LingBaW. Linguistics Beyond and Within 11 (2025), 23–51

The Teacher and the Frog: Unveiling the Morphosyntax of Gender Shifts in Czech with Nanosyntax

Daniel Boucník
Masaryk University, Czech Republic

Abstract

This study examines the formation of nouns in Czech that show grammatical gender through stem morphemes and three productive suffixes: *-a*, *-k-(a)*, and *-ák*. These suffixes are the main morphological exponents across three noun classes – monomorphemic (e.g. *kmotr/kmotr-a*), bimorphemic (*učitel/učitel-k-a*), and stem–suffix compounds (*žáb-a/žab-ák*). Using nanosyntax, the study shows that gender alternations arise from hierarchical syntactic structures rather than lexical or semantic defaults.

By investigating noun pairs such as *učitel* ('teacher.Masc') / *učitel-k-a* ('teacher.Fem') and *žáb-a* ('frog.FEM') / *žab-ák* ('frog.MASC'), the study shows that masculine forms contain Class and Masc, while feminine forms emerge through the addition of Fem. Suffixes such as *-k-a* and *-ák* function as lexicalization solutions, structured through movement-based operations.

This approach refines our understanding of Czech gender morphology, demonstrating that nanosyntax captures gender alternations systematically even when surface forms are unpredictable. By modeling these alternations within a unified syntactic framework, the study provides a principled account of morphological derivation and paves the way for further cross-linguistic extensions.

Keywords: Czech noun morphology; gender alternation; nanosyntax

1. The morphosyntax of gender alternation

In Czech, grammatical gender plays a crucial role in the morphology of nouns, particularly in how masculine and feminine forms are often derivable from each other, especially in subdomains rich in such gender pairs, for instance profession nouns or animal nouns. While profession nouns are often associated with masculine gender, animal nouns exhibit a more

LingBaW. Linguistics Beyond and Within, e-ISSN: 2450-5188
DOI: 10.31743/lingbaw.18225

mixed distribution, with many smaller or domesticated animals tending to be feminine (*kočka* 'cat', *liška* 'fox') and larger or wild animals often being masculine (*lev* 'lion', *medvěd* 'bear').[1]

Many non-syntactic accounts treat gender pairs as one-directional – the masculine is treated as the base and the feminine as derived, with the masculine interpreted as generic and the feminine as sex-specific (Fiebig 2023; Baggio 2022; Spathas and Sudo 2020). In this article, I adopt nanosyntax which instead derives both forms on the same structural spine: the feminine differs by the additional Fem node above Masc and other grammatical features of the nominal domain (Caha 2009, 2021; Caha, Taraldsen Medová 2022; Harley and Ritter 2002; Janků 2022). Lexicalization proceeds by matching stored trees to the built structure, following the lexicalization algorithm (De Clercq et al. 2025; Starke 2018; Starke 2009). The theory does not privilege the masculine; any perceived "direction" of derivation reflects the lexical inventory of Czech, not a theoretical default gender.

For orientation, two alternative theories would treat the nouns in this work differently, and this contrast motivates the present approach. I will show this contrast on the noun pair *kmotr* / *kmotr-a* that begins section 2 of this work and the analysis itself. In a traditional Slavic description (a lexicalist analysis), *kmotr* and *kmotr-a* are two words: the masculine nominative singular has no overt ending, while the feminine is formed with *-a* and then inflects as a feminine; the relation is stated in the lexicon, not in the structure (Acquaviva 2008). In Distributed Morphology, the root combines with gender/number/case features built by syntax, and Late Insertion – that is, inserting phonological exponents after syntax has finished building the feature structure – chooses Nom.Sg = Ø for [Masc, Sg, Nom] and Nom.Sg = *-a* for [Fem, Sg, Nom] (Halle and Marantz 1993; Harley and Noyer 1999; Kramer 2015). By contrast, nanosyntax integrates lexicalization into the syntactic derivation itself: features are merged one by one along an ordered functional sequence (fseq), and lexical items (LIs) are stored trees that can realize the newly built structure when they match it. This makes nanosyntax particularly suitable for Czech alternations like *kmotr* / *kmotr-a* and *učitel* / *učitel-k-a*, which look different at the surface (one uses *-a*, the other *-k-a*) but are structurally comparable once mapped to the same fseq; in the feminine, *-a* realizes Fem < # < Nom, while the stem shrinks accordingly to realize only the features at the bottom of this functional sequence. Details of the matching procedure are introduced in section 2.

Masculine and feminine nouns differ by exactly one additional feature – Fem – merged above Masc in the gender domain; neuters lack Masc and Fem and show only Class. What that means is that within the part of the fseq where gender features sit (the gender domain), we can find the Class feature, the Masc feature and the Fem feature. These three features represent the three genders of Czech (neuter, masculine, feminine) in the following way: neuter nouns have the Class feature, masculine have the Class and Masc features, and feminines are defined by having Class, Masc and Fem. In other words, neuters exemplify a baseline structure containing the lowermost gender feature Class; merging Masc yields

[1] Neuter nouns also exist: *sele* ('piglet'), *kuře* ('chick'), *štěně* ('puppy') or *prase* ('pig'). This analysis focuses on masculine – feminine pairings and neuters are disregarded.

Class < Masc; adding Fem on top of the two yields Class < Masc < Fem.[2] Thus, gender (and all other notions of semantics and grammar) in nanosyntax is defined syntactically, by the merging and the organization of the featural nodes in the structure of the noun. This approach accounts for both regular and irregular gender alternations, demonstrating that nouns – despite their diverse surface realizations – follow the same underlying hierarchy. By unifying masculine and feminine derivations under a single system, this analysis shows, step by step, how nanosyntax captures gender morphology without presupposing asymmetries as an inherent part of the grammar.

2. How nanosyntax derives a gender alternation

In nanosyntax, the syntax builds structure one feature at a time along an fseq; after each Merge, the derivation attempts lexicalization by matching the built structure with a stored LI (tree in the lexicon). Before proceeding, a quick reminder of the fseq notion: fseq is the ordered spine of features for a domain (here, the nominal domain, or more specifically, the gender domain), determining the order of Merge.

To show the principles of nanosyntax in action, let us start with the simplest possible example, the noun couple *kmotr* and *kmotr-a*. *Kmotr* ('godfather.Masc.Sg.Nom') is a masculine noun and *kmotr-a* ('godmother.Fem.Sg.Nom') is feminine[3]. *Kmotr-a* can be decomposed into two morphemes, the stem morpheme (which in this case is a nominal root), and a suffix, *kmotr-a*. The relationship between these two nouns is derivational: the feminine *kmotr-a* can be derived from its masculine counterpart *kmotr* by adding a suffix *-a*. This approach could be applied to other gendered noun couples, such as *Jan* and *Jan-a*, or *magistr* and *magistr-a*.[4]

In nanosyntax, each morpheme is assumed to have an underlying syntactic structure which represents specific semantic and grammatical content. This content is represented in the form of a (segment of) fseq, a universal hierarchy of features. Starting with *kmotr* and *kmotr-a*, let us look at the inner structure of the stem morpheme *kmotr* and the suffix morpheme *-a*. I claim that these – at the first glance – simple morphemes are intricate; let us use them as steppingstones illustrating the principles of nanosyntax, starting with a more detailed look at fseq.

---

[2] I presume here that in neuter nouns, only the Class feature is present out of the Class < Masc < Fem trio, following Caha's (2021) proposal for neuters.

[3] *Kmotr* / *kmotr-a* are general human nouns. Neither one is the default form, they both are specific, one of them usable for women and the other for men. Still, the relative simplicity of this noun pair makes it a perfect candidate to explain the workings of nanosyntax.

[4] While *kmotr* is used only for male referents and *kmotr-a* for females, *magistr* ('graduate_student.M') is the umbrella term for the group of individuals (see previous footnote) but *magistr-a* ('female_graduate_student.F') is only for female referents. The question of where in the fseq such genericity/specificity should be located, or whether it needs to be a standalone feature at all, is yet to be answered. For more on markedness / specificity / defaultness in gender see Jakobson (1985), Corbett (1991), Bobaljik and Zocca (2011) and Kramer (2015).

Fseq is a universal hierarchy of features, seen as applicable to all the world's languages. Fseq acts as a blueprint, specifying the order in which features are added during the noun's construction and in which the features are merged in syntax. Given that this research deals with nominal phrases (nouns, really), I am interested in the nominal fseq, and the proposed structures follow works by Caha (2021), Caha and Taraldsen Medová (2022), and Janků (2022). For a masculine noun (like *kmotr*), the nominal fseq is NP < Ref < Class < Masc < # < Nom; for a feminine (*kmotr-a*), the masculine sequence is extended by Fem: NP < Ref < Class < Masc < Fem < # < Nom.

NP (Noun Phrase) is the lowest projection in the nominal structure in my diagrams (it sits at the base of the tree); higher features are merged above it along the fseq. It represents all features that would be part of a tree of a noun but are irrelevant to the analysis. Such features might be, e.g., verbal domain features in the case when a noun is derived from a verb. Each feature (Ref, Class, Masc, etc.) can project its own phrase (RefP, ClassP, MascP) unless the feature is displaced during derivation through evacuation or pied-piping. *Pied-piping* describes the movement of a larger node together with the problematic feature, allowing successful matching with a lexical item higher in the structure. The featural node "pied-pipes" the larger node to move with it. NP thus functions as the outer shell that contains all lower projections relevant to the nominal domain. Ref (Reference) encodes the noun's capacity to refer to entities (concrete and abstract), Class forms the base of the gender domain and corresponds to neuter in Czech, Masc adds masculine gender, # encodes Number (here: Singular, the grammatical count feature), and Nom expresses case (Nominative), following Caha (2009).[5]

In nanosyntax, syntax first merges features along the fseq and builds the tree that the lexicon will try to lexicalize. After each Merge, the derivation attempts "lexicalization": a stored LI (a tree) can spell out the current structure if it is a superset of that structure (Superset Effect; Starke 2009). It happens cyclically after each merging of a new feature. Unlike Distributed Morphology, which operates under the Subset Principle (a vocabulary item is inserted if its features are a subset of the position's features), nanosyntax follows the Superset Effect, allowing a single lexical item to lexicalize a structure if it is a superset of the structure (i.e. if the LI contains the structure as part of it).

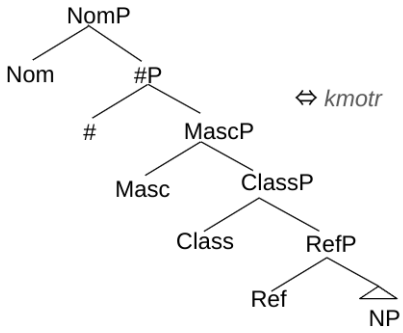
This work uses the De Clercq et al.'s (2025) *subextracting algorithm* which encodes the steps that drive the derivation process, as well as any rescue operations that take place when a matching with the lexicon is unsuccessful. The lexicalization algorithm acts as a set of rules that govern how syntactic structures are constructed to find available lexical items (LIs).

I propose (1) and (2) to be the LIs which are stored in the lexicon for *kmotr* and *-a* (with their LF and PF). They are conventionally marked down by a double arrow between the tree structure and the morpheme's surface form.

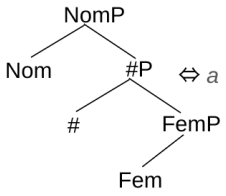
---

[5] The pairing of form and meaning – e.g. why this structure lexicalizes kmotr rather than any other masculine noun with the same feature set – follows from matching the built syntactic structure with stored lexical items in the lexicon. Once a match is found, the structure pairs with both the lexical exponent and the associated concept 'GODFATHER'. The nanosyntactic algorithm searches for structural matches, while lexical retrieval adds the conceptual match, selecting the lexical item containing the intended meaning.

(1) LI for *kmotr*



(2) LI for *-a*

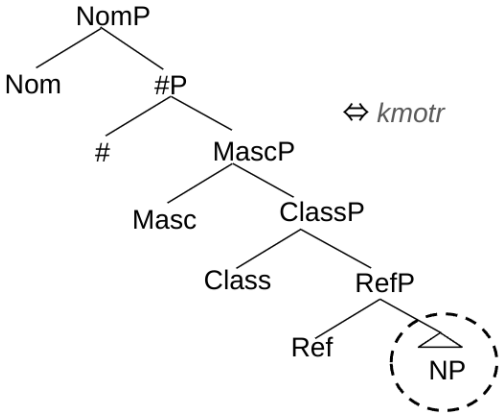


The LIs come into the picture after merging of a new feature by syntax. Each time syntax interfaces with the Lexicon, syntax checks whether there is a stored lexical tree that matches the syntactic tree. In (3), NP is generated (merged), as the first step in the derivation of the noun *kmotr*. (3) represents the syntax. The triangle under NP symbolizes that it is a complex node.

(3) Merge F (the NP)

A necessary ingredient of the lexicalization process is that the matching between the lexicon and syntax operates on the Superset Effect (Starke 2009): the LI must be the superset of the syntactic structure for a match to be successful. Image (4) shows by a dotted circle that there is a subpart of the LI capable of interpreting the NP that was merged in syntax. The reason for this has to do with the matching condition (De Clercq et al., 2025) from which the Superset Effect follows: the LI in (4) is a superset of the newly merged syntactic tree, labelled NP, in (3).

(4) The LI is a superset of NP / NP is contained in the LI for *kmotr*



The lexicalization algorithm determines the steps in the lexicalization process, by steering the derivation in such a way that the language specific lexicon of Czech can be mapped onto the universal functional sequence. So far, I have explored the first step of the algorithm: Merge F and lexicalize. For the sake of clarity, it is repeated in (5).

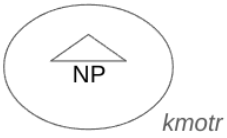
(5) First look at the lexicalization algorithm

I. Merge F and lexicalize.

II. …

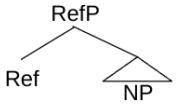
Following step I. of the algorithm, F, in the case of (3) this is NP, needs to get lexicalized. Since the syntactic structure (3) matches a subconstituent of the lexical item in (4), the lexicalization is successful. This is shown in (6). I henceforth mark successful lexicalization by encircling the lexicalized substructure.[6]

(6) Successful lexicalization



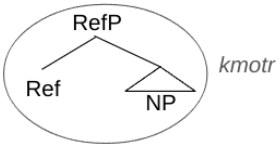
The next feature in the fseq is Reference. This feature is merged with the structure containing only the node NP, changing it to the one in (7). Afterwards, this newly merged projection RefP is lexicalized.

(7) Merge the next F on top of existing structure



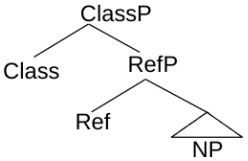
It is easily discernible that the syntactic tree in (7) is again a subconstituent of (4), therefore the LI in (4) is a superset of (7) and can lexicalize it, as visualized in (8).

(8) Successful lexicalization of the second derivation step

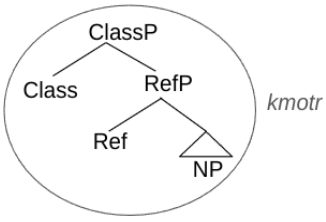


What follows is the merge of Class. This step is shown in (9) and the successful lexicalization in (10).

(9) Merge Class



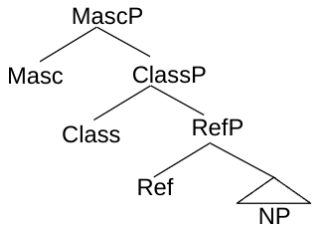
(10) Successful lexicalization of the structure



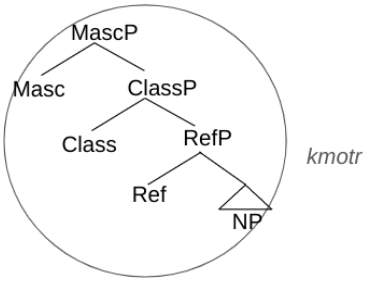
[6] Note that more than one lexical item can, in principle, match the same syntactic structure under the Superset Effect. In such cases, the choice of lexical item is free (Free Choice). See Caha, De Clercq & Vanden Wyngaerd (2019) for discussion.

Afterwards, Masculine is merged in (11), and another smooth lexicalization (12) follows. Before each lexicalization, the LI, in (4), is always checked to confirm whether it is the superset of the current result of the derivation or not. Here it is, without an issue.

(11) Merge Masc

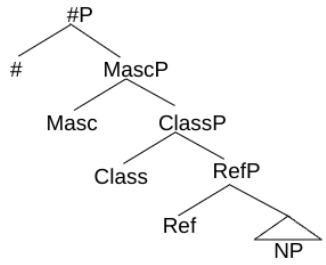


(12) Successful lexicalization

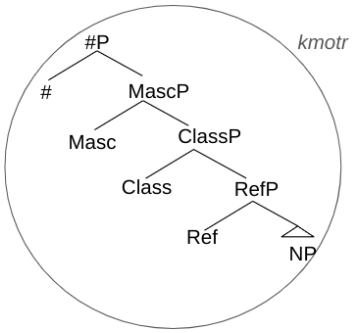


One by one, the features from the fseq NP < Ref < Class < Masc < # < Nom are merged, and a match is found in the lexicon after each Merge, ending in a successful lexicalization. This summary applies to steps (13) – (16).

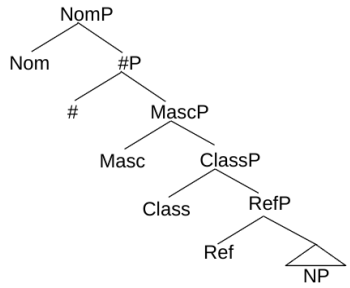
(13) Merge #



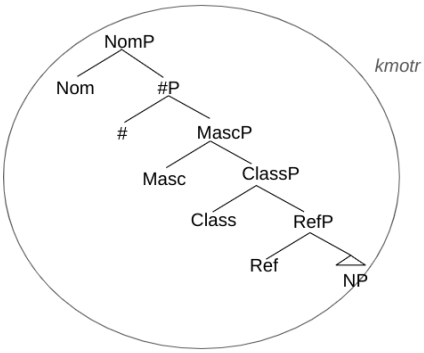
(14) Successful lexicalization



(15) Merge Nom

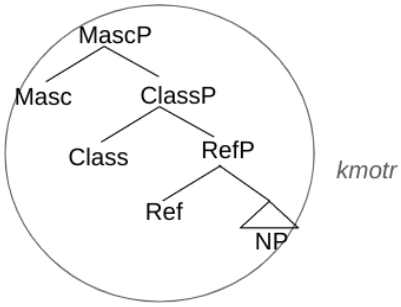


(16) Successful lexicalization



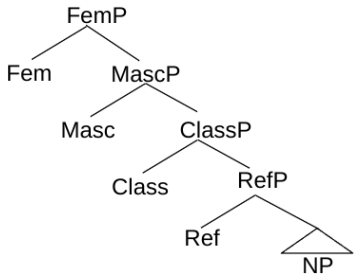
Kmotr is a straightforward example. The end of the fseq is reached, and I can conclude that *kmotr* has been derived. It is now *kmotr-a*'s turn. In deriving *kmotr-a*, I propose the following LIs: the first of them is the already defined nominal stem *kmotr* and the second is the suffix *-a*, which introduces Fem. The fseq for *kmotr-a* is going to be NP < Ref < Class < Masc < Fem < # < Nom. Merging NP, Ref, Class, and Masc follows the same steps as in *kmotr*. Let us jump forward to that point in the derivation when Masc is lexicalized as *kmotr*, (17) and the derivation continues.

(17) Lexicalize as *kmotr*

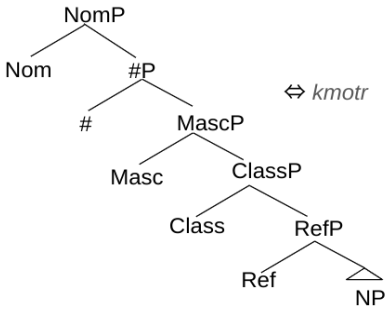


In (18), Feminine is merged, and for the first time, lexicalization is not possible. The reader can check for themselves by comparing the LI for the stem *kmotr*, repeated for convenience in (19), that there is no Fem in the tree. In other words, the LI is not a superset of the syntactic tree in (18) and hence lexicalization is not possible.

(18) Merge Fem



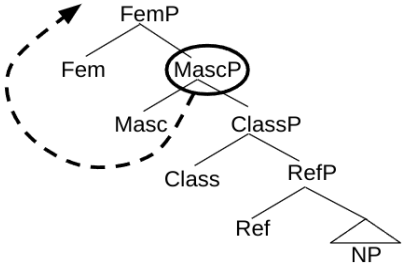
(19) LI for *kmotr*



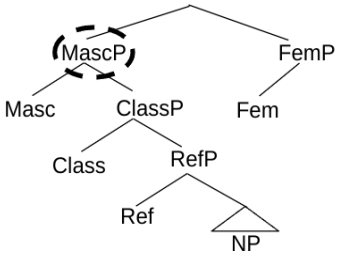
Whenever the lexicon cannot provide a matching lexicalization, the syntax will have to do a rescue operation. The rescue operations that the syntax may have to perform are ranked in the lexicalization algorithm. After merge, which I detailed as the first step in the algorithm, evacuation movement is the next possible operation. Starting from the newly merged feature and its phrasal projection, the syntax will search for the *Closest Labeled Non-Remnant Constituent. CLN-RC* (Closest Labeled Non-Remnant Constituent) identifies the nearest syntactic constituent (here: MascP) that can be evacuated when lexicalization fails. That constituent is then moved to the Specifier position of the newly merged featural head.[7] The movement is shown in (20) and its result in (21).

[7] The descriptor "Specifier position" is in nanosyntax taken from X-bar phrase structure as a shorthand for navigation in the syntactic tree, which the reader might be familiar with. The internal structure of nanosyntax and the rules for movement in it differ from the standard Spec-to-Spec movement and Spec of the Complement movement, as I will explore partly in the derivation of *kmotr-a* and partly in the derivation of *učitel* and *učitel-ka* in section 3.

(20) CLN-RC evacuation



(21) Evacuation result



(22) illustrates this rescue operation as the second step of the algorithm.

(22) Lexicalization algorithm

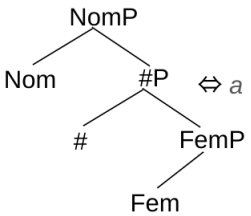
I. Merge F and lexicalize.
II. If fail, evacuate the Closest Labeled Non-remnant Constituent (CLN-RC) and lexicalize.

III. …

There is quite a lot to unpack in this second step of the algorithm: The newly merged feature has a phrasal projection FemP above it, which serves as the focus of our attention. When the algorithm says, "evacuate the *closest* … *constituent*", it means the closest in relation to the newly merged feature (Fem). Secondly, "evacuate the … *labeled* … constituent" is there to leave out non-labelled nodes, i.e. those resulting from a previous movement. Lastly, "evacuate the … *non-remnant* constituent" prevents from trying to move remnants, i.e. nodes that something has been evacuated from.

With the terminology clarified, let us focus back on the derivation in (21), where the structure has two branches. The left branch can be lexicalized by *kmotr*, but the right branch cannot. I proposed an LI for *-a* in (2) and repeated here in (23), which can lexicalize the right branch of (21), because the right branch is a subconstituent of the LI in (23).

(23) LI for *-a*



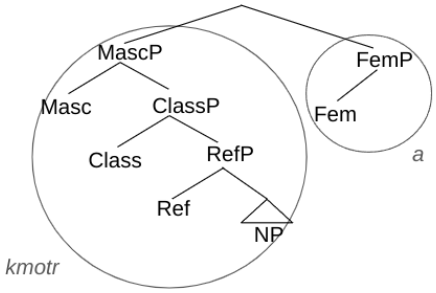
The features Fem, #, and Nom are grouped in *-a* because Czech *-a* appears exactly where these three features coincide – in the nominative singular of feminine nouns. Treating them as a single bundle reflects this co-occurrence and distinguishes this *-a* from the homophonous masculine *-a* (as in *hrdin-a*), which lacks *Fem* and serves only inflectional purposes.

While *-a* typically introduces the feminine feature in Czech, there exists a small but systematic class of masculine nouns ending in *-a*, such as *hrdin-a* ('hero.M') and *koleg-a* ('colleague.M'). These forms share some inflectional properties with feminines but pattern syntactically as masculines (the change is visible in Dative and Locative Singular but is invisible in Nominative and other grammatical cases of the Singular Number). The simplest

account is that there are two homophonous LIs: a feminine *-a* (Fem > # > Nom) and a purely inflectional *-a* (# > Nom) used in masculine paradigms. A full analysis is left for future work.

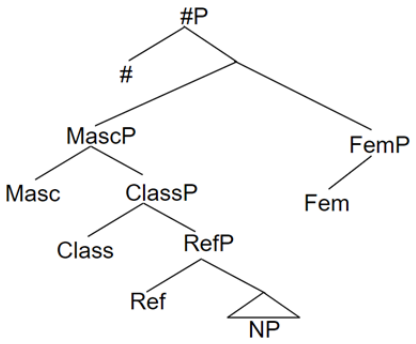
The lexicalization of *-a* is shown in (24): the left branch is realized by *kmotr*, the right branch by *-a* and the whole structure has the surface form of *kmotr-a*.

(24) Lexicalization by *kmotr* and *-a*



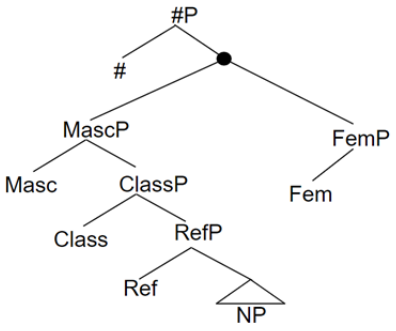
Number (#) is the next feature in the queue. It is merged on top of the structure in (25), attempt lexicalization, and find out there is no match in the lexicon.

(25) Merge #



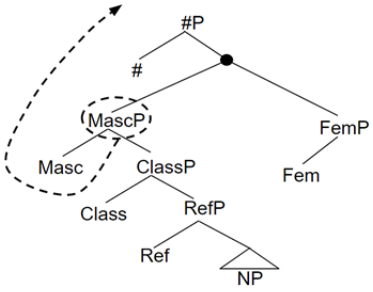
The way out is step II of the algorithm in (22), just like when merging Fem: Evacuate the closest labeled non-remnant constituent (CLN-RC) and lexicalize. For the first time in this derivation, the structure contains an unlabeled constituent because of the previous rescue movement, highlighted by a black dot in (26).

(26) Highlighting the CLN-RC

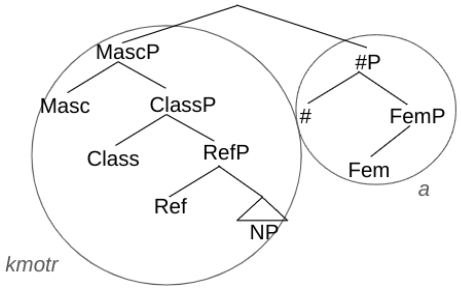


The unlabeled node is skipped and MascP is moved. The movement is in (27) and results in (28).

(27) Evacuate CLN-RC

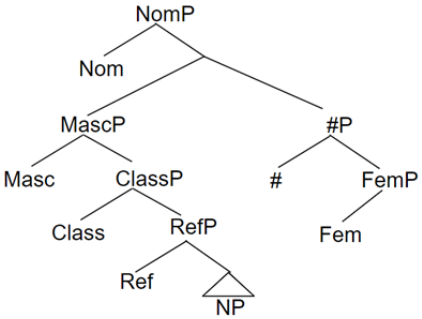


(28) Lexicalize as *kmotr-a*



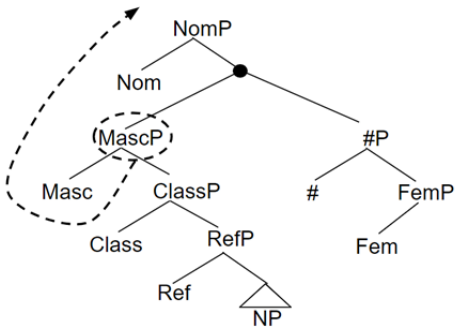
Finally, Nominative is merged in (29).

(29) Merge Nom

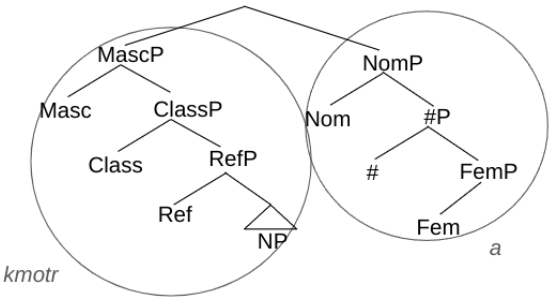


The lexicalization succeeds after a CLN-RC evacuation in (30) and the result is in (31). Since the left branch is lexicalized by *kmotr* and the right branch by *-a*, the total is *kmotr-a* (‘godmother.Fem.Sg.Nom’). The lexicalization of *kmotr-a* has been successful.

(30) Evacuate CLN-RC



(31) Lexicalize as *kmotr-a*



Another way to visualize derivations, which shows the same actors but accents their roles differently, is a lexicalization table, shown in (32). The topmost row in the table contains the fseq. The other rows represent different morphemes or phrases and the corresponding parts of the functional sequence which they lexicalize, e.g. the stem *kmotr* lexicalizes all features in the fseq (compare with LI 3 / 19). For *kmotr-a*, the lexicalization table shows that the stem morpheme *kmotr* will shrink, i.e. it will not lexicalize its full lexical potential. *Shrinking* refers to a lexical item realizing a smaller subset of the functional sequence than it potentially could, often as a result of an evacuation / pied-piping operation. As such, it will only lexicalize NP, Ref, Class and Masc; # and Nom will be lexicalized by *-a*, the morpheme that kicked in to lexicalize Fem. The colors in the table show how far every morpheme reaches and how many features it can lexicalize (i.e. *kmotr* lexicalizes all features up to Nom). The black spot shows the absence of a feature (i.e. *kmotr* does not contain Fem).

(32) A lexicalization table for *kmotr* and *kmotr-a*

ΔNP	Ref	Class	Masc	Fem	#	Nom
kmotr						
kmotr				a		

The derivation of *kmotr-a* thus illustrates the basic functioning of the algorithm. We can now assess the same mechanism on a more articulated stem.

3. The role of the complex left branch

The lexical trees in *kmotr* / *kmotr-a* were straightforward, non-branching. Now, let us look at the masculine *učitel* ('teacher'). Even though it consists of a root *učit* and a suffix *-tel* (*uči-tel*), I treat this noun as the stem / one functional morpheme for the purposes of this analysis, so in the lexicalization table (33) it realizes all the features up to Nom. As a consequence of this decision, I presume that *-tel*, if it were analyzed separately, would lexicalize the Ref feature. The table proposes the division of labor between the stem *učitel* and other functional morphemes.

(33) Lexicalization table – *učitel* / *učitel-k-a*

NP	Ref	Class	Masc	Fem	#	Nom
učitel						
učitel		k		a		

If the two masculine nouns I have presented here behaved identically to each other, *učitel*'s expected feminine form would be **učitel-a*, just like *kmotr-a*. Yet the actual feminine form for 'female teacher' is *učitel-k-a*. The answer to the problem lies again in the structure, but to get to it I first have to propose how the functional morphemes divide the fseq.

There is already a proposal for what *-a* lexicalizes: I argued it lexicalizes Fem, # and Nom. Therefore, I propose to segment the ending of *učitel-ka* into *-k* and *-a*, with -a lexicalizing Fem, # and Nom and k lexicalizing the middle features Class and Masc. The root, *učitel*, will

have to shrink to Ref, as visualized in the second row of (33). Since (33) is just a lexicalization table, one wonders how *-k* will be able to kick in and how the shrinking of *učitel* comes about?

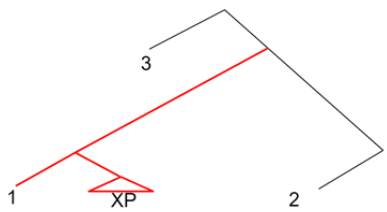
Before continuing with lexicalization, a brief note on *-k*. The suffix *-k* (often with the vowel *-e-*) is central and highly flexible in Czech word formation. It acts as a nominalizer creating concrete nouns (*úplat-ek* 'bribe.Masc'), results of actions (*zlom-ek* 'fraction', *zbyt-ek* 'remnant'), and agentive or univerbated forms (*minerální voda* 'mineral water.Fem' → *minerál-k-a* 'mineral_water.Fem'). Because it lexicalizes the low nominal domain – Class and Masc – it can surface in feminine (*řidič-k-a* 'driver.Fem') and diminutive (*nož-k-a* 'foot.Fem.Dim') contexts alike. Traditionally, *-k* is considered synonymous with *-ice* in feminine derivation (*děln-ice* 'female_worker') (Dokulil 1986: 218–231; Štícha 2018: 160). In what follows, we will see how this flexibility enables *-k* to enter the derivation of *učitel-k-a*, adding a uniquely nominal layer.

Although *-k* appears in several environments (feminization, diminutives, …), in this paper it remains an affixal exponent of the low nominal domain: it realizes Class and Masc (and, when word-final, may co-realize #/Nom). Crucially, I do not assume that *-k* carries a referential layer. Its output becomes a full noun because the derivation builds that layer elsewhere (e.g., through higher functional structure or another suffix), not because *-k* itself introduces Ref. In other words, *-k* is multifunctional by distribution, not by raising the noun to referentiality on its own.[8]

Returning to the derivation, complex left branches (CLB) are now crucial. In a CLB, one or more nodes of the lexical tree are in a position that is higher than their original position. This is a consequence of evacuation movement, i.e. a feature at a lower level has moved to a higher level (Blix 2022).[9] Since all lexical items in nanosyntax consist of well-formed syntactic objects, and since we know that tree structures with evacuation movements are well-formed syntactic objects, it is not surprising that this type of object gets stored in the lexicon. Janků (2022) and Cortiula (2023) show how lexical items with complex left branches are extremely useful in the analysis of nouns and verbs respectively. In this paper, their use shows how morphemes can "shrink", which is exactly what we see with the stem *učitel* in *učitel-k-a*.

A CLB can be seen in picture (34). The order in which features have been merged is XP – 1 – 2 – 3.

(34) A complex left branch (CLB)

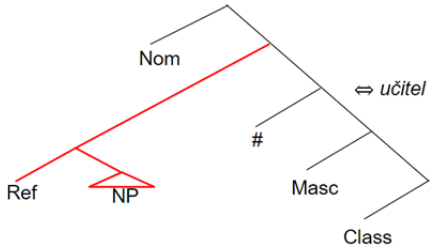


[8] A CLB for *-k* is a theoretical possibility. Future work may probe whether some *-k* items show CLB-like behavior but the derivations in the present account do not require it.

[9] CLBs admittedly give nanosyntax boost in generative power by allowing for certain ABA patterns. For more information, see Blix (2022), Caha et al. (2024), Kasenov (2023). For more on *ABA in adjectival paradigms see Bobaljik (2012) and others.

To the naked eye this seems the fseq order was violated, but such structures are still syntactically salient, and it is logically possible for LIs to vary in this way (Caha et al. 2022). Picture (35) shows this way of drawing the tree in the specific case of the noun *učitel*: the shape of this LI will allow that a syntax that has the structure of the left branch, i.e. containing NP and Ref, can be lexicalized by *učitel* without the need for the features Class, Masc and #. This is exactly what will be needed for the lexicalization of the feminine form. [10]

(35) The complex left branch (CLB) in the LI for *učitel*



Based on these general notions and observations, I propose that *-k* carries Class, the basic gender feature. Secondly, *-k* carries # and Nom because it can be word final. It is word-final in masculine nominalizations, too, and therefore an ideal carrier of Masc. I do not expect *-k* to lexicalize Fem, as it is not unique to deriving feminines, in contrast with *-a*.

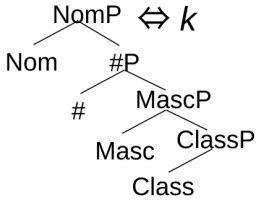
This specification also covers diminutive formations across genders: In masculine diminutives like *dědeč-ek* ('grandpa.Dim.M'), *-k* again realizes the low slice of the fseq (Class, Masc). Because the it is word-final, #/Nom can be co-realized there. The gender remains masculine because no Fem feature is introduced. In feminine diminutives like *žab-k-a* ('frog.Dim.F'), *-k* still spells out the same low slice and *-a* contributes Fem, #, Nom. In neuter diminutives like *slun-íčk-o* ('sun.Dim.N'), the cluster *-íčk-* contains *-k* as the consonantal exponent of the low slice, while *-o* provides the neuter inflectional ending (realizing #/Nom in the neuter paradigm). In all three patterns, the "diminutive effect" comes from the larger suffixal complex (the vocalic material and the final inflection), while *-k* consistently realizes the same low-domain features.[11] In neuter diminutives such as *bříšk-o*, the absence of Masc means *-k* can lexicalize only up to Class, and the neuter ending *-o* supplies the remaining #/Nom. Masc (and Fem) would be represented by a black cell in the lexicalization table.

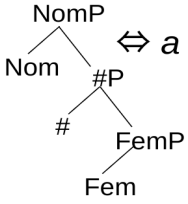
In (36) I propose the separate LIs for *učitel, -k* and *-a*. With all three together, it is now possible to derive *učitel-k-a*.

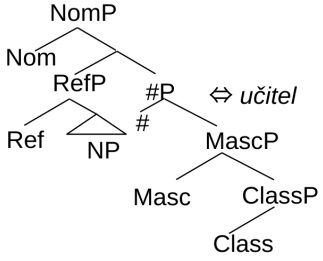
[10] As an alternative, *-a* could have a CLB. If its LI contains a left branch on which Fem can be absent (due to prior evacuation), the same item can lexicalize the right branch both with Fem (Fem < # < Nom) and without Fem (# < Nom). This covers the feminine *-a* and the masculine *-a* with a single LI by structure, not by ambiguity. However, this remains a tentative explanation: the exact tree geometry and steps that would create such a CLB are not clear, as is uncertain whether it is achievable by a well-formed derivation at all.

[11] Diminutive might also be analyzed as a separate feature of the fseq. This analysis is left to future research.

(36) LIs for a) *-k*, b) *-a* and c) *učitel*

a)


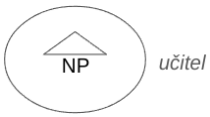
b)


c)


Even with an innovated shape for LI, the derivation of *učitel* still follows the algorithm. Merge NP (the first node in our nominal fseq) in (37). *Učitel* (36c) is a superset of NP even though it contains the NP on a left branch (cf. (36)). This LI can still lexicalize the newly merged NP as *učitel*, as visualized in (38).

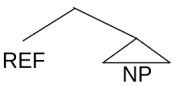
(37) Merge F (the NP)

(38) Lexicalize as *učitel*

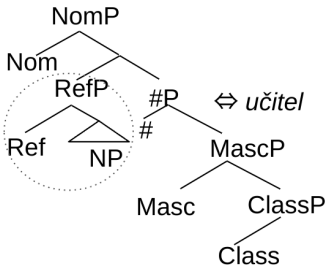


Subsequently, in (39), Ref is merged. *Učitel*, repeated in (40), matches this structure and acts as a superset of it, therefore the lexicalization goes through without an issue, summarized in (41).

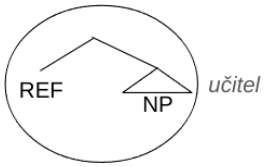
(39) Merge Ref



(40) LI for *učite*

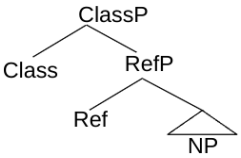


(41) Lexicalize *učitel*

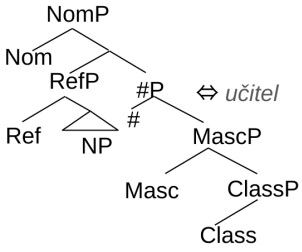


Continuing, Class is merged, with (42) being the result. But even though lexicalization is attempted, there is no constituent in the superset of the LI for *učitel*, again in (43), which would match it, and the attempt fails.

(42) Merge Class



(43) Structure of the LI *učitel*



Fortunately, the algorithm is ready for this: when it fails, the second step instructs to evacuate the closest labeled non-remnant constituent.

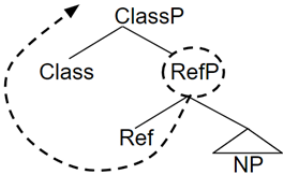
(44) Lexicalization algorithm

I. Merge F and lexicalize.
II. If fail, evacuate the Closest Labeled Non-remnant Constituent (CLN-RC) and lexicalize.

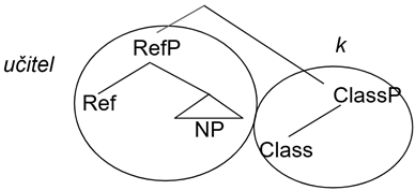
III. …

The CLN-RC evacuates above the newly merged feature. (45) visualizes the movement step and (46) the results of the movement, highlighting the part available for lexicalization by *učitel*. This step is one of the first operations that offers additional options when failure is imminent.

(45) CLN-RC evacuation

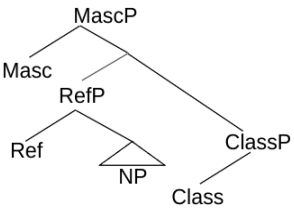


(46) Lexicalize *učitel-k*

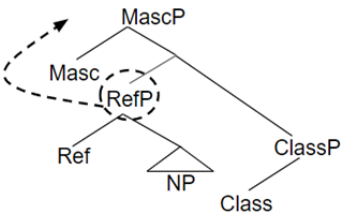


(46) might seem to lead to failure even so. The Class feature on the right branch is, after all, lexicalized by a morpheme *-k*, proposed back in (36a), which does not occur in the masculine *učitel*. This item can, nonetheless, lexicalize this right branch and only thanks to it can the derivation continue. This does not lead to failure, because there are still features to merge. The derivation continues with Masc in (47). *Učitel* does not find a match and a CLN-RC evacuation follows. The constituent in question is highlighted by a dashed line in (48).

(47) Merge Masc

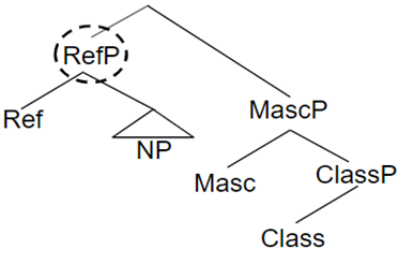


(48) Evacuate CLN-RC

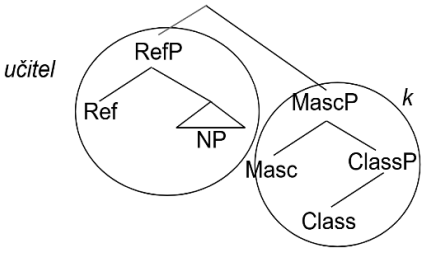


The resulting structure in (49) has two branches now: a left one, containing a complex moved constituent, and a right one, which is a remnant constituent with Class and Masc. The left branch can be interpreted by *učitel,* and lexicalization is successful, in (50).

(49) The result of CLN-RC evacuation

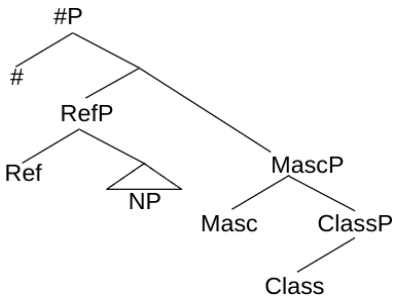


(50) Lexicalize as *učitel-k-*

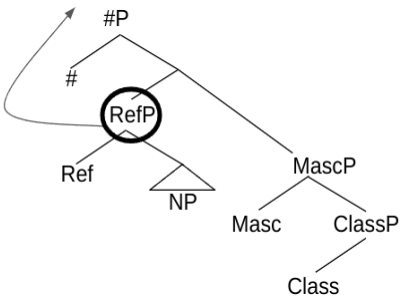


In (51), # is merged, without a match, followed by a CLN-RC movement in (52).

(51) Merge #

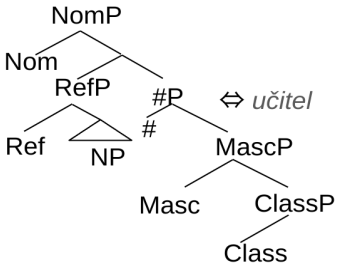


(52) Pied-pipe

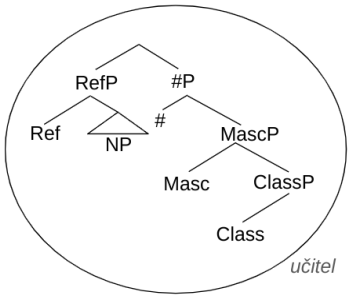


When we check the structure of *učitel*, repeated in (53), there is a clear match in it for the result of the derivation: the whole tree is lexicalized as *učitel*, with no branches left out. Check for yourselves in (54).

(53) LI for *učitel*

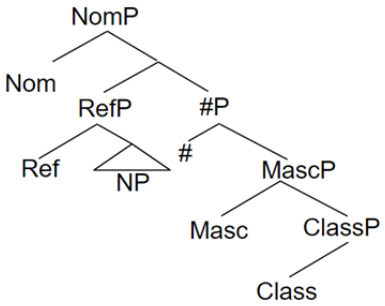


(54) Lexicalize tree as *učitel*

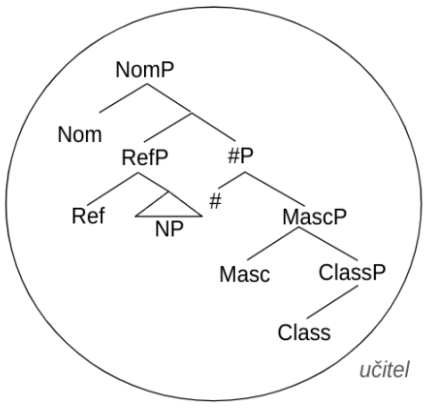


Finally, Nom is merged in (55) on top of the preceding step, and the structure exactly matches the lexical tree for *učitel*. Lexicalization is successful, as shown in (56).

(55) Merge Nom

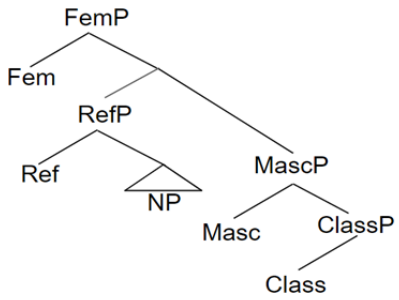


(56) Lexicalize *učitel*



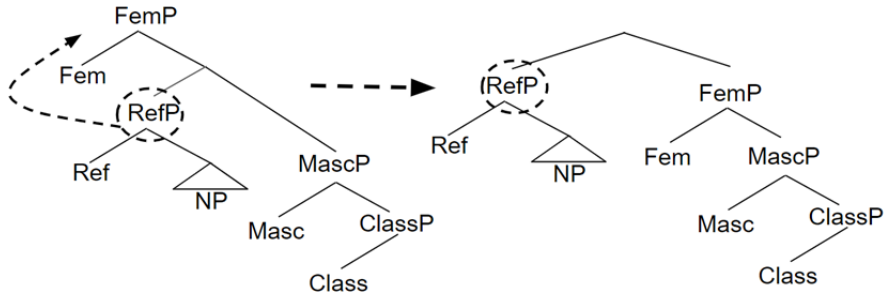
The derivation of *učitel* is finished. Since *učitel* and *učitel-k-a* both use the *učitel* derivational stem and, inevitably, a lot of steps in their derivation repeat. The crucial difference is that *učitel-k-a* is a feminine form and as such is presumed to contain the Feminine feature. In the fseq, Fem is right after Masc, so until Fem is merged in the structure, it is possible to recycle steps (37) to (50). Picture (57) therefore fast-forwards to the crucial moment when Fem is merged, and we divert from *učitel*.

(57) Merging Fem



Fem is added with no match from the lexicon for the whole tree, and the structure resulting from a CLN-RC movement is without a match either. For that step see (58).

(58) Evacuation of the Closest Labeled Non-Remnant Constituent to NP



There is one more rescue movement to try out if moving the CLN-RC does not yield results: Pied-piping the next higher (unlabeled) node. This strategy is now mentioned in line III of the lexicalization algorithm in (59).

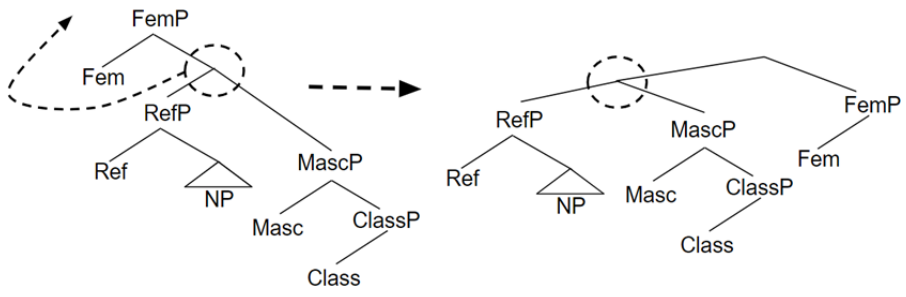
(59) Lexicalization algorithm

I. Merge F and lexicalize.
II. Evacuate the closest labeled non-remnant constituent (CLN-RC) and lexicalize.
III. Pied-pipe a node one notch higher and lexicalize.

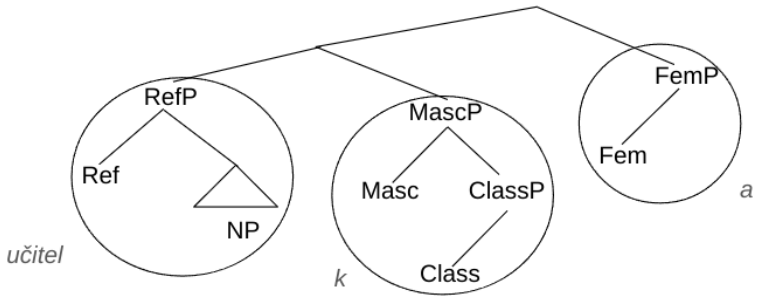
IV. …

Pied-piping starts from the unsuccessful movement in step (58), targets a node one notch higher in the tree, and moves it above the phrasal projection of the newly merged feature (see (60) for both the movement and the result). It produces a syntactic structure with three branches: the left one containing NP and Ref, the middle one with Class and Masc, and the one on the right with Fem.

(60) Pied-piping a higher node

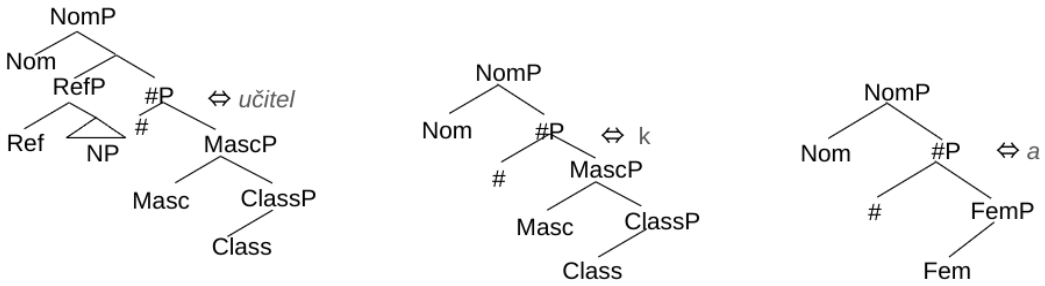


(61) Lexicalize *učitel-k-a*



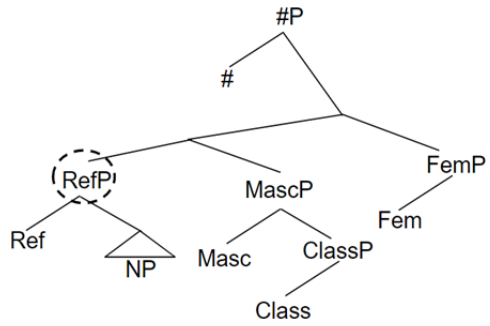
If we compare the three branches of (61) to the LIs, repeated in (62), we can see that LI for *učitel* is the superset of the left branch, the LI for *-k* is the superset of the middle branch and *-a* is the superset of the right branch.

(62) LIs for *učitel*, *-k* and *-a*



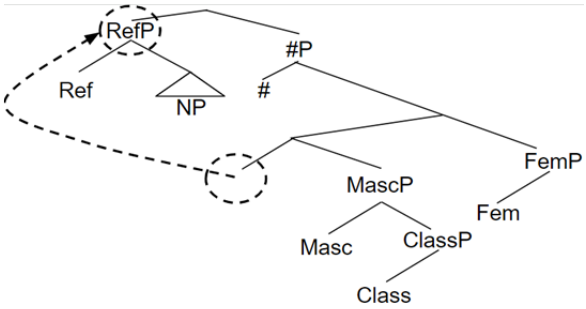
Two more features are left to merge: # and Nom. # is merged first, on top of the previous step of the derivation. Immediate lexicalization is not possible, but for a rescue movement to work, syntax needs to identify the closest labeled non-remnant constituent. Since both the right and the middle branches have been evacuated from, they are remnant constituents, therefore they do not fit the algorithm's rule. The derivation skips the unlabeled nodes, as they also have been created by the evacuation movement operations. (63) shows both the structure at the merging of # and the CLN-RC, RefP, highlighted by a dashed oval shape.

(63) Merge #



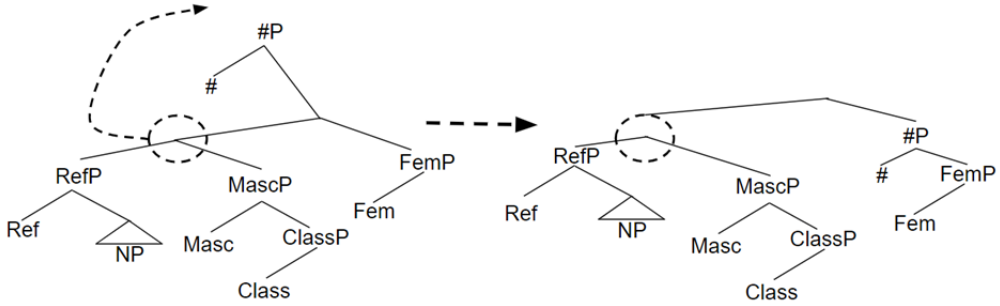
This movement produces the complex structure in (64) which does not find a match in the lexicon.

(64) Evacuate CLN-RC



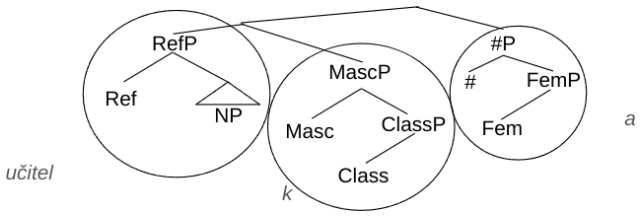
Therefore, in (65), the next highest notch is pied-piped.

(65) Pied-piping a higher node



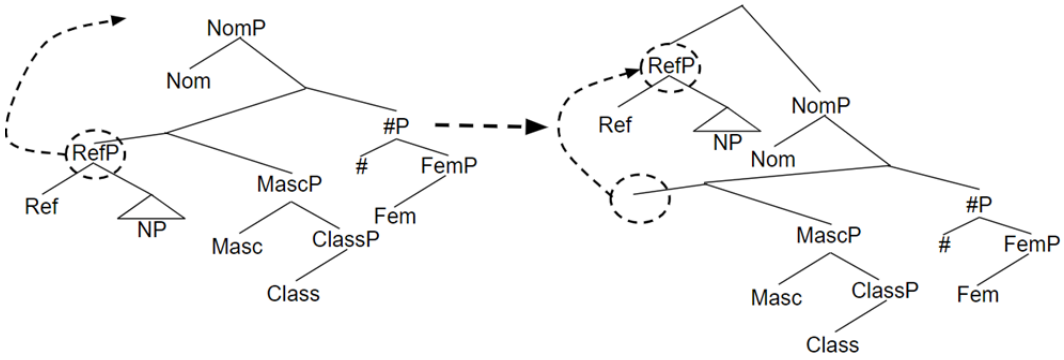
Finally, in (66) there is a match for each one of the branches.

(66) Lexicalization successful



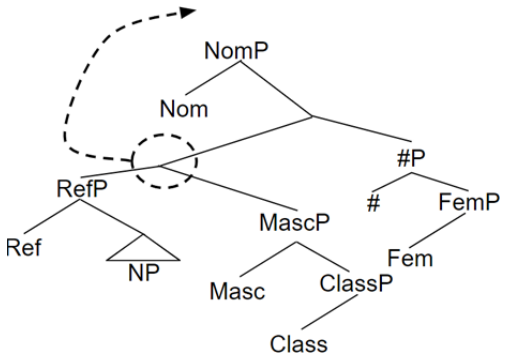
The last step, at least to derive *učitel-k-a*, Nominative is merged as the first case feature, with no luck in immediate lexicalization, followed by a CLN-RC movement in (67), also without acceptable matching results.

(67) Evacuation of CLN-RC leads to failure



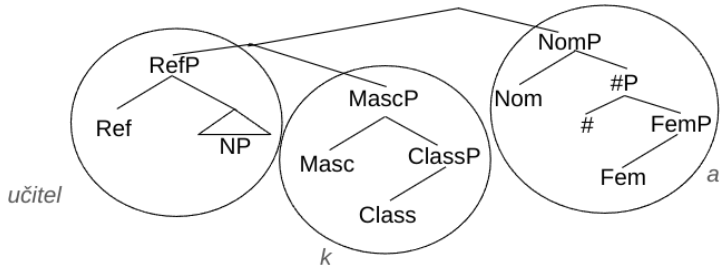
A higher node is pied-piped in (68), just like in the previous steps of the derivation, and the outcome is successful lexicalization.

(68) Pied-piping a node one notch higher than RefP



All features from the fseq have been merged, and the form *učitel-k-a* has been derived, leading to the picture in (69).

(69) Result of the pied-piping and successful lexicalization of *učitel-k-a*



To summarize, this paper has so far contained a detailed explanation of how masculine and feminine forms of nouns are derived when following the algorithm of nanosyntax. Structures for several LIs have been proposed, built on the idea of a universal hierarchy of features in the form of an fseq. The structure inventory consists of two separate stem morphemes, *kmotr* and *učitel*, the latter exemplifying a complex left branch in its tree. Furthermore, there are two separate suffixal items that have been used to derive feminine forms of *kmotr-a* and *učitel-k-a*, but whose usefulness and applicability is yet to be assessed. Such an assessment comes in the form of deriving more nouns that contain similar morphology, and that is why in the following paragraphs nominal forms with more complex stems will be introduced, which undergo gender alternation. The next section shows the complementary case of *-ák*: unlike *-k*, *-ák* carries a CLB with Ref and therefore licenses a referential noun directly.

4. Nouns with bi-morphemic stem

The aforementioned noun *žab-ák* (‘frog.Masc’) pairs with its feminine correlate *žáb-a* (‘frog.Fem’). An analogy with *kmotr* → *kmotr-a* might suggest adding *-a* to *žabák* and

obtaining *žabák-a*, which is in fact the accusative case form of the same masculine noun.[12]. That expectation is incorrect. The reason is structural: in the masculine *žab-ák**, the suffix* *-ák* comes with a lexical entry that does not introduce Fem and leads to a different distribution of coverage for the root. When Fem is merged, the matching proceeds differently: the LI *-a* lexicalizes out the slice Fem < # < Nom, and the stem *žab-* shrinks to its lower portion in order to fit that match. The feminine correlate is therefore *žáb-a*, not *žabák-a*. The table in (70) and the trees in (71)–(73) display the two distributions side by side, so the reader can see exactly which slices are realized in the masculine and in the feminine.[13]

The role that *-a* plays is complicated by some masculines having *-a* as a suffix, e.g. *hrdin-a* ('hero.Masc'). In this analysis, *-a* is the carrier of the Fem feature which is associated only with feminine nouns.[14] When masculine nouns end in *-a*, I assume a distinct, homophonous LI lacking Fem in their fseq as the simplest account.[15] The reasons for getting a different feminine form in *žáb-a* than in *učitel-k-a* might be multiple, but this nanosyntactic analysis pinpoints the specific part of the syntactic structure that influences why we get *-a* and not *-k-a* or *-ič-k-a*. For this part of the derivation, the stem morpheme *žab*, and the new suffixes *-a* and *-ák* will be necessary. These morphemes are represented in the lexicalization table (70).

(70) Lexicalization table for *žab-ák* and *žáb-a*

ΔNP	Reference	Class	Masculine	Feminine	Number	Case
žab	ák				...	
žab				a		

From the lexicalization table of *žab-ák* and *žáb-a* in (70) it can be deduced that the morpheme *žab* is followed by *-ák*, *žab* only takes care of NP, but the stem morpheme plays a much larger lexicalization role when Fem is introduced into the picture with *-a*.

I propose the functional morpheme *-ák* or its phonological variant *-ak*[16] as a standalone morpheme. This morpheme derives denominal nouns for conceptual or spatial relation (*škol-ák*, 'schoolgoer.Masc'), animal names (*plameň-ák*, 'flamingo.Masc'), derived versions of feminines (the aforementioned *žab-ák*), and many others. The multiplicity of meanings suggests that the suffix *-ák* is linked to multiple concepts in the lexicon. Based on that, it is closer to a stem (like *kmotr*) than to a suffix (like *-k*). Because of this I propose that *-ák*

[12] Since this paper focuses on laying the foundation for the analysis, it skips over the issue of what role the inflectional endings of different cases play.

[13] On a more general level, it is becoming clear that how LIs work and how noun derivation works are two different issues. Even though one noun, e.g. učitel-k-a, is derived from a different one, e.g. učitel, it is not true that one LI is derived from another (e.g. -ák is not derived from -a or -k). Nouns and endings are the things observed by speakers on the surface, while LIs are the deep structural representations.

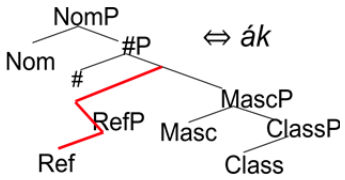
[14] The LI for a (repeated in (73)) contains Fem at its foot which means it is not capable of lexicalizing any structure which does not likewise contain Fem. Remember that a LI only lexicalizes a structure of which it is a superset.

[15] A CLB-based single-entry analysis has been considered (see footnote 10), but I leave its viability open.

[16] Length alternation is a common phonological effect in Czech nouns, even though it most often applies to words ending with the much more frequent *-ík* (e.g. *děln-ík* / *děln-ice*, *domovn-ík* / *domovn-ice*).

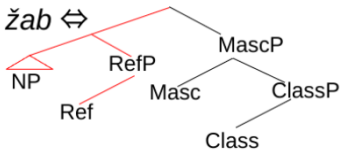
contains a CLB, similarly to some of the presented stems. The LI for *-ák* is shown in (71). The foot of the tree lacks the complex NP node reserved for stems; therefore, it is a suffix morpheme.

(71) LI for *-ák*

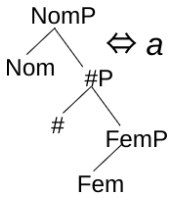


The Ref feature appears on the CLB of *-ák* because this suffix creates nouns that can stand alone as referring expressions (*žab-ák*, *škol-ák*), rather than only extending the meaning of the base. *Žab* and *-a* are in (72) and (73). The red branches in *žab*'s and *-ák*'s trees are CLBs. It is thanks to the CLB that *žab* can play a small role in *žab-ák* and a bigger role in *žáb-a*. The reader can see a left branch showing up in *-ák* morpheme as well in (71).

(72) LI for *žab*

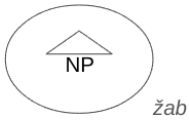


(73) LIs for *-a*

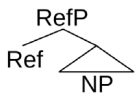


Let us derive *žab-ák* first. The algorithm dictates the first step: Merge NP the first node. It can be lexicalized by *žab* without an issue, (75), because *žab* is a superset of this node and I want to talk about a [FROG]. This is followed by merging Ref on top in (76).

(75) Lexicalize as *žab*

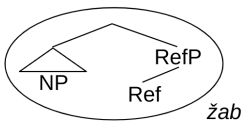


(76) Merge Ref



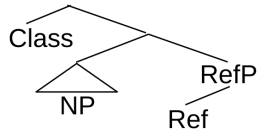
The (76) structure cannot be lexicalized by *žab* in (72) because it is not (72)'s subconstituent. That is because the LI for *žab* contains a CLB. The NP is evacuated above Ref. The closest constituent (NP) of the merged feature is moved up, resulting in (77), which is lexicalized by *žab*. The visualizations of the evacuation will be skipped for brevity.

(77) Lexicalize as *žab*



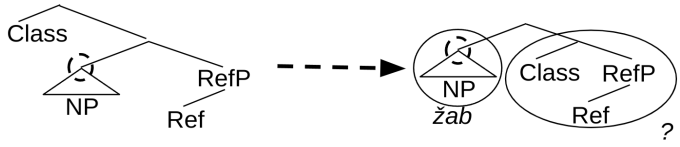
The derivation continues by merging Class (78) and finding out that lexicalization straight up is impossible.

(78) Merge Class



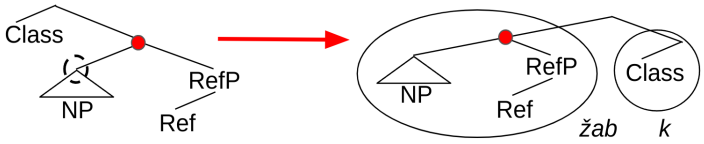
Rescue movement of the closest constituent (CLN-RC) does not yield lexicalization either, as can be seen in (79). Because of how *-ák* and *žab* are shaped, *-ák* cannot lexicalize the right branch of the rightmost tree in (79) (symbolized by a question mark).

(79) CLN-RC evacuation



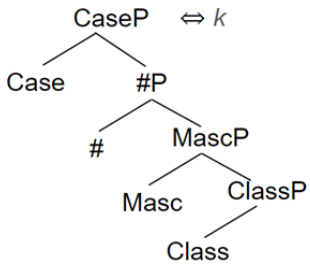
A higher node therefore needs to be pied-piped. That node is highlighted by a red ball, and the result of the pied-piping is visible in (80).

(80) Pied-piping of a node one notch higher



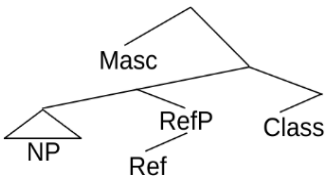
The lexical tree for *-k*, proposed originally for *učitel-k-a*, will help here, as it can lexicalize the rightmost branch of (80). See picture (81) for the repetition of the LI.

(81) LI for *k*



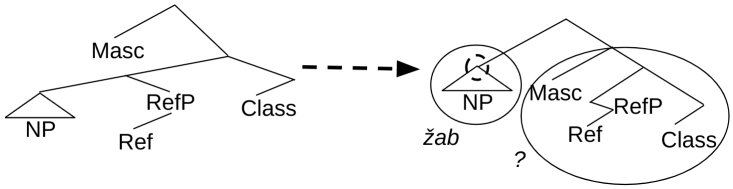
Even though the result (80) at this point is *žab-k*, this is an intermediate result, and it does not predict what the final structure is going to be. Masc is merged in (82), with no matches found in the lexicon and lexicalization being impossible right away.

(82) Merge Masc

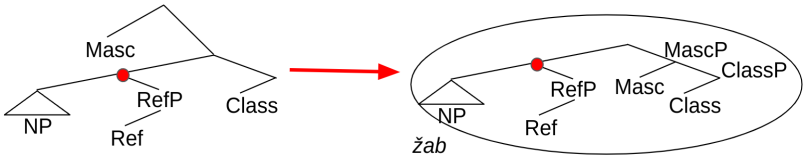


Just like in the previous merge step, CLN-RC evacuation does not help (83), but pied-piping a higher node brings us to the structure (84) which can be lexicalized just by the single morpheme of *žab*, taking *-k* out of the equation for now.

(83) CLN-RC evacuation

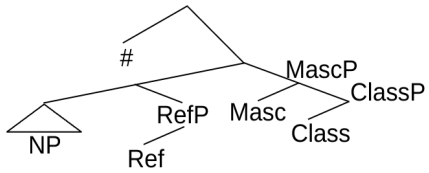


(84) Pied-pipe a higher node



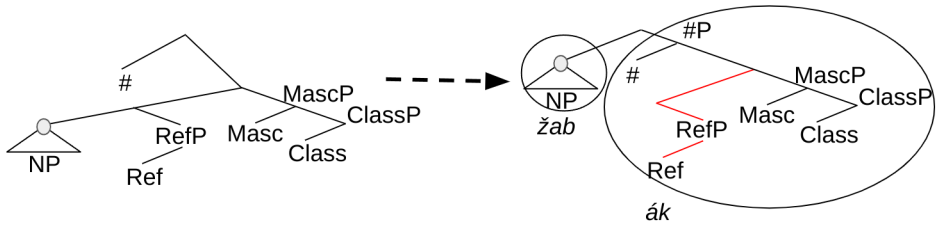
Since this is a derivation of a masculine noun, Number (#) is merged right after the Masc feature. This moment is crucial. When # is merged in (85), the total cannot be lexicalized, and in (86) only CLN-RC evacuation can result in the right branch perfectly matching the hypothesized LI for *-ák*.

(85) Merge #



The node that undergoes the rescue operation is highlighted by the white ball in (86).

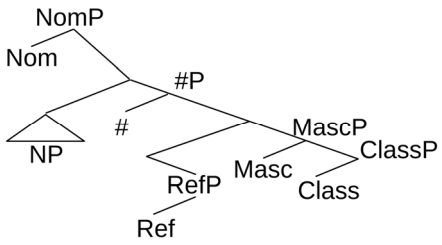
(86) Evacuate CLN-RC



Lexicalization is successful, with only one more feature to go. In the process of the most recent movement, the Ref feature has been stranded by NP, but since it arrived at its current place in the tree as a result of evacuation, the shape of the branch is left crooked to signal that Ref is, in fact, still hanging on a complex left branch. By designing the LIs to already contain movement in them, just like the LI for *-ák* does, their usefulness increases. Such a LI can lexicalize just the feature(s) on its CLB, just the features on its right branch, or the constituent containing both.

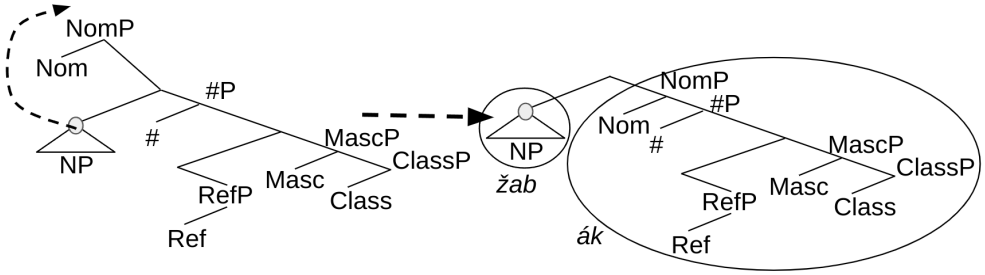
The last merge in the fseq line is Nom in (87).

(87) Merge Nom



No possible lexicalization of the tree in (87) warrants the evacuation of the CLN-RC in (88), highlighted by the white ball.

(88) *Žab-ák* is lexicalized



Nominative is the last feature present in the form of the masculine noun *žab-ák*, ergo the whole derivation has been successfully concluded. Over the course of the derivation, the first three steps of the algorithm were used and the practicality of Complex Left Branches (CLBs) was explored.

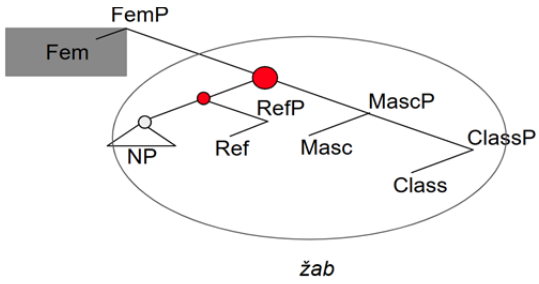
The feminine noun *žáb-a* is the next in line. Table (89) repeats the relevant lexicalization table. We can see that *žab* lexicalizes a larger part of the sequence than it did in *žab-ák* ('frog.M'), and right after the Feminine feature is merged, the morpheme *-a* joins the fray.

(89) Lexicalization table for *žab-ák* and *žáb-a*

ΔNP	Reference	Class	Masculine	Feminine	Number	Case
žab	ák				...	
žab				a		

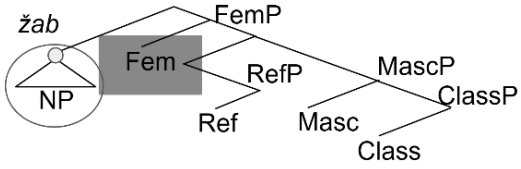
The feminine form *žáb-a* uses the same features as *žab-ák*, with the only difference being the Fem feature. Because of the shape of the LI of *žab*, with the NP standing on the left branch higher up, this LI can lexicalize either the NP on its own or the whole sequence NP < Ref < Class < Masc. For that reason, the derivation of *žáb-a* is identical to *žab-ák*'s in steps (74) – (84). When Fem is merged afterwards, its presence allows for a matching of the tree with the *-a* ending. (90) visualizes Fem merged in syntax. The tree requires evacuation movement to achieve lexicalization.

(90) When the derivation of *žáb-a* ('frog.Fem') deviates from the derivation of *žab-ák* ('frog.Masc') – Merging the Fem feature

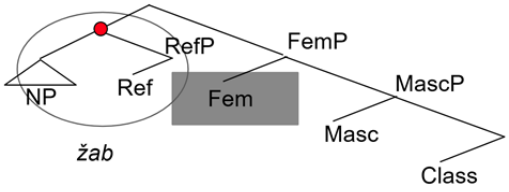


At that moment, the typical rescue operations (CLN-RC evacuation in (91) – the white node – pied-piping in (92) – the red node – try to save the derivation, yet they do not produce viable results.

(91) CLN-RC evacuation of NP

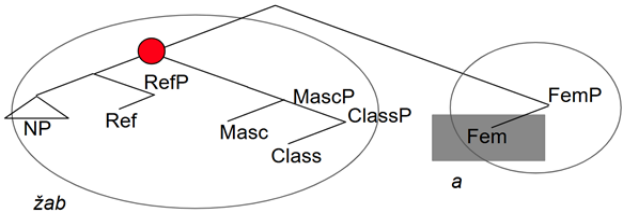


(92) Pied-piping of a higher node



Finally, when an even higher node is pied-piped in (93), the tree finds a match amongst the LIs, and lexicalization is successfully reached.

(93) *Žab-a* is lexicalized



This concludes the step-by-step derivations of *žáb-a* ('frog.F') and *žab-ák* ('frog.M'). They work because the LIs of *žab* and *-ák* were designed to contain Complex Left Branches (CLBs). CLBs are a tool of syntactic geometry to increase the usefulness of LIs without postulating new rules for syntactic derivation. They follow from the logic of the algorithm and its evacuation movements.

5. Conclusion and discussion

This study has demonstrated how nanosyntax can be applied to the morphology of Czech nouns, specifically in modeling gender alternations through the interaction of stems and suffixes, including cases where multiple suffixes appear in sequence. By utilizing the functional sequence (fseq), matching, and the lexicalization algorithm, the analysis accounts for patterns such as *učitel* → *učitel-k-a* and *žáb-a* → *žab-ák* without appealing to lexical defaults or semantic-pragmatic labels.

This study illustrated it on Czech, but nanosyntax can systematically model gender morphology across languages. The framework naturally extends to defining structural patterns and variations, offering a precise approach to lexicalization within syntax. Future research could apply this model to further languages, to languages with more or fewer genders, particularly those with systematic gender alternations or derivational processes tied to gender, even when on the surface there do not seem to be strong patterns. By tracing the step-by-step operation of the algorithm, this analysis clarifies the role of nanosyntax in deriving complex morphological patterns.

Finally, the analysis of multifunctional endings in Czech (here applied on *-k* and hinted at for *-a*) shows that nanosyntax can capture morphological variability without extra rules. Russian exhibits comparable reuse of exponents: *-ka* can form both feminines (*student* → *student-ka*) and diminutives (*kniga* → *knizh-ka*). Likewise, *-a* is a common feminine ending but some masculines end in *-a*, as well (*dyady-a* 'uncle.M', *yunosh-a* 'youth.M') – parallel to Czech *hrdin-a*. Romanian similarly reuses the same vowel endings across gender/number: *-u* realizes both M.Sg and N.Sg, *-a* realizes F.Sg, and *-e* realizes both N.Pl and F.Pl (e.g., *le-u* 'lion.M.Sg', *teatr-u* 'theater.N.Sg', *teatr-e* 'theater.N.Pl', *fete* 'girl.F.Pl'). This suggests that the Czech patterns described here are not isolated and the nanosyntactic analysis would be warranted.

References

Acquaviva, P. 2008. *Lexical plurals: A morphosemantic approach*. Oxford Studies in Theoretical Linguistics. Oxford and New York: Oxford University Press.

Baggio, P. 2022. Getting rid of MASC: The encoding of gender in Italian. *LingBuzz*. August 2022. https://lingbuzz.net/lingbuzz/006753.

Blix, H. 2022. Interface legibility and nominal classification: A nanosyntactic account of Kipsigis singulatives. *Glossa: A Journal of General Linguistics* 7. https://doi.org/10.16995/glossa.5825.

Bobaljik, J. D., and C. Levart Zocca. 2011. Gender markedness: The anatomy of a counter-example. *Morphology* 21(2): 141–166. https://doi.org/10.1007/s11525-010-9156-3.

Caha, P., K. De Clercq, and G. Vanden Wyngaerd. 2019. The fine structure of the comparative. *Studia Linguistica* 73(3): 470–521. https://doi.org/10.1111/stul.12107.

Caha, P. 2009. *The nanosyntax of case*. Ph.D. diss., University of Tromsø. https://munin.uit.no/bitstream/handle/10037/2203/thesis.pdf.

Caha, P., and L. Taraldsen Medová. 2022. On the feminine singular declension of the Russian demonstrative. https://www.muni.cz/vyzkum/publikace/2222757.

Corbett, G. G. 1991. *Gender*. Cambridge Textbooks in Linguistics. Cambridge: Cambridge University Press. https://doi.org/10.1017/CBO9781139166119.

Cortiula, M. 2023. The nanosyntax of Friulian verbs. *LingBuzz*. https://ling.auf.net/lingbuzz/007617.

De Clercq, K., P. Caha, M. Starke, and G. Vanden Wyngaerd. 2025. Nanosyntax: State of the art and recent developments. In *Nanosyntax and the lexicalization algorithm*. Oxford: Oxford University Press. https://lingbuzz.net/lingbuzz/007744.

Dokulil, M. 1986. Tvoření slov. In *Mluvnice češtiny*, vol. 1. Praha: Academia.

Fiebig, N. 2023. Asserted and presupposed gender on animal nouns. Paper presented at NELS 53, Göttingen, January 12, 2023. https://drive.google.com/file/d/1XDufsIvnqfg6qjzdE3fNwin9TX6B-Nos/view.

Halle, M., and A. Marantz. 1993. Distributed morphology and the pieces of inflection. In K. Hale and S. J. Keyser (eds.), *The view from Building 20: Essays in linguistics in honor of Sylvain Bromberger*, 111–176. Cambridge, MA: MIT Press.

Harley, H., and R. Noyer. 1999. State-of-the-article: Distributed morphology. *Glot International* 4(4): 3–9.

Harley, H., and E. Ritter. 2002. Person and number in pronouns: A feature-geometric analysis. *Language* 78(3): 482–526. https://muse.jhu.edu/pub/24/article/19419.

Jakobson, R. 1984. Structure of the Russian verb. In M. Halle and L. R. Waugh (eds.), *Russian and Slavic grammar*, 1–14. Berlin: De Gruyter Mouton. https://doi.org/10.1515/9783110822885.1.

Janků, L. 2022. *The nanosyntax of Czech nominal declensions*. Ph.D. diss., LingBuzz. https://lingbuzz.net/lingbuzz/006705.

Jurafsky, D. 1996. Universal tendencies in the semantics of the diminutive. *Language* 72(3): 533–578. https://doi.org/10.2307/416278.

Kasenov, D. 2023. Pseudo-ABA patterns and the generative power of nanosyntax. *LingBuzz*. September 2023. https://ling.auf.net/lingbuzz/007590.

Kramer, R. 2015. *The morphosyntax of gender*. Oxford: Oxford University Press. https://doi.org/10.1093/acprof:oso/9780199679935.001.0001.

Scheer, T. 2005. Slavic vowel-zero alternations and government phonology: Two approaches, one solution. In S. Franks, F. Y. Gladney, and M. Tasseva-Kurktchieva (eds.), *Annual Workshop on Formal Approaches to Slavic Linguistics: The South Carolina Meeting*, 13: 300–311. Michigan Slavic Materials 50. Ann Arbor: Michigan Slavic Publications. http://tscheer.free.fr/papers/Scheer_05_Slavic_v-zero&GP_FASL13.pdf.

Spathas, G., and Y. Sudo. 2020. Gender on animal nouns in Greek. *Catalan Journal of Linguistics* 19: 25. https://doi.org/10.5565/rev/catjl.297.

Starke, M. 2009. Nanosyntax: A short primer to a new approach to language. *Nordlyd* 36(1): 1–6. https://doi.org/10.7557/12.213.

Starke, M. 2018. Complex left branches, spellout, and prefixes. In L. Baunaz, L. Haegeman, K. De Clercq, and E. Lander (eds.), *Exploring nanosyntax*, 239–249. New York: Oxford University Press. https://doi.org/10.1093/oso/9780190876746.003.0009.

Štícha, F. (ed.). 2018. *Velká akademická gramatika spisovné češtiny*. Vol. 1. Praha: Academia.

Cite this article as:

Boucník, D. (2025). The Teacher and the Frog: Unveiling the Morphosyntax of Gender Shifts in Czech with Nanosyntax. *LingBaW. Linguistics Beyond and Within, 11*, 23–51.