

Mismatches in free relatives - grafting nanosyntactic trees

Fenna Bergsma

Goethe-Universität Frankfurt

`bergsma@em.uni-frankfurt.de`

July 15, 2019

Abstract

German free relative constructions allow for case requirement mismatches under two types of circumstances. The first is when the case required in the embedded clause is more complex ($\text{NOM} < \text{ACC} < \text{GEN} < \text{DAT}$) than the case required in the main clause, and the relative pronoun takes the form of the embedded clause case. The second type of circumstance is when the form that corresponds to the two required cases is syncretic. I propose an analysis that combines Caha's (2009) case hierarchy in nanosyntax with Van Riemsdijk's (2006a) concept of grafting. By placing case features as separate heads in the syntax, a less complex case can be grafted into a different clause, explaining the first type of circumstance. The second type makes reference to the fact that syncretic forms are inserted via the same lexical entry (Superset Principle). A cross-linguistic comparison shows that it is language-specific whether a more complex case requirement in the main or embedded clause causes non-matching non-syncretic free relatives to be grammatical. For all languages it holds that the relative pronoun appears in the most complex case required, which provides additional evidence for case being complex and more complex cases being able to license less complex cases.

1 Introduction

A free relative construction is a type of relative clause that occurs without an antecedent. In the headed relative clause in (1a), the antecedent of the rela-

tive pronoun *den* ‘who.ACC’ is the homophonous demonstrative pronoun *den* ‘the.ACC’. In (1b), the free relative counterpart of (1a), the relative pronoun *wen* ‘who.ACC’ does not have an antecedent.^{1,2,3}

- (1) a. Ich mag den, den ich eingeladen habe.
 I like_{acc} the.ACC who.ACC I invited_{acc} have
 ‘I like the one, who I have invited.’
 b. Vogel (2001: 344)
 Ich lade ein, wen auch Maria mag.
 I invite_{acc} who.ACC also Maria likes_{acc}
 ‘I invite whoever Maria also likes.’

(1b) is an example of a matching free relative. Both *lade ein* ‘invite’ in the main clause and *mag* ‘likes’ in the embedded clause require their object to be in accusative. *Wen* ‘who.ACC’ satisfies both these accusative case requirements.⁴

In German free relatives, the case requirements in both clauses do not need to match: mismatching free relatives are permitted under particular circumstances. The first type of circumstance is when the case required in the embedded clause is more complex (or more oblique) than the case required in the main clause (according to the complexity scale NOM < ACC < GEN < DAT < ...). Additionally, the relative pronoun has to surface in the more complex case (cf. Pittner 1991; 1995; Vogel 2001; Grosu 2003). I illustrate this in (2) and (3) below. The examples in (2a) and (2b) only differ in the relative pronoun that is used. The predicate *vertraut* ‘trusts’ in the embedded clause requires its object to be in dative, and *lade ein* ‘invite’ in the main clause requires it to be in accusative. German has two distinct forms for accusative and dative to denote *who*: *wen* ‘who.ACC’ and *wem* ‘who.DAT’ respectively. The sentence is grammatical as long as the (more complex) dative relative pronoun

¹ All examples in the paper are from German, unless indicated differently.

² The annotation of case on the verb indicates the case that the verb requires the free relative pronoun to be in, e.g. *acc* on *like* in (1b) indicates that *like* requires its object to be in accusative case.

³ The German data and the description of the generalizations in the German data in this paper originate in Vogel (2001).

⁴ Matching free relatives in German do not only exist with accusative - accusative pairs, but also with double nominative, genitive and dative case requirements.

wem ‘who.DAT’ is used, as shown in (2a). (2b) shows that the construction is ungrammatical if the accusative relative pronoun *wen* ‘who.ACC’ is used.^{5,6}

(2) Vogel (2001: 344)

- a. Ich lade ein, *wem* auch Maria vertraut.
 I invite_{acc} who.DAT also Maria trusts_{dat}
 ‘I invite whoever Maria also trusts.’
- b. * Ich lade ein, *wen* auch Maria vertraut.
 I invite_{acc} who.ACC also Maria trusts_{dat}
 ‘I invite whoever Maria also trusts.’

In (3) the case requirements are reversed: the embedded clause predicate *mag* ‘likes’ requires its object to be in accusative and the one in the main clause *vertraue* ‘trust’ requires it to be in dative. This means that the case required in the embedded clause is less complex than the case required in the main clause. The sentence in (3) is ungrammatical, independent of which relative pronoun (*wen* ‘who.ACC’ or *wem* ‘who.DAT’) is used.

(3) Vogel (2001: 345)

- * Ich vertraue, *wen/* *wem* auch Maria mag.
 I trust_{dat} who.ACC/ who.DAT also Maria likes_{acc}
 ‘I trust whoever Maria also likes.’

The pattern observed here does not only appear with accusative - dative case combinations. As long as the embedded clause requires a more complex case than the main clause, and the more complex relative pronoun is used, a mismatching free relative is grammatical. If the embedded clause requires a less complex case than the main clause, the result is ungrammatical. For example, if the embedded clause requires a genitive case, the main clause can have either an accusative or a genitive case requirement. What follows is that German does not have mismatching free relatives with the embedded clause requiring a nominative, because nominative is the least complex case. For the relevant examples, see Appendix 1. In this paper, I illustrate my proposal using the accusative - dative case examples, but the analysis carries over to the other case combinations.

⁵ In Polish (Citko 2013) (and in some varieties of German (Vogel 2001)), mismatches with distinct lexical entries are never permitted, and (2a) is ungrammatical. Presumably these speakers require a stricter form of matching. I return to this point in Section 5.2.

⁶ Other languages that show (different types of) hierarchy effects are Gothic (Harbert 1982) and Modern Greek (Daskalaki 2011), to which I return to in Section 5.2.

The second type of circumstance in which mismatching free relatives are permitted is when the phonological form that corresponds to two cases is syncretic, and the syncretism resolves the mismatch (cf. Groos & van Riemsdijk 1981; Zaenen & Karttunen 1984; Dyta 1984; Pullum & Zwicky 1986; Ingria 1990; Dalrymple & Kaplan 2000; Sag 2003). In (4), the case required in the embedded clause is less complex than the case required in the main clause. The example in (4) is grammatical, although it also has the less complex case required in the embedded clause, just like the ungrammatical example in (3). The crucial difference between (3) and (4) is that the latter has a syncretic form that corresponds to the two cases, and the former does not. *Gefällt* ‘pleases’ in the embedded clause requires a nominative subject, and *erzähle* ‘tell’ requires its object to be in accusative. *Was* ‘what.NOM/ACC’ is syncretic between both nominative and accusative; it has the same phonological form in both cases. Therefore, it is able to satisfy both the nominative and accusative case requirements, and the sentence is grammatical.

- (4) Vogel (2001: 344)
 Ich erzähle, was immer mir gefällt.
 I tell_{acc} what.NOM/ACC ever me pleases_{nom}
 ‘I tell whatever pleases me.’

For completeness, I give an example of a free relative with a syncretic form and the less complex case required in the embedded clause, which is also grammatical. In (5), the embedded clause predicate *weiß* ‘know’ requires its object to be in accusative, and *macht* ‘makes’ requires a nominative subject. *Was* ‘what.NOM/ACC’ is able to satisfy both the nominative and accusative case requirements.

- (5) Vogel (2001: 363)
 Was ich nicht weiß, macht mich nicht heiß.
 what.NOM/ACC I not know_{acc} makes_{nom} me not hot
 ‘What I don’t know doesn’t excite me.’

Unlike the German-specific effect in (2a), resolving a case mismatch by syncretism is not specific to German. (6) gives three examples from different Slavic languages that show the same effect. In (6a), the Russian *kogo* ‘who.ACC/GEN’ satisfies the accusative case requirement from *iskal* ‘sought’ and the genitive case requirement from *bylo* ‘was’. (6b) shows a conversational Bulgarian example in which *kojto* who.NOM/ACC is syncretic between nominative and accusative, and the accusative case requirement of *celuna* ‘kiss’ and the nominative one of *dojde* ‘comes’ are satisfied. The Polish *kogokolwiek* ‘who-

ever.ACC/GEN’ is both accusative and genitive, and satisfies the genitive case requirement from *unika* ‘avoids’ and the accusative case requirement of *obrazil* ‘offended’.

- (6) a. *Russian*, Levy & Pollard (2002: 222)
 Kogo ja iskal, ne bylo doma.
 who.ACC/GEN I sought_{acc} not was_{gen} home
 ‘Who I was looking for wasn’t at home.’
 b. *conversational Bulgarian*, Izvorski (1997: 279)
 Šte celuna kojto dojde prâv.
 will kiss_{acc} who.NOM/ACC comes_{nom} first
 ‘I will kiss whoever comes first.’
 c. *Polish*, Himmelreich (2017: 17) after Citko (2013)
 Jan unika kogokolwiek wczoraj obrazil
 Jan avoids_{gen} whoever.ACC/GEN yesterday offended_{acc}
 ‘Jan avoided whoever he offended yesterday.’

A summary of the patterns discussed in this section is shown in Table 1. As long as the case required in the embedded clause is more complex, and the form corresponding to this more complex case is inserted, the free relative is grammatical (the left column). If the case required in the embedded clause is less complex, the phonological form starts to play a role. If German has two different forms for the different cases, the free relative is ungrammatical (see upper-right cell). It can be saved, however, if there is a single syncretic form that corresponds to the two required cases (see lower-right cell).

Table 1: Grammaticality pattern of German mismatching free relatives.

	case in embedded clause	
	more complex	less complex
distinct forms	✓	×
syncretism	✓	✓

In lexicalist frameworks, the effect that a more complex case is required in the embedded clause was generally claimed to originate from the stipulation that more marked cases are able to license less marked cases (cf. Pittner 1995). The power of syncretism was mostly explained by underspecified lexical entries: the presyntactic lexicon contains phonological forms that are specified for more than one case (cf. Ingria 1990). Lately, theories of late insertion, such as Distributed Morphology (Halle & Marantz 1993), have become more prominent. At first sight, the fact that syncretism resolves mismatches is prob-

lematic for such approaches. If there is no presyntactic lexicon that contains phonology, it is unclear how phonological forms can influence grammaticality. Nevertheless, in recent years, the resolution of case mismatches has been accounted for using different theories within late insertion. [Asarina \(2011\)](#) provides an analysis that combines underspecified lexical entries in Distributed Morphology with multidominance. Her analysis of Russian can be extended to the syncretic cases in German, but does not capture the effect of requiring a more complex case in the embedded clause (which does not exist in Russian). [Himmelreich \(2017\)](#) proposes an agree-based account, which captures both the syncretic cases and the more and less complex cases. However, she needs two different mechanisms to derive the effects. In this paper, I provide a unified analysis that accounts for both types of circumstances in which mismatching free relatives are permitted, crucially drawing on case containment ([Caha 2009](#)). This paper provides further evidence that case is internally complex, a more complex case can license a less complex case, and that syntactic structure can be shared between clauses.

The paper is organized as follows. Section 2 presents the internal structure I assume for the free relative pronoun. I introduce the nanosyntactic ([Starke 2009](#)) tools for the analysis, the projections within the relative pronoun, and I show how a relative pronoun is formed in a derivation. One of the projections within the relative pronoun is the KP (case phrase), which contains the case hierarchy ([Caha 2009](#)) which is crucial for the analysis in Section 4. Section 3 discusses the external syntax of a free relative construction. A free relative pronoun is associated with two syntactic positions, which can be analyzed by a grafting operation ([Van Riemsdijk 2006a](#)): embedded features in one structure can be remerged into a different structure. In Section 4, I present the analysis as a three step derivation. In the first step, the embedded clause predicate combines with the required case node on the relative pronoun. In the second step, the relative pronoun moves to the left edge of the clause. In the third step, the main clause predicate combines with the case node it requires. If this case node is not available, the highest case node combines with more complex case nodes first, until the main clause predicate can be merged. German is subject to the restriction that only case features that have the same spellout can be added. This restriction makes a crucial distinction between syncretic and non-syncretic mismatching free relatives. Section 5 extends the proposal to three other languages besides German by identifying two restrictions which can be present or absent in a language. Section 6 concludes.

2 Internal syntax of free relatives

In this section I make explicit what I assume to be the internal syntax of free relative pronouns. I argue that free relative pronouns consist of a combination of the element *w*- and an element which expresses different mass type and case distinctions. First, I introduce the basic concepts from nanosyntax that allow for a discussion, and I discuss the feature make-up of the different projections. Towards the end of the section, I show how the lexical entries that I propose correspond to free relative pronouns as they surface.

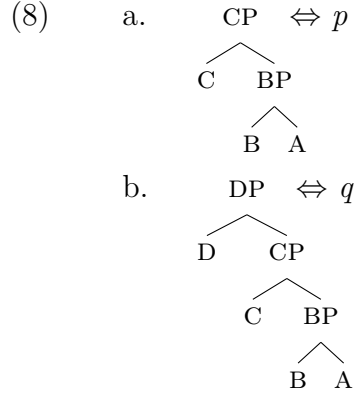
2.1 Nanosyntax

Nanosyntax (Starke 2009; Caha 2009; Baunaz & Lander 2018b) is a generative approach in which syntactic features correspond to their own terminal nodes and are merged into syntactic trees. Nanosyntax has a postsyntactic lexicon that consists of lexical trees which are linked to phonological and conceptual representations. Spellout in nanosyntax is phrasal: instead of spelling out individual terminal nodes, lexical entries target phrasal (non-terminal) nodes in the syntactic tree. Spellout is also cyclic, and each successful spellout overrides all previous spellouts that it dominates (Principle of Cyclic Override Starke 2009: 4). After each instance of merge, the syntactic tree is spelled out. No parts of the syntactic structure can remain unlexicalized (known as Cyclic Exhaustive Lexicalization, Fábregas 2007). Spellout only takes place under strict constituenthood, i.e. only constituents can be targeted for spellout. The two principles in (7) govern lexical insertion.

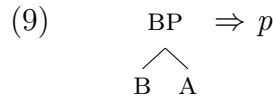
- (7) a. **Superset Principle:** a lexically stored tree matches a syntactic node iff the lexically stored tree contains the syntactic node (Starke 2009: 3)
- b. **Elsewhere Condition:** if several lexical items match the root node, the candidate with the least unused nodes wins (Starke 2009: 4)

The Superset Principle in (7a) ensures that a lexical tree matches a syntactic tree if the lexical tree is a superset (proper or not) of the syntactic tree. This means that a lexical tree can match a syntactic tree if the former contains features that the latter does not, but not vice versa. The Elsewhere Condition as in (7b) makes sure that, when more than one lexical tree can lexicalize the same syntactic tree (by the Superset Principle), the lexical tree with the least amount of superfluous material is chosen.

I illustrate with abstract examples how these principles together select a single lexical entry. In (9) to (11), the structure $[\text{DP}[\text{CP}[\text{BP}]]]$ is created in three derivational steps with lexicalizations after each step. (8) gives two abstract lexical entries.⁷ The lexical tree $[\text{CP}[\text{BP}]]$ corresponds to the phonological structure p , and the lexical tree $[\text{DP}[\text{CP}[\text{BP}]]]$ corresponds to the phonological structure q .

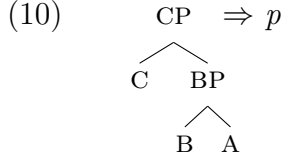


In the first step of the syntactic derivation, A is merged with B, creating BP in (9). Both the lexical trees $[\text{CP}[\text{BP}]]$ in (8a) and $[\text{DP}[\text{CP}[\text{BP}]]]$ in (8b) are a superset of BP (Superset Principle). $[\text{CP}[\text{BP}]]$ has less superfluous material than $[\text{DP}[\text{CP}[\text{BP}]]]$ (Elsewhere Condition), and therefore the phonological form p is inserted.

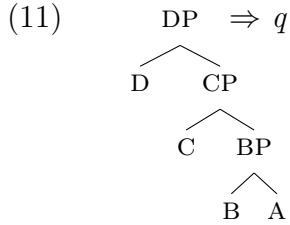


Next, BP merges with C, and $[\text{CP}[\text{BP}]]$ in (10) is created. Here again, both lexical trees $[\text{CP}[\text{BP}]]$ in (8a) and $[\text{DP}[\text{CP}[\text{BP}]]]$ in (8b) are a (proper) superset of the syntactic tree (Superset Principle). As $[\text{CP}[\text{BP}]]$ has no superfluous material but $[\text{DP}[\text{CP}[\text{BP}]]]$ does, the lexical entry (8a) matches with the syntactic structure via the Elsewhere Condition. In this example, $[\text{CP}[\text{BP}]]$ and BP are syncretic: they correspond to the same lexical entry and have the same phonological form p .

⁷ Throughout the paper, \Leftrightarrow indicates the pairing between a lexical tree and a phonological form in a lexical entry, and \Rightarrow indicates how a node in the syntactic structure is spelled out.



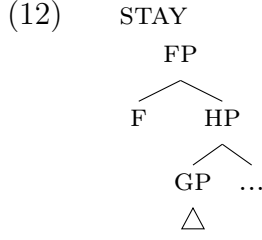
Finally, $[\text{CP}[\text{BP}]]$ merges with D , as shown in (11). The lexical tree $[\text{CP}[\text{BP}]]$ in (8a) is no longer a superset of the syntactic tree, as it does not contain D (Superset Principle), so it is not a candidate for a match. $[\text{DP}[\text{CP}[\text{BP}]]]$ in (8b), however, still contains all features. The earlier spellout of (8a) is overridden by (8b) and the phonological structure q is inserted (Principle of Cyclic Override).



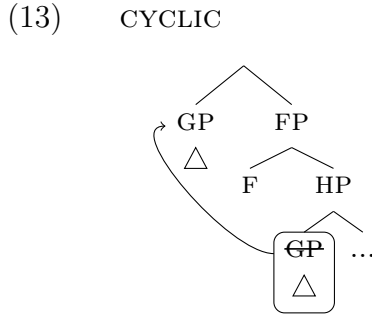
The syntactic structures given above could all be spelled out after a new feature has been merged. However, this is not always the case. If there is no lexical tree that matches the syntactic tree after a new feature is merged, spellout driven movement takes place. Spellout-driven movement is governed by the so-called spellout algorithm. There are several steps in the algorithm, of which I only discuss the first two in this paper (see for additional steps not required for this paper Baunaz & Lander 2018b; Starke 2018; Caha et al. 2018; De Clercq 2019).

The first step in this algorithm has already been used to illustrate the principles governing lexical insertion in (7). This step is called **STAY**: the syntactic structure stays as it is after the merge of a new feature, and the lexicon is checked for a lexical entry. If there is no lexical entry available that matches, the first movement possibility takes place. This is **CYCLIC**: the leftmost daughter of the sister of the last added feature is moved to the left of the last added feature. The lexicon is consulted again after this movement.

I illustrate the spellout algorithm with an abstract example in (12) to (13). I start with the first step: **STAY**. The structure stays at it is, and the lexicon is consulted for a lexical entry with the syntactic structure, shown in (12).



If there is no lexical entry available for this configuration, then GP, the leftmost daughter of the sister of F, moves to the left of F (CYCLIC). The lexicon is checked for an entry that contains the syntactic structure FP. This FP contains HP, but it does not have a GP in it, shown in (13).



This kind of movement does not leave any traces (Baunaz & Lander 2018b; Starke 2018). In the next section, I discuss the internal feature make-up of the free relative pronouns. In Section 2.3, I apply the tools introduced in this section to show how the free relative pronouns are built.

2.2 The feature inventory

In this section I make my assumptions for the internal structure of German free relative pronouns explicit. An overview of the pronouns is given in Table 2.⁸

I follow a large body of literature that argues that pronouns have complex internal structure and that they correspond to phrases (cf. Cardinaletti 1994; Weerman & Evers-Vermeul 2002; Boef 2013; Hachem 2015). Like Hachem (2015) and Baunaz & Lander (2018a), I decompose relative pronouns into smaller units. Looking at the paradigm, there is reason to assume that pronouns consist of a form *w-* and a form that expresses gender and case.

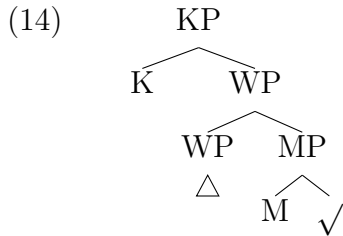
⁸ For discussion on the incompleteness of the paradigm of *was*, see Hachem (2015: 162) and references therein.

Table 2: German free relative pronouns.

	‘who’	‘what’
NOM	wer	was
ACC	wen	was
GEN	wessen	-
DAT	wem	-

Moreover, the latter form also appears on other elements that show gender and case distinctions, such as definite determiners and demonstratives (compare *d-er* ‘the-M’ and *w-er* ‘what-M’, *d-as* ‘the-N’ and *w-as* ‘what-N’). Taking this formal identity seriously (like cf. [Leu 2008](#); [Hachem 2015](#)), I assume that they are the same morphosyntactic object.

In my analysis, the internal structure of the free relative pronoun looks as in (14). Following [Hachem \(2015\)](#), I assume that the WP corresponds to the *w*-element. The case phrase (KP) contains the case features and will be split up further following [Caha 2009](#). The mass phrase (MP) contains different types of mass (that correspond to the distinctions better known as different grammatical genders) ([Hachem 2015](#)). The MP and KP together correspond to a single phonological element. In the structure in (14), the KP and the MP do not form a constituent that can match a lexical tree. Spellout-driven movement (of the WP into the specifier of the KP) will ensure that KP and MP form a constituent, such that they can be spelled out together.



In the next sections, I discuss each of the projections in (14) in turn.

2.2.1 Case (KP)

[Caha \(2009\)](#) (building on [Blake 1994](#)) proposes that case features are organized in the functional sequence (fseq) given in (15).

- (15) NOM — ACC — GEN — DAT — INS — COM

This hierarchy is not specific to nanosyntax; it is motivated typologically (Blake 1994), and a version of it has been incorporated in work on Distributed Morphology (Smith et al. 2018).⁹

The evidence that *Caha* provides for the hierarchy comes from amongst others syncretisms in morphological case suffixes and case compounding. He proposes that case syncretism only targets contiguous regions (nodes that are adjacent) in the case hierarchy. *Caha* investigates possible syncretisms in the Russian noun system. Of all possibilities, 15 syncretisms are predicted to exist by the case hierarchy (because they target contiguous regions) and 42 are not (because they do not). 8 of the 15 predicted patterns occur in Russian. Not all 15 contiguous patterns arise because syncretism mostly covers only two contiguous elements and not ‘long’ stretches. Only one of the 42 non-contiguous patterns occurs, which can be explained as accidental homophony (Caha 2009: 13-14). Table 3 (Caha 2009: 12) shows some examples of nouns with syncretisms in their case suffixes that target contiguous regions.

Table 3: Case syncretisms in Russian.

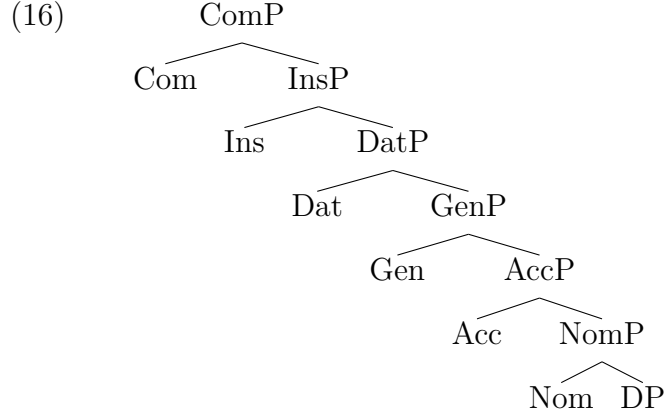
	‘window’	‘teachers’	‘two’	‘book’	‘100’
NOM	okn- o	ucitel-ja	dv-a	knig-a	st-o
ACC	okn- o	ucitel- ej	dv-a	knig-u	st-o
GEN	okn-a	ucitel- ej	dv- ux	knig-y	st- a
PREP	okn-e	ucitel-jax	dv- ux	knig- e	st- a
DAT	okn-u	ucitel-am	dv-um	knig- e	st- a
INS	okn-om	ucitel-ami	dv-umja	knig-oj	st- a

As can be seen in Table 3, *-o* in *okno* ‘window.NOM/ACC’ is syncretic for nominative and accusative, *-ej* in *ucitelej* ‘teachers.ACC/GEN’ is syncretic between accusative and genitive, etcetera. Syncretism only targets contiguous regions, and no non-contiguous syncretisms exist, e.g. nominative and genitive to the exclusion of accusative.

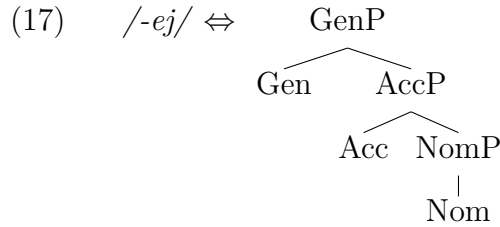
Working within nanosyntax, *Caha* implements the case hierarchy in terms of syntactic structure, given in (16). The higher, more complex cases contain the smaller, less complex cases. A nominative consists of a DP merged with nominative features. An accusative consists of a DP merged with nom-

⁹ There is some controversy about the positioning of the genitive in this ordering, as some languages (such as Icelandic, see Caha 2009: 273ff) point to an ACC — DAT — GEN ordering instead of ACC — GEN — DAT. Starke (2017) solves this problem by refining the case hierarchy, arguing that there are two types of accusative and dative, both above and below the genitive. As I do not discuss data with a genitive, I maintain the bare ordering in (15).

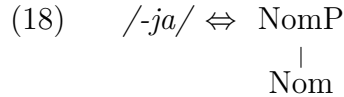
inative and accusative features. The accusative contains, by definition, the nominative, and so forth.



In nanosyntax, the syncretic forms in Table 3 involve lexical entries that can spell out multiple syntactic structures. For example, for *-ej* in *ucitelej* ‘teachers.ACC/GEN’, there is a single lexical entry that contains nominative, accusative and genitive features, as in (17).



If there is a genitive in the syntax, this lexical entry is selected, as the lexical tree exactly matches the syntactic tree. If there is an accusative in the syntax, the same lexical entry is chosen, as the lexical tree is a superset of the syntactic tree (via the Superset Principle), and there is no more specific lexical entry (Elsewhere Condition). The reason why */-ej/* is not inserted as nominative is because there is a lexical entry only for the nominative, as given in (18).



In Table 4, further evidence from case compounding is exemplified (Caha 2009: 69, after Gippert 1987).

As shown in (16), a genitive does not only contain genitive features, but also the accusative (and nominative). Table 4 shows that this containment

Table 4: Case compounding West Tocharian.

	‘horses’	‘men’
NOM	yakwi	eikwi
ACC	yakwe- m̥	eikwe- m̥
GEN/DAT	yäkwē- m̥ -ts	eikwē- m̥ -ts

relation is morphologically visible in West Tocharian: the genitive marker *-m-ts* formally contains the accusative marker *-m*.

When implemented in nanosyntax, case compounding looks as follows. I give two lexical entries for West Tocharian in (19).

- (19) a. $/-m/ \Leftrightarrow \text{AccP}$
 |
 Acc
 b. $/-ts/ \Leftrightarrow \text{DatP}$
 / \
 Dat GenP
 |
 Gen

If a genitive is spelled out, it spells out all features up to the genitive (so also the accusative features). The accusative features are spelled by */-m/* in (19a), and the genitive features are spelled out by */-ts/* in (19b). It is crucial here that the lexical entry for the genitive and dative does not contain accusative features, so the phonological structure of the accusative is not canceled out by the Principle of Cyclic Override. Instead, both phonological structures in (19) are combined as *-mts*.

2.2.2 Mass types (MP)

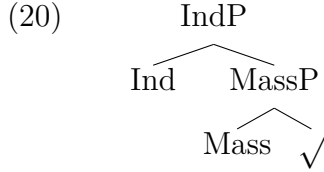
In the previous section I discussed the y-axis of Table 2 (case). This section concerns the x-axis of the table, the choice between *wer* ‘who’ and *was* ‘what’. In this paper, I follow [Hachem \(2015\)](#) who argues that the distinction between *wer* ‘who’ and *was* ‘what’ is one of ‘mass type’. Parallel to the KP for case, I use the MP for mass type. [Hachem](#) proposes that mass type is the notion that should replace, at least for German and Dutch, what has been formerly known as grammatical gender. According to her, the morphological distinctions that can be seen are not semantically empty, but they are used to classify noun phrases into different types of mass. In her work, [Hachem](#) discusses so-called *d*- and *w*-elements in German and Dutch, such as *d*-er ‘the/that’ and *w*-er

‘who’. She analyzes *-er* in *d-er* and *w-er* as the same morphosyntactic element. She lets this element combine with *d-*, which denotes a definite interpretation, or *w-*, which introduces a set of alternatives. I return to the element *w-* in the next section. In this section I discuss the feature content of the elements that combine with *w-*: *-er* and *-as*.

Hachem (2015) argues that mass type distinctions provide a much better understanding of the nature of *w*-elements than gender distinctions. First, in terms of traditional gender distinctions, *-er* would be masculine gender and *-as* would be neuter. There is no principled reason for the absence of the feminine variant that could refer to a feminine antecedent. Second, gender distinctions do not shed any light on why *was* ‘what’ represents a general lack of information and *wer* ‘who’ refers only to humans (Hachem 2015: 164). Instead, Hachem (2015) introduces the idea that *-as* and *-er* denote different types of mass. The element *-as* (former neuter) expresses unbounded mass, which is defined as ‘homogeneous undifferentiated stuff without any certain shape or precise limits’ (Koptjevskaja-Tamm 2004: 1067). The element *-er* (former masculine) refers to concentration/boundedness/individuation which ‘includes referents that have clear conceptual boundaries, such as natural objects, artifact, or body parts’ (Audring 2009: 69). As the highest point on the individuation scale is a human, *wer* has a default interpretation as a set of human individuals. Answering the question ‘Who are you thinking about?’ will always generate a set of human individuals. *Was*, on the other hand, refers to unboundedness, basically anything. An answer to the question ‘What are you thinking about?’ can be anything: an activity, a state, but also a person. This indicates that ‘what’ and ‘who’ do not refer to inanimates and animates respectively, but that ‘who’ refers to a subset of what ‘what’ refers to. If the person who asked the question knows that the answer is going to be a person, it would be infelicitous to ask ‘what’ and instead ‘who’ should be used, because this is more specific.

Hachem (2015) merges a root (which includes encyclopedic knowledge, see Hachem 2015: 97-98) with unbounded mass features (into a MassP), and with individuation mass features (into an IndP), creating an fseq that looks as in (20).¹⁰ The syntactic containment relation is reflected in the semantics: every individual is made up of mass (Hachem 2015: 108).

¹⁰ Hachem (2015) actually places a CollP (denoting a collective) and a DivP (dividing the collective into individuals, making a plural) on top of the IndP. I only discuss the two lowest elements of the hierarchy, as only these two are relevant for this paper.



Hachem (2015) argues that the unbounded mass - individuable distinction is still reflected in Modern German in remnants of multiple gender assignment. A single noun can combine with multiple determiners of different mass types, rendering distinct meanings. An example is *das Erbe* ‘the heritage’ in which the definite determiner contains the unbounded mass type marker *-as*. It is a mass noun that denotes inherited property. *Der Erbe* ‘the heir’ has the definite determiner with an individuable mass type marker *-er*. It is a count noun that denotes the person that receives the heritage. See Hachem (2015) for more arguments for the fseq she proposes, which include amongst others how nouns in Indo-European combine with different gender suffixes.¹¹

I adopt Hachem (2015)’s terminology and refer to the syntactic node that spells out *-as* as MassP (unbounded mass) and the one that spells out *-er* as IndP (individuable), which always contains the MassP.

2.2.3 W-element (WP)

As I noted at the end of the previous section, Hachem (2015) proposes an account for German and Dutch *w*- and *d*- elements. Both *w*- and *d*- combine with the same morphosyntactic objects expressing mass type (and case) distinctions. In Hachem’s work, *d*- is an element that is responsible for establishing a definite reference. The *w*-element, on the other hand, triggers the construction of a set of alternatives in the sense of Rooth (1985; 1992). This set of alternatives is restricted by whatever is in the complement of the WP (Hachem 2015: 180). For Hachem this is always the MP. This means that a *w*-item can refer to a set of alternative individuals (in the case of *wer* ‘who’) or a set of alternative unbounded masses in the case of *was* ‘what’.¹² I do not discuss the internal structure of the WP, but I assume it is a complex element that contains more than a single feature.

¹¹ Hachem (2015) points out that not all article-noun pairs exhibit the expected mass type, e.g. *das Haus* ‘the house’. According to her, most article-noun pairs in German and Dutch are nowadays idioms, fixed expressions that need to be learned. See Hachem 2015: 130-132 for discussion.

¹² I assume that the MP is a sister of the WP, as shown in (14). In the next section I show how the WP needs to be moved (via spellout driven movement) from its base position, such that the MP is not the sister of the WP anymore. Instead, the WP is a sister of KP, and the set of alternatives introduced by WP is restricted by the KP (which contains MP).

Summing up, I split the KP up in different cases, according to the fseq in (16) (Caha 2009). I split the MP up in different types of mass, according to the fseq in (20) (Hachem 2015). Lastly, I assume that the WP introduces a set of alternatives, which is restricted by the KP that contains the MP.

2.3 The lexical entries

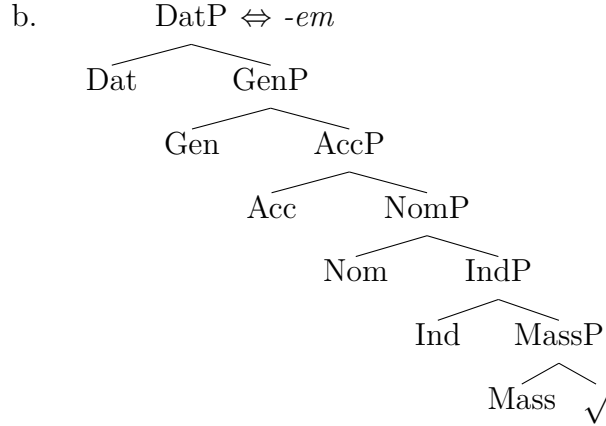
In the previous two sections, I presented nanosyntactic lexical insertion and spellout-driven movements. I also discussed the fseqs of the KP and the MP, and I introduced the WP. In this section, I give the lexical entries that I use in this paper, and I illustrate the construction of a free relative pronoun in the course of a derivation.

I start with the WP. As already pointed out in the previous section, I follow Hachem (2015) in that the WP corresponds to *w-*. The lexical entry is given in (21).

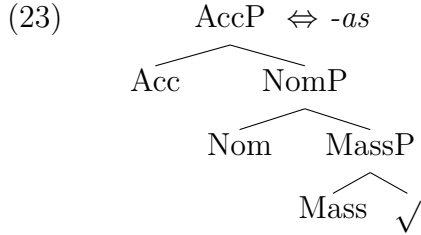
$$(21) \quad \text{WP} \Leftrightarrow w- \\ \triangle$$

Throughout this paper I illustrate my proposal using examples with the relative pronouns *wen* ‘who.ACC’, *wem* ‘who.DAT’ and *was* ‘what.NOM/ACC’. Therefore, I only give the lexical entries for *-en*, *-em* and *-as*. First, *-en* and *-em* express individuable mass, as they refer to individuals. Therefore their lexical entries contain mass type features up to the IndP. The relevant case features are built on top of the mass features. For *-en* this is the fseq up to the AccP. The lexical entry for *-em* also contains genitive and dative. The lexical entries for *-en* and *-em* are given in (22).

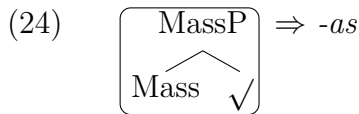
$$(22) \quad \text{a.} \quad \begin{array}{c} \text{AccP} \Leftrightarrow -en \\ \swarrow \quad \searrow \\ \text{Acc} \quad \text{NomP} \\ \quad \swarrow \quad \searrow \\ \quad \text{Nom} \quad \text{IndP} \\ \quad \quad \swarrow \quad \searrow \\ \quad \quad \text{Ind} \quad \text{MassP} \\ \quad \quad \quad \swarrow \quad \searrow \\ \quad \quad \quad \text{Mass} \quad \checkmark \end{array}$$



The marker *-as* expresses unbounded mass (i.e. basically anything) in nominative and accusative case. Specifying the lexical entry for accusative (which contains nominative) makes it possible to insert it in both contexts by means of the Superset Principle. The lexicon does not contain a lexical entry specifically for unbounded mass in nominative, so the lexical entry for unbounded mass in accusative is inserted.¹³ I give the lexical entry in (23).



In (24) to (28) I give a step by step derivation of the creation of *was* ‘what.NOM’. In the first step, the root is merged with Mass, creating MassP, shown in (24). The lexical entry that can be inserted is (23), since the lexical tree in (23) is a superset of the structure in (24), and it has less superfluous material than the entries for *-en* and *-em* (STAY).



In a separate workspace, the WP is built, and spelled out as *w-*.

¹³ The individuating mass type in nominative case (*-er*, with lexical tree [NomP[IndP[MassP[✓]]]]) will not be inserted because trees are only allowed to shrink from the top.

$$(25) \quad \begin{array}{c} \text{WP} \Rightarrow w- \\ \triangle \end{array}$$

In the next step, the WP merges with the MassP, both having their own spellout.¹⁴

$$(26) \quad \begin{array}{c} \text{WP} \\ \swarrow \quad \searrow \\ \boxed{\begin{array}{c} \text{WP} \\ \triangle \end{array}} \quad \boxed{\begin{array}{c} \text{MassP} \\ \swarrow \quad \searrow \\ \text{Mass} \quad \checkmark \end{array}} \end{array} \Rightarrow -as$$

WP merges with Nom, resulting in the structure in (27).

$$(27) \quad \begin{array}{c} \text{NomP} \\ \swarrow \quad \searrow \\ \text{Nom} \quad \text{WP} \\ \swarrow \quad \searrow \\ \boxed{\begin{array}{c} \text{WP} \\ \triangle \end{array}} \quad \boxed{\begin{array}{c} \text{MassP} \\ \swarrow \quad \searrow \\ \text{Mass} \quad \checkmark \end{array}} \end{array} \Rightarrow -as$$

No lexical entry exists for the structure as it is (STAY). There is no lexical entry for NomP that contains MassP with WP in its specifier. Following the spellout algorithm, the WP cyclically moves to the left of Nom (CYCLIC), as shown in (28).

$$(28) \quad \begin{array}{c} \text{NomP} \\ \swarrow \quad \searrow \\ \boxed{\begin{array}{c} \text{WP} \\ \triangle \end{array}} \quad \boxed{\begin{array}{c} \text{NomP} \\ \swarrow \quad \searrow \\ \text{Nom} \quad \text{MassP} \\ \swarrow \quad \searrow \\ \text{Mass} \quad \checkmark \end{array}} \end{array} \Rightarrow -as$$

Now (23) matches the structure again, and MassP is spelled out as *-as*.

This concludes the section on the internal structure of free relative pronouns. In the next section, I discuss the external syntactic structure of the free relative construction.

¹⁴ I assume the WP is created in a separate workspace and it merges as a complex specifier that projects. As the exact mechanics of how this happens are not relevant for the analysis in this paper, I do not discuss the details here. See Starke cf. 2018; Caha et al. cf. 2018 for discussion.

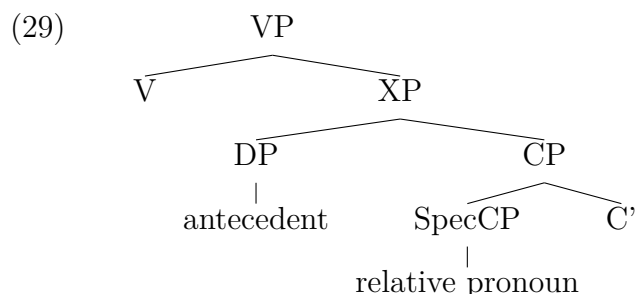
3 External syntax of free relatives

This section discusses the external syntax of free relatives. I show that the free relative pronoun is both the DP in the main clause and in the SpecCP of the embedded clause. I introduce [Van Riemsdijk \(2006a\)](#)'s concept of grafting to account for this. Grafting entails that embedded features in one structure are remerged into a different structure. Applying this to free relatives means that the free relative is connected to the main and embedded clause via the relative pronoun. I also show how nanosyntactic structures can be grafted.

3.1 Two syntactic positions

In this section I discuss the syntactic position of the relative pronoun in the main and embedded clause. I provide evidence that indicates that the relative pronoun is part of the embedded clause and also of the main clause. I conclude, following [Van Riemsdijk \(2006a\)](#), that relative pronouns are simultaneously part of both clauses, and that they should be analyzed with grafting, i.e. embedded features from one structure are remerged into a different structure.

Two positions are often considered for the relative pronoun in a free relative construction: the DP in the main clause and the SpecCP in the embedded clause ([Bresnan & Grimshaw 1978](#); [Groos & van Riemsdijk 1981](#)). These options originate from the comparison between headed relative constructions and free relative constructions, as these are the positions of the antecedent and the relative pronoun in a headed relative clause. A syntactic structure of a headed relative clause is shown in (29).



Consider the headed relative clause in (30a) and its free relative counterpart in (30b).

- (30)
- a. I like the book that you have finished reading.
 - b. I like what you have finished reading.

If one assumes that headed and free relative constructions have the same underlying syntax, the question that follows is whether *what* corresponds to the antecedent *the book* (the DP in the main clause) or to the relative pronoun *that* (in the SpecCP) in (29). The evidence that I discuss below suggests it should be in both.

Bresnan & Grimshaw (1978) argue that the relative pronoun is part of the main clause. One of the arguments she provides is that the relative pronoun triggers number agreement in the main clause, so it should be part of that clause. Himmelreich (2017) shows that this argument Bresnan & Grimshaw use for English also holds in German. In what follows I illustrate their argumentation, and compare a regular main clause, an indirect interrogative clause and a free relative construction.

In a regular main clause as in (31), a plural wh-subject requires plural number agreement.

- (31) Himmelreich (2017: 166)
Welche Bücher haben/ *hat dir gefallen?
which book.PL have.PL/ have.SG 2SG.DAT liked
‘Which books did you like?’

The construction is only grammatical if the plural subject *welche Bücher* ‘which books’ is able to agree with the plural form of the predicate (*haben* ‘have.PL’), and it is ungrammatical when the predicate has the singular form *hat* ‘have.SG’.

In contrast to that, wh-phrases that occur in embedded clauses (e.g. indirect interrogative clauses) do not affect the number agreement in the main clause, which is illustrated in (32).

- (32) Himmelreich (2017: 167)
Welche Bücher ihm gefallen, ist/ *sind unklar.
which book.PL 3SG.DAT like, be.SG/ be.PL unclear
‘It is unclear which books he likes.’

Welche Bücher ‘which books’ does not agree in number with the main clause predicate: the construction that contains the agreeing plural form *sind* ‘be.PL’ is ungrammatical, and the sentence becomes grammatical if the singular form of the predicate *ist* ‘be.SG’ is used. Thus, forms that are part of the embedded clause do not affect verbal agreement in the main clause.

If the relative pronoun patterns with the wh-element in a regular main clause as in (31) and shows agreement with the main clause verb, this is an indication that it is part of the main clause. If the relative pronoun does not

show agreement with the verb in the main clause, and it patterns with the wh-element in indirect interrogative clauses as in (32), this indicates that it is part of the embedded clause. The data show that the relative pronoun patterns with the wh-element in a regular main clause: in free relative constructions with a plural (complex) wh-phrase, plural agreement is required.¹⁵ *Welche Bücher* ‘which books’ in (33) has to agree in number with *haben* ‘have.PL’, and the use of the singular form (*habe* ‘have.SG’) renders an ungrammatical result.

- (33) Himmelreich (2017: 167)
 Welche Bücher ich auch immer gelesen habe, haben/ *hat
 which book.PL 1SG.NOM ever read have, have.PL/ have.SG
 mir gefallen.
 1SG.DAT liked
 ‘I liked whatever books I read.’

This suggests that the relative pronoun is part of the main clause.

Note here though that this is positive evidence for the relative pronoun being part of the main clause, and not negative evidence indicating that the relative pronoun is not part of the embedded clause. In other words, (33) excludes the possibility that the relative pronoun is only part of the embedded clause and not of the main clause. This argument is compatible with the relative pronoun being simultaneously part of both the main and embedded clause, which is what I argue for, following Van Riemsdijk (2006a).

Groos & van Riemsdijk (1981) argue that the relative pronoun should be in the embedded clause. One of the arguments they use comes from relative clause extraposition. They examine the positional behavior of the DP in the main clause and the relative clause in headed relative clauses, and compare this to the position of the relative pronoun in free relative constructions. The headed relative clauses that Groos & van Riemsdijk discuss are given in (34).

- (34) Groos & van Riemsdijk (1981: 185)
 a. Der Hans hat das Geld, das er gestohlen hat, zurückgegeben.
 the Hans has the money which he stolen has returned
 ‘Hans has returned the money that he has stolen.’

¹⁵ Himmelreich gives examples of complex wh-phrases (*welche Bücher* ‘which books’) in German, since simple wh-phrases (e.g. *was* ‘what’) always have singular number agreement. In Spanish, number agreement with relative pronouns also occurs with simple plural wh-phrases (Himmelreich 2017: 168).

- b. Der Hans hat das Geld zurückgegeben, das er gestohlen
 the Hans has the money returned which he stolen
 hat.
 has
 ‘Hans has returned the money that he has stolen.’
- c. * Der Hans hat zurückgegeben, das Geld, das er gestohlen
 the Hans has returned the money which he stolen
 hat.
 has
 ‘Hans has returned the money that he has stolen.’

The example in (34a) shows a headed relative clause with the DP *das Geld* ‘the money’ in the main clause and the relative clause *das er gestohlen hat* ‘which he has stolen’ in base position. In (34b), the relative clause is extraposed to the right edge of the sentence, and the DP in the main clause remains behind. This is possible in German and the sentence is grammatical. In (34c), both the relative clause and the DP in the main clause are extraposed, and the sentence is ungrammatical. The conclusion is that only the relative clause can be extraposed, but the DP from the main clause cannot.

The question is what happens to the relative pronoun if a relative clause in a free relative construction is extraposed. If the relative pronoun were part of the main clause, the relative pronoun would be expected to take the position of the DP in the main clause: it should be possible to be stranded on its own in the main clause (as in (34b)), and it should be impossible to extrapose to the right edge of the clause (as in (35)).

- (35) Groos & van Riemsdijk (1981: 185)
 Der Hans hat zurückgegeben, was er gestohlen hat.
 the Hans has returned what he stolen has
 ‘Hans has returned what he has stolen.’

However, (36) shows that a sentence is ungrammatical if the relative pronoun is stranded, and (35), in which *was* ‘what’ is extraposed to the right edge of the clause, together with the rest of the relative clause, is grammatical.

- (36) Groos & van Riemsdijk (1981: 185)
 * Der Hans hat was zurückgegeben, er gestohlen hat.
 the Hans has what returned he stolen has
 ‘Hans has returned what he has stolen.’

This suggests that the relative pronoun is part of the embedded clause. To summarize, there is evidence that the relative pronoun is part of the main clause, and there is evidence that it is part of the embedded clause. Given these observations (and because relative pronouns seem to take case requirements from the main and embedded clause into account, see Section 1), different proposals have been made to let the relative pronoun simultaneously be the DP in the main clause and in the SpecCP of the embedded clause. These proposals can be grouped into two categories: (1) having a single element in each syntactic position, and realizing only one of them phonologically, or (2) having a single syntactic element that is part of both clauses because of a sharing relation (cf. multidominance or grafting).

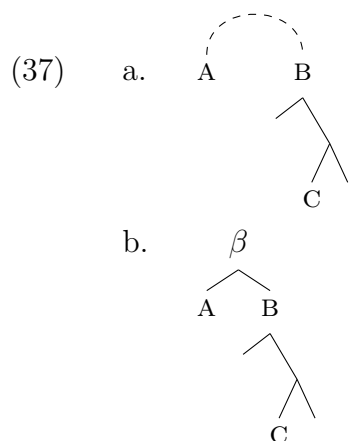
An example of the first type of proposal is the one by [Himmelreich \(2017\)](#). She argues that there are two elements (one in the main clause and one in the embedded clause) that bear the same syntactic feature values. The sharing of features comes about by several non-standard Agree relations. The two elements are always present, and one of the elements is always phonologically empty. In order to avoid having non-standard Agree relations and having to stipulate the presence of a phonologically empty element, I do not adopt [Himmelreich \(2017\)](#)'s approach. A proposal that fits into the second category is [Asarina \(2011\)](#)'s account for right node raising in Russian. She proposes that a case conflict can be resolved if conflicting case features on a single node can be realized by the same lexical entry. Her proposal does not extend to free relatives that are sensitive to more or less complex case. For these cases, there is not a single underspecified lexical entry available that corresponds to two cases. In the system of [Asarina \(2011\)](#) these sentences are predicted to crash, which is not what happens in German. However, [Asarina \(2011\)](#)'s and my analysis share the idea of having a single syntactic element that is simultaneously part of both the main and embedded clause via a sharing configuration. In my proposal, I adopt [Van Riemsdijk \(2006a\)](#)'s proposal of grafting, on which I elaborate in the next section.¹⁶

¹⁶ An anonymous reviewer raised the questions what the motivation is for adopting grafting and not (any other type of) multidominance. It goes beyond the scope of the paper to discuss the differences and similarities between sharing accounts such as multidominance and grafting. The intuition that I adopt for this paper is that embedded syntactic structure can be remerged into a different clause (external remerge), which can be achieved by both grafting and (at least some versions of) multidominance. For expository reasons, I frame this proposal using [Van Riemsdijk \(2006a\)](#)'s grafting approach.

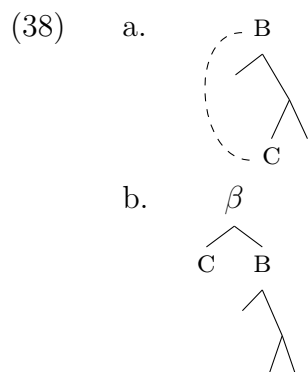
3.2 Grafting

Van Riemsdijk (2006a) has argued for a grafting analysis, in which the relative pronoun is simultaneously the DP in the main clause and in the SpecCP of the embedded clause. In his proposal embedded features in one structure are remerged into a different structure. He argues that this special type of merge must exist, when all logical combinations of internal and external merge are taken into account. Grafting essentially combines properties of internal and external merge. The reasoning in Van Riemsdijk (2006a) goes as follows.

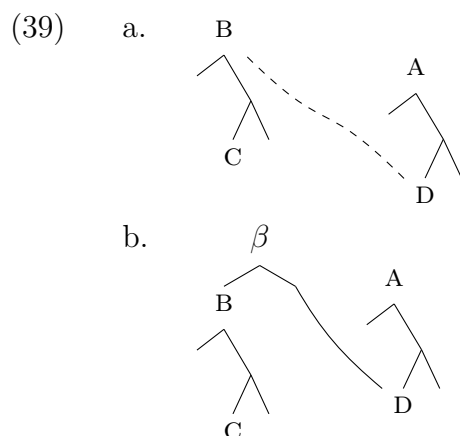
In (37a), A and B are combined with external merge. The two distinct structures are taken and combined to form the structure in (37b).



Internal merge takes a subpart of an existing structure as one of the two objects. In (38a), a subpart of a structure called C is combined with B via internal merge. The result is its movement, as shown in (38b).



In grafting, the properties of internal and external merge are combined. Just like in external merge, two distinct structures are combined, and, like in internal merge, a subpart of an existing structure is one of the merging objects. In (39a), merge applies to B and D. D is a subpart of an existing structure and it remerges with the distinct structure B (i.e. it is grafted). In (39b), D is the element that is shared between the two structures. It is a sister of B but still preserves the structural relations within its own structure (Van Riemsdijk 2006a: 22).



3.3 Grafting and nanosyntactic structures

In this section, I show the analysis of free relative constructions with matching case requirements. The derivation presented is identical (apart from some simplifications) to what is proposed in Van Riemsdijk 2006b, except that in this proposal the relative pronoun consists of a complex syntactic structure rather than a single syntactic node. Just like in Van Riemsdijk 2006b, the derivation proceeds in three steps.¹⁷

- (40) Derivational steps for German matching free relatives
- a. Externally merge the embedded clause predicate with the relative pronoun
 - b. Internally remerge the relative pronoun to the left edge of the clause


¹⁷ Non-spellout driven movement, such as the movement of the relative pronoun to the SpecCP position, falls outside the scope of this paper. See De Clercq 2019: 21 for an approach in which syntactic movement is motivated by spellout. The idea is that a feature can (right after being merged) attract a constituent from the derivation to its specifier, if it provides the feature with a spellout. This movement precedes spellout-driven movement operations.

- c. Externally remerge (graft) the main clause predicate with the relative pronoun

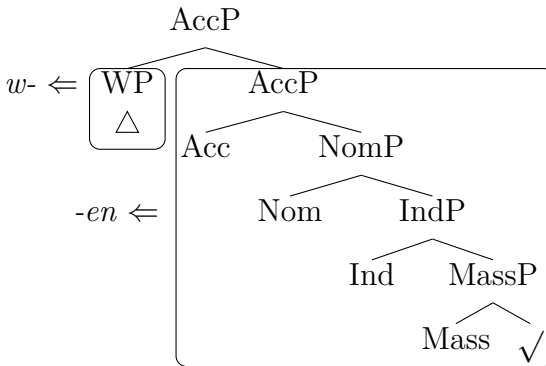
The sentence under examination is given in (41), repeated from the introduction. Both *lade ein* ‘invite’ and *mag* ‘likes’ require their objects to be in accusative case.

- (41) Vogel (2001: 344)
 Ich lade ein, wen auch Maria mag.
 I invite_{acc} who.ACC also Maria likes_{acc}
 ‘I invite whoever Maria also likes.’

The derivation starts with the embedded clause. *Mag* ‘likes’ combines with an accusative, as shown in (42).¹⁸

- (42)
- 

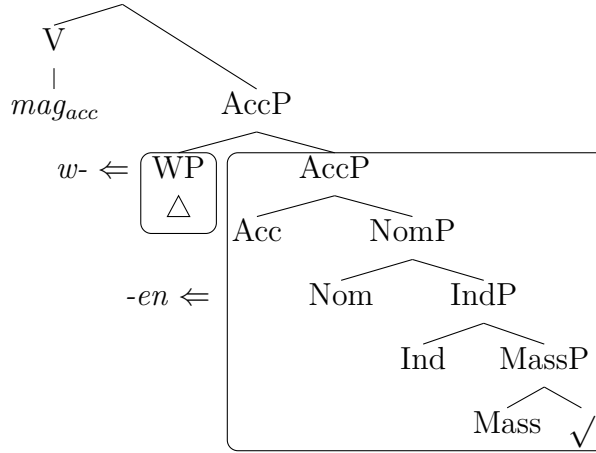
The accusative relative pronoun in (41) has an individuable mass type interpretation, so this structure needs to be built. This happens in a similar way to how *was* ‘what’ was built in Section 2.3. I leave out the derivation of the pronoun *wen* ‘who.ACC’, and I give the result in (43).

- (43)
- 

The embedded clause predicate *mag* ‘likes’ merges with the accusative relative pronoun (the AccP), as shown in (44).

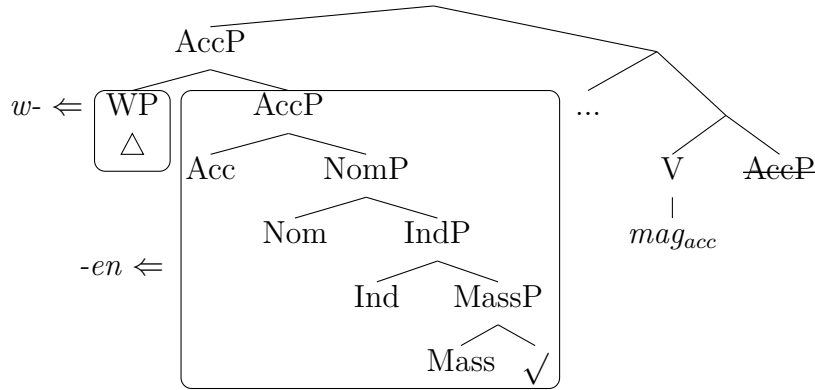
¹⁸ In this paper I do not discuss argument selection. For now I assume that part of the syntactic structure associated with the predicate selects for the case it requires, and merges with this particular case node in the syntax. In this paper I abstract away from this, and the predicate is simply merged directly with the required case node.

(44)



I ignore the construction of the rest of the structure, and continue with the second derivational step: the AccP is internally remerged to the left edge of the clause. The dots in (45) refer to the material between the leftmost position in the clause and the predicate. The position where the relative pronoun is moved from is ~~AccP~~.

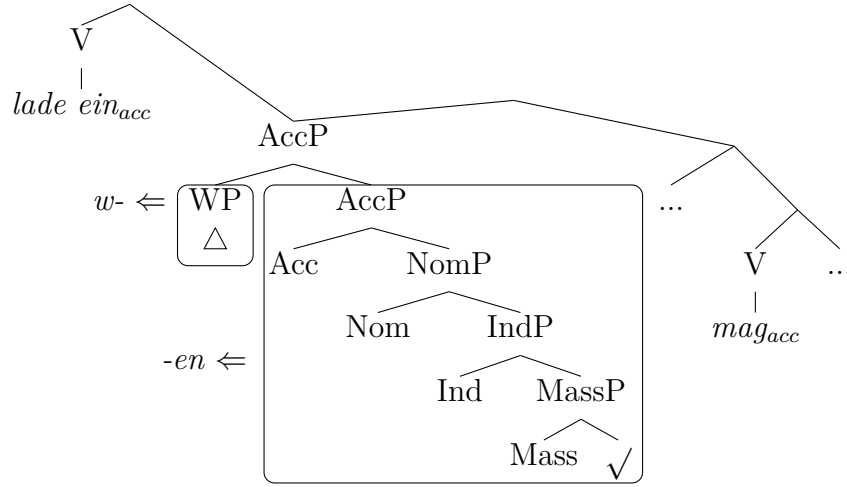
(45)



Lade ein ‘invite’ requires its object to be in accusative, so it has to combine with the AccP. Grafting occurs: the AccP externally remerges with the main clause predicate, as shown in (46). As a result, *wen* ‘who.ACC’ is part of both the main and embedded clause.¹⁹

¹⁹ There are some issues that need to be resolved, such as the linearization of these structures and theta role assignment. I do not have anything to offer here, and leave it open to future research.

(46)



To summarize, Section 3 discussed the external syntax of free relative constructions. I argued for a grafting approach (Van Riemsdijk 2006a), and I showed how nanosyntactic structures can be grafted to account for a matching free relative. With the background from the last two sections in mind, I derive the mismatching free relatives in the next section.

4 Deriving the patterns

In this section, I outline my analysis, accounting for the patterns in Table 1, repeated in Table 5.

Table 5: Grammaticality pattern for German.

	case in embedded clause	
	more complex	less complex
distinct forms	✓	×
syncretism	✓	✓

I propose that mismatching free relatives are derived in the three steps described in (40). First, the relative pronoun is externally merged with the embedded clause predicate. Second, the relative pronoun is internally remerged to the edge of the clause. In the third step, several options are possible, all of which involve the grafting of two elements. If the embedded clause predicate requires a more complex case than the main clause predicate (the left column in Table 5), the required embedded case node is grafted into the main clause. I

go through the derivations in Section 4.1. If the main clause predicate requires the more complex case, the required case node is not available for the main clause predicate to externally remerge with. Instead, the highest case node is externally remerged with the next case node in the fseq. Case features are added until the case node required by the main clause predicate is merged. The adding of case feature is subject to a restriction, which correctly rules out non-syncretic forms and predicts syncretic forms to be grammatical. I discuss the relevant derivations (those of the configurations in the right column in Table 5) in Section 4.2.

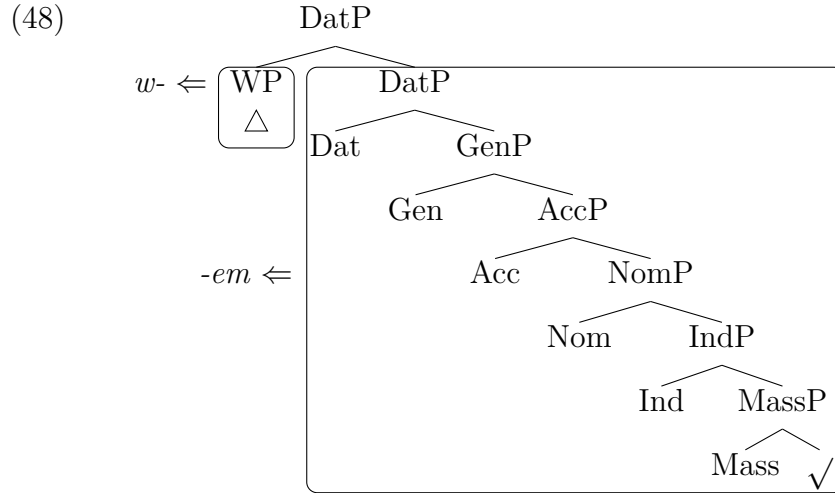
4.1 Grafting a less complex case

I start by discussing free relative constructions in which the case required in the embedded clause is more complex than the case required in the main clause. These constructions are grammatical, independent of whether the forms corresponding to the cases are syncretic or not. I first discuss the examples with non-syncretic forms. Then, I show how the same analysis derives the syncretic variant. In the case of non-syncretic forms, the relative pronoun needs to appear in the form that corresponds to the more complex case, as shown again in (47). The use of *wem* ‘who.DAT’ grammatical, the use of *wen* ‘who.ACC’ is not.

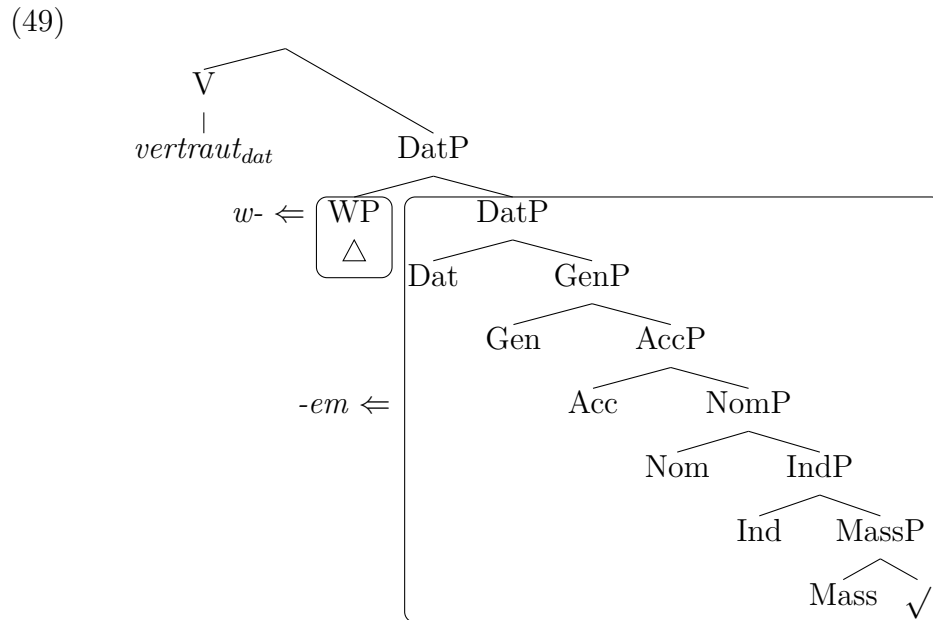
(47) Vogel (2001: 344)

- a. Ich lade ein, wem auch Maria vertraut.
 I invite_{acc} who.DAT also Maria trusts_{dat}
 ‘I invite whoever Maria also trusts.’
- b. * Ich lade ein, wen auch Maria vertraut.
 I invite_{acc} who.ACC also Maria trusts_{dat}
 ‘I invite whoever Maria also trusts.’

The derivation starts with the embedded clause. *Vertraut* ‘trusts’ in (47a) combines with a relative pronoun with an individuated mass type interpretation in dative case. This relative pronoun is built as shown in (48).

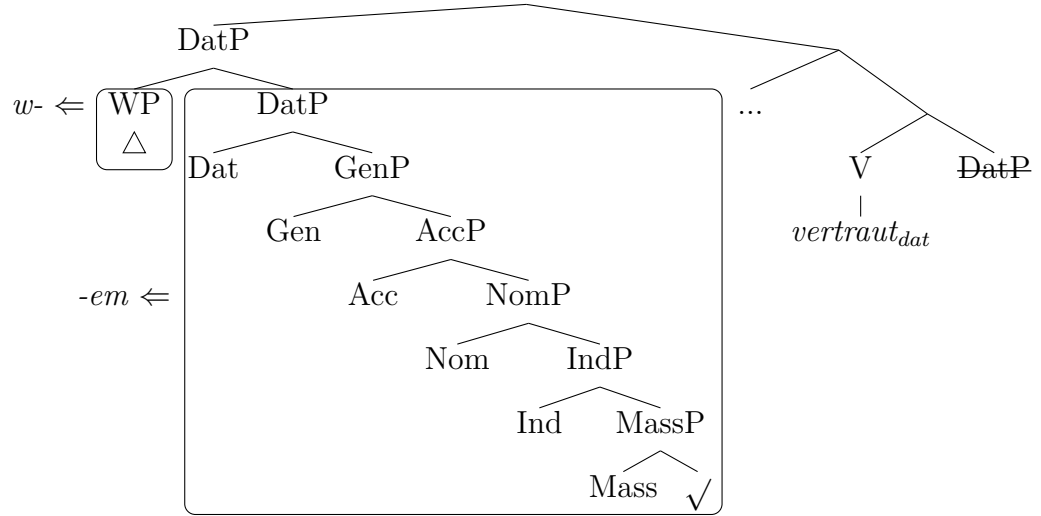


It becomes clear why the accusative relative pronoun *wen* ‘who.ACC’ cannot be used in this configuration, and why (47b) is ungrammatical: *wen* ‘who.ACC’ has been overwritten by *wem* ‘who.DAT’ during the creation of the dative relative pronoun. First, the predicate *vertraut* ‘trusts’ externally merges with the *DatP*, as shown in (49).



In the next step, the *DatP* is internally remerged to the left edge of the clause, as shown in (50).

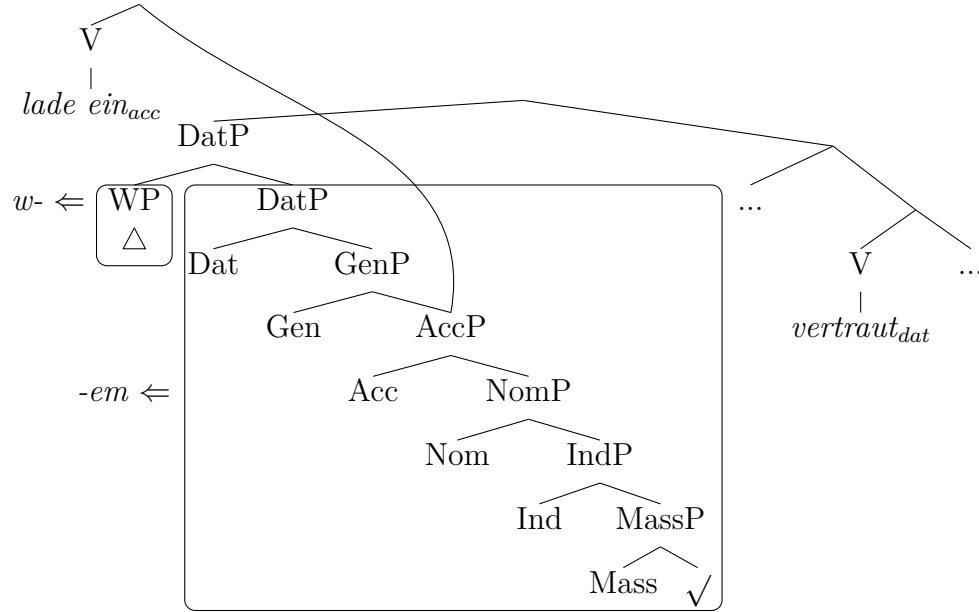
(50)



Finally, the relative pronoun is externally remerged with the main clause predicate. The main clause predicate is *lade ein* ‘invite’, which requires its object to be in accusative. Therefore, it is not the highest node, the **DatP**, that is externally remerged, but it is the **AccP**, an embedded case node. What is crucial here is that the **DatP** always contains the **AccP** in the fseq (Caha 2009). This is the reason why the **DatP** can also satisfy an accusative case requirement. In Section 5.2 I show that it is language-specific whether only the highest case node or also with an embedded case node can be grafted. German groups with languages that allow for grafting embedded case nodes.

The last step of the derivation is illustrated in (51): **AccP** is grafted into the main clause.

(51)



Grafting targets embedded features; DatP and AccP are both embedded in the structure (i.e. none of them is the root), so the grafting operation is the same. In (52), I give a revised version of the derivational steps, in which case nodes on the relative pronoun are included.

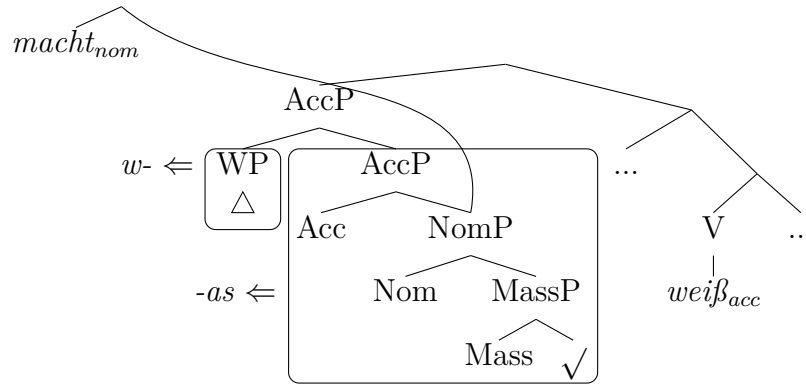
- (52) Derivational steps for non-matching free relatives in German (to be revised)
- Externally merge the embedded clause predicate with the required case node
 - Internally remerge the relative pronoun to the left edge of the clause
 - Externally remerge (graft) the main clause predicate with the required case node

For completeness, I also include the analysis for the syncretic variant in the left column of Table 5. Again, the more complex case is required in the embedded clause, but now a syncretic form corresponds to the different cases, as in (53).

- (53) Vogel (2001: 363)
- | | | | |
|----------------|-------------------------|----------------------|------------------|
| Was | ich nicht weiß, | macht | mich nicht heiß. |
| What.NOM/ACC I | not know _{acc} | makes _{nom} | me not hot |
- ‘What I don’t know doesn’t excite me.’

The analysis is identical to the one described in this section, illustrated in (54). First, *weiß* ‘know’ combines with the AccP. In the next step, the AccP is moved to the left edge of the clause. Last, *macht* ‘makes’ externally remerges with the NomP (which is contained in the AccP).

(54)



In this section, I discussed configurations in which the embedded clause predicate requires a more complex case than the main clause predicate.

4.2 Grafting a more complex case

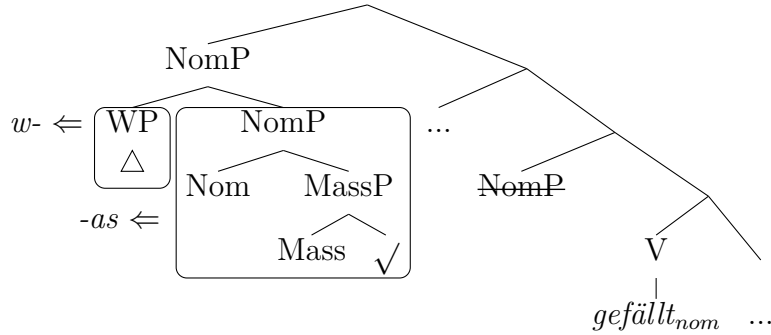
In this section, I discuss configurations in which the embedded clause predicate requires a less complex case than the main clause predicate. The first two steps in (52) can proceed as in the previous section, but the external remerge of the relative pronoun with the main clause predicate is not possible. I illustrate this with the example in (55). The embedded clause predicate *gefällt* ‘pleases’ combines with a nominative subject, and the main clause predicate *erzähle* ‘tell’ combines with an accusative object.

(55) Vogel (2001: 344)

Ich erzähle, was immer mir gefällt.
 I tell_{acc} what.NOM/ACC ever me pleases_{nom}
 ‘I tell whatever pleases me.’

The derivation starts in the embedded clause, where *gefällt* ‘pleases’ merges with the NomP, and the NomP moves to the left edge of the clause. (56) shows the structure after this step.

(56)



At this point, *erzähle* ‘tell’ is supposed to externally remerge with an AccP. However, as can be seen in (56), there is no AccP available in the structure (NomP does not contain AccP). Fortunately, there is another way for the derivation to proceed. Not only the main clause predicate can externally remerge case nodes, but other elements, such as other case features, can do this as well. Whichever two elements are externally remerged, the grafting operation remains the same. I suggest that case features are merged, until the case node required by the main clause is available, and the main clause predicate can be merged.

I show that German is subject to a certain restriction with respect to merging additional case features. The added features are not permitted to cause an overriding of the spellout. I call this restriction ‘Keep spellout’. This restriction is going to make the crucial distinction between syncretic and non-syncretic forms: in the syncretic case, the same lexical entry, (the same spellout) is used, but in the non-syncretic case, a different lexical entry (a different spellout) is used. Normally, the Principle of Cyclic Override applies if newly added features are spelled out and the previous spellout is overridden. However, in this case, a single element is part of two syntactic structures. It seems that in a grafting situation, Cyclic Override needs to be restricted. In Section 5.2 I show that it is language-specific whether ‘Keep spellout’ holds. German groups with languages that does not allow overriding. (57) describes the derivational steps for German non-matching free relatives, taking this restriction into account.

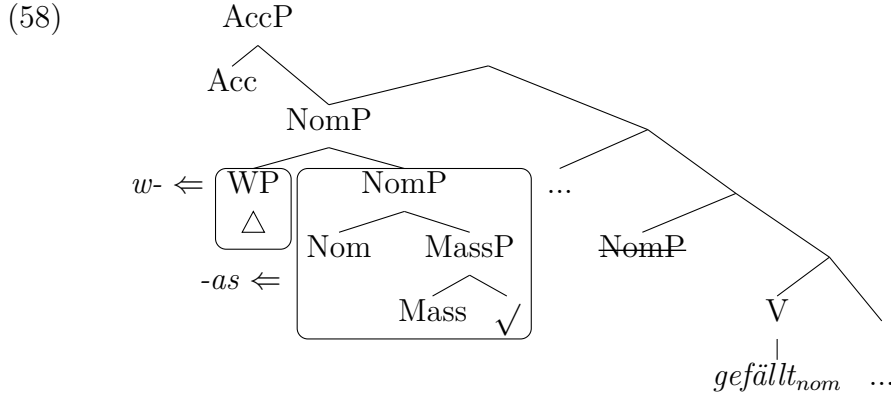
- (57) Derivational steps for German non-matching free relatives
- a. Externally merge the embedded clause predicate with the required case node
 - b. Internally remerge the relative pronoun to the left edge of the clause
 - c. Externally remerge (graft) the main clause predicate with the required case node

- (i) if the required case node is not available, externally remerge (graft) the highest case feature with the next case feature in the fseq, and externally merge the main clause predicate with the required case node
 - (ii) if the required case node is not available, continue to externally merge the additional case features following the fseq (until the required case node has merged), and externally merge the main clause predicate with the required case node
- *Restriction ‘Keep spellout’*: only case features within the boundaries of the inserted lexical entry can be merged

In the remainder of this section I show how the revised derivational steps derive the syncretetic and non-syncretetic configurations.

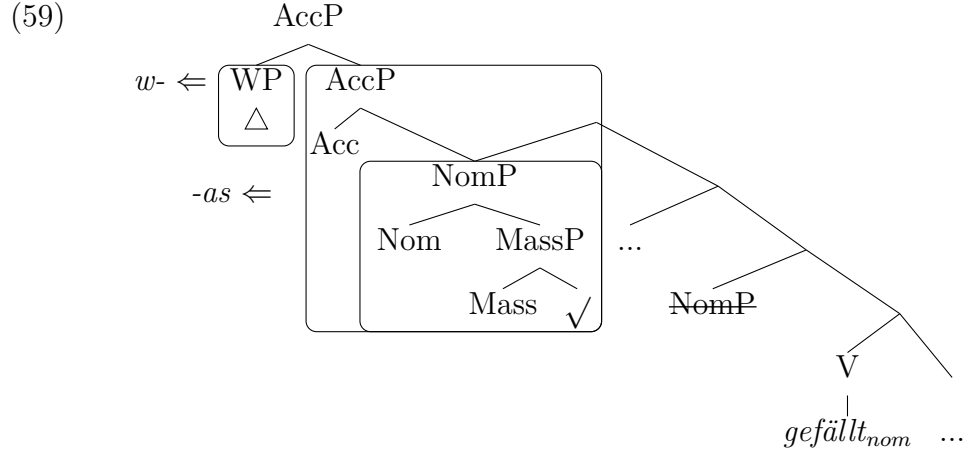
4.2.1 Syncretetic forms

I showed that if the main clause predicate *erzähle* ‘tell’ requires a more complex case than the embedded clause predicate (as in (55)), external remerge cannot take place between the predicate and the required case node. Instead, the next case feature in the fseq externally remerges the highest case node. The required case node is AccP, so accusative case features are merged. The highest case node is the NomP, so accusative features are merged to create an AccP, as shown in (58).



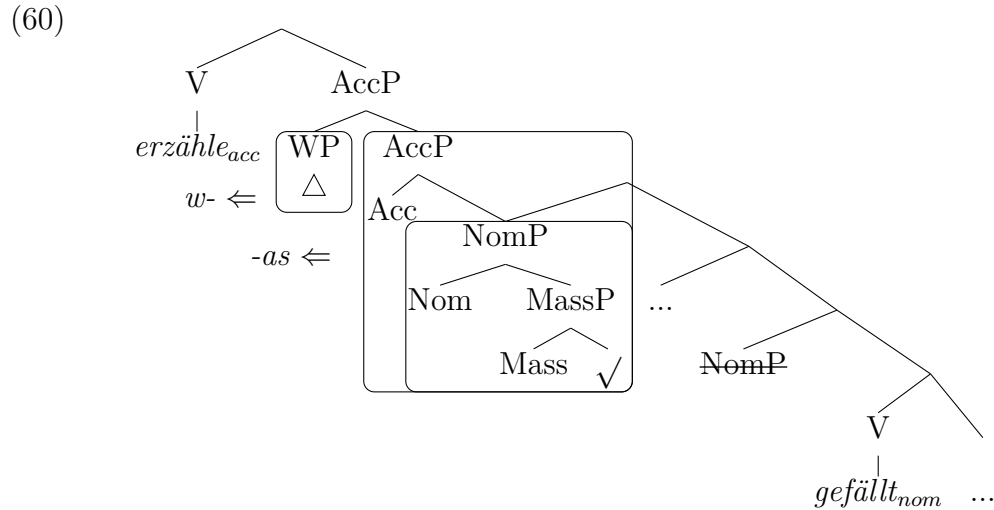
The AccP needs to receive a spellout. There is no lexical entry available for the AccP that contains the WP and NomP. Following the spellout algorithm,

the WP cyclically moves to the left of Acc (CYCLIC). At this point, the AccP can be realized as *-as* (see (23)).²⁰



This insertion adheres to the restriction ‘Keep spellout’. The nominative and accusative in unbounded mass are syncretic in German, so there is a single lexical entry that is used both as nominative and as accusative. In (59), it is still the lexical entry *-as* that is inserted into the AccP, just as it was for the NomP.

With the AccP available, the main clause predicate *erzähle* ‘tell’ can merge with the accusative case node it requires, shown in (60).



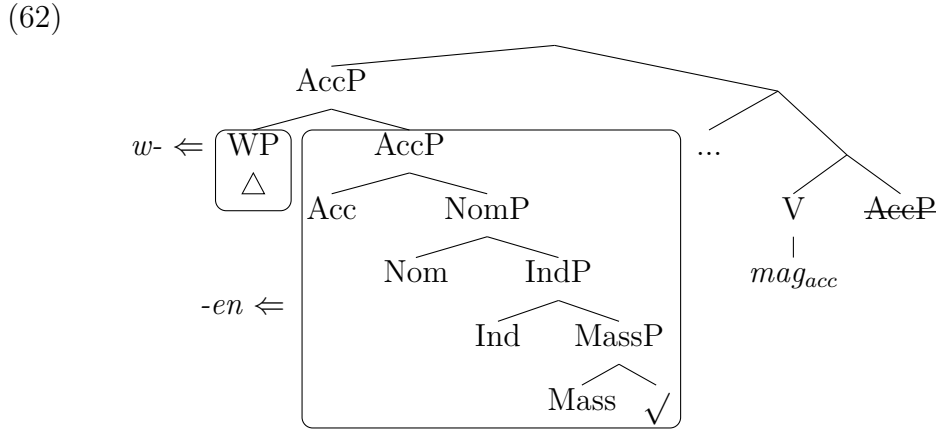
²⁰ I assume here that *-as* can be inserted because the Acc, Nom, Mass and ✓ form a constituent: Acc dominates Nom, Mass and ✓. The embedded clause does not play a role in the insertion process.

4.2.2 Non-syncretic forms

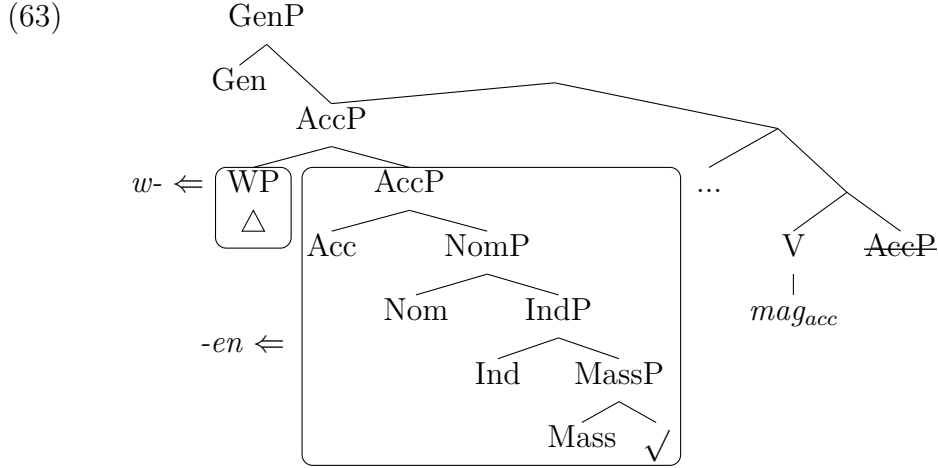
In the previous section I discussed the grammatical example in which the embedded clause predicate requires a less complex case, and there is a syncretic form for both required cases (the lower right cell in Table 5). This section discusses the ungrammatical example in which the embedded clause predicate requires a less complex case than the main clause predicate, and the phonological forms that correspond to the two cases differ. I repeat the example in (61).

- (61) Vogel (2001: 345)
 * Ich vertraue, wen/ wem auch Maria mag.
 I trust_{dat} who.ACC/ who.DAT also Maria likes_{acc}
 ‘I trust whoever Maria also likes.’

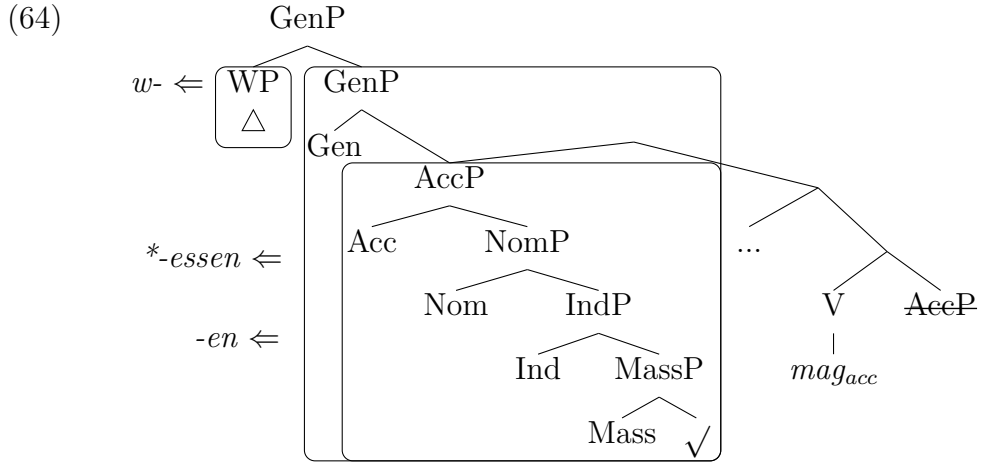
Again, up to the second step, the derivation proceeds normally. The derivation starts in the embedded clause. First, *mag* ‘likes’ externally merges with the AccP, and the relative pronoun is internally remerged to the left edge of the clause. (62) shows the structure after this step.



Vertraue ‘trust’ is supposed to externally remerge with a DatP. However, there is no DatP available in the structure (AccP does not contain DatP), so grafting between the predicate and the required case node cannot take place. Instead, the next case feature in the fseq externally remerges the highest case node. Ultimately, the structure needs to be built up to the DatP. The first step is to externally remerge the AccP with genitive features, creating a GenP, as shown in (63).



The GenP needs to receive a spellout. There is no lexical entry available for the GenP containing the WP and the AccP. Following the spellout algorithm, the WP cyclically moves to the left of Gen (CYCLIC). The GenP is realized as *-essen* (which is the genitive individuated mass).



However, the insertion of *-essen* violates the restriction ‘Keep spellout’. The phonological form *-essen* overrides the earlier spellout *-en*. Because this overriding is not permitted, the derivation crashes and the construction is ungrammatical.

To summarize, in this section I showed that satisfying two different case requirements happens in three steps. First, the relative pronoun is externally merged with the embedded clause predicate. Second, the relative pronoun is internally remerged to the left edge of the clause. In the third step, grafting takes place. In case the embedded clause requires the more complex case,

the main clause predicate externally remerges with an embedded case node. When the main clause requires the less complex case, the highest case node and the next case feature in the fseq are externally remerged. Case features are merged until the required case feature is added, and then the main clause predicate combines with the required case node. German is subject to the restriction ‘Keep spellout’: case features can be added as long as the spellout of the relative pronoun remains the same.

5 Cross-linguistic differences

In the previous section I showed how the grammaticality pattern of German can be captured using case containment and grafting. This section discusses patterns that appear in other languages. First, I briefly return to the point that syncretism resolving case conflicts seems to be universal. Then, I switch to non-syncretic forms, and I show that some languages behave differently from German in this respect. Two aspects of the analysis for German are shown to be language-specific, and are formulated in terms of restrictions. Varying the presence of these restrictions provides different grammaticality patterns observed in four different languages.

5.1 Syncretism across languages

In Section 4.2.1, I showed how the German *was* ‘what.NOM/ACC’ can satisfy nominative case requirement from the embedded clause and an accusative case requirement from the main clause. I argued that this is possible because the next case feature (accusative) can be externally remerged with the highest case node available in the structure (NomP). This is permitted as long as the phonological form of the relative pronoun does not change, i.e. as long as the same lexical entry is used. This possibility to freely insert features within the boundaries of a lexical entry is not specific to German. The examples from (6), repeated here in (65), show that syncretism is able to resolve feature conflicts in free relatives cross-linguistically.

- (65) a. *Russian*, Levy & Pollard (2002: 222)
 Kogo ja iskal, ne bylo doma.
 who.ACC/GEN I sought_{acc} not was_{gen} home
 ‘Who I was looking for wasn’t at home.’

- b. *conversational Bulgarian*, Izvorski (1997: 279)
 Šte celuna kojto dojde prâv.
 will kiss_{acc} who.NOM/ACC comes_{nom} first
 ‘I will kiss whoever comes first.’
- c. *Polish*, Himmelreich (2017: 17), after Citko (2013)
 Jan unika kogokolwiek wczoraj obraził
 Jan avoids_{gen} whoever.ACC/GEN yesterday offended_{acc}
 ‘Jan avoided whoever he offended yesterday.’

The analysis laid out in this paper holds for syncretic non-matching free relatives cross-linguistically.

5.2 Language-specific restrictions

So far, the proposal holds cross-linguistically for the syncretic cases, but it does not for the non-syncretic cases. This section extends the proposal such that it also accounts for these patterns. I have shown that German non-matching non-syncretic free relative constructions are only grammatical when the more complex case is required in the embedded clause. This section shows that Gothic permits the constructions independent of which clause requires the more complex case, Polish does not allow any non-matching non-syncretic free relatives, and in Greek non-matching non-syncretic free relatives are only grammatical if the more complex case is required in the main clause. An overview is given in Table 6.

Table 6: Grammaticality patterns non-matching non-syncretic free relatives.

	embedded clause case	
	more complex	less complex
German	✓	×
Gothic	✓	✓
Polish	×	×
Greek	✓	✓

The variation between the languages can be reduced to the presence or absence of two restrictions. The first restriction (‘Keep spellout’), which has been shown to hold for German, is that the spellout of the relative pronoun needs to remain the same, and it cannot be overridden.²¹ The second restriction, which has not been discussed yet because it does not apply to German,

²¹ It was suggested to me by an anonymous reviewer this ban on overriding is not hardwired in language, but something that differs per language. Overriding does not seem to hold for

is that external remerge can only take place with the highest case node (and not with an embedded one). I call this restriction ‘Only graft highest node’. I propose that languages differ in whether or not the restrictions hold. The presence or absence of two restrictions generates four possible patterns that arise in four different languages: German, Gothic, Polish and Greek. (66) shows an updated version of the derivational steps, in which the restrictions per language are incorporated.

- (66) Derivational steps for non-matching free relatives
- a. Externally merge the embedded clause predicate with the required case node
 - b. Internally remerge the relative pronoun to the left edge of the clause
 - c. Externally remerge (graft) the main clause predicate with the required case node
 - *Restriction ‘Only graft highest node’ for Polish and Greek: only the structurally highest case node can be grafted*
 - (i) if the required case node is not available, externally remerge (graft) the highest case feature with the next case feature in the fseq, and externally merge the main clause predicate with the required case node
 - (ii) if the required case node is not available, continue to externally merge the additional case features following the fseq (until the required case node has merged), and externally merge the main clause predicate with the required case node
 - *Restriction ‘Keep spellout’ for German and Polish: only case features within the boundaries of the inserted lexical entry can be merged*

Table 7 shows which restriction holds for which language. First, Polish has both restrictions: it is not allowed to override the spellout of the relative pronoun, and only the highest case node can be grafted. German only has one restriction: it is required to keep its spellout, but grafting of the highest and embedded nodes is permitted. Greek is the opposite of German in that it allows for overriding of the spellout, but it does not permit grafting an embedded case node. Gothic is the most free of the four languages: none of the restrictions hold.

languages with case attraction, such as Gothic. I thank the anonymous reviewer for this suggestion.

Table 7: Two possible restrictions in four languages.

		‘Only graft highest node’	
		yes	no
‘Keep spellout’	yes	Polish	German
	no	Greek	Gothic

In the remainder of this section I show the relevant data and I derive the patterns with the derivational steps in (66).

5.2.1 Gothic

The first language I discuss is Gothic. In Gothic, the relative pronoun takes the case of the more complex predicate, independent of whether it is the main clause or the embedded clause predicate that requires it.²² (67a) gives an example of a predicate requiring a more complex case in the embedded clause and a predicate requiring a less complex one in the main clause. *Lag* ‘lay’ takes a dative in the embedded clause, and *ushafjands* ‘picking up’ takes an accusative in the main clause. In (67b), the case requirements are reversed: *qipip* ‘say’ takes an accusative in the embedded clause and *taujaui* ‘do’ takes a dative in the main clause. In both (67a) and (67b) the dative relative pronoun is used.

- (67) *Gothic*, Harbert (1978: 339,434)
- a. *ushafjands ana þamm-ei lag*
picking up_{acc} on DAT-compl lay_{dat}
‘picking up that on which he lay’
 - b. *hva nu wileip ei taujaui þamm-ei qipip þiudan Iudaie?*
what now want that do_{dat} DAT-compl say_{acc} king of Jews
‘What now do you wish that I do to him whom you call King of the Jews?’

The derivation of (67a), in which the embedded clause predicate requires the more complex case, is identical to the one for German described in Section 4.1. First, the embedded clause predicate *lag* ‘lay’ externally merges with the dative case of the relative pronoun. Next, the relative pronoun is internally remerged to the left edge of the clause. Last, the main clause predicate *ushafjands* ‘picking up’ externally remerges with the embedded accusative case node.

²² The generalizations regarding Gothic originate in Caha 2014.

The construction in (67b) is grammatical in Gothic but its German counterpart is not. Therefore, the derivation for the Gothic (67b) differs from what is described in Section 4.2.2 for German. I suggested that the difference between the languages can be reduced to whether or not it is allowed to override the spellout of the relative pronouns after grafting (i.e. the presence or absence of ‘Keep spellout’). This restriction does not hold for Gothic. The first step in the derivation is that the embedded clause predicate *qibip* ‘say’ externally merges with the accusative case node. Next, the relative pronoun internally remerges to the left edge of the clause. In the third step, the dative case node is not available for the main clause predicate *taujaui* ‘do’ to be externally remerged with. Therefore, the highest case node (AccP) externally remerges with the next case feature in the fseq (creating a GenP), and with the next feature after that (creating a DatP), such that the required (dative) case node has been merged. The spellout of the accusative relative pronoun is overridden, first by the genitive, and then by the dative. This is the point at which Gothic and German crucially differ: overriding the spellout of the relative pronoun is not allowed in German (illustrated by (64)), but it is in Gothic. Finally, the main clause predicate *taujaui* ‘do’ merges with the dative case node. In this derivation, the spellout of the Gothic relative pronoun is overridden, which is the crucial difference between Gothic and German.

5.2.2 Polish

The next language I discuss is Polish. In Section 1, I mentioned that Polish (and a different variety of German) requires a stricter form of matching than the variety of German I discussed in this paper. Non-syncretic non-matching free relatives are ungrammatical, independent of whether the embedded or the main clause requires the more complex case.²³ In (68a), the embedded clause predicate *dokucza* ‘teases’ requires a dative object, and the main clause predicate *lubi* ‘likes’ requires an accusative. In (68b), the embedded clause predicate *wpuścił* ‘let’ requires an accusative object, and the main clause predicate *ufa* ‘trusts’ requires a dative. The constructions are ungrammatical, independent of which relative pronoun is used.

- (68) *Polish*, Himmelreich (2017: 17), after Citko (2013)
- a. * Jan lubi kogokolwiek/komukolwiek dokucza.
 Jan likes_{acc} whoever.ACC/whoever.DAT teases_{dat}
 ‘Jan likes whoever he teases.’

²³ The generalizations regarding the Polish data originates in Himmelreich (2017).

- b. * Jan ufa kogokolwiek/komukolwiek wpuścił do domu.
 Jan trusts_{dat} whoever.ACC/whoever.DAT let_{acc} to home
 ‘Jan trusts whoever he let into the house.’

The construction in (68a) is ungrammatical in Polish, but its counterparts in German and Gothic are not. Therefore, the derivation has to differ from what is described in Section 4.1 and Section 5.2.1. I suggest that the difference between German and Gothic on the one hand and Polish on the other is that for Polish the restriction ‘Only graft highest node’ holds, whereas it does not for German and Gothic. In the first step of the derivation of (68a), the embedded clause predicate *dokucza* ‘teases’ externally merges with the DatP on the relative pronoun. Next, the relative pronoun is internally remerged to the left edge of the clause. Then, the third step of the derivation cannot take place: the main clause predicate *lubi* ‘likes’ requires accusative case, which is not the highest case node but an embedded one. Remerging an embedded node is not allowed because of the ‘Only graft highest node’ restriction.²⁴

The derivation of (68b) is identical to the one for German described in Section 4.2.2. Just like for German, Polish has the restriction ‘Keep spellout’. In the first step of the derivation, the embedded clause predicate *dokucza* ‘teases’ externally merges with the AccP on the relative pronoun. Next, the relative pronoun is internally remerged to the left edge of the clause. The third step cannot take place. First, there is no dative case node available for the main clause predicate *ufa* ‘trusts’ to externally remerge with. Second, no extra case features can be externally remerged with the highest case feature without changing the spellout of the relative pronoun (‘Keep spellout’).

5.2.3 Greek

Greek is the opposite of German in that non-syncretic non-matching free relatives are grammatical if the case required in the main clause is the more complex one (whereas it is the embedded clause in German). Greek and Ger-

²⁴ As shown in (65c), syncretism saves a case conflict in Polish. At first sight, the restriction ‘Only graft highest node’ seems to rule out the syncretic case when the embedded clause requires a more complex case than the main clause. One possibility concerns the nature of the element that is merged with the predicate of the main clause. The concept that keeps reoccurring in the discussion about syncretic forms is ‘the same lexical entry’. In the case of a syncretism, the selected phonological form in the embedded clause is ambiguous: it can be either the biggest case it corresponds to, or the smaller one. If it would be only the phonological form that is available for remerge in the main clause, there is no way of telling whether it was the bigger case or the smaller case that was required in the embedded clause. In a sense, the syncretic form is able to ‘fool’ the main clause predicate. I leave it for future research to determine whether this option is viable.

man have with each other in common that it is the relative pronoun in the more complex case that has to be used.²⁵

In (69a), *ḑósi* ‘given’ takes a recipient in genitive case in the embedded clause, and *efχarístisan* ‘thanked’ requires nominative case in the main clause. Neither the use of *ópjon* ‘who.GEN.PL’ nor *ópji* ‘who.NOM.PL’ provides a grammatical result (see (69a)). In (69b), the embedded clause predicate *voíθise* ‘helped’ requires a nominative, and the main clause predicate *eḑósa* ‘gave’ requires a genitive recipient. If the relative pronoun appears in genitive case (the most complex and main clause case), the sentence is grammatical. If the relative pronoun appears in nominative case (the less complex and embedded clause case), the sentence is ungrammatical.

- (69) *Greek*, Daskalaki (2011: 79-80)
- a. * Me efχarístisan ópjon/ ópji íχα
CL.1SG.ACC thanked.3PL_{nom} who.GEN.PL/ who.NOM.PL had.1SG_{gen}
ḑósi leftá.
given money
‘Whoever I had given money to, thanked me.’
 - b. Eḑósa leftá ópju/ *ópjos me voíθise.
gave.1SG_{gen} money who.GEN/ who.NOM CL.1SG.ACC helped.3SG_{nom}
‘I gave money to whoever helped me.’

The derivation of (69b) is identical to the one described for Polish in 5.2.2. Greek has the restriction ‘Only graft highest node’, just as Polish (but unlike German and Gothic). The first step of the derivation is that the embedded clause predicate *ḑósi* ‘given’ externally merges with the genitive case node of the relative pronoun. Next, the relative pronoun is internally remerged to the left edge of the clause. In the third step, the main clause predicate *efχarístisan* ‘thanked’ should be externally remerged with the nominative case node. However, the nominative case node is an embedded node, and in Greek it is not permitted to merge with anything but the highest case node. The third step in the derivation cannot take place, and the derivation crashes.

The derivation for (69b) is identical to the derivation of the Gothic example (67a) in 5.2.1. Greek does not have the restriction ‘Keep spellout’, just as Gothic (but unlike German and Polish). The first step in the derivation is that the embedded clause predicate *voíθise* ‘helped’ externally merges with the nominative case node of the relative pronoun. Second, the relative pronoun internally remerges to the left edge of the clause. In the third step, the genitive case node is not available for the main clause predicate *eḑósa* ‘gave’ to be

²⁵ The Greek data and generalizations are taken from Daskalaki (2011).

externally remerged with. Therefore, the highest case node (NomP) merges with the next case feature in the fseq (creating an AccP) and with the next case feature after that (creating a GenP), such that the required case node has been merged. Finally, the main clause predicate *eđósa* ‘gave’ merges with the genitive case node. The spellout of the relative pronoun is overridden in this derivation, just like what happened for Gothic.

To sum up, languages differ in whether non-matching non-syncretic free relative are grammatical. In some languages they are grammatical if the embedded clause requires the more complex case (German and Gothic), if the main clause requires the more complex case Gothic and Greek), or they are never grammatical (Polish). It holds for all languages that in grammatical non-matching non-syncretic free relatives, the relative pronoun in the more complex case is used. In this section I reduced the cross-linguistic differences to two restrictions that may or may not be present in a language. The restrictions are incorporated in the derivational steps that I proposed for German non-matching free relatives.

6 Conclusion

This paper started out with the observation that case mismatches in German free relative constructions are permitted under two types of circumstances. First, if the case required in the embedded clause is more complex than the case in the main clause, and the relative pronoun has the form of the more complex case (cf. Pittner 1991; 1995; Vogel 2001; Grosu 2003). Second, when the form that corresponds to the two different required cases is syncretic (cf. Groos & van Riemsdijk 1981; Zaenen & Karttunen 1984; Dyta 1984; Pullum & Zwicky 1986; Ingria 1990; Dalrymple & Kaplan 2000; Sag 2003). Mismatching free relatives are ungrammatical when the embedded clause predicate requires the less complex case, and there are two distinct relative pronouns for the required cases.

I provided an analysis for these two patterns by taking case containment (Caha 2009) as a point of departure, and combining it with Van Riemsdijk’s (2006a) grafting (or external remerge). This made it possible to make a distinction between more and less complex cases in the syntax. It no longer comes as a surprise that a relative pronoun can satisfy the case requirement of a predicate which is less complex than what the relative pronoun is marked for. Cases are organized in a containment structure and more complex cases by definition contain less complex cases. Therefore, at the point in the derivation that the less complex case is required, the main clause predicate can merge

with the case node contained in the more complex case. When the main clause requires a less complex case than the embedded clause, the required case node is not available. Additional case features can be merged as long as they have the same spellout. This makes the crucial distinction between syncretic and non-syncretic constructions: syncretic forms are grammatical because they have the same spellout, but non-syncretic forms are not grammatical because they do not. This paper adds to the claim that case is internally complex. Moreover, it shows that cases can have two roles at the same time: they fulfill the role of the case they surface in, and the role of a case they contain. In this type of situation, it is always the most complex case that surfaces, as has also been shown in [Caha 2015](#).

In the final section I discussed different grammaticality patterns that are observed in non-matching non-syncretic free relatives across languages. It differs per language whether they allow for the embedded clause predicate or the main clause predicate to require the more complex case. It holds for all languages that the relative pronoun has to appear in more complex case. I reduced these cross-linguistic differences to two restrictions that may or may not hold for the language. The first restriction, ‘Only graft highest node’, is whether the main clause is permitted to graft an embedded case feature or only the structurally highest case node. The second restriction, ‘Keep spellout’ is whether the spellout of the relative pronoun can be overridden or not. The grammaticality patterns of German, Polish, Gothic and Greek follow from these four derivational steps, taking the two restrictions into account.

- (70) Derivational steps for non-matching free relatives
- a. Externally merge the embedded clause predicate with the required case node
 - b. Internally remerge the relative pronoun to the left edge of the clause
 - c. Externally remerge (graft) the main clause predicate with the required case node
 - *Restriction ‘Only graft highest node’ for Polish and Greek: only the structurally highest case node can be grafted*
 - (i) if the required case node is not available, externally remerge (graft) the highest case feature with the next case feature in the fseq, and externally merge the main clause predicate with the required case node
 - (ii) if the required case node is not available, continue to externally merge the additional case features following the fseq

- (until the required case node has merged), and externally merge the main clause predicate with the required case node
- *Restriction ‘Keep spellout’ for German and Polish: only case features within the boundaries of the inserted lexical entry can be merged*

Future research should identify other constructions with similar mechanisms in which the same restrictions surface, which could help understand the underlying motivations behind the identified restrictions.

Abbreviations

1, 2, 3 = 1, 2, 3 person, NOM = nominative, ACC = accusative, GEN = genitive, DAT = dative, N = neuter, M = masculine, SG = singular, PL = plural, CL = clitic

References

- Asarina, Alevtina A. 2011. *Case in Uyghur and beyond*. Massachusetts: Massachusetts Institute of Technology dissertation.
- Audring, Jenny. 2009. *Reinventing pronoun gender*. Utrecht: Netherlands Graduate School of Linguistics dissertation.
- Baunaz, Lena & Eric Lander. 2018a. Deconstructing categories syncretic with the nominal complementizer. *Glossa: a journal of general linguistics* 3(1). <https://doi.org/10.5334/gjgl.349>.
- Baunaz, Lena & Eric Lander. 2018b. Nanosyntax: The basics. In Lena Baunaz, Karen De Clercq, Liliane Haegeman & Eric Lander (eds.), *Exploring nanosyntax*, 3–56. Oxford: Oxford University Press. <https://doi.org/10.1093/oso/9780190876746.001.0001>.
- Blake, Barry. 1994. *Case*. Cambridge: Cambridge University Press. <https://doi.org/10.1017/cbo9781139164894>.
- Boef, Eefje. 2013. *Doubling in relative clauses: Aspects of morphosyntactic microvariation in dutch*. Netherlands Graduate School of Linguistics.
- Bresnan, Joan & Jane Grimshaw. 1978. The syntax of free relatives in English. *Linguistic Inquiry* 9(2). 331–391.
- Caha, Pavel. 2009. *The nanosyntax of case*. Tromsø: University of Tromsø dissertation.
- Caha, Pavel. 2014. Funky genitives.

- Caha, Pavel. 2015. Czech numerals and no bundling. In Ur Shlonsky (ed.), *Beyond functional sequence: The cartography of syntactic structures, volume 10*, Oxford University Press. <https://doi.org/10.1093/acprof:oso/9780190210588.001.0001>.
- Caha, Pavel, Karen De Clercq & Guido Vanden Wyngaerd. 2018. The fine structure of the comparative. *Studia Linguistica* <https://doi.org/10.1111/stul.12107>.
- Cardinaletti, Anna. 1994. On the internal structure of pronominal DPs. *The Linguistic Review* 11(3-4). 195–220. <https://doi.org/10.1515/tlir.1994.11.3-4.195>.
- Citko, Barbara. 2013. Size matters: multidominance and DP structure in Polish. *44th Poznań Linguistic Meeting*.
- Dalrymple, Mary & Ronald M. Kaplan. 2000. Feature indeterminacy and feature resolution. *Language* 76(4). 759–798. <https://doi.org/10.2307/417199>.
- Daskalaki, Evangelia. 2011. Case mis-matching as kase stranding. In Lauren A. Friedman (ed.), *University of Pennsylvania Working Papers in Linguistics*, vol. 17. 77–86. Philadelphia: Penn Linguistics Club.
- De Clercq, Karen. 2019. French negation, the superset principle and feature conservation. In *Cycles in language change*, Oxford University Press.
- Dyta, Stefan. 1984. Across-the-board dependencies and case in Polish. *Linguistic inquiry* 15(4). 701–705.
- Fábregas, Antonio. 2007. The exhaustive lexicalisation principle. *Nordlyd* 34(2). <https://doi.org/10.7557/12.110>.
- Gippert, Jost. 1987. Zu den sekundären Kasusaffixen des Tocharischen. *Tocharian and Indo-European Studies* 1. 22–39.
- Groos, Anneke & Henk van Riemsdijk. 1981. Matching effects in free relatives: A parameter of core grammar. In Luciana Brandi Adriana Belletti & Luigi Rizzi (eds.), *Theory of markedness in generative grammar*. Pisa: Scuola Normale Superiore.
- Grosu, Alexander. 2003. A unified theory of ‘standard’ and ‘transparent’ free relatives. *Natural Language and Linguistic Theory* 21(2). 247–331. <https://doi.org/10.1075/la.55.07gro>.
- Hachem, Mirjam. 2015. *Multifunctionality. The Internal and External Syntax of d- and w-Items in German and Dutch*. Utrecht: Utrecht University dissertation.
- Halle, Morris & Alec Marantz. 1993. Distributed morphology and the pieces of inflection. In Ken Hale & Jay Keyser (eds.), *The view from building 20*, 111–176. Cambridge, Massachusetts: MIT Press.
- Harbert, Wayne. 1982. On the nature of the matching parameter. *The Linguistic Review* 2(3). 237–284.

- Harbert, Wayne Eugene. 1978. *Gothic syntax: a relational grammar*. Urbana-Champaign: University of Illinois dissertation.
- Himmelreich, Anke. 2017. *Case matching effects in free relatives and parasitic gaps: A study on the properties of agree*. Leipzig: Universität Leipzig dissertation.
- Ingria, Robert. 1990. The limits of unification. In *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics*. 194–204. <https://doi.org/10.3115/981823.981848>.
- Izvorski, Roumyana. 1997. Subject free relatives in null-subject languages: evidence from Slavic. In *Formal Approaches to Slavic Linguistics: The Cornell Meeting*, 267–288. Ann Arbor: Michigan Slavic Publications.
- Koptjevskaja-Tamm, Maria. 2004. Mass and collection. In Geert Booij, Christian Lehmann & Joachim Mugdan (eds.), *Morphology: A handbook of inflection and word formation*, 1016–1031. Berlin: de Gruyter.
- Leu, Thomas. 2008. *The Internal Syntax of Determiners*. New York: New York University PhD Thesis.
- Levy, Roger & Carl Pollard. 2002. Coordination and Neutralization in HPSG. In *The Proceedings of the 8th International Head-Driven Phrase Structure Grammar Conference*. 221–234. CSLI Publications.
- Pittner, Karin. 1991. Freie Relativsätze und die Kasus-hierarchie. *Neue Fragen der Linguistik* 1. 341–347. <https://doi.org/10.1515/9783111353180.341>.
- Pittner, Karin. 1995. The case of German relatives. *The linguistic review* 12(3). 197–231. <https://doi.org/10.1515/tlir.1995.12.3.197>.
- Pullum, Geoffrey & Arnold Zwicky. 1986. Phonological resolution of syntactic feature conflict. *Language* 62. 751–773. <https://doi.org/10.2307/415171>.
- Rooth, Maths. 1992. A Theory of Focus Interpretation. *Natural Language Semantics* 1(1). 76–116. <https://doi.org/10.1007/bf02342617>.
- Rooth, Mats. 1985. *Association with Focus*. Amherst: University of Massachusetts PhD Thesis.
- Sag, Ivan A. 2003. Coordination and underspecification. In *Proceedings of the 9th HPSG conference*. 267–291. Stanford: Stanford University.
- Smith, Peter, Beata Moskal, Ting Xu, Jungmin Kang & Jonathan David Bobaljik. 2018. Case and number suppletion in pronouns. *Natural Language & Linguistic Theory* 1–73. <https://doi.org/10.1007/s11049-018-9425-0>.
- Starke, Michal. 2009. Nanosyntax: A short primer to a new approach to language. *Nordlyd* 36. 1–6.
- Starke, Michal. 2017. Resolving (DAT= ACC) \neq GEN. *Glossa: a journal of general linguistics* 2(1). <https://doi.org/10.5334/gjgl.408>.
- Starke, Michal. 2018. Complex left branches, spellout, and prefixes. In Lena Baunaz, Karen De Clercq, Liliane Haegeman & Eric Lander (eds.), *Ex-*

- ploring nanosyntax*, 239–249. Oxford: Oxford University Press. <https://doi.org/10.1093/oso/9780190876746.003.0009>.
- Van Riemsdijk, Henk. 2006a. Free relatives. In Martin Everaert & Henk van Riemsdijk (eds.), *The Blackwell Companion to Syntax* (2), 338–382. Oxford: Blackwell Publishing. <https://doi.org/10.1002/9780470996591.ch27>.
- Van Riemsdijk, Henk. 2006b. Grafts follow from merge. In Mara Frascarelli (ed.), *Phases of interpretation*, 17–44. Berlin: Mouton de Gruyter. <https://doi.org/10.1515/9783110197723.2.17>.
- Vogel, Ralf. 2001. Case conflict in German free relative constructions: An optimality theoretic treatment. In Gereon Müller & Wolfgang Sternefeld (eds.), *Competition in syntax*, 341–375. Berlin: Mouton de Gruyter. <https://doi.org/10.1515/9783110829068.341>.
- Weerman, Fred & Jacqueline Evers-Vermeul. 2002. Pronouns and case. *Lingua* 112. 301–338. [https://doi.org/10.1016/s0024-3841\(01\)00049-3](https://doi.org/10.1016/s0024-3841(01)00049-3).
- Zaenen, Annie & Lauri Karttunen. 1984. Morphological non-distinctiveness and coordination. In Gloria Alvarez, Belinda Brodie & Terry McCoy (eds.), *Proceedings of the First Eastern States Conference on Linguistics*. 309–320. Somerville, Massachusetts: Cascadilla Press.