

Discontinuous constituency in Segment Grammar

Koenraad De Smedt and Gerard Kempen

Abstract. Segment Grammar (SG) is a grammar formalism which is especially suited to model the incremental generation of sentences. SG is characterized by a dual level of syntactic description: f-structures, which are unordered functional structures composed out of syntactic segments, and c-structures, which represent left-to-right order of constituents. True discontinuities in SG are viewed as differences between immediate dominance (ID) relations in c-structures and those in corresponding f-structures. Constructions which are treated in this way include clause union, right dislocation, and fronting. Separable parts of words such as verbs and compound prepositions are not viewed as true discontinuities but as lexical entries consisting of separate syntactic segments.

1. Word order in Segment Grammar

1.1 Introduction

Segment Grammar (SG) was originally proposed by Kempen (1987) under the name of Incremental Grammar. It is a unification-based formalism which is especially suited to model the incremental generation of sentences. In order to account for the fact that human speakers normally produce utterances to some extent in a piecemeal fashion, certain requirements are imposed on a grammar formalism. Specifically, the grammar must define the left-to-right order of partial sentences as well as of complete ones. Moreover, the grammar must allow a partial utterance to be extended, if possible, in vertical as well as horizontal orientation (De Smedt - Kempen 1987: 369-370).

A stronger requirement for incremental generation is that the grammar must fit into a detailed predictive model of language behaviour which explains how utterances are actually produced by speakers. In this chapter, we will, for instance, not be content to merely describe that certain continuous constituents have discontinuous counterparts, but we will also try to explain what are the conditions in the sentence generation process favouring discontinuities and

what are the conditions obstructing them. Various timing factors within the incremental generation process will play a crucial role in this account.¹

We have developed an SG for Dutch and we have implemented a computer simulation program on a Symbolics Lisp Machine which uses this grammar to construct Dutch sentences in an incremental mode. A number of simulations have been run so far which have produced some of the discontinuous constructions described in this chapter. In the remainder of this section, we will briefly describe the way in which left-to-right order of constituents is determined in SG. After that, we will turn our attention to the treatment of discontinuities in SG.

1.2 F-structures and c-structures in Segment Grammar

Somewhat like a lexical-functional grammar (LFG) as proposed by Kaplan - Bresnan (1982: 175-231), an SG assigns two distinct descriptions to every sentence of the language which it generates. The *constituent structure* (or “c-structure”) of a sentence is a conventional phrase structure (PS), which is represented as an ordered tree-shaped graph. It indicates the surface grouping and ordering of words and phrases in a sentence. *The functional structure* (or “f-structure”) provides a more detailed representation of grammatical relations between words and phrases, as traditionally expressed by subject, direct object, etc. The representation in f-structures also accounts for agreement, and it does so by using features like number, gender, etc.

When an SG is used for generation, semantic and discourse information is mapped into f-structures, which in turn are mapped into c-structures. C-structures are then subjected to morpho-phonological processing, producing phonetic strings which are eventually uttered as speech sounds. This overall process is depicted in Figure 1.

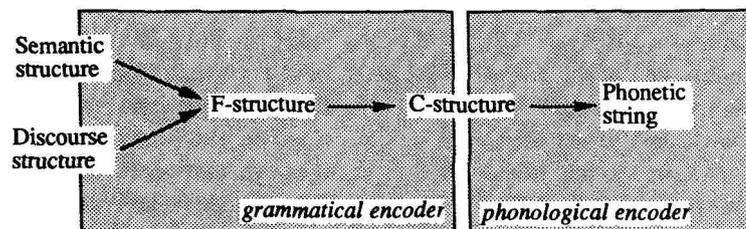
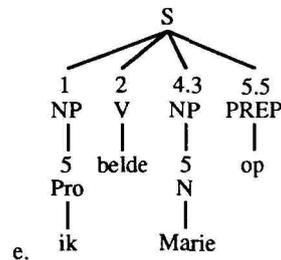
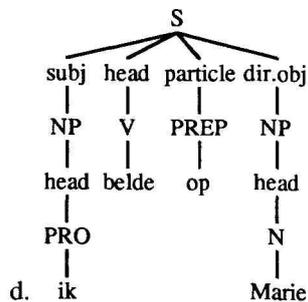
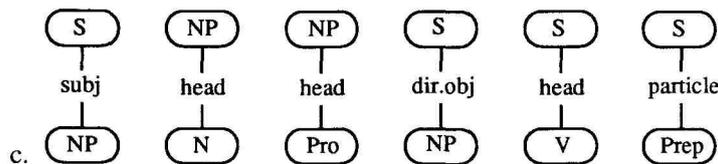


Figure 1. The generation of successive linguistic descriptions during sentence formulation.

1.3 Syntactic segments and ID/LP format

In order to encapsulate grammatical knowledge into units small enough for incremental sentence generation, Kempen (1987) proposes that a grammar consist solely of a set of *syntactic segments*, each representing a single immediate dominance (ID) relation. A segment consists of two nodes representing grammatical categories linked by an arc labeled with a grammatical function. They are graphically represented in vertical orientation, where the top node is called the *root* and the bottom node the *foot*. Syntactic segments join to form a syntactic structure (f-structure) by means of a general unification operation. The f-structure (1d) for the Dutch examples (1a) as well as (1b) consists of six segments (1c).

- (1) a. *Ik belde Marie op.*
‘I called Marie up’
- b. *Marie belde ik op.*
 Marie called I up
 ‘Marie I called up’



The assignment of left-to-right positions to constituents is modeled as the piecemeal derivation of a different kind of structure - a c-structure. By way of example, c-structure (1e) is assigned to (1a).

Somewhat like the ID/LP format for PS rules, SG handles ID relations and linear precedence (LP) relations separately. This enables the grammar to account for systematic variations of word order in a more general way. For example, both (1a) and (1b) could be assigned the same f-structure (1d). However, there are two crucial differences. First, whereas a PS-based system specifies a relative ordering of sister nodes, SG assigns a position to a constituent independently of its sisters; therefore, a system of absolute positions is used, as shown by the numbered slots in (1e). Second, the assignment of LP in SG may be attended with a revision of ID relations. Consequently, the ID relations in the f-structure and the c-structure for a sentence may not be isomorphic. We will therefore explain in some more detail how left-to-right positions are assigned in SG.

1.4 Destinations

The procedure which assigns left-to-right positions works in a bottom-up fashion: the foot node of a segment is attached in the c-structure directly under its *destination*. The destination of a constituent is determined by its matrix in the f-structure, that is, the node which is root of the segment where the constituent is the foot. Normally, the address which the matrix constituent assigns as destination of its dependents is the matrix constituent itself, that is, ID relations in the c-structure are by default the same as those in the corresponding f-structure. Figure 2 is a schematic representation of this process.

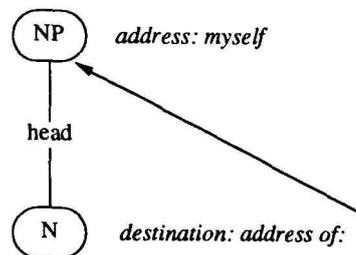


Figure 2. Finding the destination of a node via the address of its mother

Such indirect determination of the destination may seem complicated, but it guarantees that the root node of a segment in f-structure exerts control over the ID relation of the foot node. This will prove useful in the treatment of

constructions where nodes go to higher-level destinations, as discussed below.

1.5 Holders and word order variation

Since f-structures are constructed in a piecemeal fashion, it is natural to assign word order incrementally as well. As soon as a node has been attached to its mother in the f-structure, SG attempts to assign it a left-to-right position in the corresponding c-structure. Because not all constituents are available at the same time, it is difficult to encode word order relative to other constituents. Therefore, SG prefers an absolute order of constituents. For this purpose, a *holder* is associated with each phrase. A holder is a vector of numbered slots that can be filled by constituents. Figure 3 schematically shows some holders associated with c-structure (1e).

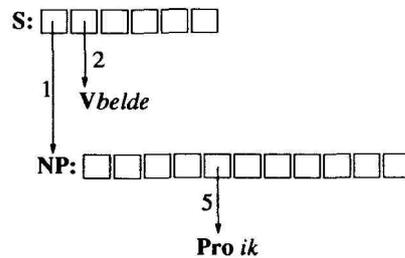


Figure 3. Diagram showing some holders for (1e); the first and second positions of the S and the fifth position of the NP have just been occupied.

The foot node of each segment in the grammar has a feature called *positions* which lists all possible slot positions that the node can occupy in its destination. Word order variation is accounted for by listing more than one possibility where appropriate. For instance, in the grammar of Dutch it is specified that the foot of a S-subject-NP segment may go to holder slots 1 or 3. Constituents will try to occupy the first available slot in this list. For instance, when the foot of a S-subject-NP segment is assigned a position in the holder of its destination, it will first attempt to occupy position 1. Suppose, for instance, that the first slot in the holder has already been occupied (indicated by the crossed out slot in Figure 4); the NP will consequently attempt to insert a pointer to itself into the

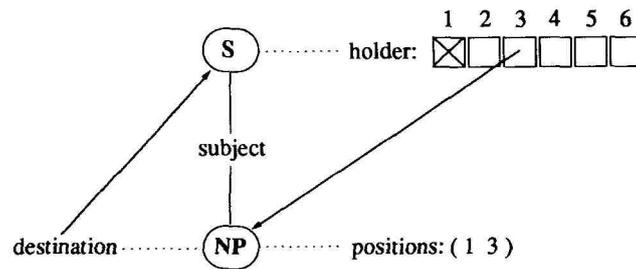


Figure 4. Destination and linearization processes: assign NP to the third slot in the holder of its destination when the first slot is already occupied.

third slot (indicated by the arrow in Figure 4). This situation may give rise to the word order in (1b), where the subject *ik* ‘I’ takes third position rather than first.

Several constituents can make attempts to fill the same slot; SG presupposes a “first come, first serve” principle in dealing with these situations. This principle, in combination with the previously mentioned mechanisms, may give rise to different word order choices in different circumstances. If the utterance has proceeded beyond the point where a constituent can be added, a syntactic dead-end occurs and a self-correction or restart may be necessary. Alternatively, the constituent may end up somewhere else. For instance, the final slot in the S holder is a “dump” for late material (as occurs in right dislocation). The relative clause in (2a) is an instance of such a construction. In spontaneous speech, right dislocation sometimes occurs even if the result is not quite grammatical (2b).

- (2) a. *Marie belde ik op, die ziek was.*
 Marie called I up, who ill was
 ‘I called up Marie, who was ill’
 b. * *Dat was prettig, redelijk.*
 That was nice, quite ‘
 That was quite nice’

Although human speakers do not always make perfect sentences, and sometimes produce clear ordering errors, it seems generally possible, at least in languages like English and Dutch, to determine the order of single fragments one after the other incrementally during sentence generation. In fact,

it seems that incremental production is only possible if for the assignment of left-to-right position to a constituent, the simultaneous presence of all other constituents in the phrase is not required. If this empirical claim is true, the necessary knowledge to determine word order can be encoded locally on the level of single segments, as is done in SG.

The number of holder slots in a Dutch clause is substantial. In order to keep an overview, positions within positions are sometimes used. The Dutch sentence can be divided into six main parts, each having its own internal ordering. Decimal notation is used to represent such slots; for instance, the number 3.2 denotes the second slot in the third main slot. Some holder slots can be occupied by a single constituent only; others may be occupied by an unspecified number of constituents, for instance, an indefinitely long list of adjectival phrases in front of a noun. We will not further elaborate on general aspects of word order in an SG for Dutch. For further discussion of SG and its role in generation we refer to De Smedt (1990b) and De Smedt–Kempen (1991). We will now turn to those aspects dealing with discontinuous constituency.

2. Discontinuous constituency in Segment Grammar

As indicated above, the assignment of left-to-right positions to constituents in a sentence may be accompanied by changes in the ID relations. Thus, a c-structure need not be isomorphic to the corresponding f-structure. SG accounts for various kinds of discontinuous constituents - including right dislocation, S-O raising, and Wh-fronting - by assigning different ID relations in the c-structure.

2.1 Overview of discontinuities

Even languages with a relatively fixed word order allow constituents of a phrase to be non-adjacent. In (3-7), five important kinds of discontinuities in Dutch are summed up (see also Bunt 1988).

The examples in (3) contain broken-up constituents (indicated by means of underlining) which in part have been dislocated to the right, across another constituent.

- (3) a. *Ik heb een auto gekocht met zes deuren.*
I have a car bought with six doors ‘I have bought a car with six doors’
- b. *Een van zijn vrienden kwam, die in Brussel woont.*
One of his friends came, who in Brussels lives
‘One of his friends came, who lives in Brussels’
- c. *Een betere film dan ik verwachtte draaide gisteren in Calypso.*
A better film than I expected was shown yesterday in Calypso
‘A better film than I expected was shown at Calypso yesterday’
- d. *Een betere film draaide gisteren in Calypso dan ik verwachtte.*
A better film was shown yesterday in Calypso than I expected
‘A better film than I expected was shown at Calypso yesterday’

Example (3a) shows a discontinuous NP with an extraposed PP. Extraposition is optional here and tends to occur more often in spontaneous speech. Example (3b) shows a similar construction with a right dislocated relative clause rather than a PP. Again, extraposition is optional, but tends to be more acceptable as the relative clause is longer and the rest of the sentence (*kwam*) is shorter. In (3c), it is an adjectival phrase (ADJP) which is discontinuous; this right dislocation is obligatory and can extend not only to the NP level (3c) but also to the S level (3d).

A second kind of discontinuity consists of compound words which are split up, for instance the verb *opbellen* in (4a) and (1a,b) and the preposition *doorheen* (through) in (4b).

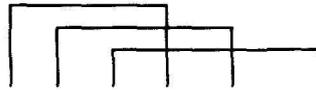
- (4) a. *Bel me morgen op om dit te bevestigen.*
Call me tomorrow up in-order-to this to confirm
‘Call me up tomorrow in order to confirm this’
- b. *Het vliegtuig gaat nu door de geluidsbarriere heen.*
The airplane goes now through-1 the sound barrier through-2
‘The airplane now breaks the sound barrier’

In SG, these cases are not really considered discontinuities, but the “split” elements are listed in the lexicon as already consisting of several segments. Consequently, they are realized as separate constituents on all levels of rep-

resentation. They may be assigned left-to-right positions in such a way that other constituents may intervene. The French negative *ne ... pas* is also an instance of this kind of lexical entry.

Examples of *clause union* are given in (5). In (5a), the constituents of the infinitival clause (underlined) are not kept together as a whole but are assigned positions in the main clause. Thus, objects are grouped irrespective of the clause where they functionally belong, and likewise for non-finite verbs. Clause union may result in crossed dependencies in Dutch, as shown schematically in (5b).

- (5) a. *Ik heb Otto een appel zien eten.*
 I have Otto an apple seen eat
 ‘I have seen Otto eat an apple’



- b. *Ik dacht dat Jan Piet Marie zag helpen zwemmen.*
 I thought that Jan Piet Marie saw help swim
 ‘I thought that Jan saw Piet help Marie swim’

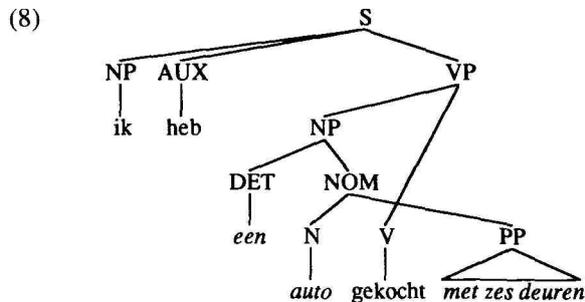
A fourth kind of discontinuities involves unbounded dependencies, for instance, *wh*-fronting in (6a) and fronting of a focused element from subordinate clauses in (6b).

- (6) a. *Wie dacht je dat ik opbelde?*
 Who thought you that I up-called
 ‘Who did you think I called up?’
 b. *Dat blonde meisje dacht ik dat je opbelde.*
 That blond girl thought I that you up-called
 ‘That blond girl I thought you called up’

A fifth kind of discontinuity contains the pronoun *er* ‘there’ and similar “R-words”. When the object in a PP does not have a person as its antecedent, it is pronominalized by means of the special pronoun *er* (often also called a pronominal adverb), which is placed before the preposition. This combination of *er* and a preposition may be interrupted by some other constituents, as in (7).

- (7) *De vloeistof gaat er nu in.*
 The liquid goes there now in
 ‘The liquid now goes in it’

Since the various kinds of discontinuous constructions which are summed up above are more or less problematic for PS-based grammars, it has been proposed to amend the definition of an ordinary PS tree to accommodate discontinuities. For sentence (1a), this could result in the modified tree structure (8).



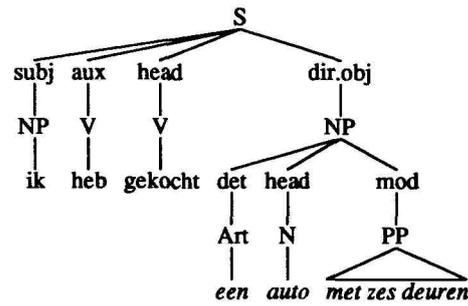
In order to generate such structures within a context-free framework, Bunt (1988) proposes Discontinuous Phrase Structure Grammar (DPSG) which introduces and formalizes the notion of adjacency into PS grammars. DPSG is motivated by the claim that other generation algorithms for a language with discontinuities would first have to generate a continuous ordered tree representation and then identify and apply the transformations which produce the correct word order for discontinuities. However, this need not be the case for a grammar which distinguishes between an unordered functional structure (f-structure) and an ordered surface structure (c-structure). In SG, the correct word order is produced directly.

In the remainder of this section, it is shown how various discontinuities are handled by changes in ID relations at the time when left-to-right order is determined.

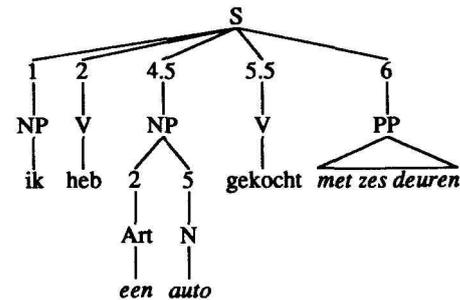
2.2 Right dislocation

At the level of f-structure, SG assigns the right dislocated PP in (3a) a functional relation to the NP *een auto* as shown in f-structure (9a). However, the PP is not part of the NP in the c-structure. Rather, it has an ID relation to the S, as shown in c-structure (9b). Since the f-structure is unordered, the computation of a c-structure is not a transformation in the TGG sense, but an assignment of left-to-right order accompanied by a simple reassignment of an ID relation.

(9) a.



b.

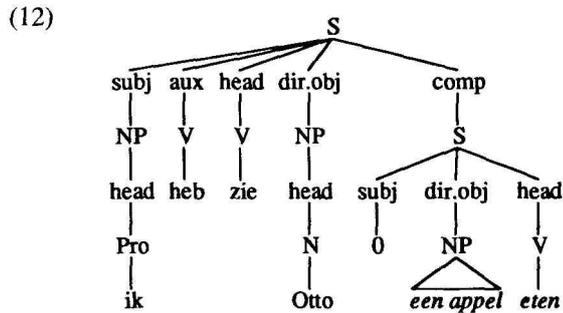


The right dislocation of a PP fits naturally into the incremental generation process. It is triggered by the fact that the extraposed PP cannot be added incrementally to the holder of the NP, because the utterance has already proceeded beyond that point. Therefore the PP is exceptionally allowed to move to the S level, which has a holder slot where some kinds of “late” constituents can be placed. Preferably, this slot (numbered 6) contains at most one constituent; more than one constituent is not impossible though, as in (10a,b). The character of such constituents as “afterthoughts” becomes clearer if more constituents are added.

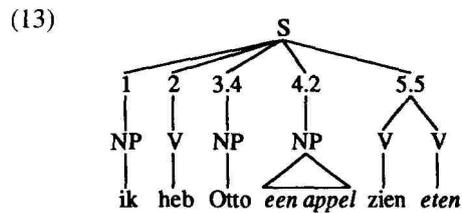
- (10) a. *Ik heb een auto gekocht vorige week met zes deuren.*
 I have a car bought last week with six doors
 ‘I have bought a car last week with six doors’
 b. *Ik heb een auto gekocht met zes deuren vorige week.*
 I have a car bought with six doors last week
 ‘I have bought a car with six doors last week’

Right dislocation is sometimes obligatory, for instance (3c,d). Such obligatory right dislocation must be specified in the lexicon. In SG this is achieved by

subject in the complement S is not realized because non-finite clauses never have subjects.



The discontinuity of the infinitival clause, as shown by means of underlining in (5), is accounted for by means of *clause union*: the complement S forms one surface unit with the matrix S. That is, the constituents of the embedded S are assigned positions in the holder of the matrix S. Although both infinitives collocate in one positional slot, the infinitives from deeper clauses - which are positioned later - are added at the end. The resulting c-structure is shown in (13).



Clause union is brought about by the same destination mechanism which constructs “normal” c-structures. As mentioned in Section 4.4, the destination of a constituent is determined by its mother node in the f-structure. Normally, the mother assigns a dependent constituent a position in its own holder. However, clauses whose constituents are “raised” do not function as destinations. Rather, they use their mother nodes as the destination addresses for their constituents,² as shown in Figure 5.

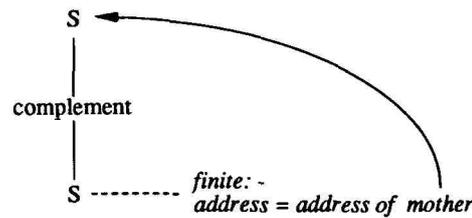


Figure 5. Segment for object complement clause.

Raised constituents may themselves contain raised constituents: pointers are followed step by step upward in the tree until a node is found which will function as the destination address. The infinitival complement S node itself occurs nowhere in the c-structure; since all its dependent constituents are put elsewhere, there is no need to assign it a position.

S-O raising requires substantial planning ahead: if only the subject and head are planned ahead and realized early, the sentence will come out as (14), where the first part cannot be complemented by an S-O raising construction, but can - in this case - be complemented with a finite subordinate clause (a *that*-clause). In order for S-O raising to be successful in an incremental mode of generation, it is therefore necessary that the thematic roles involved in this construction are established well in advance.

- (14) *Ik heb gezien dat hij een cake bakt.*
 I have seen that he a cake bakes
 'I have seen that he bakes a cake'

The requirement of semantic coreferentiality of the direct object of the matrix clause with the subject of the embedded clause may force a passivization, as shown in (15). Suppose that in an incremental mode of generation, the direct object of the matrix clause is generated first. If this constituent is coreferential with the direct object of the (active) embedded clause, then that clause cannot be realized as an active one, because coreferentiality with its subject is required. The lexicalization process may then apply lexical rules to the lemma of the embedded clause. Passivization will produce a lemma where the subject is coreferential with the direct object of the matrix clause.

- (15) *Jan ziet Piet door Marie gekust worden.*
 Jan sees Piet by Marie kissed be
 'Jan sees Piet being kissed by Marie'

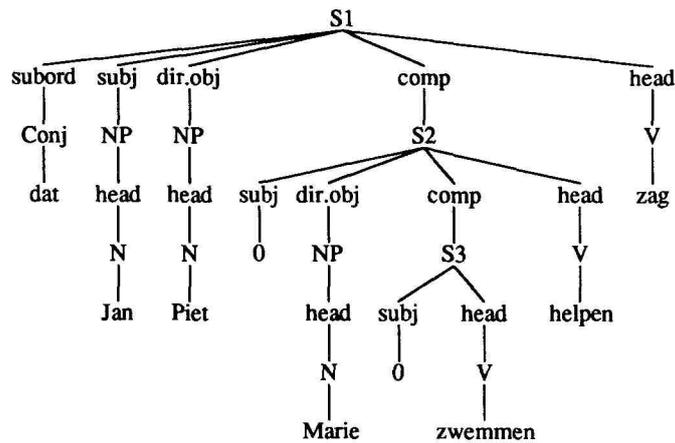
2.4 Cross-serial dependencies

When multiple instances of clause union are embedded in a finite subordinate clause in Dutch, the c-structure may exhibit so-called cross-serial dependencies. Example (16a) is taken from Bresnan et al. (1982). The horizontal brackets indicate dependency relations between NPs and main verbs. In (16b), the vertical brackets indicate the grouping of constituents in surface positions. The German translation equivalent (16c) shows that the ordering of the infinitives is language-specific.

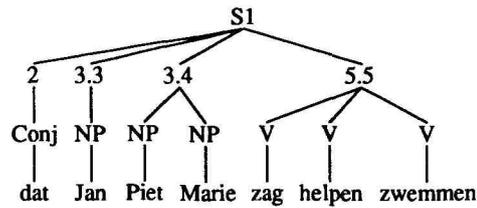
- (16) a. *dat Jan Piet Marie zag helpen zwemmen.*
 that Jan Piet Marie saw help swim
 ‘that Jan saw Piet help Marie swim’
- b. *dat Jan [Piet Marie] [zag helpen zwemmen]*
- c. *daß Jan [Piet Marie] [schwimmen helfen sah]*

As shown in f-structure (17a), example (16a) is a doubly embedded S-0 raising construction. The collocation of the raised objects in one surface position, as well as the collocation of the infinitives in one surface position, cf. (17b), is accomplished by clause union, as explained in Section 5.3.

(17a)



(17b)



The relative ordering of objects and verbs remains to be explained. Recall that a single position in a holder can be occupied by a list of several constituents. For instance, the position of the clause-final verb cluster contains an ordered list of verbs. The relative ordering of the verbs follows from a rule which specifies that, in Dutch, constituents are by default added to the end of the list. Thus, the cross-serial pattern emerges quite automatically, since deeper embedded constituents are added later than shallower ones. For a language like German, which is quite similar but accumulates the verbs in the reverse order, the opposite rule which adds verbs to the front of the list is postulated (as suggested by Kempen - Hoenkamp 1987: 230). Finally, the English equivalent is simply accounted for by the absence of clause union, so that the embedded clauses are retained in the c-structure. It can be concluded that the same f-structure can easily account for the different surface phenomena in the three languages mentioned.

2.5 Unbounded dependencies

Interrogatives in Dutch are characterized by a marked word-order. Yes-no questions, for instance (18a), show subject-verb inversion. In *wh*-questions, the interrogative pronoun is normally fronted (18b) although this is not necessary (18c). *Wh*-fronting in itself is not seen as a discontinuity in SG. However, certain verbs allow a *wh*-constituent to escape from an embedded clause in order to be fronted in the matrix clause; this results in a discontinuity which will be called *wh-extraction* (18d). Optional fronting and possible resulting discontinuities are also observable with focused elements (18e), which suggests that *wh*-extraction and focus extraction can be treated in a similar fashion.

- (18) a. *Eet Otto een appel?* Eats
 Otto an apple 'Is Otto
 eating an apple?'

- b. *Wat eet Otto?*
 Wat eats Otto
 ‘What is Otto eating?’
- c. *Otto eet wat?*
 Otto eats what
 ‘Otto is eating what?’
- d. *Wat denk je dat Otto eet?*
 What think you that Otto eats
 ‘What do you think Otto is eating?’
- e. *Een appel denk ik dat Otto eet.*
 An apple think I that Otto eats
 ‘An apple I think Otto is eating.’

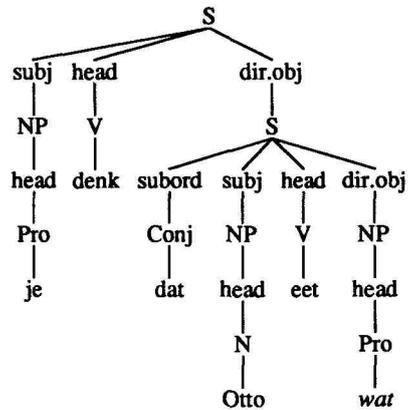
The treatment proposed below roughly follows the lines suggested by Kempen - Hoenkamp (1987: 231-238) but works more incrementally and is extended to cover focus fronting as well; actually, wh-constituents will be considered focused. The discontinuities are accounted for in terms of a non-isomorphism between corresponding f-structures and c-structures. A treatment similar to that of clause union may cause an embedded constituent to be “raised” to a higher level in the c-structure, where it occupies the clause-initial position, which is reserved for focused constituents. Let us now have a closer look at this process.

As with other word order variations, the temporal properties of the generation process are considered primarily responsible for the marked word order of focused constituents. It is assumed that those parts of the semantic input which are to be realized as focused constituents, are passed to the grammatical encoder at a very early stage in the sentence generation process. This causes them to occupy a sentence-initial position. However, if for some reason the focused semantic elements are not accessible in time, the grammatical encoder may already have assigned another constituent a position in the first holder slot. This may result in a question with unmarked (declarative) word order as in (18c).³

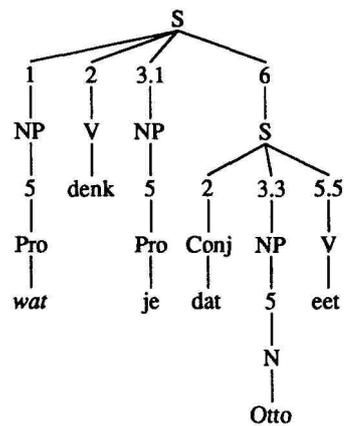
For sentences involving wh-extraction (18d) and focus extraction (18e), the moment when focused elements are accessible is important as well, but some additional machinery is necessary to allow the extraction of a constituent from an embedded clause to a higher level clause. This possibility must be indicated at the level of the lexical entry. Unlike clause union, where the destination of all constituents of a clause is transferred to a higher level, we must be more selective now. A special feature called *focus-destination*⁴ is proposed, which

handles the destination of focused elements. A focused element will first attempt to occupy a specified spot (position 1) in the holder of its focus-destination, otherwise it will go to its normal (default) destination. An analysis of (18d) is presented as f-structure (19a) and c-structure (19b).

(19) a.



b.



In lexical entries which allow focus extraction (and wh-extraction), this feature is specified on the segment for the object complement clause which is associated with the entry. If the lexical entry allows focus extraction, for instance Dutch *zeggen* 'to say' or *zien* 'to see', then the feature in the foot node of the segment S-direct-object-S will refer to the feature in the root node. This is schematically shown in Figure 6. If the lexical entry does not allow

focus extraction, for instance Dutch *weten* ‘to know’, then the feature will be absent.

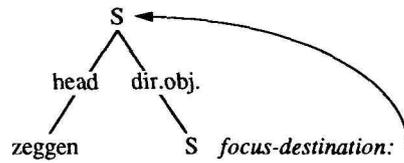


Figure 6. Dutch *zeggen* (to say) is a lexical entry allowing focus extraction.

In a fashion similar to the treatment of clause union, the value of the feature *focus-destination* may recursively refer upward in multiple embedded clauses. A remaining question is then, when and how to stop referring upward. In declarative clauses and direct questions, the final destination for focused elements is clearly the main clause. However, the mechanism should also work for indirect questions, for instance (20a,b), where the final destination of the focused element is a subordinate clause.

- (20) a. *Ik weet wat je ziet dM Otto eet.*
 _ I know what you see that Otto eats
 ‘I know what you see Otto is eating’
 b. *Ik weet dat je ziet wat Otto eet.*
 _ I know that you see swhat Otto eats
 ‘I know that you see what Otto is eating’

It seems to be necessary for grammatical encoding to know exactly which clause is being questioned. This can be indicated by means of a feature *interrogative* on the S in question. It is assumed that this feature has been set as a consequence of processing the semantic input structure. When such a feature is present, the *focus-destination* of an S refers to that S itself rather than upward.

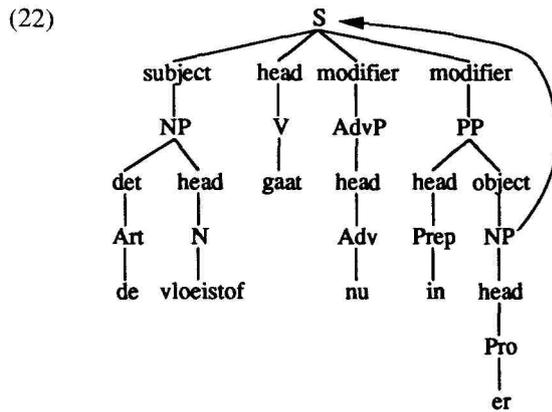
2.6 Pronominal adverbs

The Dutch adverbs *er*, *daar* ‘there’ and *hier* ‘here’ sometimes serve as variants on the pronouns *het* ‘it’, *dat* ‘that’ and *dit* ‘this’ respectively, because the latter pronouns are not tolerated by many prepositions, for instance (21a). This use

of pronominal adverbs may result in a discontinuity of the prepositional phrase (cf. 21b).

- (21) a. * *De vloeistof gaat nu in het.*
 The liquid goes now in it
 ‘The liquid now goes into it’
 b. *De vloeistof gaat er nu in.*
 The liquid goes there now in
 ‘The liquid now goes into it’

Apparently *er* and the other “R-words” are part of the S at the level of c-structure and thus must be allowed to escape from the PPs where they belong in f-structure. The destination of *er* is in this case not the default, that is, its mother node in the f-structure, but the next higher node. This is a property of the lexical entries for the pronominal adverbs; so this exception does not interfere with the general mechanism. F-structure (22) shows the destination of the pronominal adverb in example (21b). The S holder uses slot 3.5 for this constituent.



- (23) a. *Er gaat een vloeistof in de fles.*
 There goes a liquid into the bottle
 ‘A liquid is poured into the bottle’
 b. *Er gaat een vloeistof in.*
 There goes a liquid in
 ‘A liquid is poured into it’

2.7 Concluding remarks

We have presented discontinuities not only from a structural viewpoint but also from a processing viewpoint. Several kinds of discontinuities seem to offer advantages for an incremental strategy in sentence generation. This holds especially for the optional dislocations. Right dislocation allows the generator to utter constituents which are ready, and to postpone uttering more complex (or “heavy”) ones, which are still being processed, to a later stage. In addition, right dislocation allows the incorporation of new semantic input as afterthoughts. Fronting of focused constituents is also natural in an incremental mode of generation if we assume that prominent concepts are passed on to the grammatical encoder earlier than other parts of the semantic input. In contrast, S-O raising benefits less from an incremental mode because it seems to require some planning ahead.

True discontinuities in SG are viewed as differences between ID relations in c-structures and those in corresponding f-structures. Constructions which are treated in this way include clause union, right dislocation, and fronting. Separable parts of words such as verbs and compound prepositions are not viewed as true discontinuities but have their origin in lexical entries consisting of multiple segments.

The use of c-structures in SG is somewhat similar to LFG, but contrasts with other approaches such as DPSG which are based on PS grammar. Whereas DPSG attempts to fit both functional relations and surface constituency into one structure, SG distinguishes between an unordered functional structure and an ordered constituent (surface) structure.

We make the following tentative generalizations about SG mechanisms for discontinuities. The destination of a constituent must always be a node which dominates it - but not necessarily immediately. There seem to be two major variants of the destination mechanism allowing constituents to go to nonimmediately dominating destinations. The first is root initiated. In these cases, for instance clause union, a node refers its constituents to another

dominating node. This operation may be recursive. The second mechanism is foot initiated. In these cases, for instance, PPs with pronominal adverbs, a node directly presents itself to a higher-level destination.

Notes

1. A similar enterprise was undertaken by De Smedt (1993) for some aspects of word order variation.
2. The implementation of this rule is shown in De Smedt (1990a: 137-138).
3. Kempen - Hoenkamp (1987: 233) account for "declarative" word order in questions by assuming that wh-fronting occurs only in the presence of a special ?X tag. The treatment proposed in the current work does not rely on this special tag but exploits the incremental assignment of word order to choose between alternatives.
4. This feature takes the role of the variable wh-dest in IPG (Kempen - Hoenkamp 1987: 232).

References

- Bresnan, J., R. Kaplan, S. Peters and A. Zaenen
 1982 Cross-serial dependencies in Dutch. *Linguistic Inquiry* 13, 613-635.
- Bunt, H.
 1988 DPSG and its use in sentence generation from meaning representations. In: M. Zock and G. Sabah (ed.), *Advances in natural language generation*, Vol 2. London, Pinter Publishers, 1-26.
- De Smedt, K.
 1990a *Incremental sentence generation: a computer model of grammatical encoding*. Ph. D. dissertation. *NICI technical report 90-01*, NICI, University of Nijmegen, Nijmegen.
- De Smedt, K.
 1990b IPF: An incremental parallel formulator. In: R. Dale, C. Mellish and M. Zock (eds.), *Current research in natural language generation*. London, Academic Press, 167-192.
- De Smedt, K.
 1991 Parallelism in incremental sentence generation. In: G. Adriaens and U. Hahn (eds.), *Parallel models of natural language computation*. Norwood, NJ, Ablex.

- De Smedt, K. and G. Kempen
1987 Incremental sentence production, self-correction and coordination. In: G. Kempen (ed.), *Natural language generation: new results in Artificial Intelligence, psychology and linguistics*. Dordrecht/Boston/Lancaster, Kluwer Academic Publishers, 365-376.
- De Smedt, K. and G. Kempen
1991 Segment Grammar: a formalism for incremental sentence generation. In: C. Paris, W. Swartout and W. Mann (eds.), *Natural language generation in Artificial Intelligence and Computational Linguistics*. Boston/Dordrecht/London, Kluwer Academic Publishers, 329-349.
- Horn, G.
1985 Raising and complementation. *Linguistics* 23, 813-850.
- Kaplan, R. and J. Bresnan
1982 Lexical-functional grammar: A formal system for grammatical representation. In: J. Bresnan (ed.), *The mental representation of grammatical relations*. Cambridge, MA, MIT Press, 173-281.
- Kempen, G.
1987 A framework for incremental syntactic tree formation. *Proceedings of the 10th IJCAI, Milan*, Los Altos, Morgan Kaufmann, 655-660.
- Kempen, G. and E. Hoenkamp
1987 An incremental procedural grammar for sentence formulation. *Cognitive Science* 11, 201-258.