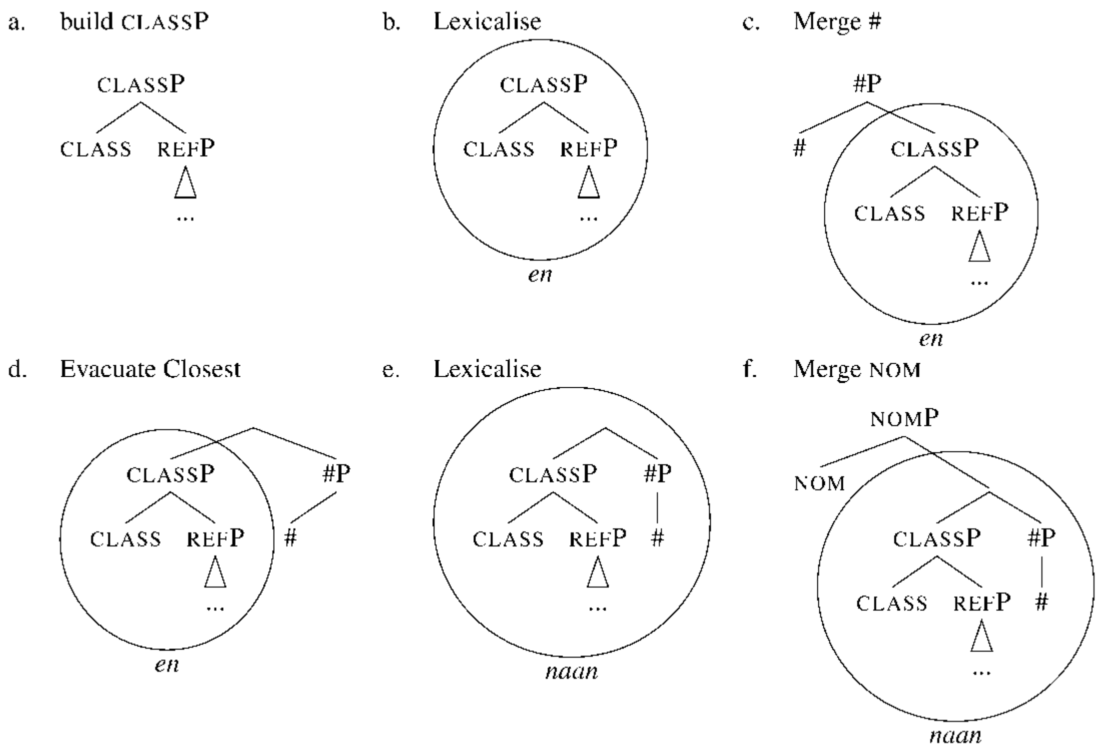
(28)



With the entries in place, let us turn to the derivation. We start following it when CLASSP is constructed, see (29a). Since CLASSP is contained in the entry of *en*, see (28c), lexicalisation is successful, see (29b).

Since lexicalisation was successful, we can merge another feature, namely #, see (29c). There is no matching item for the lexicalisation of #P, which means that the clause (27b) of the algorithm is activated. The clause tells us that we should move the closest non-remnant labelled constituent, which is CLASSP. CLASSP is therefore evacuated from inside #P, yielding (29d).

(29)



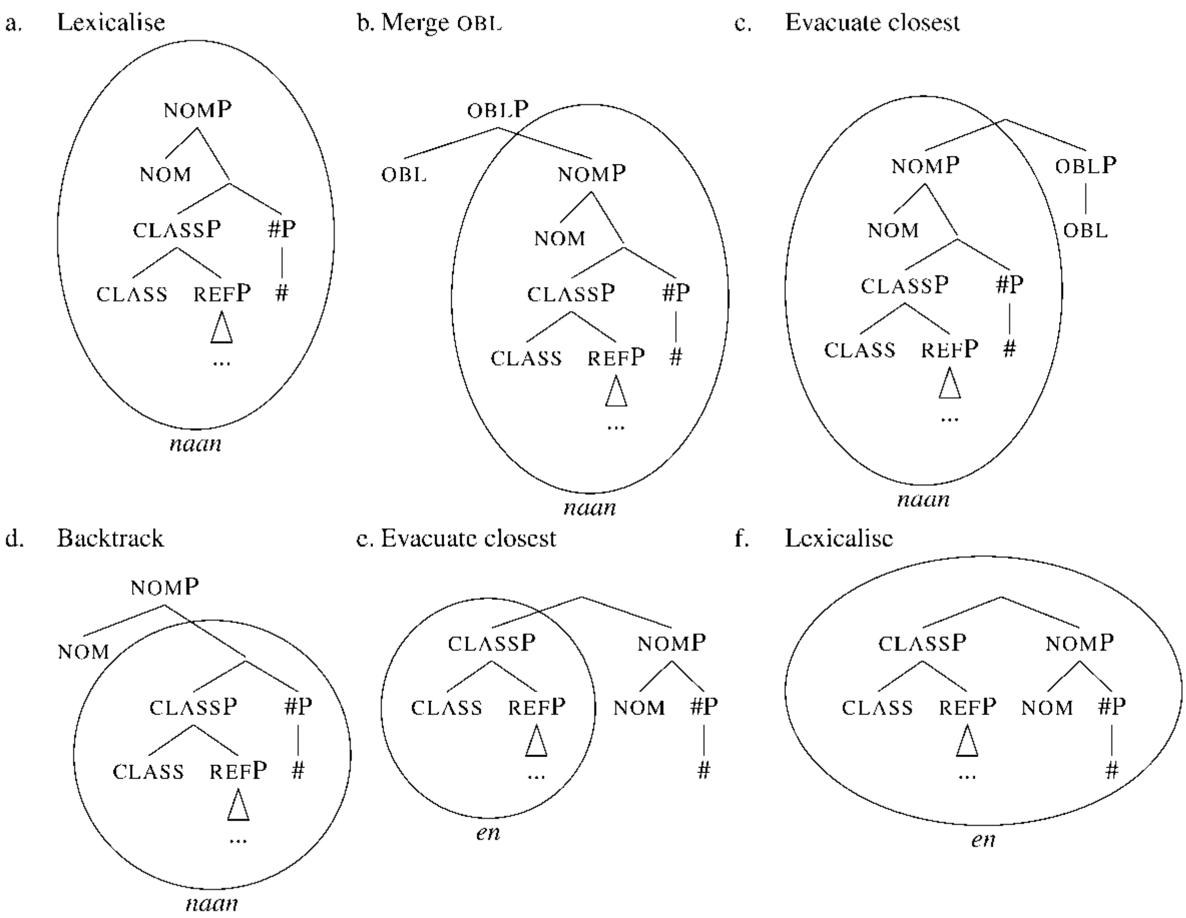
Notice that in (29d), we left out the trace in the complement position of #, as is customary in Nanosyntax. This reflects the idea that lexicalisation movements do not leave traces, or the trace is irrelevant for the purpose of matching, hence eliminated to avoid clutter. Also, since the root node was created by lexicalisation movement, it is not labelled.

Once the movement has been executed, matching is tried again. We can see in (29e) that it succeeds since the whole constituent is contained in the entry of *naan*, recall (28a).

Since lexicalisation was successful, a new feature is merged, labelling the root node, see (29f). This structure is again matched by the entry of *naan*, in fact, the entry (28a) is identical to the structure (29f). Successful lexicalisation is depicted in (30a): this is the correct form of the singular nominative: *naan*.

If the nominative were to be derived, the derivation would terminate. If the oblique case (or the dative) should be derived, we must merge the feature OBL, yielding (30b). There is no direct match for (30b), we must therefore evacuate the closest labelled non-remnant constituent, which is the complement, i.e., NOMP. The evacuation yields the structure in (30c), but there is still no match for the remnant OBLP.

(30)



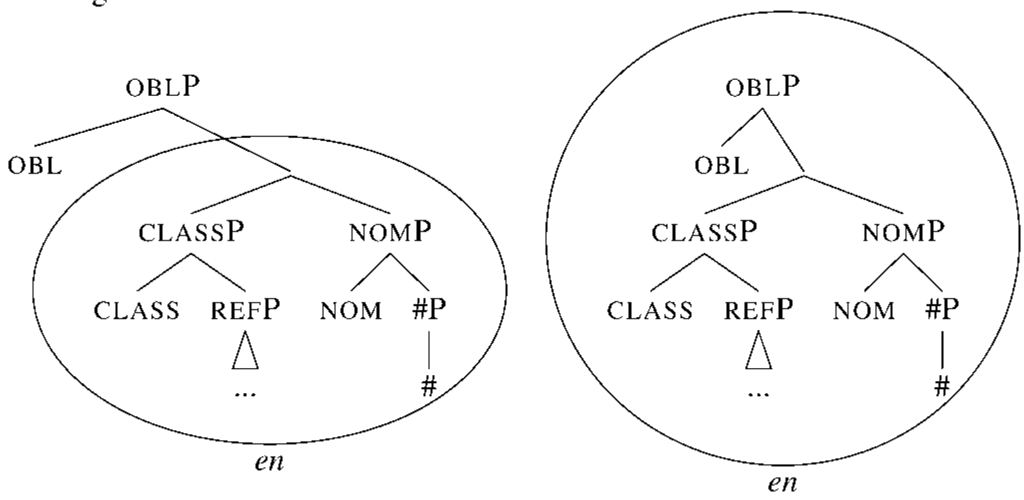
Because of this, we should now try to pied-pipe the immediately dominating constituent (due to (27c)), but since that brings us to the root node, there is no such movement possible.

Therefore, we have to backtrack, following the step (27d). What that means is that we have to go back to the previous cycle and try the next option for that cycle.

The "current cycle" is defined by the feature OBL. Going to the previous cycle thus means going to the cycle when nominative was merged, see (30d). Originally, we lexicalised this structure without any movement, yielding the correct nominative form *naan*, recall (30a). However, that structure led to a dead end when we merged OBL on top of it, so what needs to be done now is that we try to lexicalise the nominative differently. This means that instead of lexicalising it without movement, we try evacuating the closest labelled non-remnant constituent, which is CLASSP. When we move CLASSP, we get (30e). This whole structure is contained in the entry of *en*, recall in (28c). Therefore, the structure is successfully lexicalised as *en*, see (30f).

Note, however, that this is not the correct form of the nominative, we are only following this derivational track because our initial goal is to derive the oblique case. To do so, we must merge the OBL feature on top of (30f), yielding (31a).

(31) a. Merge OBL b. Lexicalise

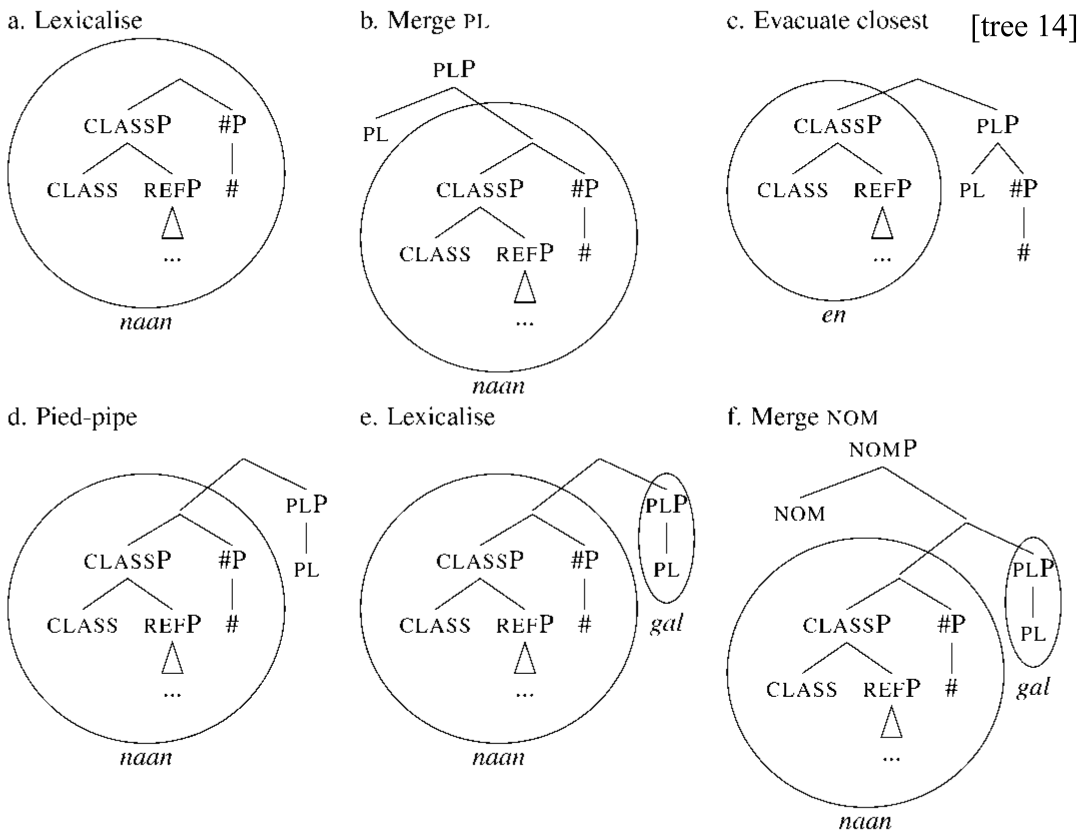


(31a) is contained in the entry of *en* as given in (28c), and we therefore successfully lexicalise the whole structure using this lexical item, see (31b). We have thus correctly derived the oblique singular *en* (without any need for a zero suffix).

The derivation of the dative is not shown here for reasons of space, but it is fairly trivial: we merge DAT on top of (31b), no match. We evacuate the closest labelled non-remnant node, which is the whole complement of DAT (i.e., the whole oblique structure). This structure can be lexicalised as *en-akku*.

Let me now turn to the plural. We start following the derivation at the stage when #P is derived, recall (29e), repeated in (32a). In the singular, we could merge case features directly on top of (32a), but in the plural, we must merge PL first, see (32b). There is no match, so we evacuate the closest labelled non-remnant constituent, yielding (32c). Note that when CLASSP moves out, *en* is no longer a candidate for the realisation of this constituent, and *en* has therefore replaced *naan* as the lexicalisation of CLASSP. Still, there is no match in (32c), so we must go back to (32b) and pied-pipe the node immediately above CLASSP, yielding (32d).

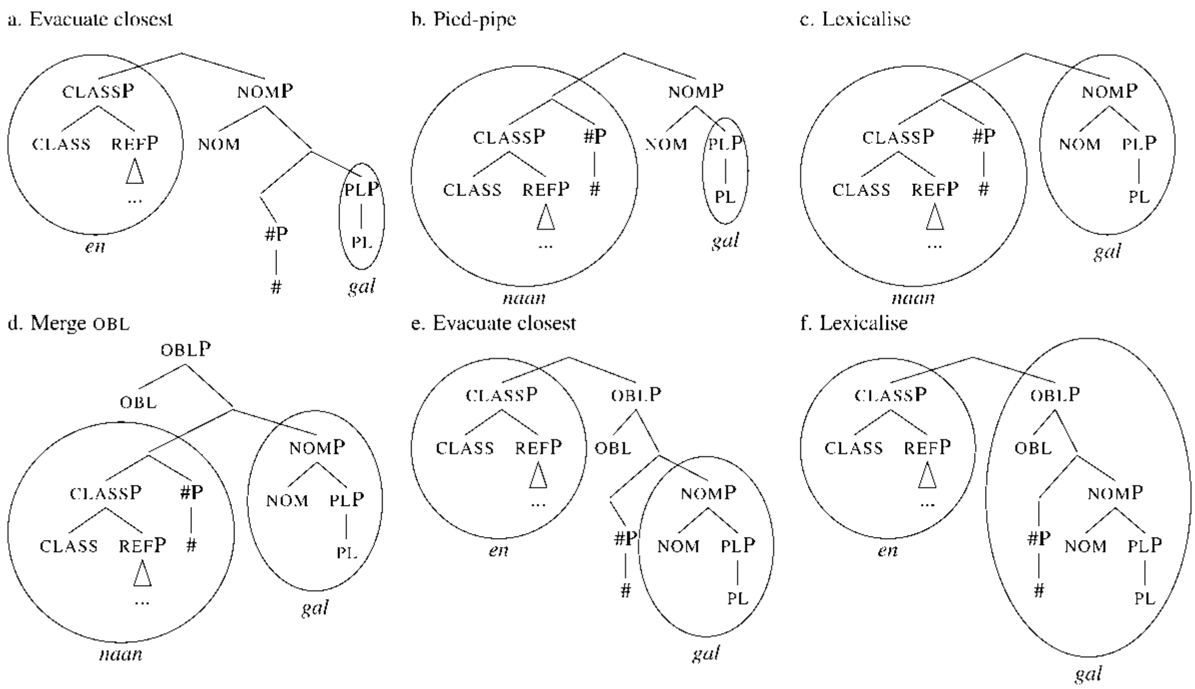
(32) [tree 14]



In (32d), we can lexicalise the remnant PLP by *gal*, since *gal* contains this constituent, see (32e). (The entry of *gal* is as given in (28b).) When we successfully lexicalise the PLP, we merge the NOM feature, see (32f).

Since (32f) has no match in the lexicon, we must evacuate the closest non-remnant labelled constituent. That happens to be CLASSP, yielding the structure (33a). However, there is no match for the remnant NOMP in (33a), so we have to pied-pipe the node immediately above CLASSP in (32f). When we move this node across NOM, we get the structure (33b), where the NOMP can be lexicalised by *gal*, see (33c). This is the correct NOM.PL form.

(33)



If we want to derive the oblique plural, we merge the feature OBL, yielding (33d). Lexicalisation fails, so we move the closest labelled non-remnant node, which is CLASSP. Evacuating this constituent yields (33e). Recall that CLASSP is realised as *en*. After the movement, *gal* is a match for the remnant OBLP, yielding (33f). This is the correct oblique form, namely *en-gal*. Once again, the derivation of the dative is trivial, and it is not shown here for reasons of space.

Summarising, this section provided a walk-through to the derivations that produce the lexicalisation table discussed for Tamil in Section 5. This proves that the pattern can be generated using the proposed lexical entries (given in (28)) and the subextracting lexicalisation algorithm due to Starke (2022), see (27).

7. Conclusions

This paper discussed some empirical issues related to the adjacency condition on allomorphy. The adjacency condition is an apparently intuitive restriction on allomorphy, which says that in a sequence of morphemes, the form of a given morpheme may only be influenced by adjacent morphemes. The reason for proposing this condition is both empirical and theoretical. On the empirical side, it captures a number of cases where allomorphy is blocked by intervening morphemes. On the conceptual level, it reduces the hypothesis space that the child may entertain when acquiring lexical items, thus providing one ingredient to a solution of Plato's problem.

This paper looked at one particularly challenging piece of data coming from Tamil, where this intuitive notion of adjacency is not satisfied: it seems clear to the naked eye that in Tamil, case markers condition the form of the base across the invariant plural marker.

The solution I have offered in this paper shows that at the theoretical level, this example can be explained without invoking a long-distance interaction of morphemes. The machinery I have used for this is a version of Nanosyntax that uses the subextracting algorithm. In this account, allomorphy is determined strictly locally, namely by the number of features that each morpheme realises. In the lexicalisation tables I have provided, all morphemes realise contiguous feature sets, and interact only with the neighbouring morphemes in terms of how many features each of them lexicalises. They never interact long distance or lexicalise noncontiguous stretches of features. Such a scenario does not arise in my analysis of Tamil, and, in fact, it cannot arise under the theory explored in this chapter.

The broader conclusion to be drawn from this is that as our theory of adjacency moves from a superficial to a theoretical level, we may discover interesting discrepancies between the intuitive notion and the theoretical one. One of the conclusions of this paper is that the more abstract, theoretical notion of adjacency is the correct one, and may, in fact, be satisfied even when this seems to contradict the facts that we see with the naked eye.

References

Blix, Hagen. 2022. "Interface Legibility and Nominal Classification: A Nanosyntactic Account of Kipsigis Singulatives." *Glossa: A Journal of General Linguistics*, *7*(1).

Bobaljik, Jonathan D. 2012. *Universals in comparative morphology: Suppletion, superlatives, and the structure of words*. MIT Press.

Caha, Pavel. 2009. *The nanosyntax of case*. PhD thesis, Tromsø.

Caha, Pavel. 2013. "Explaining the structure of case paradigms by the mechanisms of Nanosyntax: The Classical Armenian nominal declension." *Natural Language & Linguistic Theory*, *31*, 1015-1066.

Cinque, Guglielmo. 2005. "Deriving Greenberg's Universal 20 and its exceptions." *Linguistic Inquiry*, *36*(3), 315-332.

Embick, David. 2010. *Localism versus globalism in morphology and phonology*. MIT Press.

Filatov, K. 2020. "Nominal Paradigms in Kina Rutul." *Higher School of Economics Research Paper No. WP BRP 99/LNG/2020*.

Harley, Heidi and Ritter, Elisabeth. 2002. "Person and number in pronouns: A feature-geometric analysis." *Language*, 482-526.

Kibrik, Aleksandr E. 2003. "Nominal inflection galore: Daghestanian, with side glances at

Europe and the world.” In *Noun Phrase Structure in the Languages of Europe*, edited by Frans Plank, pp. 37–112. Mouton de Gryuter, Berlin.

Lamontagne, Greg and and Travis, Lisa. 1987. “The syntax of adjacency.” In: *Proceedings of the West Coast conference on formal linguistics*. Vol. 6. Stanford, CA: Stanford Linguistics Association.

Lehmann, Thomas. 1989. *A grammar of modern Tamil*. Pondicherry: Pondicherry Institute of Linguistics and Culture.

McFadden, Thomas. 2018. “*ABA in stem-allomorphy and the emptiness of the nominative.” *Glossa: A Journal of General Linguistics*, *3*(1).

Moskal, Beata and Smith, Peter W. 2016. “Towards a theory without adjacency: Hyper-contextual VI-rules.” *Morphology*, *26*, 295-312.

Picallo, Carme M. 1991. Nominals and nominalization in Catalan.” *Probus* 3, 279–316.

Siegel, Dorothy. 1978. “The Adjacency Constraint and the Theory of Morphology.” In *Proceedings of NELS 8*, edited by M. Stein, 189–97. Amherst: GLSA.

Smith, Peter W., Moskal, Beata, Xu, Ting, Kang, Jungmin and Bobaljik, Jonathan D. 2019. “Case and number suppletion in pronouns.” *Natural Language & Linguistic Theory*, *37*, 1029-1101.

Starke, Michal. 2009. “Nanosyntax: A short primer to a new approach to language.” *Nordlyd,* 36(1), 1-6.

Starke, Michal. 2018. “Complex left branches, spellout, and prefixes.” In *Exploring nanosyntax*, edited by Lena Baunaz, Karen De Clercq, Liliane Haegeman and Eric Lander, 239– 249. Oxford: Oxford University Press.

Starke, Michal. 2022. Nanoseminar. Online lecture series at Masaryk university, Brno.

Vanden Wyngaerd, Guido. 2018. “The feature structure of pronouns: a probe into multidimensional paradigms.” In *Exploring Nanosyntax*, edited by Lena Baunaz, Karen

De Clercq, Liliane Haegeman and Eric Lander, 277–304. Oxford: Oxford University Press.

Vanden Wyngaerd, Guido, De Clercq, Karen, and Caha, Pavel. 2021. “Late insertion and root suppletion.” *Revista Virtual de Estudos da Linguagem - ReVEL*, *19*(18), 81–123.

Wiland, Bartosz. 2019. *The spell-out algorithm and lexicalization patterns: Slavic verbs and complementizers*. Language Science Press.

Zompì, Stanislao. 2019. “Ergative is not inherent: Evidence from *ABA in suppletion and syncretism.” *Glossa: A Journal of General Linguistics*, *4*(1).