Free Relatives, Feature Recycling, and Reprojection in Minimalist Grammars

Richard Stockwell(✉)

University of California, Los Angeles, USA
rstockwell15@ucla.edu

Abstract. This paper considers how to derive free relatives—e.g. *John eats [$_{DP}$ what Mary eats]*—in Minimalist Grammars. Free relatives are string-identical to indirect questions—e.g. *John wonders [$_{CP}$ what Mary eats]*. An analysis of free relatives as nominalised indirect questions is easy to implement, but empirical evidence points instead to wh-words 'reprojecting' in free relatives. Implementing a reprojection analysis in Minimalist Grammars requires innovations to revise the stipulation that the probe always projects the head, and to allow features to be reused non-consecutively.

Keywords: Free relatives · Matching effects · Minimalist Grammars · Reprojection · Resource sensitivity

1 Introduction

This paper considers how to derive free relatives (FRs) (1) in Minimalist Grammars (MG) [24, 26]. Section 2 illustrates MG with an analysis of indirect questions (IQs) (2), which are string-identical to FRs. An analysis of FRs as nominalised IQs is easy to implement, but the evidence presented in Sect. 3 points instead to wh-words 'reprojecting' in FRs [12]. In order to implement a reprojection analysis of FRs, I propose two innovations to MG in Sect. 4: one, a Reproject operation, revises the stipulation that the probe always projects the head; while a second, feature recycling, allows for features to be reused non-consecutively. I explore these innovations in Sects. 5 and 6 before concluding in Sect. 7.

(1) John eats [$_{\text{DP}}$ what he eats].

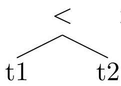
(2) John wonders [$_{\text{CP}}$ what Mary eats].

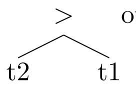
2 Minimalist Grammars, Indirect Questions, and Free Relatives

An MG analysis specifies a lexicon, pairing words with ordered lists of syntactic features. Matches between the first elements in these lists license applications of

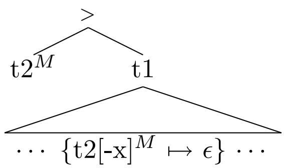
J. Sikos and E. Pacuit (Eds.): ESSLLI 2018, LNCS 11667, pp. 157–170, 2019.
https://doi.org/10.1007/978-3-662-59620-3_10

the structure building operations Merge and Move. We write *t[f]* when the head of a tree—found by following the headedness arrows > and < down to a leaf node—has a sequence of syntactic features whose first element is *f*, and *t* for that tree with feature *f* erased. Merge (3) is licensed by matching category X and selector $=X$ features on the head of a pair of trees *t1* and *t2*. If the selector *t1* is lexical, it is linearized to the left < and *t2* is called the complement; otherwise *t1* is linearized to the right > and *t2* is called the specifier. Move (4) is licensed by matching probe $+x$ and goal $\text{-}x$ features on a tree *t1* containing a subtree *t2*. The probe *t1* takes as a specifier the maximal projection of *t2*, $t2^M$, which is made phonetically null in its original position.[1] The matching features that license Merge and Move are 'checked' and deleted.

(3) Merge (t1[=X], t2[X]) = 

if t1 is lexical, 

otherwise.

(4) Move (t1[+x]) = 

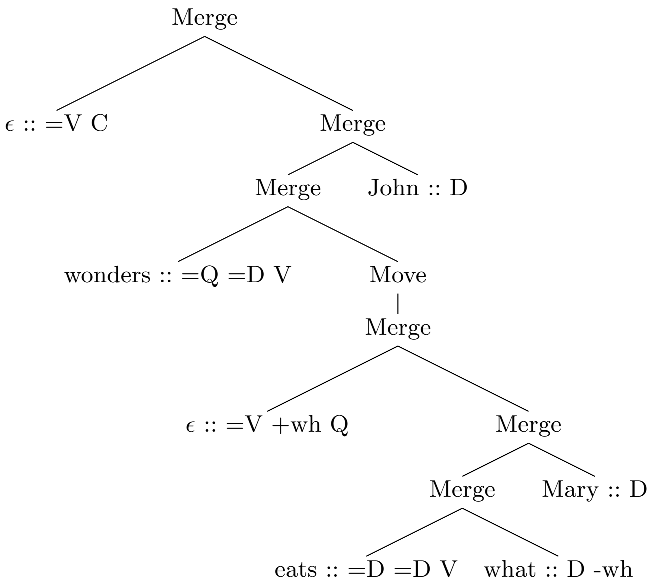
Two aspects of these definitions will be especially relevant for the analysis of FRs. First, projection: the selector in Merge and the probe in Move *t1* projects the head, pointed to by > and <, whose remaining features drive further structure building. Second, resource sensitivity: the matching features that license Merge and Move are checked and deleted, so are not available to drive any further structure building. While the second is a matter of technical implementation, the first is more central to Minimalist [8] reasoning in enforcing the endocentricity requirement of X-bar theory [6,18] and bare phrase structure [7]: each phrase has a head, one lexical item inside it that determines its distribution. However, MG Merge and Move are especially strict in stipulating the selector as the projector—projecting the selectee would still be endocentric.

To illustrate MG, an uncontroversial analysis of indirect questions (IQs) (2) can be implemented with the lexicon in (5), along with derivation (6) and derived (7) trees.

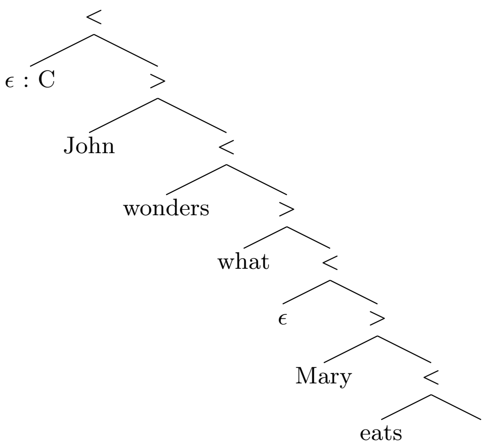
(5)	John :: D	Mary :: D	wonders :: =Q =D V	eats :: =D =D V
	ϵ :: =V C	ϵ :: =V +wh Q	what :: D -wh	

[1] Move is also subject to the shortest move constraint, which will not concern us here.

(6)



(7)



In two Merge steps, *eats* checks its *=D* selector features against the category *D* features of *what*, then *Mary*. Omitting consideration of the tense layer, the phonologically null question complementiser $\epsilon ::= V + wh\, Q$ merges next. Move checks *+wh* against *-wh* to complete the indirect question, whose category Q is what *wonder* selects for. Construction of the main clause ends with the null complementiser $\epsilon ::= V\, C$ of the start category C.

The lexicon in (5) cannot derive free relatives (FRs) (1). Substituting *eats* for *wonder* does not converge, since *eats* selects for a complement of category D, not Q. A simple solution supplements the lexicon with $\epsilon ::= Q\, D$—a null D that selects a Q complement. Merge of $\epsilon ::= Q\, D$ with the output of Move in (6) converts the IQ from category Q to D, which *eats* can then take as its complement. Several versions of this null head analysis of FRs have been proposed, e.g. [15,16]. However, as shown in the next section, there is a good deal of evidence against it.

3 Dual Role of Wh-Words in Free Relatives

MG Merge and Move conform to the general syntactic notion of headedness [6,7,18]. On the null head analysis, the derivation of a FR proceeds via an IQ of category Q. Projecting Q and merging $\epsilon ::= Q\ D$ seals off the wh-word inside the IQ, preventing it from informing the rest of the derivation; instead the null head will determine the FR's distribution. However, evidence suggests that the wh-word itself is the head of the FR, since the behaviour of a FR is keyed to the wh-word that forms it. This section shows that this is so for category distribution and case matching.

First, FRs distribute with the category of their wh-word (8), cf. [2]. A FR with *what* (1) distributes as a DP; but FRs formed with *where* distribute as PPs rather than DPs (8a), and those formed with *how* as AdvPs in not being able to intervene between a verb and its object (8b). On the null head analysis, this would require null, Q-complement-taking lexical items of many categories, e.g. $\epsilon ::= Q\ P$, $\epsilon ::= Q\ Adv$. And even then, nothing would enforce category matching between the null lexical item and the wh-word inside the FR, as required to rule out (9); in other words, we would expect mixtures like $what :: D\ \text{-}wh$ and $\epsilon ::= Q\ P$ to be grammatical.

(8) a. i. Mary put the book [$_{\text{PP}}$ on the shelf] / [$_{\text{PP}}$ where she keeps it].
ii. *Mary put the book [$_{\text{DP}}$ the shelf] / [$_{\text{DP}}$ what she built].
b. i. John speaks [$_{\text{AdvP}}$ quickly] / [$_{\text{AdvP}}$ how you speak].
ii. *John takes [$_{\text{AdvP}}$ quickly] / [$_{\text{AdvP}}$ how you write letters] notes.

(9) *Mary put the book [$_{\text{PP}}$ $\emptyset_{\text{P}}$ [$_{\text{QP}}$ what$_{\text{D}}$ John built]].

Second, in languages with morphological case, e.g. German [22], the wh-word in a FR must satisfy the case requirements of both the relative and matrix clauses. (10) is grammatical, since the nominative wh-word is the subject of the FR, which is the subject of the sentence. But (11) is ungrammatical due to the competing case requirements placed on the wh-word inside the FR, where it is an accusative object, and the FR as a whole, which is the nominative subject of the sentence. This conflict cannot be resolved—neither the accusative nor the nominative form of the wh-word will do. Since the null head analysis involves two distinct lexical items of category D—*what* and $\epsilon ::= Q\ D$—it offers no explanation for why they should match in case.

(10) [$_{\text{DP}_{\text{NOM}}}$ Wer$_{\text{NOM}}$ nicht stark ist] muss klug sein.
who not strong is must clever be.
'Who is not strong must be clever.'

(11) *[$_{\text{DP}_{\text{NOM}}}$ $\left\{ \begin{matrix} \text{Wen}_{\text{ACC}} \\ \text{Wer}_{\text{NOM}} \end{matrix} \right\}$ Gott schwach geschaffen hat] muss klug sein.
who God weak created has must clever be.
'Who God has created weak must be clever.'

Matches between FRs and their wh-words in distribution and case argue that the moving wh-word serves as the head of FRs, thereby determining their behaviour in the rest of the derivation. A number of researchers have reached this conclusion [2,10], a promising line of analysis being one where the wh-word itself projects the head of FRs [4,5,12].

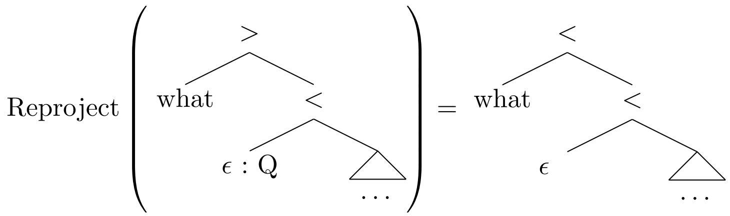
The rest of this paper considers what it would take to modify the MG formalism to allow such a reprojection analysis to be expressed.[2] There are two issues to be overcome. First, while retaining endocentricity, a reprojection analysis directly contradicts the standard stipulation that a probe always projects over the goal it attracts to its specifier [1,9,11,21]—a stipulation that is baked into the definition of Move in (4). Second, even if the wh-word can be made the head, it will lack any features to drive the rest of the derivation. The next section proposes amendments to MG that allow a reprojection analysis of FRs to be implemented.

4 Implementing Reprojection in Minimalist Grammars

This section seeks to implement a reprojection analysis of FRs in MG. It does so by proposing two innovations: (i) Reproject, a new structure-building operation that revises the stipulation that the probe always projects; and (ii) feature recycling, a way for the category feature of the wh-word to be reused in the face of the resource sensitivity of MG. Consequences of these innovations and further directions are explored in Sects. 5 and 6.

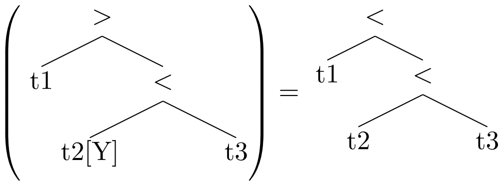
4.1 Reproject

In revising the stipulation that a probe always projects, I propose to add Reproject to MG's inventory of structure building operations. We want Reproject to apply as in (12), reversing headedness to make *what* the head, thereby allowing *what* to determine the future of the derivation; and deleting the category feature Q, which would otherwise be left unchecked and cause the derivation to crash.[3]

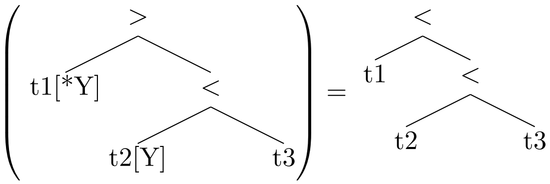
(12)


A general definition of Reproject is given in (13). A unary operation[4] applying to a tree with specifier *t1*, head *t2*, and complement *t3*, Reproject switches headedness to *t1* and deletes the category feature of *t2*, leaving *t3* unchanged.

[2] cf. Reprojecting head movement, e.g. [19], about which I will have nothing to say.
[3] The question of what features are on *what* in (12) is postponed to the next subsection.
[4] cf. [20] for a unary HPSG schema for free relatives in German.

(13) Reproject 

However, the way Y is checked without matching another feature in (13) would make Reproject very different from Merge and Move, which symmetrically check pairs of matching features.[5] Reproject is defined symmetrically in (14), where it applies to a tree where a reprojection feature **Y* on the specifier *t1* matches the category of the head *t2*. Both features are checked, and headedness switches to *t1*. Using (14) means adding reprojection features **Y* to the inventory of syntactic features, and FR-specific reprojecting versions of wh-words to the lexicon; e.g. *what* :: *D -wh *Q*, *where* :: *P -wh *Q*.[6]

(14) Reproject 

However, as things stand the outcome of (14) has no features.[7] *t1* is the head, but all its features have been checked en route to it becoming the specifier of *t2*. In deriving the FR in (1), *what* :: *D -wh *Q* has its *D* checked by Merge with *eats*, *-wh* by Move, and **Q* by Reproject, rendering its feature list empty, i.e. *what : ϵ*. The next subsection proposes a way for category features to be reused so that the wh-word can serve as the head of FRs after Reproject.

4.2 Feature Recycling

After Reproject the wh-word is the head, ready to determine how the derivation will proceed. In order to account for the matching effects observed in Sect. 3, we would like *what* to play a dual role in deriving (1) by contributing its category feature twice: first as complement to *eats*; then again after Move and Reproject to categorize the FR. However, MG structure building is resource sensitive: the matching features that license Merge and Move are checked and deleted. *D* of *what* :: *D -wh *Q* is expended in Merge with *eats*, and is subsequently unavailable.

---

5 Even with persistent features [25], as discussed in the next subsection, while checking is not necessarily symmetric, structure building is still licensed by pairs of matching features.

6 Wh-clustering [14]—see Sect. 5.3 below—provides a precedent for Reproject in being triggered by a feature on a specifier rather than a head. Clustering also involves complex specifiers, whereas I restrict attention here to trees with exactly one specifier.

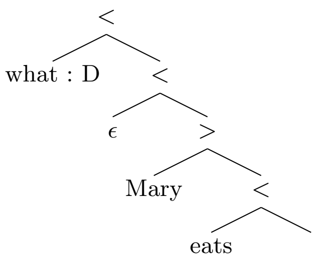
7 Recall from the illustration of MG in Sect. 2 that in order to converge, the derivation must reach the start category *C*.

Endowing *what* with a second D feature ordered after *-wh* and **Q*, i.e. *what* :: D *-wh *Q* D, (cf. *where* :: P *-wh *Q* P) invites the same empirical challenges as the null head analysis: with two separate category features, nothing enforces category and case matching between FRs and their wh-words. Instead, we would like one and the same D feature to contribute twice to the derivation.

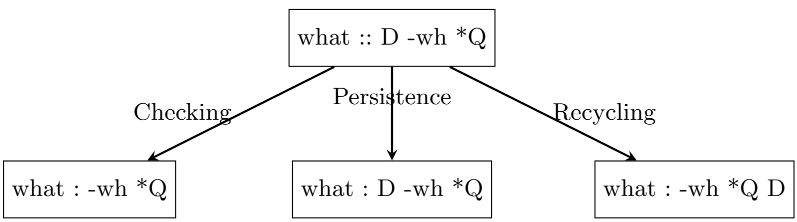
Persistent features are an existing innovation that allow category features to be used multiple times [25]. Merge continues to be licensed symmetrically by matching features, but persistent features (underlined $\underline{F}$) do not have to delete. Persistent features were motivated for implementing the movement theory of control [17], allowing the same $\underline{D}$ to occupy multiple argument positions by satisfying multiple $=D$ features. However, persistence in *what* :: $\underline{D}$ *-wh *Q* does not help in deriving FRs, since the two desired uses of D are non-consecutive. Move and Reproject must apply after Merge of *what* with *eats* but before *what* categorizes the FR. Hence $\underline{D}$ would have to delete to allow Move to be triggered by *-wh* before having the chance to provide the category of the FR.

I therefore propose feature recycling. Beyond persisting at the head of a list, features can live on in the derivation by cycling to the end of the list after licensing Merge. Feature recycling allows the D of *what* :: D *-wh *Q* to play a dual but non-consecutive role, as shown in the derived tree of the FR from (1) in (15). After licensing Merge with *eats*, D cycles to the end of the feature list; and after Move checks *-wh* and Reproject checks **Q*, the recycled D is back at the head of *what*'s feature list to serve as the head of the FR. The diagram in (16) summarizes the differences between standard resource sensitive feature checking, persistent features, and feature recycling with respect to *what*'s D feature.

(15)



(16)



Thus we have implemented a reprojection analysis of FRs, cf. [4,5,12]. However, the analysis has come at the cost of two innovations. The first, Reproject, reverses headedness to the wh-word and deletes the category feature of the embedded clause, which would otherwise be left unchecked and cause a crash.

The output of Reproject would lack any features were it not for the second innovation, feature recycling, which provides a way for the wh-word's category feature to be reused non-consecutively as the head of the FR. I explore these innovations further in the next two sections.

5 More on Reproject

This section considers the Reproject operation in greater detail, with discussion of the location of the triggering feature, Reproject's relationship to Move, and multiple-wh FRs.

5.1 Wh-Word Trigger

This subsection attempts to justify making the wh-word the trigger for Reproject. Whereas we could have put the triggering feature *Y* on the head that is reprojected over, the definition of Reproject in (14) has the triggering feature on the wh-word, e.g. $what :: D$ *-wh *Q*. The argument is based on the 'restrictor restriction' on FRs: wh-words can only form FRs if they lack a complement restrictor, as shown by the ungrammaticlity of (17).

(17) *John eats [$_{\mathrm{DP}}$ what food Mary eats].

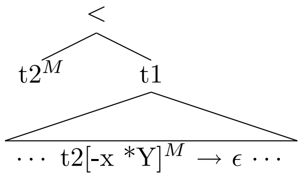
The restrictor restriction is much easier to state if the wh-word is the trigger for Reproject. We can exclude from the lexicon wh-words with both a selector feature and a Reproject feature, e.g. **what* $:: =N\ D$ *-wh *Q*. By the time $\epsilon ::\ +wh\,Q$ interacts with the wh-word in the Move step, on the other hand, it would be unable to discriminate between a wh-word with or without a restrictor; in either case the wh-word will now have *-wh* as its first feature, $=N$ having long since been checked. Since the relevant information to distinguish between good FRs and (17) is not available to $\epsilon ::\ +wh\,Q$, the restrictor restriction on FRs argues that the wh-word should trigger Reproject.

However, this conclusion is provisional. Much stronger evidence would be cases of wh-words reprojecting over lexical items other than $\epsilon ::\ +wh\,Q$. Only then could we be sure that the wh-word is the trigger for Reproject, rather than the head reprojected over.

5.2 Reprojecting Move

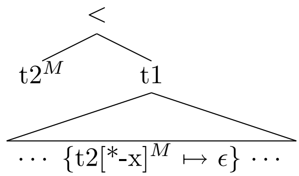
Reproject as in (14) involves the heads $what :: D$ *-wh *Q* and $\epsilon ::\ =V\ +wh\,Q$. It is licensed symmetrically by matching between the reprojection feature **Q* and category feature *Q*. But the two heads match in more than *Q*—they also match for *wh*. In Sect. 4.1, *wh* was checked by an application of Move before Reproject. But nothing said there enforces the co-occurrence in a lexical item's feature list of a Move licensee like *-wh* and a Reproject trigger like **Q*. If Reproject is always fed by Move, we would be missing a generalization. Instead, we could recast Reproject as a version of Move, as in (18).

(18) Reprojecting Move (t1[+x Y]) =



The definition in (18) enforces a dependency between Move licensees and Reproject triggers. Continuing to assume that the wh-word is the trigger for Reproject, we could further strengthen the dependency between moving and reprojecting by collapsing Move licensing and Reproject triggering into a single feature, **-x*. The reprojecting lexical item *what* :: *D *-wh* would then trigger Reprojecting Move, as defined in (19).

(19) Reprojecting Move (t1[+x Y]) =



However, the definition in (19), reintroduces the asymmetry problem from our first definition of Reproject in (13). The category feature *Y* on *t1* is asymmetrically checked, deleting without having matched with another feature. Beyond this technical point, I cannot see how to decide between (18) and (19) as the definition of Reprojecting Move.

More generally, Reprojecting Move raises problems regardless of which of (18) or (19) we choose. For one, it increases the size of the 'moving window' on feature lists from one to two. Whereas Merge (3) and Move (4) apply based only on the first feature of the head, Reprojecting Move requires sight of the first two. A second problem might be that we have to restate all the restrictions on Move for Reprojecting Move, like the shortest move constraint and islands. Thus while reducing redundancy in enforcing a dependency between Move licensees and Reproject triggers, Reprojecting Move might increase redundancy elsewhere. However, redundancy only arises to the extent that movement as it feeds reprojection is subject to the same constraints as ordinary movement. If empirical investigation finds it to be subject to different constraints, we would have a strong argument for Reprojecting Move as its own operation. As with the previous subsection, we end by wondering whether wh-words reproject over lexical items beyond $\epsilon :: +wh\, Q$.

5.3 Clustering and Reproject

Languages with overt multiple-wh movement to Spec, CP—like Bulgarian and Romanian (20) [3]—also allow multiple-wh FRs [23]:

(20) Ti-am dat [ce unde când a trebuit instalat].
CL2-have.1SG given what where when has needed installed
'I have given you the things that needed to be installed in the appropriate place at the appropriate time.'

Adopting the MG analysis of multiple wh-movement, or clustering, in [14], only the topmost wh-word participates in a Move operation; the rest participate in Cluster. Since multiple-wh FRs distribute with the category of the topmost wh-word—*D* in (20)—the crucial dependency is between Reproject triggers and Move licensees, not Cluster licensees.[8]

6 More on Reusing Features

Section 3 emphasised matches between FRs and their wh-words in order to argue that the moving wh-word projects the head of FRs. Section 4.2 proposed feature recycling as a way for the wh-word's category feature to be reused to implement a reprojection analysis of FRs. This section explores ways in which a FR and the wh-word that forms it can behave differently—if only very slightly. Slight differences regarding case syncretism, complement/adjunct *where* FRs, and A-bar features suggest that the notion we need may not be recycling but refreshing, returning to the lexicon to pick another list of features compatible with the morphological form of the word.

6.1 Case Syncretism

This subsection considers the case matching facts of Sect. 3 in greater detail. In MG, abstract case licensing is implemented by k(ase) features, with all lexical items of category *D* also bearing *-k*. The ungrammaticality of (21) shows that *-k* must recycle along with *D* in deriving FRs, since the FR as a whole must be in a case position.

(21) *It seems [$_{\mathrm{DP}}$ what John eats] to be nice.

English wh-words do not differ morphologically for case,[9] which might suggest a generic *-k* feature for English rather than more articulated *-nom*, *-acc*, etc. Support for generic *-k* comes from the lack of case matching effects in English FRs. (22) is grammatical, despite the wh-word being assigned accusative internal to the FR, while the FR is nominative in the sentence overall. This contrasts with the German mismatch from Sect. 3, repeated here as (23).

[8] Working with the definitions of Reproject in (14) and Cluster in [14], the lexical entries for the wh-words in (20) would be *ce* :: $D \bigtriangledown wh$ *-wh* **Q*, *unde* :: $P \bigtriangledown wh \bigtriangleup wh$ and *când* :: $P \bigtriangleup wh$. The reprojection trigger **Q* would co-occur with *-wh*, not the Cluster licensor $\bigtriangledown wh$ or licensee $\bigtriangleup wh$.

[9] I set aside *whom* as an archaism.

(22) [$_{\text{DP}_{\text{NOM}}}$ What$_{\text{ACC}}$ John ate] killed him.

(23) *[$_{\text{DP}_{\text{NOM}}}$ $\left\{ \begin{smallmatrix} \text{Wen}_{\text{ACC}} \\ \text{Wer}_{\text{NOM}} \end{smallmatrix} \right\}$ Gott schwach geschaffen hat] muss klug sein.
who God weak created has must clever be.

'Who God has created weak must be clever.'

(24) [$_{\text{DP}_{\text{NOM}}}$ Wer$_{\text{NOM}}$ nicht stark ist] muss klug sein.
who not strong is must clever be.

'Who is not strong must be clever.'

For English, then, we can say that generic *-k* of *what* :: *D -k -wh *Q* is licensed inside the FR by a case-assigner, recycled along with *D*, and licensed again by another case-assigner in the main clause. In German, on the other hand, the morphological differences among wh-words for case might suggest lexical items of category *D* differ among *-nom*, *-acc*, etc. In (24), *wer* :: *D -nom -wh *Q* is licensed for nominative inside the FR, recycled with *D* in forming the FR, and licensed again for nominative in the main clause.

Switching between *-acc* and *-nom* is ungrammatical in (23); but this is not always the case in German for a FR configuration that mixes nominative and accusative. Such mismatches are possible when there is case syncretism, as in (25) [22], where neuter gender *was* can realise either nominative or accusative.

(25) [$_{\text{DP}_{\text{NOM}}}$ Was$_{\text{ACC}}$ du gekocht hast] ist schimmlig.
What you cooked have is moldy.

'What you have cooked is moldy.'

Starting with *was* :: *D -acc -wh *Q* inside the FR, we cannot switch to *was* :: *D -nom -wh *Q* in the main clause via feature recycling, incorrectly predicting (25) to be bad. We could salvage feature recycling by changing our assumptions about case features, claiming that there is only one lexical item *was* :: *D -nomacc -wh *Q* whose underspecified *-nomacc* case feature can be checked by either a nominative or accusative case-assigner.

Alternatively, we could account for (25) by refreshing rather than recycling the features of *was*. After moving to specifier position and reprojecting, *was* :: *D -acc -wh *Q* has exhasted its list of features. Rather than pre-empting this problem with feature recycling, *was* : ϵ could refresh its features by reaching back into the lexicon for a list of features compatible with its morphological form. This reuse of the string rather than its features allows for the refreshed feature list to be slightly different—including *-nom* rather than *-acc* in deriving (25).[10]

While either underspecified *-nomacc* or feature refreshing would account equally well for German case syncretism, refreshing appears to be the only plausible option for the topic of the next subsection.

[10] The length of the reused string is finitely bounded, in that only lexical items and not phrases can be reused – recall the restrictor restriction on FRs from Sect 5.1.

6.2 Complement vs. Adjunct *Where* Free Relatives

The previous subsection showed that syncretism allows wh-words and the FRs they form to differ in case. This section argues that sycretism also allows differences in category.

Following a prominent analysis of adjunction in MG [13], PPs have very different categories depending on whether they appear in complement or adjunct position: *where* :: P *-wh* is a complement to verbs like *put* :: $=P\ =D\ =D\ V$, whereas *where* :: $\approx V$ *-wh* adjoins to category V, which continues to be the head. In (26), *where* is an adjunct to *eats* inside the FR, while the FR as a whole is a complement to *put*. Thus *where* has different category features internal and external to the FR, which would not follow from feature recycling.

(26) Mary put the book $[_{\mathrm{PP}_{\mathrm{COMP}}}$ where$_{\mathrm{PP}_{\mathrm{ADJ}}}$ John eats].

It is difficult to countenance an underspecification analysis among two different feature lists for *where* along the lines of underspecified *-nomacc* in the previous subsection. That leaves us with feature refreshing: in deriving (26), *where* :: $\approx V$ *-wh* is exhausted to *where : ϵ* in deriving the FR, before refreshing as *where* :: P *-wh* for the main clause.

6.3 A-bar Features

However feature reuse is to be characterised – whether features are recycled or refreshed – A-bar features are not reused. Despite being headed by a wh-word, a FR cannot itself undergo wh-movement, as in (27).

(27) *$[_{\mathrm{DP}}$ What John eats] does Mary eat t?

Unlike category and case features, which play a dual role in deriving FRs, *-wh* is definitively consumed in moving *what* inside the FR, so would have to be barred from recycling. In terms of refreshing, meanwhile, we could say that features are refreshed based on the non-wh part of the word, assuming decomposition of e.g. German *wer* into wh *w-* + *-er* nominative D.

Yet FRs can embark on other A-bar movements, e.g. topicalisation in (28).

(28) $[_{\mathrm{DP}}$ What John eats], I eat t.

Still, the movement in (28) cannot result from reusing a feature. Assuming topicalisation is licensed by *-top*, it must be added to the FR after it has been fully formed: while the FR as a whole is topicalised in (28), the wh-word does not undergo topicalisation inside the FR, so *-top* cannot have been present on *what* at the start of the derivation. The opposite behaviour of *-wh* and *-top* in being active only internal vs. external to the FR tracks the difference between intrinsic vs. optional features [8, p. 231].

7 Conclusion

This paper set out to derive FRs in MG. Reviewing category and case matching effects motivated implementing a reprojection analysis. Doing so came at the cost of two innovations. Reproject, a new structure-building operation, revised the stipulation that the probe always projects. Answers to the technical questions of whether the trigger is the wh-word, and whether Reproject is a special case of Move, depend on the empirical question of whether wh-words reproject over lexical items other than $\epsilon :: +wh\, Q$. The second innovation—feature recycling—provided a way for the category feature of the wh-word to be reused nonconsecutively as the head of the FR in the face of the resource sensitivity of MG. The slight relaxation of matching effects where there is syncretism suggested features might be refreshed rather than recycled, though A-bar features cannot be reused.

References

1. Boeckx, C.: Bare Syntax. Oxford University Press, Oxford (2008)
2. Bresnan, J., Grimshaw, J.: The syntax of free relatives in English. Linguist. Inq. **9**(3), 331–91 (1978)
3. Caponigro, I., Fălăuş, A.: The functional nature of multiple wh-free relative clauses in Romanian. Poster Presented at SALT 28, MIT (2018)
4. Cecchetto, C., Donati, C.: Relabeling heads: a unified account for relativization structures. Linguist. Inq. **42**, 519–560 (2011)
5. Cecchetto, C., Donati, C.: (Re)labeling. MIT Press, Cambridge (2015)
6. Chomsky, N.: Remarks on nominalization. In: Jacobs, RA., Rosenbaum, D.H. (eds.) Reading in English Transformational Grammar, pp. 184–221. Ginn, Waltham (1970)
7. Chomsky, N.: Bare phrase structure. In: Webelhuth, G. (ed.) Government and Binding Theory and the Minimalist Program, pp. 385–439. Blackwell, Oxford (1995)
8. Chomsky, N.: The Minimalist Program. MIT Press, Cambridge (1995)
9. Chomsky, N:. On phases. In: Foundational Issues in Linguistic Theory: Essays in Honor of Jean-Roger Vergnaud, pp. 133–166 (2008)
10. Citko, B.: Missing labels. Lingua **118**(7), 907–944 (2008)
11. Collins, C.: Eliminating labels. In: Epstein, S., Seely, T. (eds.) Derivation and Explanation in the Minimalist Program, pp. 45–61. Blackwell, Oxford (2002)
12. Donati, C.: On wh-head movement. In: Wh-Movement: Moving on, pp. 21–46 (2006)
13. Frey, W., Gärtner, H.M.: On the treatment of scrambling and adjunction in minimalist grammars. In: 2002 Proceedings Formal Grammar (2002)
14. Gärtner, H.M., Michaelis, J.: On the treatment of multiple-wh-interrogatives in minimalist grammars. In: Language and Logos, pp. 339–366 (2010)
15. Groos, A., Van Riemsdijk, H.: Matching effects in free relatives: a parameter of the core grammar. In: Theory of Markedness in Generative Grammar, pp. 171–216 (1981)
16. Grosu, A.: Three Studies in Locality and Case. Routledge, London (1994)
17. Hornstein, N.: Movement and control. Linguist. Inq. **30**(1), 69–96 (1999)

18. Jackendoff, R.: X-bar-Syntax: A Study of Phrase Structure. MIT Press, Cambridge (1977)
19. Koeneman.: The flexible nature of verb movement. Ph.D. dissertation, Universiteit Utrecht (2000)
20. Müller, S.: An HPSG-analysis for free relative clauses in German. Grammars **2**(1), 53 (1999)
21. Pesetsky, D., Torrego, E.: Probes, goals and syntactic categories. In: Otsu, Y. (ed.) Proceedings of the seventh annual Tokyo conference on psycholinguistics, pp. 25–60. Hituzi Syobo, Tokyo (2006)
22. van Riemsdijk, H.: Free Relatives. The Blackwell Companion to Syntax. Blackwell, Oxford (2007)
23. Rudin, C.: Multiple wh-relatives in Slavic. In: Compton, R., Goledzinowska, M., Savchenko, U. (eds.) FASL, pp. 282–307 (2007)
24. Stabler, E.: Derivational minimalism. In: Retoré, C. (ed.) LACL 1996. LNCS, vol. 1328, pp. 68–95. Springer, Heidelberg (1997). https://doi.org/10.1007/BFb0052152
25. Stabler, E.: Sidewards without copying. In: Formal Grammar, vol. 11, pp. 133–146 (2006)
26. Stabler, E.: Computational perspectives on minimalism. In: Boeckx, C. (ed.) Oxford Handbook of Linguistic Minimalism, pp. 617–642 (2011)