AN LFG APPROACH TO NESTED DEPENDENCIES IN DUTCH

Marjolein Poortvliet
University of Oxford

Proceedings of the LFG15 Conference

Miriam Butt and Tracy Holloway King (Editors)

2015

CSLI Publications

http://csli-publications.stanford.edu/

Abstract

This paper argues that, contrary to the prevailing idea that Dutch is a cross-serial dependency language, Dutch also uses nested dependency constructions in embedded clauses. In these nested dependency constructions, the order of the verb cluster is reversed compared to the order of the noun cluster, much like the order of German embedded clauses. I show that the Dutch nested dependency construction is restricted to passive constructions with perception verbs and causative verbs as the main predicate. I propose an LFG treatment for these restricted nested dependency structures.

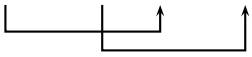
1 Introduction

A unique syntactic phenomenon that is typical of Dutch and German is the serial dependency structure.[1] This structure occurs in embedded clauses, in which all verbs appear clause-finally and are preceded by all the verbs' arguments. Unlike the English embedded clauses, which consist of sequential argument-verb pairs (see (1a)), Dutch and German embedded clauses are such that all nominal material comes first, followed by all verbal material. The sentences in (1) provide a comparison of English (see (1a)), Dutch (see (1b)) and German (see (1c)), in which the arrows are used to show the dependency relations between the arguments and their verbs.

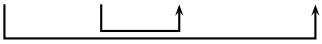
1. (a) ...that Peter saw Marie swim (English)

   (b) *...dat Peter Marie zag zwemmen* (Dutch)
   ...that Peter Marie saw swim

   (c) *...dass Peter Marie schwimmen sah* (German)
   ...that Peter Marie swim saw

Dutch is said to be a cross-serial dependency language, demanding the same internal left-to-right order both in the noun cluster and in the verb cluster. German, on the other hand, is said to be a nested-dependency language, in which the order of the verbs in the verb cluster is reversed compared to the order of the arguments in the noun cluster. Even though these claims about Dutch and German still stand in

[†]For helpful comments and suggestions, I thank Mary Dalrymple, Ash Asudeh, Carryn Yong, the audience at LFG15, and the anonymous reviewers.

[1]The terms 'Dutch' and 'German' are used to denote Standard Dutch and Standard German respectively. Other varieties of these languages are also known to have serial dependency structures. For instance, a variety that has raised interest for its serial dependencies is Swiss German, see Zaenen and Kaplan (1995) and Shieber (1985).

most cases, certain highly constrained constructions in Dutch suggest that Dutch also has nested dependencies. Such a construction is shown in (2), in which the past participle of the verb cluster is not in the expected clause-final position as given in (3), but appears before the main predicate *zag* 'saw'.

2. *...dat Jan een lammetje gevoed zag worden*
   ...that Jan a lamb fed saw to-become
   '...that Jan saw a lamb being fed"

3. **...dat Jan een lammetje zag worden gevoed*
   ...that Jan a lamb saw to-become fed

Serial dependency structures have received much attention in the literature from various perspectives: they have been discussed within syntactic frameworks such as Lexical Functional Grammar (e.g. Bresnan et al. 1982, Zaenen and Kaplan 1995, Kaplan and Zaenen 2003), Head-driven Phrase Structure Grammar (e.g. Rentier 1994 and Hinrichs and Nakazawa 1998) and Tree Adjoining Grammar (e.g. Rambow 1992), as well as from psychological perspectives (e.g. Kaan and Vasic 2004) and from comparative perspectives (e.g. Bach et al. 1986). Furthermore, cross-serial dependencies have played an important role in discussions of language complexity and have functioned as evidence for the argument that natural languages are not context-free (see Shieber 1985 for evidence from Swiss German, and Bresnan et al. 1982 for evidence from Dutch).

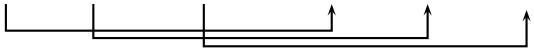
In this paper, I show how nested dependency structures in Dutch can be treated within the framework of Lexical Functional Grammar by making a number of modifications to the rules proposed for crossed dependencies in Dutch by Kaplan and Zaenen (2003) and Zaenen and Kaplan (1995). The outline is as follows: first, previous research on cross-serial dependency constructions will be revisited, followed by an overview of the rules proposed for these constructions. Secondly, I will introduce a passive construction (see (2)) which cannot be accounted for by the rules proposed and which therefore demand modifications to the existing rules. Finally, I show that this construction is not only limited to past participles, but can also include resultative adjectives and resultative PPs.

2 Background

2.1 Crossed Dependencies in Dutch

A cross-serial dependency construction consists of a noun cluster and a verb cluster. The noun cluster is formed by the arguments of the main predicate and the arguments of the complement or complements of the main predicate. The noun cluster is ordered according to the embeddedness of the complements, i.e. N1 N2 N3. This noun cluster is followed by a sentence-final verb cluster, formed by the main predicate and the verbs of the complement. The order of the verb cluster

is also ordered according to the embeddedness of the complements, i.e. V1 V2 V3. This means that both clusters have the same relative order of elements. If we were to draw dependency lines between the arguments and verbs, these lines would cross. Therefore, an overarching term for these constructions is *cross-serial dependency structures*. An often cited example (e.g. Seuren and Kempen 2003, Kaplan and Zaenen 2003, and a slightly modified version in Zaenen and Kaplan 1995) of such a sentence is given in (4), in which the matrix verb *wil* 'want' is linked to the subject *Jan*, followed by the complement consisting of the verb *laten* 'let' linked to its object *Marie*, followed by a third complement consisting of the verb *studeren* 'study' linked to its object *geneeskunde* 'medicine'.

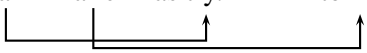
4. *...dat Jan Marie geneeskunde wil laten studeren*
   ...that Jan Marie medicine wants to-let to-study
   
   '...that Jan wants to let Marie study medicine'

There are two types of cross-serial dependency structures in Dutch: *the verb raising construction* as first noted by Evers (1975), and *the third construction* as coined by Besten et al. (1988). A number of verbs in Dutch can appear in both constructions, as is shown for the verb *trachten* 'try' in (5) and (6). In both examples, the infinitive verb *helpen* 'help' from the complement appears to the right of the matrix verbs *heeft trachten* or *heeft getracht* 'has tried', creating the verb cluster V1 V2. Similarly, the argument of the complement, *Marie*, appears to the right of the argument of the matrix verb, creating the noun cluster N1 N2.

5. Verb Raising

   ...dat Jan Marie heeft trachten te helpen
   ...that Jan Marie has try.INF to help
   
   '...that Jan has tried to help Marie'

6. Third Construction

   ...dat Jan Marie heeft getracht te helpen
   ...that Jan Marie has try.PART to help
   
   '...that Jan has tried to help Marie'

There are two differences between the verb raising construction and the third construction. First of all, the auxiliary in the verb raising construction selects an infinitive instead of the expected past participle that is found in the third construction. This phenomenon is called the IPP-effect (*Infinitivus pro Participio*) and only occurs if the verb selected by the auxiliary takes a complement itself. The IPP-effect

only takes place in combination with certain verbs, i.e. causatives, modals, perception verbs, duratives, and is optional for inchoatives and control verbs. Secondly, the verb raising constructions can employ both subject and object control relations, but the third construction can only have a subject control verb.

2.2 Nested Dependencies in German

The phenomenon of serial dependencies is also found in German (see Vogel et al. 1996, Bach et al. 1986, for the third construction in German see Rambow 1992, Hinrichs and Nakazawa 1998), but differs from Dutch in that the order of the verbs within the verb cluster in German is different from the order in Dutch: whereas in Dutch a lower verb follows its matrix verb, in German a lower verb precedes its matrix verb. When dependency lines are drawn between the verbs and their arguments in the German sentence, this shows that the dependencies are nested (nouns ordered N1 N2 N3, while verbs are ordered V3 V2 V1). This type of structure is therefore called a nested dependency structure.

7. *...dass Jan Marie Medizin studieren lassen will*
   ...that Jan Marie medicin to-study to-let to-want

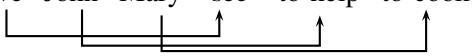
   '...that Jan wants to let Marie study medicin'

2.3 Kaplan and Zaenen's LFG analysis for Dutch

Dutch dependency structures have received a considerable amount of attention within the framework of lexical-functional grammar, and after its first treatment by Bresnan et al. (1982) further research was carried out by Zaenen and Kaplan (1995) and Kaplan and Zaenen (2003). This section provides an overview of the LFG treatment and the rules proposed for cross-serial dependencies in Dutch (leaving the third construction out of the discussion), and is mainly based on the most recent work by Kaplan and Zaenen (2003).

According to the c-structure and f-structure proposed by Kaplan and Zaenen (2003), a sentence such as (8) has the c-structure and the f-structure shown in Figure 1. The c-structure shows that the V node dominating *helpen* 'help', its mother node V'_1 and the NP node *Jan* combined make up the SUBJ and PRED of $XCOMP_1$, whereas the V node dominating *koken* 'cook', its mother node V'_2 and the NP *Marie* make up the SUBJ and PRED of $XCOMP_2$.

8. *...dat we Jan Marie zien helpen koken*
   ...that we John Mary see to-help to-cook

   '...that we see John help Mary to cook'

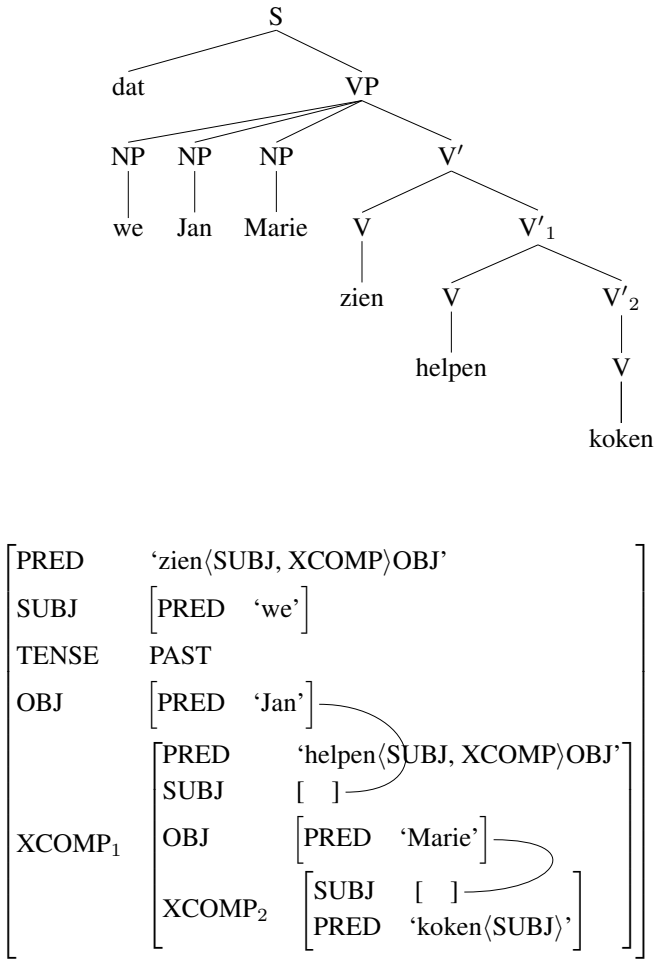


Figure 1: C-structure and f-structure for (8)

The phrase structure rules for Dutch dependency structures as proposed by Kaplan and Zaenen (2003) are given in (5) below. These rules account for the various complementation patterns found in Dutch. Before explaining the technicalities of the equations in (5), I will briefly discuss these complementation patterns. As a starting point, let us take a look at (9), which consists of a simple construction without a complement. The c-structure and f-structure of such a construction without a complement is given in Figure 2, which shows that the NP in the c-structure corresponds to SUBJ at f-structure and that the V′ corresponds to the entire f-structure.

9. *...dat Jan helpt*
   ...that Jan helps
   ‘...that Jan helps’

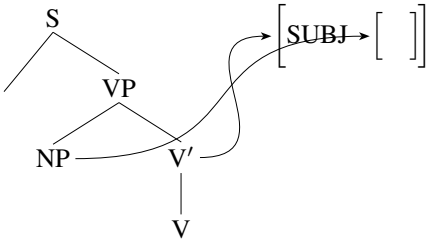


Figure 2: Constructions with matrix verb only

A slightly more complex embedded clause with one complement is given in (10). Such a construction, containing an XCOMP, would have the partial c-structure and f-structure given in Figure 3.[2] For Figure 3 shows the correspondences that were given above in Figure 2, but with the additional correspondences of the second NP to OBJ of XCOMP at f-structure, and the second V′ to XCOMP. More XCOMPs can be added and would be added in the exact same way.

10. *...dat Jan Marie probeert te helpen*
    ...that Jan Marie tries to help
    '...that Jan tries to help Marie'

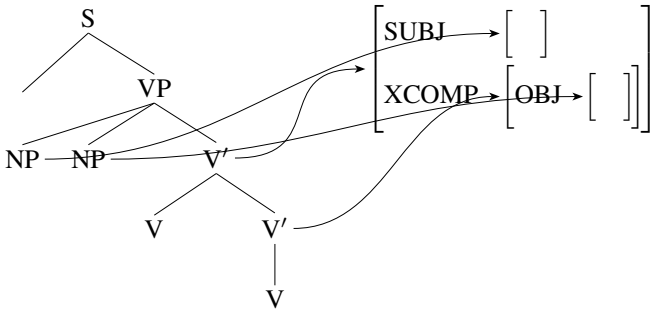


Figure 3: Constructions with matrix verb and one XCOMP

Furthermore, Dutch has the option of extraposing complements within an embedded clause. This is shown in (11), in which the complement *Marie te helpen* is extraposed to the right. Note the difference in position of the NP *Marie* between (11) and (10). The partial c-structure and f-structure of such a construction is given in Figure 4.[3] The f-structure appears very similar to the f-structure in Figure 3, but note that the complement is an COMP instead of an XCOMP, following Zaenen and Kaplan (1995) and Schuurman (1991) for Dutch and Berman (2000) for German.[4]

[2]For the sake of simplicity, the XCOMP's SUBJ is not included in the f-structure, but it is assumed that the main predicate shares its SUBJ with its XCOMP.

[3]Again, the COMP's SUBJ is not included in the f-structure, but based on Zaenen and Kaplan (1995)'s analysis, a 'pro' value for the PRED of the COMP's SUBJ is assumed.

[4]Zaenen and Kaplan (1995) take as evidence for the analysis of the extraposed complement as a

11. *...dat Jan probeert Marie te helpen*
...that Jan tries Marie to help
'...that Jan tries to help Marie'

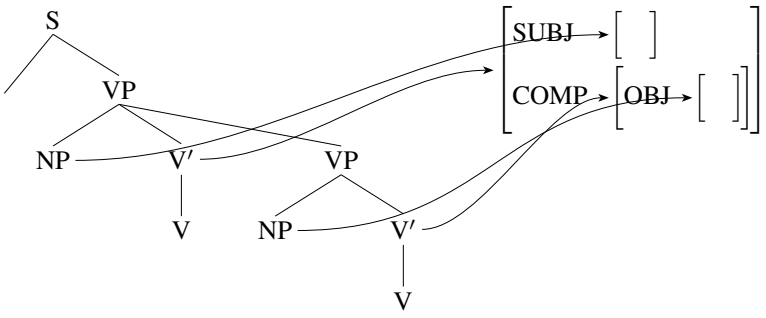


Figure 4: Extraposed constructions

From what has been discussed so far, it is clear that all arguments must precede all verbs, and that both the argument cluster and the verb cluster have specific orders within their cluster. Kaplan and Zaenen (2003), further building on Zaenen and Kaplan (1995), propose the phrase structure rules given in Figure 5. These rules state that a VP can expand to zero or more (as indicated by the Kleene star) NPs, which can serve as some nominal grammatical function (as indicated by NGF) of the complementizer phrases. The functional uncertainty equation (↑XCOMP * (COMP) NGF)=↓ indicates that each grammatical function can be reached in the f-structure by an indeterminate number of XCOMPs. This is followed by a V′ and an optional VP for extraposed structures. Furthermore, the rule in Figure 5 shows that V′ always expands to a V and optionally to an additional V′. The shuffle operator ',' indicates that both the orders V, V′ and V′, V are possible, in order to ensure that both constructions in (12) are permitted.

12. *...dat Jan Marie heeft gezien/gezien heeft*
...that Jan Marie has seen/seen has
'...that Jan has seen Marie'

COMP the fact that impersonal passives are possible with extraposed constructions (i), but not with verb raising constructions (ii):

i ...omdat er (door iedereen) werd getracht Marie te helpen

ii ∗...omdat er (door iedereen) Marie werd trachten te helpen

The functional control equation that is required by the XCOMP function can only specify the subject or object of the higher predicate. Since the XCOMP's SUBJ in the raising construction in (ii) is identified with an oblique agent function (i.e. *door iedereen*), this construction is ungrammatical. The COMP function does not have such a requirement. Therefore, an analysis of extraposed constructions taking COMPs instead of XCOMPs would explain why the impersonal passive is permitted in the extraposed construction in (i).

VP	→	NP*		V′		(VP)	
		(↑ XCOMP* (COMP) NGF)=↓				(↑ XCOMP* COMP)=↓	

V′	→	(PostP)	(V)		[V ,		(V′)]
		(↑ XCOMP$^+$ PRT)=↓	(↓ VFORM)=PART				(↑ XCOMP)=↓
			(↑ XCOMP$^+$)=↓				(↑ XCOMP$^+$ NGF) $\neg <_f$ (↑ NGF)
			↓ $\neg <_f$ (↓ PRT)				

Figure 5: Phrase structure rules for VP and V′

Zaenen and Kaplan (1995) and Kaplan and Zaenen (2003) propose a flat structure of NPs, the embeddedness of which is reflected in the f-structure. The correlation between the level of embedding of the NP and that of its verb is provided by the functional notation below V′ in Figure 5. These annotations under the V′ state that the next lower verb becomes the head of the XCOMP assigned at each level. In order to constrain the cross-serial dependency relations between the NPs and Vs, a rule of f-precedence (i.e. $\uparrow$ XCOMP^{+} NGF) $\neg <_{f}$ ($\uparrow$ NGF) has been added to this rule, which states that an NP that belongs to a lower verb cannot precede an NP belonging to a higher verb.

The main additions to Zaenen and Kaplan (1995) in Kaplan and Zaenen (2003) are the optional PostP and V (as shown in Figure 5) to account for leftward creepers: Dutch has leftward creeping particles (as in (13)) and leftward creeping past participles (see (14)). The extension PostP permits particles of separable verbs to creep leftwards and to assign the feature PRT to the f-structure of their head, which is always at least one level further embedded than the PostP itself, hence the Kleene Plus. The other extension is the optional V when it is a participle, which maps onto an f-structure that is the XCOMP of its mother node. Again, this XCOMP is always at least one level further embedded than the past participle itself. The f-precedence constraint simply states that the past participle cannot come before its particle.

13. (a) *...dat Jan het liedje zal hebben* ***mee****gezongen*
...that Jan the song will have along.sung
'...that Jan will have sung along with the song'

(b) *...dat Jan het liedje zal* ***mee*** *hebben gezongen*
...that Jan the song will along have sung
'...that Jan will have sung along with the song'

(c) *...dat Jan het liedje* ***mee*** *zal hebben gezongen*
...that Jan the song along will have sung
'...that Jan will have sung along with the song'

14. (a) *...dat Jan het liedje zal hebben* ***gezongen***
...that Jan the song will have sung'
'...that Jan will have sung the song'

(b) *...dat Jan het liedje zal* ***gezongen*** *hebben*
...that Jan the song will sung have
'...that Jan will have sung the song'

(c) *...dat Jan het liedje* ***gezongen*** *zal hebben*
...that Jan the song sung will have
'...that Jan will have sung the song'

3 Nested Dependencies in Dutch

3.1 The Fourth Construction

The phrase structure rules discussed so far account for crossed dependencies in Dutch. The literature on dependency relations in Dutch and German (e.g. Bach et al. 1986) has implied that crossed dependency structures are a feature of Standard Dutch, whereas nested dependency structures are a feature of Standard German. However, it appears that the facts are not necessarily this straightforward. Let us now turn to an up-to-now not discussed case of a nested dependency construction in Dutch. The construction I address here is neither a case of verb raising nor of the third construction, and I will refer to it as *the fourth construction* following the tradition that Besten et al. (1988) started. An example of this fourth construction is given in (15). Even though this structure appears to be similar to verb raising constructions, it differs from the verb raising construction in that only a part of the verbal cluster of the complement appears in the position to the right of the matrix verb, instead of the whole verbal cluster.[5] So in (15), the past participle *gevoed* 'fed' is separated from its auxiliary *worden* 'become' (even though they are part of the same complement) by the main predicate *zag* 'saw'. This has consequences for the dependency structure, which as a result is nested: the dependency lines in (15) show the dependency relation between the past participle *gevoed* 'fed' and its argument *een lammetje* 'a lamb', and the dependency relation between the main predicate *zag* 'saw' and its argument *Jan*.

15. *...dat Jan een lammetje gevoed zag worden*
    ...dat Jan a lamb fed saw to-become
    '...that Jan saw a lamb being fed'

I take a non-raising analysis of the passive, following Falk (2003), and I will assume that the auxiliary and the lexical verb correspond to the same minimal f-structure. Under this analysis, the lexical verb is the functional head of the clause and the auxiliaries serve simply as elements that provide voice information. This means that the dependency lines are drawn between the arguments and the lexical verbs and not between the arguments and the auxiliaries.

In other words, the past participle in these constructions does not appear to behave according to the traditional rules of cross-serial dependency constructions. At first sight, the past participle in (15) might appear to be an instance of a cluster creeper, just like other past participles. As said before, past participles are cluster creepers, which are elements in the cluster that creep to various positions within the

[5]This is the case for Standard Dutch, and varieties of Dutch may allow different constructions. One reviewer points out that their southern variety would allow the past particle adjacent to its auxiliary: *...dat Jan een lammetje zag gevoed worden* and would even prefer this order for the verb *laten* 'let': *...dat Jan een lammetje laat gevoed worden*.

cluster, as shown in (14), in which the particle *gezongen* 'sung' can be separated from the verb and appear in two other positions. However, the past participle in the fourth construction cannot be analyzed as a cluster creeper. When comparing a cluster creeper particle to the past participle found in the fourth construction, the difference between the two elements becomes clear: the sentences in (16) show that the past participle *gevoed* (in boldface) can only appear next to its argument and not in any of the other positions that cluster creepers can appear in.

16. (a) *...dat Jan een lammetje* ***gevoed*** *zag worden*
...that Jan a lamb fed saw to-become
'...that Jan saw a lamb being fed'

(b) **...dat Jan een lammetje zag* ***gevoed*** *worden*
...that Jan a lamb saw fed to-become

(c) **...dat Jan een lammetje zag worden* ***gevoed***
...that Jan a lamb saw to-become fed

3.2 Restrictions

There are a few further constraints on this construction. First of all, the fourth construction only occurs in sentences with a perception verb or a causative verb as the matrix verb. Since these verbs are at the same time the only verbs in Dutch that appear with the AcI-construction (Accusativus cum Infinitivo), I refer to them as the AcI-verbs (see *zag* in (17) and (19)). Secondly, fourth the construction will only occur if the XCOMP consists of a passive, which in Dutch is constructed with either the auxiliary *worden* 'become' or *zijn* 'be' in combination with a past participle (see *gevoed worden* 'be fed' in (17) and *geslagen worden* 'be hit' in (18)).

The sentences below show that the matrix verb has to be either a perception verb or a causative verb, and another verb type will be ungrammatical. Only a passive construction (as in (18)) or only a perception verb (as in (19)) is not enough for the fourth construction to be grammatical.

17. Passive, **AcI Verb**

...dat Jan een lammetje gevoed ***zag*** *worden*
...that Jan a lamb fed saw to-become
'...that Jan saw a lamb being fed'

18. Passive, No AcI Verb

**...dat Jan Marie geslagen wilde worden*
...that Jan Marie hit want to-become

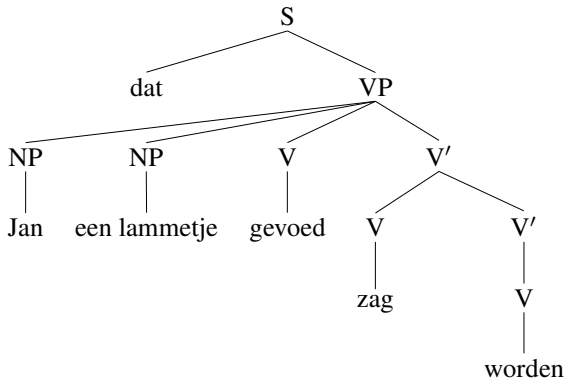
19. No Passive, **AcI Verb**

...dat Jan de vrouw gestoft* ***zag* *hebben*
...that Jan the woman dusted saw to-have

4 Modifications to Kaplan and Zaenen

The c-structure and f-structure that correspond to the fourth construction in (20) are given in Figure 6, in which the passive is analyzed as the functional head of the clause.

20. *...dat Jan een lammetje gevoed zag worden*
...that Jan a lamb fed saw to-become
'...that Jan saw a lamb being fed'



$$\begin{bmatrix} \text{PRED} & \text{'zien } \langle \text{SUBJ, XCOMP} \rangle \text{ OBJ'} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{'Jan'} \end{bmatrix} \\ \text{TENSE} & \text{PAST} \\ \text{VFORM} & \text{TENSED} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{'een lammetje'} \end{bmatrix} \\ \text{XCOMP} & \begin{bmatrix} \text{PRED} & \text{'voeden} \langle \text{SUBJ} \rangle \text{'} \\ \text{SUBJ} & [\ ] \\ \text{VFORM} & \text{PART} \\ \text{AUX-FORM} & \text{WORDEN} \\ \text{PASSIVE} & + \end{bmatrix} \end{bmatrix}$$

Figure 6: C-structure and f-structure for (20)

The c-structure shows the unique aspect of the fourth construction: the separated verbs from the XCOMP verb cluster (i.e. the verb *gevoed* 'fed' and its auxiliary *worden* 'become'), are on either side of the matrix verb *zag* 'saw'. In this analysis, the past participle is placed directly under the VP, in order to ensure this fixed position for the past participle of a passive construction when the main clause's PRED is a perception verb or causative. In order to arrive at this order, the phrase structure rules discussed so far will have to be modified, as shown in Figure 7.

VP	→	NP*	**(V)**	V′	(VP)
		(↑ XCOMP* (COMP) NGF)=↓	(↓ **VFORM)=$_c$PART**		(↑ XCOMP* COMP)=↓
			(↑ **XCOMP)**=↓		

V′	→	(PostP)	(V)	[V ,	(V′)]
		(↑ XCOMP$^+$ PRT)=↓	(↓ VFORM)=$_c$PART		(↑ XCOMP)=↓
			(↑ XCOMP$^+$)=↓		(↑ XCOMP$^+$ NGF) $\neg <_f$ (↑ NGF)
			↓ $\neg <_f$ (↓ PRT)		
			¬ (↑ **PASSIVE)**		

Figure 7: Modified phrase structure rules for VP and V′

The newly introduced item in the first rules in Figure 7 is the optional V, which is constrained to a past participle that is the head of the XCOMP.

The modification to the second rule in Figure 7 is the constraint $\neg(\uparrow$PASSIVE), which needs to be added to the leftmost V in the V-bar rule, with the assumption that the auxiliary *worden* 'become' in (20) will add [PASSIVE $= +$] to the XCOMP. This constraint is necessary to ensure that the past participle under V-bar cannot be passive. This will correctly disallow the past participle of (20) to appear in the three possible positions of *gezongen* 'sung' as in (14), but permit any other past participle to do so.

The lexical entries for the verbs in (20) are the following:

21. zag
($\uparrow$ PRED) = 'zien $<$SUBJ, XCOMP$>$ OBJ'
($\uparrow$ TENSE) = PAST
($\uparrow$ VFORM) = TENSED
($\uparrow$ OBJ) = ($\uparrow$ XCOMP SUBJ)
($\uparrow$ XCOMP VFORM) $=_c$ PART
$\uparrow <_f$ ($\uparrow$ XCOMP AUX-FORM)

22. gevoed
($\uparrow$ PRED) = 'voeden $<$SUBJ$>$'
($\uparrow$ AUX-FORM) = WORDEN
($\uparrow$ VFORM) = PART

23. worden
($\uparrow$ PASSIVE) = $+$
($\uparrow$ AUX-FORM) = $\downarrow$
$\downarrow$ = WORDEN

The lexical entry for the verb *zag* 'saw' in (21) shows that the verb takes a subject and an XCOMP, and an athematic object, following Kaplan and Zaenen (2003). Its tense is past, its form is tensed, its object is the subject of the XCOMP, emphasizing the object raising nature of these verbs. The form of the predicate of the XCOMP is past participle. The f-precedence rule makes sure that the main predicate will precede the XCOMP's AUX-FORM. The lexical entry for the verb *gevoed* 'fed' in (22) shows that this verb takes a subject. Its auxiliary is *worden* 'become' and its form is that of the past participle. Finally, the lexical entry for the auxiliary *worden* 'become' is given in (23) and contributes the feature PASSIVE to the f-structure of its head. The last two equations are presented in what appears to be a rather indirect manner. However, in order to allow for the f-precedence rule for (21) to work (as f-precedence rules only hold between two f-structures), the value of the AUX-FORM feature needs to be assigned an f-structure. This f-structure corresponds to *worden*.

5 Resultative Adjectives

Interestingly, it appears that this fixed position of the past participle corresponds to the position of resultative adjectives (see (24a)), and resultative PPs (see (24b)).

These resultative adjectives and resultative PPs within an XCOMP appear before the main predicate, whereas the XCOMP's PRED appears after the main predicate. These adjectives cannot appear in any other position. This position appears to be reserved for particles as well (for instance, see Neeleman 1994).[6]

24. (a) *...dat Jan het hek* ***groen*** *wil verven*
...that Jan the fence green wants to-paint
'...that Jan wants to pain the fence green'

(b) *...dat Jan het hout* ***in stukken*** *wil hakken*
...that Jan the wood in pieces want to-chop
'...that Jan wants to chop the wood in pieces'

It therefore appears that the fixed position under VP is not just reserved for past participles of passive constructions, but also for resultatives. This link between past participles and resultatives is reminiscent of Kibort (2005)'s discussion of the *resultative* and *passive* in English. Her term *resultative* refers to a resultative participle (e.g. *broken* in *The vase appeared broken*) that is the complement of the main PRED (e.g *be/seem/appear*), whereas the term *passive* refers to a resultative participle (e.g. *broken* in *The vase got broken*) that fulfills the role of the main PRED itself, taking an auxiliary (e.g. *be/become/get*). These descriptions appear to match those for the past participle form of Dutch passives (e.g. *gevoed*), which contribute their own PRED, on the one hand, and for the Dutch resultative adjectives and resultative PPs, which are XCOMPs of other lexical verbs, following Simpson (2006)'s approach to resultative attributes, on the other.

The Dutch data presented here contribute to this discussion on the blurred distinction between adjectives and participles, as resultative adjectives and past participles appear to share the same syntactic position in Dutch. Whether this shared position follows from some shared property (e.g. the position where predicative elements go) remains an unanswered question and provides a direction for further research.

6 Conclusion

In conclusion, I have shown that the rules by Kaplan and Zaenen (2003) need to be modified to account for the nested dependency structure found in sentences with perception verbs/causatives as the main predicate and a passive construction in the XCOMP. By implementing small changes to the existing phrase structure, the fixed position of the past participle is ensured. The lexical entries in addition will make sure the relative order within the verb cluster is correct and that all the necessary features are assigned to the f-structure.

[6]One reviewer notes that the generalization to resultatives proposed here also holds for their southern variety of Dutch: just as the past participle can appear to the right of the main predicate in this variety, so can the resultative.

I have assumed a non-raising analysis of passives, following Falk (2003), unlike previous work on serial dependency relations, which has always taken a raising approach to modals and auxiliaries.

Finally, I have highlighted a direction for further research, namely the shared syntactic position between the past participle of perception verbs and causative verbs on the one hand, and resultative adjectives and resultative PPs on the other.

References

Bach, Emmon, Colin Brown, and William Marslen-Wilson. 1986. Crossed and Nested Dependencies in German and Dutch: A Psycholinguistic Study. *Language and Cognitive Processes*, 1(4):249–262.

Berman, Judith. 2000. *Topics in the Clausal Syntax of German*. PhD thesis, University of Stuttgart.

Besten, Hans den, Jean Rutten, Tonjes Veenstra, and Joop Veld. 1988. Verb Raising, Extrapositie en de Derde Constructie. Manuscript, UvA.

Bresnan, Joan, Ron M. Kaplan, Stanley Peters, and Annie Zaenen. 1982. Cross-Serial Dependencies in Dutch. *Linguistic Inquiry*, 13:613–36.

Evers, Arnold. 1975. *The Transformational Cycle of Dutch and German*. PhD thesis, University of Utrecht.

Falk, Yehuda. 2003. The English Auxiliary System Revisited. In Butt, Miriam and Tracy Holloway King, (eds.), *Proceedings of the LFG03 Conference*. Stanford: CSLI Publications.

Hinrichs, Erhard and Tsuneko Nakazawa. 1998. Third Construction and VP Extraposition in German. In Erhard Hinrichs, Andreas Kathol and Tsuneko Nakazawa, (eds.), *Complex Predicates in Nonderivational Syntax*. New York: Academic Press.

Kaan, Edith and Nada Vasic. 2004. Cross-Serial Dependencies in Dutch: Testing the Influence of NP-Type on Processing Load. *Memory & cognition*, 32(2): 175–184.

Kaplan, Ron M. and Annie Zaenen. 2003. West-Germanic Verb Clusters in LFG. In Seuren, Pieter and Gerard Kempen, (eds.), *Verb Constructions in German and Dutch. The Series is Current Issues in Linguistic Theory, 242*.

Kibort, Anna. 2005. The Ins and Outs of the Participle-Adjective Conversion Rule. In Butt, Miriam and Tracy Holloway King, (eds.), *Proceedings of the LFG05 Conference*, pages 205–225. Stanford, CA: CSLI Publications.

Neeleman, Ad. 1994. *Complex Predicates*. Utrecht: OTS.

Rambow, Owen. 1992. A Linguistic and Computational Analysis of the German 'Third Construction'. In *Proceedings of the 30th Annual Meeting on Association for Computational Linguistics*.

Rentier, Gerrit. 1994. Dutch Cross Serial Dependencies in HPSG. In *Proceedings of the 15th Conference on Computational Linguistics*, pages 818–822.

Schuurman, Ineke. 1991. Functional Uncertainty and Verb-Raising Dependencies. In Wim Kosmijer, Werner Abraham and Eric Reuland, (eds.), *Trends in Germanic Syntax*. Mouton de Gruyter.

Seuren, Pieter and Gerard Kempen. 2003. *Verb Constructions in German and Dutch*. Amsterdam: Benjamins.

Shieber, Stuart. 1985. Evidence Against the Context-Freeness of Natural Language. *Linguistics and Philosophy*, 8:333–343.

Simpson, Jane. 2006. Resultatives. In Butt, Miriam and Tracy Holloway King, (eds.), *Lexical semantics in LFG*. University of Chicago Press.

Vogel, Carl, Ulrike Hahn, and Holly Branigan. 1996. Cross-Serial Dependencies Are Not Hard to Process. In *Proceedings of the 16th Conference on Computational Linguistics*, pages 157–162.

Zaenen, Annie and Ron Kaplan. 1995. Formal Devices for Linguistic Generalizations: West Germanic Word Order in LFG. In *Formal issues in Lexical-Functional Grammar*, pages 215–239. Stanford: CSLI Publications.