

Quantifier Scope in German: An MCTAG Analysis

Laura Kallmeyer

University of Tübingen
Collaborative Research Center 441
lk@sfs.uni-tuebingen.de

Maribel Romero

University of Pennsylvania
Department of Linguistics
romero@ling.upenn.edu

Abstract

Relative quantifier scope in German depends, in contrast to English, very much on word order. The scope possibilities of a quantifier are determined by its surface position, its base position and the type of the quantifier. In this paper we propose a multicomponent analysis for German quantifiers computing the scope of the quantifier, in particular its minimal nuclear scope, depending on the syntactic configuration it occurs in.

1 Introduction: The data

- (1) A man loves every woman.
 $\exists > \forall, \forall > \exists$

In English, in sentences with several quantificational NPs, in principle all scope orders are possible independent from word order. (1) for example has two readings, the $\exists > \forall$ reading and the inverse scope $\forall > \exists$ reading. This is different in German where word order is crucial for scope possibilities.

- (2) a. Viele Männer haben mindestens eine
many men_{nom} have at least one
Frau hofiert.
woman_{acc} flattered.
'Many men have flattered at least one woman.'
*viele > eine, *eine > viele*
- b. Mindestens eine Frau haben viele
at least one woman_{acc} have many
Männer hofiert.
men_{nom} flattered.
'Many men have flattered at least one woman.'
viele > eine, eine > viele

In German, for quantifiers in base order, the surface order determines scope.¹ (2a) has only the scope order *viele > eine* corresponding to surface order, that is, the inverse order *eine > viele* is not available. In contrast to this, if the word order differs from the base order, ambiguities are possible. (2b) for example displays both scope orders, *viele > eine* and *eine > viele*.

In the literature, the following generalizations have been noticed for German: For two quantifiers Q_1, Q_2 with Q_1 preceding Q_2 in the surface order of a sentence, the scope order $Q_1 > Q_2$ is always possible. Furthermore, the inverse reading $Q_2 > Q_1$ is possible if

- (Q1) Q_1 has been moved so that Q_2 c-commands the trace of Q_1 ((Frey, 1993)), and
- (Q2) Q_1 is a weak quantifier (e.g., *irgendein* 'some', *viele* 'many', cardinals) ((Lechner, 1998)).

Evidence for (Q2) –and further evidence for (Q1)– are the examples in (3)–(4). In (3), the (a)-example is in base order and thus has only surface scope, but moving the weak quantifier over the dative quantifier in the (b)-version results in scope ambiguity. This contrasts with (4). In (4), the (a)-version with base order has only surface scope, as before. But now we move the strong quantifier over the dative quantifier, and this does not yield ambiguity. That is, even though the dative quantifier c-commands the trace of the moved quantifier both in (3b) and in (4b), only when the moved

¹Throughout the paper we assume an unmarked intonation. With a different intonation, other scope orders become available because of the change in information structure. But this lies outside the scope of this paper. The base order depends on the verb; in most cases it is Subject - (Indirect Object) - Direct Object.

element is a weak quantifier do we obtain scope ambiguity.

- (3) a. ... dass er [fast jedem Verlag]
 ... that he almost every publisher
 [mindestens ein Gedicht] anbot.
 at least one poem proposed_to.
 ‘... that he proposed some poem to almost every
 publisher.’
*jedem > ein, *ein > jedem*
- b. ... dass er [mindestens ein Gedicht]₁
 ... that he some poem
 [fast jedem Verlag] *t*₁ anbot.
 almost every publisher proposed_to.
jedem > ein, ein > jedem

- (4) a. ... dass er [mindestens einem Verleger]
 ... that he at least one publisher
 [fast jedes Gedicht] anbot.
 almost every poem proposed_to
 ‘... that he proposed almost every poem to at least
 one publisher.’
*jedes > einem, *einem > jedes*
- b. ... dass er [fast jedes Gedicht]₁
 ... that he almost every poem
 [mindestens einem Verleger] *t*₁
 at least one publisher
 anbot.
 proposed_to.
*jedes > einem, *einem > jedes*

(Kiss, 2000) claims that if two quantifiers have been moved such that among themselves they remain in base order, inverse scope is not possible between them. Because of this, he argues for a non-movement-based theory of German quantifier scope. However, Kiss’ claim is not true as can be seen with the example (5) from (Frey, 1993):

- (5) a. weil der freundliche Museumsdirektor
 because the friendly curator_{nom}
 [mindestens einer Frau]₁
 at least one woman_{dat}
 [fast jedes Gemälde]₂ gezeigt hat
 almost every painting_{acc} has shown
 ‘because the friendly curator has shown almost ev-
 ery painting to at least one woman’
*Q₁ > Q₂, *Q₂ > Q₁*
- b. weil [mindestens einer Frau]₁ [fast jedes
 Gemälde]₂ der freundliche Museumsdi-
 rektor *t*₁ *t*₂ gezeigt hat
Q₁ > Q₂, Q₂ > Q₁

In both cases, (5a) and (5b), the two quanti-
 fiers are in base order. According to Kiss there
 should be, contrary to fact, no ambiguity in (5b).
 The difference between the two is that in (5a) the
 quantifiers are in base position while in (5b) both
 of them have been scrambled with the result that
 Q_2 c-commands the trace of Q_1 . We assume with
 (Frey, 1993) that this is why the inverse scope or-
 der becomes available.

We therefore stick to the above-mentioned gen-
 eralizations (Q1) and (Q2) and try to capture them
 in our LTAG analysis. This means that, in order to
 capture (Q1), we need a syntactic analysis of Ger-
 man NPs that takes into account movement and
 base positions.

2 English quantifier scope in LTAG

We use the LTAG semantics framework from
 (Kallmeyer and Romero, 2004; Kallmeyer and
 Romero, 2005). Semantic computation is done on
 the derivation tree. Each elementary tree is linked
 to a semantic representation (a set of Ty2 formu-
 las and scope constraints). Ty2 formulas (Gallin,
 1975) are typed λ -terms with individuals and situ-
 ations as basic types. The scope constraints of the
 form $x \geq y$ specify subordination relations be-
 tween Ty2 expressions. In other words, $x \geq y$
 indicates that y is a component of x .

A semantic representation is equipped with a
 semantic feature structure description. Semantic
 computation consists of certain feature value iden-
 tifications between mother and daughter nodes in
 the derivation tree. The feature structure descrip-
 tions do not encode the semantic expressions one
 is interested in. They only encode their contribu-
 tions to functional applications by restricting the
 argument slots of certain predicates in the seman-
 tic representations: They state which elements are
 contributed as possible arguments for other se-
 mantic expressions and which arguments need to
 be filled. They thereby simulate lambda abstrac-
 tion and functional application. A sample feature
 for this simulation of functional application is the
 feature I that serves to pass the individual con-
 tributed by an NP to the predicate taking it as an
 argument. Besides this functional application as-
 pects, the feature structure descriptions also con-
 tain features that determine the scope semantics,
 i.e., features specifying boundaries for the scope
 of different operators. Sample features for scope
 are MINS and MAXS encoding the minimal and

maximal scope of attaching quantifiers.

Features can be global (feature GLOBAL, here abbreviated with GL) or they can be linked to specific node positions (features S, VP, ...). The latter are divided into top (T) and bottom (B) features. The equations of top and bottom features linked to specific node positions in the elementary trees are parallel to the syntactic unifications in FTAG (Vijay-Shanker and Joshi, 1988). The global features that are not linked to specific nodes can be passed from mothers to daughters and vice versa in the derivation tree.

(6) Everybody laughs.

As a sample derivation let us sketch the analysis of quantificational NPs in English from (Kallmeyer, 2005). Fig. 1 shows the LTAG analysis of (6). More precisely, it shows the derivation tree with the semantic representations and feature structure descriptions of *laughs* and *everybody* as node labels. The feature identifications are depicted by dotted lines. The semantic representation of the NP *everybody* contains the generalized quantifier *every* that binds the variable x and that has a restrictive scope [4] and a nuclear scope [5]. Furthermore, it contains the proposition *person*(x) that must be part of the restrictive scope (constraint [4] \geq l_3). Concerning functional application, the NP provides the individual variable x in the global feature I as a possible argument for the verb predicate *laugh*.

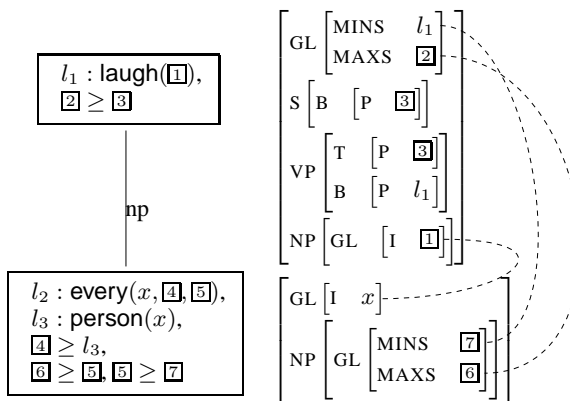


Figure 1: LTAG analysis of (6) *everybody laughs*

Quantificational NPs in English can in principle scope freely; an analysis of quantifier scope must guarantee only two things: 1. the proposition corresponding to the predicate to which a quantifier attaches must be in its nuclear scope, and 2. a quantifier cannot scope higher than the first finite

clause. (Kallmeyer and Romero, 2005) model this by defining a scope window delimited by some maximal scope (global feature MAXS and some minimal scope (global feature MINS) for a quantifier. In Fig. 1, the nuclear scope [5] of the quantifier is delimited by the maximal and minimal scope boundaries provided by the verb the quantifier attaches to (constraints [6] \geq [5], [5] \geq [7]). The feature identifications in Fig. 1 lead then to the constraints [2] \geq [5], [5] \geq l_1 .

Applying the assignments following from the feature identifications and building the union of the semantic representations leads to the underspecified representation (7):

$$(7) \quad \begin{array}{l} l_1 : \text{laugh}(x), \\ l_2 : \text{every}(x, [4], [5]), l_3 : \text{person}(x) \\ [2] \geq l_1, \\ [4] \geq l_3, [2] \geq [5], [5] \geq l_1 \end{array}$$

As the only possible disambiguation, we obtain [2] \rightarrow l_2 , [4] \rightarrow l_3 , [5] \rightarrow l_1 which yields the semantics *every*(x , *person*(x), *laugh*(x)).

3 Syntax of German quantificational NPs

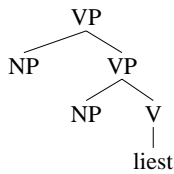
Recall that, according to criterion (Q1), not only the position of an NP but also -if the NP was moved- the position of its trace are crucial for the scope properties. In order to capture this, our analysis needs to take into account movements (scrambling, topicalization, etc.) of NPs including traces at base positions. We therefore cannot adopt the analyses proposed by (Rambow, 1994) in V-TAG where the slot for the NP is generated at the surface position and there is only one initial tree for NPs, whether moved or not.²

- (8) a. ... dass jeder/irgendeiner
 ... that everybody/someone
 irgendein Buch/jedes Buch liest
 some book/every book reads
 '... that everybody/someone reads some
 book/every book'
 SUBJ > DOBJ
- b. ... dass [jedes Buch]₁ irgendeiner t_1 liest
 ... that every book someone reads
 DOBJ > SUBJ

²To avoid misunderstandings, let us emphasize that in LTAG, there is no movement outside the lexicon. Therefore, either the NP or the slot of the NP must be localized together with the corresponding trace inside one elementary structure. This elementary structure can be a tree or, in MCTAG, a set of trees.

- c. ... dass [irgendein Buch]₁ jeder t₁ liest
 ... that some book everybody reads
 SUBJ > DOBJ, DOBJ > SUBJ

To illustrate our analysis, in this and the following section, we restrict ourselves to the sentences in (8). For the syntax, we adopt a multicomponent analysis for NPs that have been moved consisting of an auxiliary tree for the moved material and an initial tree for the trace. Our analysis can be adopted using V-TAG (Rambow, 1994) or something in the style of SN-MCTAG (Kallmeyer, 2005). Note that, in order to account for scrambling, we need some type of MCTAG anyway, independent from quantifier scope.



for each NP, e.g., *irgendein Buch*:

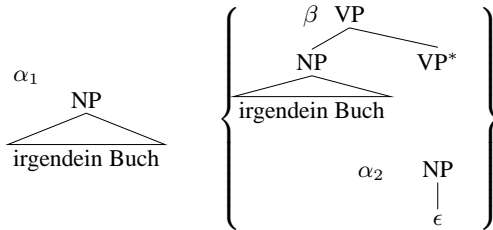


Figure 2: Elementary trees for (8)

The elementary trees for (8) are in Fig. 2. α_1 is used for NPs in base position, while the set $\{\alpha_2, \beta\}$ is used for moved NPs. We assume that, if possible, α_1 is used. I.e., starting from the verb, trees of type α_1 are substituted to its left as long as possible. $\{\alpha_2, \beta\}$ sets are used when α_1 could not possibly yield the desired surface word order. Fig. 3 shows a derivation of a sentence of type (8a) (with no movement). Fig. 4 shows the derivation of (8b). ((8c) is similar to (8b).)

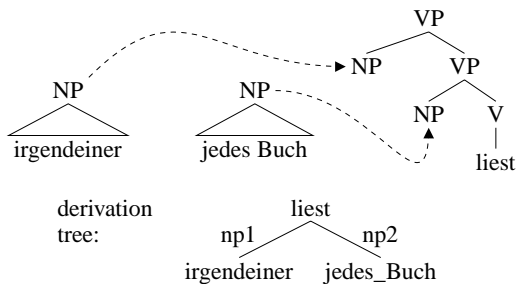


Figure 3: Derivation for (8a)

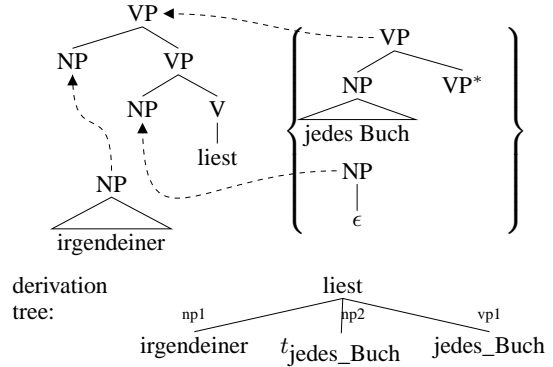


Figure 4: Derivation for (8b)

Note that, in the derivation trees, each node represents a single elementary tree, not a set of elementary trees from the grammar. An MCTAG derivation tree as defined in (Weir, 1988) with each node representing a set is available only for tree-local or set-local MCTAG, not for the MCTAG variants we need (SN-MCTAG or V-TAG). Therefore we take the underlying TAG derivation tree as the derivation structure semantics will be computed on.

4 Semantics of German quantificational NPs

Because of the generalizations above, the following must be guaranteed: i) Strong quantifiers scope over the next element in surface order (take scope where they attach).³ ii) The minimal nuclear scope of a weak quantifier is the closest “unmoved” element following its base position. Consequently, we need different lexical entries for weak and strong quantifiers.

We characterize the scope possibilities of a quantifier in terms of its minimal scope. Consider first the verb tree for *liest* ‘read’ in Fig. 5. In contrast to English, MINS is not a global feature since, depending on the position where the quantifier attaches, its minimal scope is different. In the *liest*-tree, MINS appears in the feature structure of different nodes, with each MINS value determined in the following way: the value of MINS at the NP₂ address is the label l_1 of the verb; the value of MINS at the NP₁ address depends on what is attached at NP₂ (see variables $\boxed{4}$ and $\boxed{0}$, which in this case will be identified with each other); and the value of MINS at the top VP address depends on what is attached at NP₁ ($\boxed{5}$).

³But see section 5, where more complex examples show that this generalization needs to be refined.

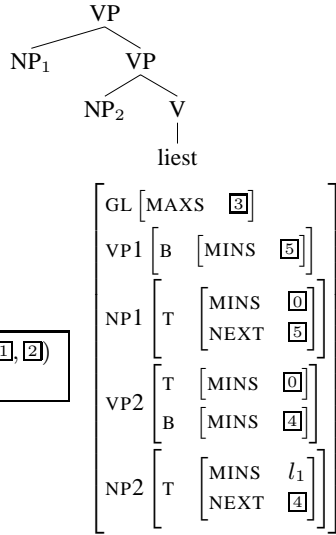


Figure 5: Semantics for *liest*

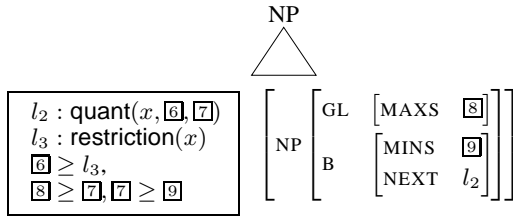


Figure 6: Quantifiers in base position

The idea is that, when an NP (part) is attached at a given address, the label of that NP is the new MINS to be passed up the verb tree; when a trace (part) is attached instead, the MINS of the verb address is passed up unmodified. This feature passing is technically achieved by articulating the VP spine with the feature MINS (much like the use of the P feature in English for adverbial scope in Kallmeyer and Romero, 2005), and by adding the feature NEXT for passing between NP substitution nodes (since substitution nodes do not have T and B features that allow feature percolations between mothers and daughters).

The lexical entries for the three types of quantifiers we must distinguish (non-moved quantifiers, weak moved quantifiers and strong moved quantifiers) are shown in Fig. 6–8. Quantificational NPs that have not been moved (Fig. 6) receive their MINS boundary (variable 9) simply from their attachment position. Weak and strong quantifiers that have been moved differ in how their own MINS is determined: Strong quantifiers (see Fig. 7) get their MINS from the VP node they attach to, i.e., from their surface position (see variable 13). In contrast to this, weak quantifiers (see Fig. 8) get their MINS from the base order position, i.e., from

their trace position (see variable 18).

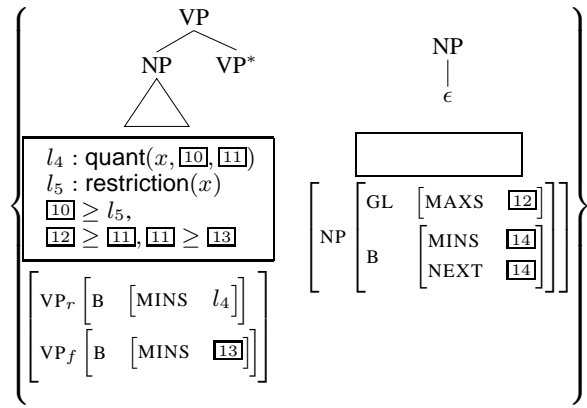


Figure 7: Strong quantifiers that have been moved

As sample analyses consider Fig. 9 and Fig. 10 showing the analyses of (8b) and (8c) where the accusative object quantifier has been moved. (The features of the internal VP node are omitted since they are not relevant here.) In the first case, it is a strong quantifier, in the second case a weak quantifier. For Fig. 9, we obtain the identifications 12 = l1 = 4 = 8, 5 = l2 = 11 (depicted with dotted lines). Consequently, the only scope order is wide scope of *jedes Buch*: l4 > 10 ≥ l2 > 7 ≥ l1. In Fig. 10, we obtain 11 = l1 = 4 = 8, 5 = l2 which leads to the scope constraints l2 > 7 ≥ l1 and l4 > 10 ≥ l1. Consequently, we have an underspecified representation allowing for both scope orders.

The analysis proposed in this section has demonstrated that some features –in this case MINS– are global in some languages (e.g. English) while being local in other languages (e.g. German). We take this as further evidence that the distinction between the two kinds of features, advocated in (Kallmeyer and Romero, 2005) is em-

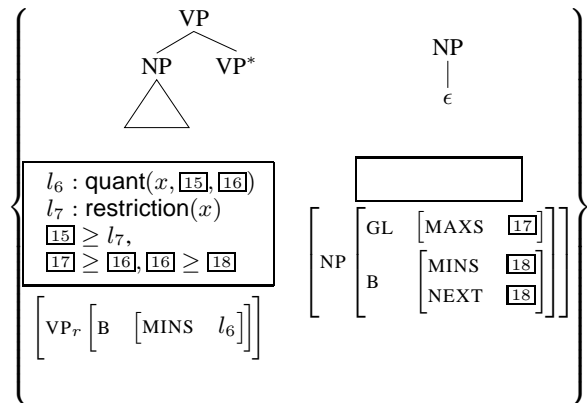


Figure 8: Weak quantifiers that have been moved

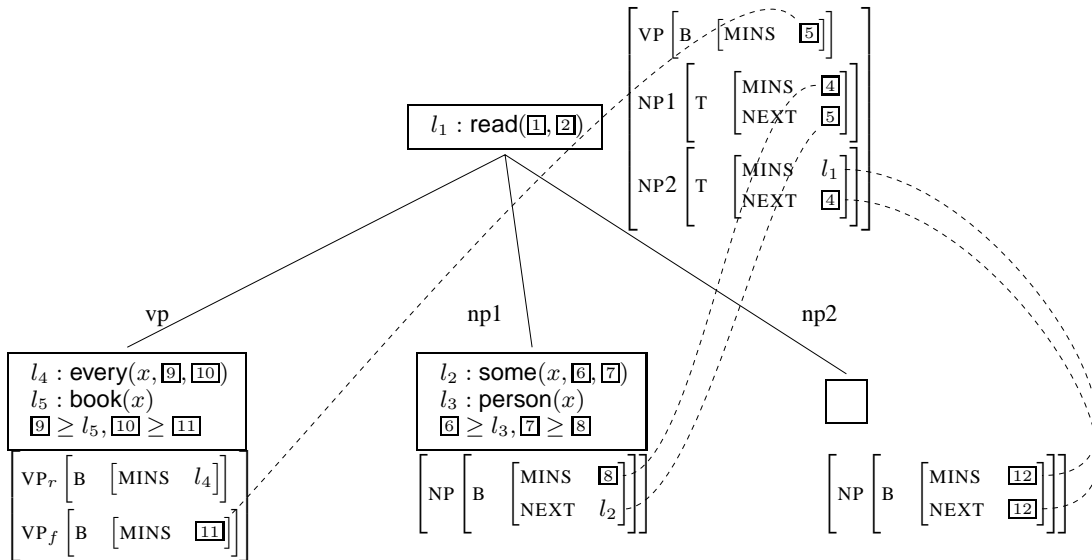


Figure 9: Analysis of *dass [jedes Buch]₁ irgendeiner t₁ liest*

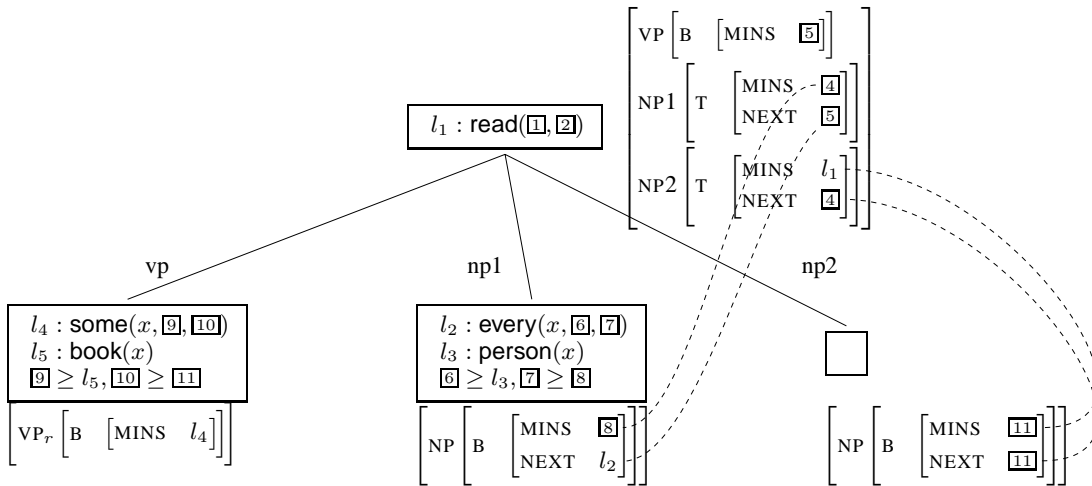


Figure 10: Semantic analysis of *dass [irgendein Buch]₁ jeder t₁ liest*

pirically justified.

5 Long-distance scrambling and quantifier scope

So far we have examined cases where local scrambling affects quantifier scope order. In this section, we will demonstrate how our analysis carries over to long-distance scrambling.

- (9) ... dass [irgendein Lied]₁ Maria
 ... that some song_{acc} Maria_{nom}
 [fast jedem]₂ [t₁ zu singen]
 almost everybody_{dat} to sing
 versprochen hat
 promised has
 'that Maria has promised almost everybody to sing
 some song'
 $Q_1 > Q_2, Q_2 > Q_1$

In (9) both scope orders are possible.

Fig. 11 shows the syntactic analysis for (9). According to the treatment of weak quantifiers proposed above, the minimal nuclear scope of *irgendein Lied* is determined by the position of the trace; it is therefore the proposition of *singen*. As for *fast jedem*, its minimal nuclear scope is required to include the proposition of *versprochen hat*. Nothing else is required, and consequently *irgendein* can scope over or under *fast jedem*.

A problematic configuration that can occur with scrambling concerns cases where two weak quantifiers Q_2 and Q_3 have been moved with a third quantifier Q_1 preceding them where Q_1 is either a strong quantifier or a weak quantifier in base position. Then Q_1 has scope over Q_2 and Q_3 but the scope order between Q_2 and Q_3 is unspecified. An example is (10):

