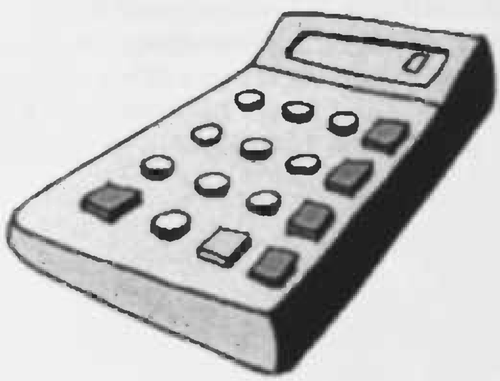
Do rules add up?

Cumulativity of constraints in the interpretation of incomplete coordinate constructions



Marieke van der Feen
s0997145

January 20th, 2006
Supervised by Dr. Petra Hendriks

Kunstmatige Intelligentie
Rijksuniversiteit Groningen

955

Abstract

In Optimality Theory, determining the optimal candidate is traditionally done on the basis of strict domination. A violation of a higher ranked constraint is always more serious than no matter how many violations of lower ranked constraints. Recently, there have been indications that strict domination is not suitable as an evaluation method for some types of linguistic data. An interesting question is whether in some linguistic fields cumulativity of constraints - in which a combination of violations of lower ranked constraints can overrule a higher constraint violation - is a more accurate way to evaluate candidates.

My research focused on the interpretation of (possible) gapping constructions. The central issue is the ambiguity in sentences such as: "Grace geeft Stan een shirt en Will een trui." (*Grace gives Stan a shirt and Will a sweater*). Will can be the person giving Stan a sweater or he can be the person receiving the sweater. The factors influencing the interpretation of this kind of sentences can be defined as OT constraints.

I implemented an OT computer model of the interpretation of gapping. This model evaluates interpretation candidates according to different constraint evaluation methods: Strict domination and three methods in which constraints interact in a cumulative manner. These four hypotheses on the evaluation of interpretation candidates were then tested in a pilot experiment, in which subjects were asked to give their interpretation of sentences. The experimental results point to an explanation in which cumulativity of constraints plays a role.

1 Introduction

At this moment, Optimality Theory (OT, Prince and Smolensky, 2004/1993) is the leading linguistic theory in phonology. In this theory, a grammar consists of a set of violable constraints. These constraints are assumed to be universal, i.e. existent in every language. The constraints are hierarchically ordered and this is where languages differ: Each language has its own hierarchy of constraints. The idea of rules that can be broken was revolutionary: Previous theories existed of rules that had to be obeyed in order to achieve acceptable output. In OT, constraints can be violated. The effect of the violation of the constraint depends on its position in the hierarchy.

A predecessor of OT was Harmony Theory (Legendre *et al.*, 1990). Harmony Theory was directly based on neural modeling. Neural networks are networks that consist of units, connected by weights. These weights can be either excitatory or inhibitory. In Harmony Theory, the constraints are the units. Each constraint is assigned a weight. The weighted summation of constraint values determines the "harmony" of a linguistic form. Harmony of a linguistic form is connected to its well-formedness. From a set of candidate inputs, the one with the highest harmony value will be chosen as the most well-formed candidate. OT borrowed the idea of violable constraints from Harmony Theory. A crucial concept of standard OT - that is not present in Harmony Theory - is "strict domination". This concept entails that there is no co-operation between constraints to determine the optimal candidate. Only the strongest of the violated constraints influences the optimality of a candidate. It does not matter at all which lower constraints have been violated, and how many times. Higher ranked constraints have strict domination over lower ranked constraints. This concept is obviously different from Harmony Theory: It is no longer the weighted summation of constraints that determines the output. To find the optimal candidate in standard OT, no calculation is necessary.

Depending on the field of application, input is presented to the model. A set of output candidates is generated. With the ordered set of constraints, a relation between input and output is determined. From the output candidates, the optimal candidate is determined: The candidate that causes the least serious constraint violations will be the optimal candidate. Optimality Theory has proven to be successful in explaining data in several linguistic fields, especially phonology, but also for example syntax.

Semantics is a relatively new field for Optimality Theory. Researchers have pointed out that semantics is fundamentally different from the fields Optimality Theory previously focused on (for example, Hendriks and de Hoop, 2001). Instead of the perspective of the speaker, the perspective of the hearer has to be taken. When Optimality Theory is applied to semantics, the input consists of a linguistic form, which has to be interpreted. A set of possible interpretations is generated, the candidate set. The optimal interpretation for that form will be the output of the model.

Perquin (1999) studied Optimality Theory and semantics. Her thesis focused on the phenomenon of gapping, which she defines as "an elliptical construction in which at least the verbal head is left out". An example of a sentence in which gapping occurs, is: "John gave Mary a cookie and Peter a book". This sentence can be interpreted in two ways; Either Peter gave Mary a book, or John gave Peter a book. There are many factors that influence the interpretation of gapping sentences. Long before Optimality Theory was developed, Kuno (1976) constructed a set of perception rules for the interpretation of gapping. Perquin (1999) adapted Kuno's perception rules and added other constraints to create an Optimality Theory model of gapping.

In recent publications it is suggested that for syntax and semantics, the principle of strict domination (the assumption that only the strongest violated constraint matters) should be abandoned. From her research into the semantics of gapping constructions, Perquin (1999) concludes that weaker constraints can "work together" and overrule a stronger constraint together. It has to be noted that - even though Perquin's Optimality Theory model was constructed to explain the interpretation of gapping - there is not a clear distinction between production and interpretation constraints in her model. Perquin's statements about constraints working together seem consistent with the findings of Jäger and Rosenbach (to appear) on English morphosyntax. They collected experimental data on the production of genitive constructions in English: When do people use the "'s" version (the boy's eyes) and when do they use the "of" version (the eyes of the boy)? Jäger and Rosenbach compared two Stochastic OT models with different predictions on cumulativity - Boersma's Stochastic OT (see Boersma and Hayes, 2001) and a Maximum Entropy model (see Berger *et al.*, 1996; Abney, 1997). Stochastic OT is a variation of Optimality Theory. In standard OT, constraints are ranked on an ordinal scale; in Stochastic OT, a numerical value is assigned to each constraint, so the constraints can be closer together or further apart from each other in the hierarchy. The final ranking of the constraints is only established after a random amount of noise is added to the numerical values of the constraints. By adding the noise, some constraints may change positions in the hierarchy. For Jäger and Rosenbach, cumulativity of constraints appeared to explain their experimental data on the production of genitive constructions best. A Maximum Entropy model - which predicts cumulativity of constraints - corresponds with these findings.

Keller (2001) developed a model for gapping based on Optimality Theory, but with some important differences. In his experiments, he let subjects judge grammaticality of gapping constructions. Therefore, in contrast to Perquin's model of gapping, Keller's model is a model of production, rather than of interpretation. A gradient acceptability pattern emerged from the experimental data: The more constraints were violated, the less acceptable the sentence was for the subjects. From his results, Keller concluded that for the explanation of his gradient grammaticality acceptability data, the assumption of cumulativity of constraints is necessary. He also adopts the idea of the existence of hard constraints (that cause strong unacceptability when violated) in addition to soft constraints (that cause weak unacceptability when violated).

Strict domination makes strong theoretical predictions. It has been very accurate in explaining linguistic, especially phonological, phenomena. Still, in some fields, it might not be sufficient.

1.2 Research question

The question this study will focus on is which model will give the best predictions on the interpretation of gapping in Dutch: An Optimality Theory model in which the strict ordering of constraints is maintained, or an Optimality Theory model in which weaker constraints can co-operate to overrule stronger rules.

Chapters 2 and 3 will deal with theoretical background on gapping and optimality theory. Then, in chapter 4, gapping and cumulativity in OT will be discussed. I developed a computer simulation of the interpretation of possibly gapped sentences. This model can give the optimal interpretation of a sentence according to different evaluation methods - strict domination and several cumulativity hypotheses. In chapter 5, this computer simulation is discussed and in chapter 6 it is evaluated with natural language sentences. To test the accuracy of the computer simulation and to test the cumulativity hypotheses laid out in chapter 4, an experiment was carried out on ten subjects. In chapter 7, the experiment and its results are described. Chapters 8 and 9 form the discussion and the conclusion of this study.

2 Gapping

In this chapter, the phenomenon of gapping will be described. A constraint set for the interpretation of possibly gapped sentences will be established, based on previous literature. In section 2.2, a preliminary constraint hierarchy will be made of the constraint set.

Gapping is a specific kind of elliptic construction. In a conjunction of two sentences, certain (given) information can be omitted in the second conjunct, while some information is left behind. Some examples of gapping are given below:

1a) Karen ontmoet Grace en Jack Stan.
Karen meets Grace and Jack Stan.
1b) Will slaat Jack met een lepel en Stan met een vork.
Will hits Jack with a spoon and Stan with a fork.
1c) Will belooft Jack om Grace te negeren en Stan om Ellen te volgen.
Will promises Jack to ignore Grace and Stan to follow Ellen.

In sentence 1a, "Jack" and "Stan" are called the remnants. The verb "ontmoet" (*meets*) is deleted in the second conjunct. Some sentences with incomplete coordinate constructions can be interpreted in different ways. For example, in sentence 1b Stan could either be the person hitting Jack with a fork, or the person that is hit with a fork by Will. In many other cases of incomplete coordinate constructions, similar ambiguities arise. There are many factors that influence the interpretation of possibly gapped sentences. People will judge different interpretations of a sentence differently in their acceptability. The rules that the interpretation of possibly gapped sentences are subject to, can be represented by OT constraints. The most acceptable interpretation will be picked as optimal. A set of constraints that gapping is subject to will be described in the next section.

From Kuno (1976): In a gapping construction, at least the matrix verb is left out, and gapping leaves behind exactly two remnants. For English this is correct. In Dutch, however, it is possible to have three (NP) remnants, as Neijt (1979) noted. An example of a grammatical incomplete coordinate structure in Dutch:

2)	Grace geeft	Will	de rekening	en	Karen	de ober	het geld.
	Grace gives	*Will*	*the bill*	*and*	*Karen*	*the waiter*	*the money.*

Apparently, the production of gapping constructions in Dutch is subject to different rules than the production of English gapping constructions as far as the number of remnants is concerned. For English, however, Kuno's "two remnant" rule has been questioned. Keller (2001) found experimental evidence that sentences that leave three remnants, are not significantly less acceptable than sentences that leave two remnants.

2.1 Gapping constraints

A set of constraints on the phenomenon of gapping was designed by Kuno (1976). These constraints cover the influences of different fields (such as syntax, pragmatics, semantics) on the interpretation of gapped sentences. Kuno's constraints were clearly designed to explain perception of gapping constructions, rather than production. For gapping, in other publications (Keller, 2001; Perquin, 1999), production constraints and interpretation constraints were used in one system. Keller's system aimed to explain the production of gapping constructions: He measured their grammatical acceptability. Like Perquin, he used Kuno's constraints (Kuno, 1976). Perquin's OT system was designed to analyze the interpretation of gapping constructions. However, the motivation for some of the constraints was (partially) based on production issues - what kind of (in)felicitous sentences could be produced with a certain set/hierarchy of constraints. In this present study, constraints will be formulated purely as interpretation constraints: For each possible interpretation it can be checked whether the constraint in question is violated or not.

Kuno based his constraints on English examples. Here, the examples will all be in Dutch, with an English translation. The applicability of the constraints is generally the same, as Perquin's research (Perquin, 1999; on Dutch) already showed. If any differences may surface, this will of course be discussed.

2.1.1 MINIMAL DISTANCE PRINCIPLE

Minimal Distance Principle (Kuno, 1976)
"The two constituents left behind by Gapping can be most readily coupled with the constituents (of the same structure) in the first conjunct that were processed last of all."

In example 3, "Jan" can either function as the subject or the indirect object of the second conjunct. Because of the Minimal Distance Principle (MinDis), 3b (where "Jan functions as the indirect object) will be the preferred interpretation for sentence 1a.

3a) Mark geeft Tom een koekje en Jan een reep.
Mark gives Tom a cookie and Jan a candy bar.
3b) Mark geeft Tom een koekje en Mark geeft Jan een reep.
Mark gives Tom a cookie and Mark gives Jan a candy bar.
3c) Mark geeft Tom een koekje en Jan geeft Tom een reep.
Mark gives Tom a cookie and Jan gives Tom a candy bar.

2.1.2 FUNCTIONAL SENTENCE PERSPECTIVE

Kuno (1976) defines Functional Sentence Perspective as follows:

Functional Sentence Perspective (Kuno, 1976)
"a. Constituents deleted by gapping must be contextually known. On the other hand, the two constituents left behind by gapping necessarily represent new information and, therefore, must be paired with constituents in the first conjunct that represent new information.
b. It is generally the case that the closer a given constituent is to sentence-final position, the newer the information it represents in the sentence.
c. Constituents that are clearly marked for nonanaphoricity necessarily represent new information in violation of (b). Similarly, constituents that appear closest tot sentence-final position necessarily represent old information (in violation of (b)) if coreferential constituents appear in the corresponding position in the preceding discourse."

It is important to look at FSP (and constraints related to it), and find out how exactly it influences interpretation. Part (a) as formulated by Kuno, is more of a gapping production constraint than a gapping interpretation constraint (deleting of constituents is done by the speaker, not by the hearer). It can be slightly reformulated to make it a true interpretation constraint:

Functional Sentence Perspective as an interpretation constraint
All remnants must be coupled to constituents in the first conjunct that represent new information.

In normal word order, without a guiding context or prosodic information, the constituents closer to sentence-final position will represent the newest information in the sentence. In these instances, the effect of MinDis and the (b) part of FSP is the same: The constituents that are processed last will be coupled to the remnants.

If the word order is different because of topicalization, the situation changes. The topicalized constituent is marked for newness and, according to the general FSP principle, will have to be paired with a remnant. Context can also have an influence on FSP. Consider the next example:

4) Wie gaf aan Piet een koekje? JAN gaf Piet een koekje en KEES een reep.
Who gave Piet a cookie? JAN gave Piet a cookie and KEES a candy bar.

The capitals mean that those words are in focus. Because the context stresses the newness of "JAN", FSP is obeyed when "JAN" and "KEES" are paired.

The influences of FSP discussed above are all of structural nature. Constituents can be lexically "marked for nonanaphoricity" (part (c) of Kuno's definition of FSP). Information is old if "coreferential constituents appear in the corresponding position in the preceding discourse". This notion corresponds with the notion of givenness, discussed by Schwartzschild (1999). He defines givenness thus: "An utterance is given iff it is entailed by prior discourse". From this, it can be inferred that pronouns count as given information: they are entailed by prior discourse. In example 5, as "hem" (*hem*) is given information, "Kees" and "Piet" are coupled.

5 Piet gaf hem een foto en Kees een cadeau.
Piet gave him a photo and Kees a present

Lexical factors (like parallelism of features) do certainly play a role in interpreting gapped sentences. However, as Carlson (2001) concludes from her experimental research on gapping, the influence of lexical parallelism seems to be less important than that of structural factors. The influence of lexical factors is further discussed in paragraph 2.1.5; Featural parallelism.

2.1.3 SUBJECT PREDICATE TENDENCY

The next rule is the tendency for subject-predicate interpretation (further abbreviated as SubPred). This rule is defined by Kuno:

Subject-predicate tendency (Kuno, 1976)
"When Gapping leaves an NP and VP behind, the two constituents are readily interpreted as constituting a sentential pattern, with the NP representing the subject of the VP."

SubPred will always be violated in the non-gapping interpretation, and never in a gapping interpretation. The following sentence can be seen as the SubPred at work:

6) Jan beloofde Piet om te stoppen en Kees om door te gaan.
Jan promised Piet to stop and Kees to continue.

There is a tendency to interpret Kees as subject of the VP in the gapped clause, although this tendency is not very strong. In the hierarchy Perquin made with - among others - Kuno's rules, the subject-predicate rule was the weakest constraint on gapping. However , looking at the interpretations of example 6, it is hard to say which interpretation is the preferred one, the gapping or the non-gapping interpretation. The gapping interpretation violates the Minimal Distance Principle, while the non-gapping interpretation goes against the subject-predicate tendency. Those two might be on the same level in the hierarchy.

2.1.4 SIMPLEX SENTENTIAL RELATIONSHIP

Requirement for a Simplex-Sentential Relationship (Kuno, 1976)
"The two constituents left over by Gapping are most readily interpretable as entering into a simplex-sentential relationship. The intelligibility of gapped sentences declines drastically if there is no such relationship between the two constituents."

The requirement for a Simplex Sentential Relationship (further abbreviated as Simplex) between the two remnants, is best explained with an example. Consider the following sentences:

7a) Kees haalde Jan over om Piet te onderzoeken en Bob Dirk.
Kees persuaded Jan to examine Piet and Bob Dirk.
7b) Kees beloofde Jan om Piet te onderzoeken en Bob Dirk.
Kees promised Jan to examine Piet and Bob Dirk.

Despite their similarity, the coupling of the remnants in the second constituent of the sentence will probably be different for these sentences. There are at least three options for reconstructing the second conjunct:

	Simplex
1) .. en Bob haalde Dirk over om Piet te onderzoeken. *..and Bob persuaded Dirk to examine Piet.*	
2).. en Bob haalde Jan over om Dirk te onderzoeken. *..and Bob persuaded Jan to examine Dirk.*	*!
3).. en Kees haalde Bob over om Dirk te onderzoeken. *.. and Kees persuaded Bob to examine Dirk.*	

Table 1

It is well possible to interpret sentence 7b as in option 2, where Bob promises Jan to examine Dirk. Option 2 is impossible as an interpretation for sentence 7a: There is no way to interpret the second part as stating that Bob persuades Jan to examine Dirk. The difference is the fact that - in the interpretation of option 2 - the remnants in sentence 7b stand in a simplex sentential relationship (Bob will be the person examining Dirk), while the remnants in sentence 7a don't.

2.1.5 FEATURAL PARALLELISM

A hypothesis on parallelism and its influence on gapping was defined by Carlson (2001):

Parallelism hypothesis (Carlson 2001)
"a. The most parallel analysis of a conjoined structure is preferred.
b. An analysis is parallel if featurally similar DP's (determiner phrases) in distinct conjuncts end up with similar syntactic roles (theta-roles and grammatical functions)."

From experimental research that Carlson performed on gapping, it turned out that constituents that had similar features (like animacy) were more likely to be coupled. She used sentences like the following - the English version is the original example:

8a) Alice bakt cakes voor toeristen en Caroline voor haar familie
Alice bakes cakes for tourists and Caroline for her family.
8b) Josh bezocht het kantoor gedurende de vakantie en Sarah gedurende de week.
Josh visited the office during the vacation and Sarah during the week.
8c) Dan verbaasde de juryleden met zijn talent en James met zijn muzikaliteit.
Dan amazed the judges with his talent and James with his musicality.

The stronger parallelism guided towards the gapping interpretation, the more often the gapping interpretation was chosen. However, the subjects were far from unanimous in judging the sentences: even sentence 8a was given a non-gapping interpretation 19% of the time. As Carlson concludes, parallelism does have an influence on interpretation, but structural influences (MinDis in this case) are dominant.

Sentence 8a is different from the others, in the sense that the non-gapping interpretation yields a semantically implausible sentence. Under this interpretation, Alice is supposed to be baking Caroline for her family. The decisive influence in this type of sentence does not come from parallelism: The fact that "bakt" (*bakes*) does not usually take a human object plays a far more important role in this case. This is what the next paragraph - 2.1.6; Selection restrictions - is about.

A special case of parallelism is what Prüst (1992) calls contrastive kinship. Pairs like "vader-moeder" (*father-mother*), "20% van de mensen-80% van de mensen" (*20% of the people-80% of the people*) display contrastive kinship. Perquin (1999) relates this phenomenon to the interpretation of gapping sentences. According to Perquin, pairs like "father-mother" display weak contrastive kinship - which can be violated - and pairs like "80% of the people-20% of the people" display strong contrastive kinship - which can never be violated. Examples of contrastive kinship (all borrowed and translated from Kuno, 1976):

9a) 20% Van de mensen bezocht Amsterdam in 2003 en 80% van de mensen in 2004.
20% Of the people visited Amsterdam in 2003 and 80% of the people in 2004.
9b) 20% Van de mensen vermeed de helft van de mensen in 2003 en 80% van de mensen in 2004.
20% Of the people avoided half of the people in 2003 and 80% of the people in 2004.
9c) Mijn zus bezocht Amsterdam in 2003 en mijn broer in 2004.
My sister visited Amsterdam in 2003 and my brother in 2004.

There is a strong tendency to couple the pairs of words that display contrastive kinship. For example, the second clause of sentence 9a is much more likely to mean "80% van de mensen bezocht Amsterdam in 2004" (*80% of the people visited Amsterdam in* 2004) than "20% van de mensen bezocht 80% van de mensen in 2004" (*20% of the people visited 80% of the people in 2004*). However, in contrast with what Perquin argues, if someone really wants to convey the latter, unlikely meaning, then - with the proper stress pattern - this is possible. Especially in example 9b where the difference in likelihood between the two interpretation alternatives is somewhat smaller, it is certainly possible to achieve either interpretation by changing the stress pattern. The same goes for example 9c. A violable constraint on parallelism, based on Carlson's Parallelism Hypothesis, can be defined as follows:

Featural Parallelism
A remnant is coupled with a constituent that it shares featural characteristics with.

A set of features (for example "animate") for lexical items is defined; if these features agree for the remnant and the constituent it is coupled with, the constraint is not violated. Otherwise it is. The way it is formulated here, is as a binary constraint (either it is violated, or it is not, with no possibilities in between).

Plausibility of the interpretation alternatives does seem to play a role. When the difference in plausibility is decreased, as in 9b, the two interpretation alternatives do not seem to be so far apart in likelihood anymore.

2.1.6 SELECTION RESTRICTIONS

The plausibility of two interpretation alternatives can be different. Verbs may impose selection restrictions on their complements. A constraint on gapping, is that these restrictions are obeyed. In sentence 8a in the previous section, the non-gapping interpretation of the sentence is highly implausible. "Caroline" fits better as a subject in this sentence, because "bakt"(*bakes*) usually requires a human subject and a nonhuman object. In their article on OT and the processing of coordination, Hoeks and Hendriks (2005) propose a constraint called Thematic Fit that can be used for coordination structures:

Thematic Fit (Hoeks and Hendriks, 2005)
"A thematic element must meet the requirements of the thematic role that is assigned to it."

This constraint can be applied to gapping. It is violated in the non-gapping interpretation of sentence 8a and explains why the gapping interpretation of this sentence is preferred.

2.1.7 WORD ORDER CONSTRAINTS

The constraints stated below can form the basis for dealing with word order in interpreting gapping constructions:

A constraint that Hoeks and Hendriks (2005) introduce is Stay:

Stay (Hoeks and Hendriks, 2005)
"Respect canonical word order."

This constraint was introduced in the context of processing coordination structures. The tendency to respect canonical word order can play a role in choosing an interpretation of incomplete coordination structures:

10) Will slaat Jack en Grace Karen.
Will hits Jack and Grace Karen.

In canonical word order, without any constituents being moved, subject comes first. Therefore, in example 10, "Grace" will be coupled with subject "Will" and "Karen" with object "Jack".

Another constraint on word order is the following:

Same Word Order (Perquin, 1999)
Remnants stand in the same word order as the constituents they are coupled with.

At first sight, including these two constraints on word order might seem a bit redundant. Consider the next example:

11)	Aan	Will	geeft	Grace	een koekje	en	Jack	Karen.
	To	*Will*	*gives*	*Grace*	*a cookie*	*and*	*Jack*	*Karen.*

A literal translation in English is given, because the example is only valid in Dutch. Ignoring the semantically anomalous interpretation where "Karen" is coupled to "een koekje" (*a cookie*) for a while, there are two other interpretations left. One with "Jack" as a subject and one with "Karen" as a subject. In case of the first interpretation, Same Word Order is violated and in case of the second interpretation, Stay. It seems that Stay is the stronger of the two, as the preferred interpretation has "Jack" as a subject. So in cases where the verbal arguments of the first conjunct are in canonical word order, Same Word Order and Stay behave exactly alike and when they conflict, Stay overrules. There does not seem to be much use for Same Word Order. However, as Perquin (1999) noticed, there seems to be an extra effect when both these constraints are violated. In the next chapters, this effect will be discussed extensively.

The next example is adapted from Perquin (1999):

12a) Jan gaf Marie een tulp en Sara een narcis.
Jan gave Marie a tulip and Sara a narcissus.
12b) Jan gaf Marie een tulp en een narcis aan Sara.
Jan gave Marie a tulip and a narcissus to Sara.

Both 12a and 12b are felicitous. However, in 12b, the interpretation of the sentence will not be consistent with an interpretation of the remnants in the same word order as the constituents in the first conjunct. This has to do with the fact that with "aan" (*to*) in front of it "Sara" is clearly marked as the indirect object of "gaf" (*gave*). Therefore, I introduce the following constraint:

Overt Syntactic Parallelism

Overt Syntactic Parallelism
The overt syntactic characteristics of a remnant must be consistent with the syntactic role of the constituent it is paired with.

This constraint is violated when, in sentence 12b, "aan Sara" (*to Sara*) is paired with "een tulp" (*a tulip*), because "een tulp" is the object of the first conjunct, and the overt form of "aan Sara" prevents it from fulfilling the role of object.

In Perquin (1999) the following example is given (in English):

13a) Kim is behoorlijk stom en Lou een grote idioot
Kim is rather foolish and Lou a complete idiot

The next example seems similar to 13a:

13b) *Kim zoent goed en Lou een mooi meisje.
**Kim kisses well and Lou a beautiful girl.*

The acceptability of sentence 13b is highly doubtful. In sentence 13a, the role of "behoorlijk stom" (*rather foolish*) is nominal predicate. An NP like "een grote idioot" (*a complete idiot*) can also function as a nominal predicate. Therefore "behoorlijk stom" and "een grote idioot" can be coupled. In 13b, "goed" (*well*) is an adverbial phrase. The NP "een mooi meisje" cannot function as an adverbial phrase and can therefore not be coupled with "goed": OvPar is violated in that case.

2.1.8 OVERVIEW OF CONSTRAINTS

Kuno's rules were used as a basis, sometimes slightly adapted for current purposes. Other constraints were added, yielding the following set of constraints (as yet not in hierarchical order!):

Minimal Distance Principle
Functional Sentence Interpretation
Subject Predicate Tendency
Simplex Sentential Relationship
Featural Parallelism
Thematic Fit
Stay
Same Word Order
Overt Syntactic Parallelism

It is needless to say that this set is not complete. However, it covers quite a few aspects of gapping and for current purposes it will suffice.

2.2 Constraint ranking

Before cumulativity effects can be investigated, a constraint ranking must be established. In this section, such a hierarchy will be made, based on single constraint violations, to exclude possible cumulativity effects. The effects of cumulativity will be analyzed later. For Kuno's (1976) constraints Minimal Distance Principle, Simplex Sentential Relationship and Subject Predicate Tendency, the ordering has earlier been established as being Simplex > MinDis > SubPred (Keller, 2001 for English; Perquin, 1999 for Dutch). The other constraints will be fit into the hierarchy, based on informal acceptability judgements.

2.2.1 Functional Sentence Perspective

A sentence in which the position of FSP in the constraint hierarchy is compared to that of Simplex, is the following:

14) Will overtuigt hem Jack te slaan en Stan Theo.
Will persuades him to hit Jack and Stan Theo.

"Him" in this sentence is old information, so coupling remnants with this constituent will violate FSP. To avoid violating FSP at all, the remants need to be coupled with new information, the constituents "Will" and "Jack" in this sentence. This violates Simplex. This latter interpretation is the preferred one for this sentence. FSP will be placed above Simplex in the hierarchy:

FSP > Simplex > MinDis > SubPred

2.2.2 Overt Syntactic Parallelism

As this is a syntactic constraint, the prediction is that it will be high in the constraint hierarchy. Will it be ranked higher than FSP?

15) Will ziet haar in het park en Karen Stan.
Will sees her in the park and Karen Stan.

The preferred coupling of remnants is "Karen" with "Will" and "Stan" with "haar" (*her*), even though it violates FSP. Interpretations in which either of the remnants is coupled with "in het park" (*in the park*) will not be chosen, because they violate OvPar. OvPar is higher ranked than FSP:

OvPar > FSP > Simplex > MinDis > SubPred

2.2.3 Stay, Same Word Order, Featural parallelism and Thematic Fit

The word order constraints have a great deal of overlap. Looking at single violations, having both Stay and Same Word Order in the hierarchy does not seem justified (see also section 2.1.7). Stay is above Same Word Order in the hierarchy:

16)	Hem	adoreert	Will	en	Stan	Jack.
	Him	*adores*	*Will*	*and*	*Stan*	*Jack.*

In Dutch, this construction is a bit odd, but acceptable with the right intonation. The most likely interpretation is that where "Stan" functions as subject and canonical word order is respected.

With the next example, Perquin (1999) ranks thematic requirements of the verb above the requirement to respect canonical word order:

17)	Een	roos	plukt	vader	en	een	tulp	moeder.
	A	*rose*	*picks*	*father*	*and*	*a*	*tulip*	*mother.*

This is the same kind of construction as example 16, but Thematic Fit plays a role here. The preference for an interpretation that respects canonical word order is gone here. The following hierarchy arises for these three constraints:

ThemFit > Stay > Same

The position of ThemFit can be compared to the position of FSP in the hierarchy:

18) Will gaf haar een koekje en Jack Karen.
Will gave her a cookie and Jack Karen.

"Will" and "een koekje" (*a cookie*) are new information and if the sentence is pronounced, these constituents are stressed. Despite the strange meaning of the sentence, the interpretation where "Jack" is coupled to "Will" and "Karen" to "een koekje" (*a cookie*) is most likely to be chosen. FSP is stronger than ThemFit. For now, it is assumed that, regarding newness of constituents, the stress pattern matches the use of pronouns of the sentence. The presence of a pronoun and a lack of stress indicate that the information is given, i.e. not new. If these two factors do not match, prosodic information is probably stronger than whether a pronoun is used or not (try this by pronouncing sentence 18 with different stress patterns). This study focuses on non-prosodic information; in section 8.2 the role of prosody (and context) will be discussed further.

The fact that FSP is stronger than ThemFit entails that FSP is also stronger than Stay and Same:

FSP > ThemFit > Stay > Same

The position of ThemFit can be compared to that of MinDis in the hierarchy:

19) Will gaf Stan een cadeau en Jack Grace.
Will gave Stan a present and Jack Grace.

Coupling "Jack" with "Will" and "Grace" with "Stan" is the most natural interpretation: ThemFit seems stronger than MinDis:

ThemFit > MinDis

Featural Parallelism does not seem to be a very strong constraint on its own. In the next example the effect of a violation of FeatPar is compared to a violation of MinDis:

20) Grace bezoekt het ziekenhuis in de zomer en Will in de winter.
Grace visits the hospital in the summer and Will in the winter.

Even though "Will" and "Grace" have more featural parallels than "Will" and "het ziekenhuis" (*the hospital*), in this example MinDis has a stronger influence than FeatPar. "Will" is most likely to be coupled to "het ziekenhuis" (*the hospital*). Therefore, FeatPar will be considered a weaker constraint than MinDis.

MinDis > FeatPar

Finding out the strength of SubPred compared to FeatPar in a strict domination hierarchy is not easy, as MinDis always plays a role where SubPred plays a role. MinDis will always be decisive in these cases.

2.2.4 Summary

The following is a summary of what we can say about the constraint hierarchy based on conflicts between single constraints:

OvPar > FSP > Simplex > MinDis > SubPred
FSP > ThemFit > Stay > Same
ThemFit > MinDis
MinDis > FeatPar

For a number of constraints (for example, Same and Simplex), it is not possible to create minimal pairs to test their comparative position in the hierarchy. The reason for this can be that for certain constraints, violations can only be forced by violating semantically based constraints (in the case of Same), or that there is always a higher constraint involved that will be decisive (in the case of SubPred, which MinDis overrules). With the information that has been collected on the constraint hierarchy, different empirically correct constraint hierarchies can be created.

3 Optimality Theory issues

In this chapter, relevant aspects of OT are discussed: Sections are dedicated to the relationship between production and interpretation constraints, and to existing literature on constraint cumulativity.

3.1 Production vs. interpretation constraints

Traditionally, the focus in OT has been on the production of language. In phonology, the main field which OT used to be applied to, the linguistic form was the output of the model. Blutner *et al.* (to appear) investigate new perspectives in OT. In OT syntax, the perspective of the speaker is taken. The input consists of a representation of meaning and the output of the optimal form for that meaning. As OT research also begins to focus on other linguistic areas (such as semantics and pragmatics), a perspective change is necessary. In OT semantics and pragmatics the perspective of the hearer needs to be taken into account. If the perspective of the hearer is taken, the input of an OT model consists of a linguistic form, and the output of the optimal interpretation of the form. The two points of view are closely related, but the input-output pattern for the corresponding OT model is different.

3.1.1 MODEL FOR GAPPING

In the case of an OT model for the interpretation of gapping, an incompete coordinate construction will be the input for the model. Interpretation constraints select the optimal interpretation from a range of possible interpretations. This optimal interpretation will be the output of the system. This information about input and output must be kept in mind for the formulation of the constraints. Every constraint must be formulated such that every interpretation candidate can easily be evaluated.

Related to this, it must also be mentioned that a model of interpretation of gapping does not deal with the grammatical acceptability of the form as such. Keller (2001) let subjects judge grammaticality of gapping constructions. The question was really what kind of gapping constructions would be considered acceptable to produce. He used a cumulative model of gapping: The more constraints were violated while producing a sentence, the less acceptable the sentence was going to be for the subjects. The concept of cumulativity will be further explained in section 3.2 and chapter 4. The application of OT in this case is traditional in the sense that it takes the production perspective, but different from earlier work, because there was only one candidate to evaluate with the constraint hierarchy: The sentence in question. This sentence formed the input of the model and the output was a label "acceptable" or "unacceptable". Keller did use Kuno's perception rules, but to construct a model of gapping production. Perquin's purpose (Perquin, 1999) was to make a model of interpretation of gapping constructions in Dutch. She also used Kuno's perception rules for gapping and added other constraints to complete the model.

Summarizing, for constructing a model of interpretation of gapping constructions, the nature of the input and output must be kept in mind. With the constraint hierarchy for interpreting incomplete coordinate structures, the optimal interpretation among the candidate interpretations must be chosen. The constraints must be formulated in such a way, that candidate interpretations can easily be evaluated when presented to the constraints.

3.2 Cumulativity: Earlier work

In Harmonic Grammar, the predecessor of OT, cumulativity is an implicit feature. Each constraint in the constraint hierarchy has a (positive or negative) weight value. For each candidate a so-called Harmony value is calculated by summing the weighted constraints. This is the cumulative aspect of the model; each constraint violation adds to the Harmony value of the candidate. From a set of candidates, the one with the highest Harmony value will be selected as the optimal candidate.

3.2.1 STRICT DOMINATION

Prince and Smolensky (2004/1993) made an observation from empirical evidence: It appeared that in many (phonological) cases, no combination of lower ranked constraint violations could decrease the harmony of a candidate in such a way that its harmony would end up lower than the harmony of a candidate that violated one higher constraint. Prince and Smolensky adopted the notion of "strict domination" for OT: Weight values can be chosen such that there is no sum of lower ranked constraint violations that can add up to be more serious than the violation of a single higher ranked constraint. In fact, it is no longer necessary to calculate a Harmony value of a candidate. It is enough to determine, per candidate, the highest constraint that is violated. All other violations of lower constraints are no longer relevant and can be ignored in determining the optimal candidate. Prince and Smolensky adopted strict domination as a standard in OT.

Strict domination hierarchies are consistent with the often-repeated credo "Grammars cannot count": On the symbolic level that OT deals with, a non-mathematical account of language processing is preferred over the numerical account of Harmonic Grammar. A functional argument for strict domination - that Legendre *et al.* (2005) mention - is the fact that grammar must be sharable: A grammar must be so robust, that it is possible to produce the same optimal candidates over and over again. A global maximum must be reached, not a series of local maximums. They also suggest that strict domination enables people to learn a grammar more efficiently.

3.2.2 GANGING UP AND COUNTING CUMULATIVITY

Jäger and Rosenbach (to appear) distinguish two kinds of cumulativity in their stochastic model: Ganging up and counting cumulativity. Ganging up cumulativity is the kind of cumulativity where single violations of lower ranked constraints can add up to be stronger than a violation of a single higher constraint, as shown in table 2.

	Constraint 1	Constraint 2	Constraint 3
Candidate x		*	*
Candidate y	*		

Table 2

In this tableau, candidate x were the optimal candidate if we would opt for strict domination. Using ganging up cumulativity, however, the added violations of constraint 2 and 3 (candidate x) will be more serious than the violation of only constraint 1 (candidate y). Candidate y will be selected as the optimal candidate.

The other notion of cumulativity that Jäger and Rosenbach describe is counting cumulativity. In this notion of cumulativity, multiple violations of a single constraint also contribute to the optimality of a candidate.

	Constraint 1	Constraint 2	Constraint 3
Candidate x		**	
Candidate y	*		

Table 3

In table 3, candidate x violates constraint 2 two times. In a traditional OT model, candidate x is the optimal candidate in this tableau. The fact that constraint 2 is violated more than once, does not play a role in determining the optimal candidate. For a model that takes counting cumulativity into account, candidate y will be optimal: The two violations of constraints add up to be stronger than the single violation of constraint 1.

3.2.3 DIFFERENT SYSTEMS FOR DIFFERENT DATA

OT is used to explain different kinds of linguistic data. The way OT was used traditionally, in phonology, was to select an optimal candidate from categorical data. It is suggested that strict domination might be the best way to explain certain kinds of data, and that, for other kinds of data, it might be necessary to use cumulativity. In Legendre *et al.* (2005), Harmonic Grammar and traditional OT, and with that cumulativity and strict domination, are compared. They suggest two systems of constraints. One "more strictly "grammatical", interacting exclusively or primarily via strict domination" and another, "a set of more pragmatically-based constraints, reflecting more directly, perhaps, statistical characteristics of experience, and interacting in a less restricted manner, via arbitrarily weighted constraints". An example of the latter category comes from Legendre *et al.* (1990). They developed a (inherently cumulative) Harmonic Grammar model to explain the acceptability of unaccusativity/unergativity constructions in French. The constraints they used were of syntactic and semantic nature. Both the unaccusativity-unergativity and the acceptability scale were gradient. Their model, using the constraints cumulatively, appeared to be able to explain the data.

Keller (2001) is also dealing with gradient data in his research on gapping. He uses OT as a model for the acceptability of gapping constructions. Keller argues that it is not enough to determine an optimal candidate and dismiss all others. Because acceptability is a gradient phenomenon, attention needs to be paid to suboptimal candidates as well. He adopts the Suboptimality Hypothesis:

a. Suboptimal candidates differ in grammaticality
b. The relative grammaticality of suboptimal candidates can be used as evidence for constraint rankings.

Intuitively, this hypothesis seems to make more sense for grammatical acceptability data than a situation where only the optimal candidate is considered and all suboptimal candidates are ignored. However, adopting this hypothesis yields problems. In traditional OT, only relative optimality values are considered: Candidates can only be compared within a candidate set. For grammaticality data, this means that it would not be possible to compare grammatical acceptability across candidate sets (Keller, 2001). A way to avoid this problem in Harmonic Grammar, is to work with absolute values - Harmony values. By using these, it would be possible to compare candidates across candidate sets. Legendre *et al.* (2005) point out that this is at odds with the connectionist basis of Harmonic Grammar. A connectionist system works with a relative Harmony value in a local context, to decide an optimal candidate. It has no access to absolute Harmony values. Another problem with the Suboptimality Hypothesis that Keller (2001) describes, is that for every difference in optimality, a difference in grammaticality is predicted. The model will predict more levels of grammaticality than is found in the data. To solve the problems with the Suboptimality Hypothesis, Keller developed the Constraint Reranking Model (Keller, 1998). In this model, the degree of grammatical acceptability of a structure depends on the number of constraints rerankings that are necessary to make the structure optimal. This way of analyzing data makes it possible to compare candidates across candidate sets. Furthermore, in the reranking model Keller distinguishes between hard and soft constraints. Hard constraints cause strong unacceptability when and soft constraints cause weak unacceptability when violated. From Keller's experiments it appeared that that there was only a difference in grammatical acceptability between violations of hard and soft constraints, not between different kinds of hard or soft constraints. This yields a large reduction in grammatical acceptability levels.

Summarizing: There are indications that different types of data call for different applications of OT. Categorical data as analyzed in phonological examples, seem to be best explained with a strict domination OT system. For semantic/syntactic data, this might well be different: Semantic and syntactic phenomena are often a lot less black and white than for example phonogical phenomena. Semantic and syntactic phenomena depend on context and context can vary endlessly. These types of data seem to display gradient properties. There are indications that for this type of data, a cumulative system might work better than a strict domination system.

4 Integrating cumulativity in OT

Several authors have suggested that in some cases, constraints interact cumulatively (Keller, 2001; Jäger and Rosenbach, to appear). In this chapter, constraint cumulativity in the specific case of processing incomplete coordination structures will be discussed: In which ways could lower ranked constraints add up to overrule a higher ranked constraint? Then it will be described how those versions of cumulativity can be implemented in a computer model that simulates the interpretation of incomplete coordination structures.

4.1 Cumulativity in gapping structures

In section 3.2.2 the concepts of counting cumulativity and ganging-up cumulativity were described, as introduced by Jäger and Rosenbach (to appear). Counting cumulativity entails that multiple violations of a single constraint add up. In the present study all constraints have been formulated in such a way that they are either violated, or they are not. For most of the constraints, this is the only imaginable way to formulate them. Take Kuno's (1976) Subject Predicate Tendency for example, that states that a combination of an NP and a VP tend to be interpreted as a subject-predicate construction. There is no possible way to violate this constraint more than once in one sentence. However, for other constraints this is different. Featural Parallelism (defined in section 2.1.5) states that a remnant is coupled with a constituent that it shares featural constituents with. As it is defined, it is a black and white constraint: Either the constituents share their featural characteristics and the constraint is obeyed, or the constituents are different in their features and the constraint is violated. Counting cumulativity could be applied to Featural Parallelism, if it was defined as a non-binary constraint. It is quite complicated to apply such a constraint: How do you count violations of parallelism? For now, this question is left open, but future experimental research could help answer it. This study focuses on ganging-up cumulativity (constraints can "gang up" to beat a stronger constraint; Jäger and Rosenbach, to appear); only binary constraints are defined.

Keller (2001) found evidence for cumulativity in his experimental research on gradient acceptability of gapping constructions. The more violations a sentence contained, the less acceptable it was to the subjects. Even though Keller's gradient acceptability approach is obviously different from the approach in this study, it can be interesting to investigate whether his findings are also reflected in the interpretation of possibly gapped sentences.

In Perquin (1999) two specific cases are given of possible instances of cumulativity in incomplete conjunction structures:

a. A violation of both a constraint that states that remnants are preferably processed in the same order as the antecedents and a constraint that remnants are preferably processed in canonical word order (introduced as respectively Same Word Order and Stay in this text) can overrule a violation of the higher ranked constraint that states that a thematic element must meet the requirements of the thematic role that is assigned to it (called Thematic Fit here). This constraint is ranked directly above the other two.

b. The combination Minimal Distance Principle, Simplex Sentential Relationship and Subject-Predicate Tendency (adjacent constraints ordered Simplex Sentential Relationship > Minimal Distance Relationship > Subject Predicate Tendency) can overrule pretty much any other constraint, except for a specific instance of Functional Sentence Perspective that Perquin introduces: Strong Contrastive Kinship (this phenomenon is described in section 2.1.5).

Experimental and theoretical research has brought forward these concrete proposals for the existence of cumulativity in incomplete coordination structures. These will form the basis of the comparison of a non-cumulative model with a cumulative model of gapping interpretation.

4.2 Cumulativity in OT: But how?

Within the framework of OT, Jäger and Rosenbach (to appear) found evidence of cumulativity in experimental data on the production of syntactic constructions. Their research focused on variations in the use of English genitive constructions. The production of two types of genitive construction is compared: The "'s" variation ("the boy's eyes) and the "of" variation ("the eyes of the boy"). Factors that influence the choice for either the "'s" or the "of" construction were used as constraints. These were animacy, topicality and possessive relation. Subjects had to choose between the two genitive variations in fragments of text. As a result, the constraints were ordered animacy > topicality > possessive relation.

	animacy	topicality	possessive relation
Candidate x	*		
Candidate y		*	*

Table 4

The violation pattern was as shown above. Jäger and Rosenbach use probabilistic models to determine the optimal candidate: In table 4, Candidate x turns out to have the highest probability of being produced. Assuming that the constraints have been formulated and ordered correctly, the only way to explain this phenomenon is to implement cumulativity in the model. But how exactly are we to implement it in an (non-probabilistic) OT system?

4.2.1 CUMULATIVITY MODELLED THROUGH WEIGHT VALUES

Harmonic grammar (Legendre, Miyata and Smolensky, 1990; see also chapter 1) formed the basis for Optimality Theory. This grammar is cumulative in nature: The harmony value is calculated by adding up the weighted sum of constraint violations. OT can be seen as a form of Harmonic Grammar where the weight values are chosen such that no combination of violations of lower ranked constraints can ever decrease the harmony of a candidate in a in such a way that its harmony will end up lower than the harmony of a candidate that violates one higher ranked constraint.

For the specific (possible) instances of cumulativity in the case of incomplete coordination structures, it might be possible to choose the weight values such that the system gives the desired result. With abstract examples, the possibilities of cumulativity will be explored. First two examples of constraint hierarchies with weight values:

Constraint hierarchy	A	B	C	D	E	F	G
Weight value	1	0.9	0.8	0.7	0.6	0.5	0.4
Candidate 1		*	*				
Candidate 2	*						

Table 5: An example of a cumulative system in which many combinations of lower ranked constraints can "overrule" higher ranked constraints.

Constraint hierarchy	A	B	C	D	E	F	G
Weight value	1	1/2	1/4	1/8	1/16	1/32	1/64
Candidate 1		*	*	*	*	*	*
Candidate 2	*						

Table 6: An example of a strict domination constraint hierarchy.

These two examples show a cumulative system and a strict domination system, modelled by choosing suitable weight values. Table 5 is an example of a cumulative system. The weight values are fairly close together; many combinations of lower ranked constraints can overrule a violation of a higher ranked constraint. Table 6 is an example of a strict domination hierarchy: No combination of violations of lower ranked constraints can overrule the violation of a higher ranked constraint. In table 5, candidate 1 violates both B and C and candidate 2 only A. Candidate 2 will be chosen as the optimal candidate: The weighted sum of constraint violations of candidate 1 is 0.9 + 0.8 = 1.7. The weighted sum of constraint values of candidate 2 is 1. For the system of table 5 many instances can be given in which a combination of lower ranked constraints can overrule a violation of a higher ranked constraint. In table 6 every weight value is half of the previous one. If it is assumed that every constraint can only be violated once, cumulativity effects are impossible with such a weight distribution. Candidate 1 violates constraints B through G: The weighted sum of violations is 0.98. So candidate 1 with all its violated constraints will be optimal, even if there is a candidate 2 that violates only constraint A (with weight value 1).
The two OT systems shown above represent two extremes: In table 5, there is a high degree of cumulativity and in table 6, there is no cumulativity at all. The weights can be adjusted such that a limited degree of cumulativity is achieved.

The system in table 6 was just an example of how strict domination can be modelled with weight values. If we choose a weight value for the highest ranked constraint, a formula can be deduced to calculate the other weight values in such a way that the system is governed by strict domination. For any constraint, the weight value of the constraint right under it in the hierarchy can be calculated by multiplying it with a factor ½ or smaller – this way strict domination is obtained. This is shown in the following calculation, where n is the rank number of the constraint and N the total number of constraints. It is known that any weight W_n must be greater than W_{n+1} plus the other weights from W_{n+2} until W_N and any weight W_{n+1} must be greater than all the weights from W_{n+2} until W_N:

(I) $$W_n > W_{n+1} + \sum_{r=n+2}^{N} W_r$$

(II) $$W_{n+1} > \sum_{r=n+2}^{N} W_r$$

To calculate cut-off values:

(III) $$W_n = W_{n+1} + \sum_{r=n+2}^{N} W_r$$

(IV) $$W_{n+1} = \sum_{r=n+2}^{N} W_r$$

Insert (IV) in (III):
(V) $W_n = 2^*W_{n+1}$

This means that:
(VI) $W_n \geq 2^*W_{n+1}$

Or, the other way around:
(VII) $W_{n+1} \leq 1/2^* W_n$

The weight values of the lowest ranked constraints, W_N and W_{N-1}, are special cases. A relationship between W_N and its successor cannot be described, simply because there is no successor. As for W_{N-1}, a relationship with its successor can be described, but it is different from the higher ranked constraints and their successors:

(VIII) $$W_{N-1} > W_N + \sum_{r=(N-1)+2}^{N} W_r$$

(IV) $$\sum_{r=(N-1)+2}^{N} W_r = 0$$

Combining (VIII) and (IV):
(X) $W_{N-1} > W_N$
(XI) $W_N < W_{N-1}$

The two examples from table 5 and 6 are basic examples of what systems with weight values can look like. We have to look at the specific needs of a system that can explain the interpretation of incomplete coordination structures.

4.3 Empirical realization of cumulativity

In section 4.1, three concrete proposals from earlier work were discussed. Three non-stochastic kinds of cumulativity can be coupled to these proposals:

I Majority cumulativity
A system where a candidate with a higher number of constraint violations is always a worse candidate than a candidate with fewer constraint violations. If candidate 1 violates four constraints and candidate 2 violates two constraints, candidate 2 will be optimal: Hierarchical relations between constraints do not play a role.

II Local restricted cumulativity
A system where two of the constraints can overrule the constraint directly above them. So in a hierarchy with the following constraints: B > C > D, C and D together are able to overrule B.

III Global restricted cumulativity
A system where a number of constraints can overrule any higher ranked constraint. For example, in a hierarchy with constraints A > B > C > D > E, if a combination of C, D and E could overrule A, then there would be global restricted cumulativity.

Can a weight value system be used to implement those varieties?

4.3.1 MAJORITY CUMULATIVITY

For majority cumulativity, the hierarchical relation between constraints is irrelevant. The candidate with the fewest constraint violations is always optimal. Of every candidate, the total number of constraint violations must be counted. If there is more than one candidate with the lowest number of violations, the best of these candidates is found by strict domination.

4.3.2 LOCAL RESTRICTED CUMULATIVITY

System II has to contain some sort of selective cumulativity. Generally, it must function like a strict domination hierarchy with two specific constraints behaving differently. Below is an example of such a system, in which constraints C and D together overrule constraint B.

Constraint hierarchy	A	B	C	D	E	F	G
Weight value	1	0.50	0.26	0.25	0.009	0.0045	0.0022

Table 7: An example of a hierarchy where constraints C and D together overrule constraint B.

Note the small weight value for E: It needs to be lower than 0.01 (the difference between C and D) to make sure that a combination of D and E cannot overrule C. From constraint E on, the hierarchy is a strict domination hierarchy again. The weight values are chosen such that every constraint's weight value is half of that of its predecessor (see section 4.2.1).

4.3.3 GLOBAL RESTRICTED CUMULATIVITY

Global restricted cumulativity is characterized by the fact that a number of constraints can overrule all other constraints. To evaluate whether this can be modelled with weight values, a set of equations that such a system needs to obey are given. Consider a system with the following constraints A > B > C > D > E, that behaves like a strict domination hierarchy, except for the fact that when C, D and E are violated, they overrule all the other constraints. These are the corresponding equations:

(1) A > B + C +D + E
(2) B > C + D + E
(3) C > D + E

(4) D > E
(5) A < C + D + E

It is not possible to find a set of weight values that matches these equations: Equations (1), (2) and (5) are conflicting. If C + D + E needs to be higher than A and B higher than C + D + E, then A can never be higher than B + C + D + E. The only conclusion that can be drawn is that, if we want to implement this type of cumulativity, a way to do it without weight values must be found.

The type of cumulativity suggested in system (III), global restricted cumulativity, cannot be captured by a system with weight values. To be able to implement this type of cumulativity, another evaluation method must be introduced. There is no way to implement this type of cumulativity that is compatible with existing OT literature. By adding a meta-constraint to the system, the desired cumulativity in system (III) can be obtained. This meta-constraint, we can call it CDE, is only violated when C, D and E are all violated. It is located above A in the hierarchy. The constraints interact via strict domination. Below is an example of a hierarchy with a meta-constraint that is violated whenever C, D and E are all violated.

Constraint hierarchy	CDE	A	B	C	D	E
Candidate a	*!			*	*	*
Candidate b		*	*	*		

Table 8: An example of a hierarchy with a meta-constraint that is violated whenever C, D and E are all violated.

This hierarchy yields output as proposed. The use of meta-constraints is incompatible with OT as it is: A second evaluation round is necessary to check the meta-constraintt (after the first round where the other constraints have been evaluated). There is no evidence in OT that constraint evaluation takes place in several phases.

5 Computer simulation

With the theoretical framework constructed in the earlier chapters, a computer simulation of the interpretation of incomplete coordinate structures will be made. In this chapter, the construction of the computer model will be discussed.

5.1 Technical details

The goal of the computer model will be to give a prediction of the interpretation of incomplete coordinate structures with four methods of evaluation: Strict domination and three types of cumulativity. Earlier work in the area of implementation of OT interpretation in Dutch was done by Bouma (2003): He developed a model for pronoun resolution. For that system, syntactic information from the Alpino Parser (Bouma *et al.*) for Dutch was used. For now, the syntactical information needed for my system will be provided by hand, with the input. In the future this system could be combined with a syntactic parser like Alpino. Then the syntactic information of the sentence will not have to be provided with the input, but can be retreived from the automatic parser. In an ideal situation, all of the information that now has to be provided with the input, can be retrieved automatically.

5.1.1 WHAT GOES IN, WHAT COMES OUT?

The input of the system will be a single sentence at the time, consisting of two conjuncts separated by a conjunction element. It is important to be sure that the right information is provided for the evaluation of an incomplete sentence. For the evaluation of the constraints formulated in chapter 2 certain information about both the sentence and its components needs to be available. This information about the sentence given with the sentence itself:

- Division of both conjuncts in syntactic components;
- Syntactic function of the components in first conjunct;
- Information on newness of the NP's in the first conjunct.

Sentence-independent information about words and constituents is stored in a lexicon (that is loaded with the program). The following information is stored in the lexicon:

- For auxiliary verbs with verbal complements: Whether they constitute a simplex relationship with all verb arguments or not. If they do, Simplex Sentential Relationship cannot be violated;
- Selection restriction information of the main verb of the first conjunct;
- (Syntactic and semantic) Featural information of the words in the sentence.

The sentential information about the first conjunct must be given with the sentence. An example of input with the sentence: "Will geeft Karen een koekje en Jack Grace" (*Will gives Karen a cookie and Jack Grace*):

21) `analyze([[will,subj,given],[geeft,vh],[karen,io],[[een, koekje],do],[en,coord],[jack],[grace]]).`

The sentence is a Prolog list, consisting of lists with information about the constituents, arranged in the following manner: [Constituent, SyntacticRole, NewnessInfo]. NewnessInfo is an optional element: If no information is provided about the newness of a constituent, it is assumed that the element is new. A constituent can either consist of one element or of a list of elements - as [een, koekje], with a determiner and a noun. In the second conjunct, the constituents are also marked with square brackets. The words in the sentence are stored in a lexicon: This lexicon contains word-specific information (for example its syntactic category).

The program is not a parser in the sense that it provides syntactic information to the user. The program has a built-in lexicon. Information about words in the sentence is stored in the lexicon; in the grammar file basic grammar rules about how a word group can be put together are stored. Some examples of lexical and word group information:

22)
```
word(theo,np,[hum]).
word(het,det,_).
word(koekje,n,[nonhum]).
```

Single words are stored with the predicate word(Word, SyntacticCategory, Features). In the grammar file some grammatical rules such as the following are stored:

23)
```
element([Det,N],np,Properties) :- word(Det,det,_),
word(N,n,Properties).
```

An element is a word or a group of words that can function as a constituent in sentence (but can also be part of a constituent). As described above, for the first conjunct, the constituents are already provided and labelled. Syntactic rules are needed to provide a label (PP, for example) for the constituents in the second conjunct. For now, most of the constituents and their label are entered into the grammar file as a whole, because the system only contains very simple grammar rules and cannot handle all constituents. In the future the grammar module could be extended or replaced by the use of another syntactic parser.

The output of the system consists of two elements. Firstly, an overview of interpretations (possible couplings of remnants with constituents from the first conjunct) and the constraints they violate, is given. For each possible interpretation, the program gives a reconstruction of the second conjunct – the remnant is placed on the location of the element it is coupled with. Secondly, per method of evaluation, the optimal interpretation is given. Below, an example of in- and output for the sentence "Will geeft Karen een koekje en Jack Grace" (*Will gives Karen a cookie and Jack Grace*) is given, where "Will" is marked as given:

23)
```
3 ?-analyze([[will,subj,given],[geeft,vh],[karen,io],
[[een, koekje],do],[en,coord],[jack],[grace]]).

The following interpretation:
[[[jack, 1, [subj, do, io]], [will, 1, subj, given]], [[grace,
2, [subj, do, io]], [karen, 2, io]]]
#jack# geeft #grace# [een, koekje]
violates these constraints: [fsp, minDis]

The following interpretation:
```

```
[[[jack, 1, [subj, do, io]], [will, 1, subj, given]], [[grace,
2, [subj, do, io]], [[een, koekje], 3, do]]]
#jack# geeft karen #grace#
violates these constraints: [fsp, featPar, themFit, minDis]

The following interpretation:
[[[jack, 1, [subj, do, io]], [karen, 2, io]], [[grace, 2,
[subj, do, io]], [will, 1, subj, given]]]
#grace# geeft #jack# [een, koekje]
violates these constraints: [fsp, stay, same, minDis]

The following interpretation:
[[[jack, 1, [subj, do, io]], [karen, 2, io]], [[grace, 2,
[subj, do, io]], [[een, koekje], 3, do]]]
will geeft #jack# #grace#
violates these constraints: [featPar, themFit]

The following interpretation:
[[[jack, 1, [subj, do, io]], [[een, koekje], 3, do]], [[grace,
2, [subj, do, io]], [will, 1, subj, given]]]
#grace# geeft karen #jack#
violates these constraints: [fsp, featPar, themFit, stay,
same, minDis]

The following interpretation:
[[[jack, 1, [subj, do, io]], [[een, koekje], 3, do]], [[grace,
2, [subj, do, io]], [karen, 2, io]]]
will geeft #grace# #jack#
violates these constraints: [featPar, themFit, stay, same]


****STRICT DOMINATION****

A reconstruction of the optimal second conjunct with the
method nonCumu is:
will geeft #jack# #grace#


****LOCAL RESTRICTED CUMULATIVITY****

The harmony value of the optimal second conjunct is 0.250001.
A reconstruction of the optimal second conjunct with the
method localCumu is:
will geeft #jack# #grace#

****GLOBAL RESTRICTED CUMULATIVITY****

A reconstruction of the optimal second conjunct with the
method globalResCumu is:
will geeft #jack# #grace#

****MAJORITY CUMULATIVITY****

There is more than one candidate with the lowest number of
violations.
```

```
Therefore, strict domination will determine the optimal
candidate.
The candidate with the lowest number of violations has 2
violation(s).
A reconstruction of the optimal second conjunct with the
method majorityCumu is:
will geeft #jack# #grace#
```

In this example, all methods yield the same optimal interpretation.

5.1.2 CONSTRAINTS

The constraints are integrated in the system as predicates.
Of course the system needs a constraint hierarchy. This hierarchy consists of a predicate with as a single argument a list of constraints. The order of these constraints can be easily changed for research purposes. An example of a constraint hierarchy:

24) `constrainthierarchy([ovPar,fsp,featPar,themFit,stay, same,simplex,minDis,subPred]).`

This constraint hierarchy has ovPar (Overt Syntactic Parallelism) as the highest ranked constraint and SubPred (Subject Predicate Tendency) as the lowest ranked constraint.

For cumulativity methods with weight values, each constraint receives a weight value, stored in a predicate such as the following:

25) `cumuweight(fsp,0.9).`

For each constraint, a predicate has been included to check whether it is violated.

5.1.3 WHAT DOES THE SYSTEM DO?

The system follows the segmented structure of OT frameworks as proposed by Prince and Smolensky (2004/1993). In the first phase, a set of possible interpretations of a sentence is generated (analogous to their "Gen") and in the second phase the optimal candidate among the interpretation candidates is determined with the constraint hierarchy (analogous to "H-eval").

From the input sentence, all constituents are collected, except for the verbal head. The constituents from the first conjunct can be coupled with the remnants. The constituents from the second conjunct are the remnants. Both the remnants and the constituents they can be coupled with are put in lists.

Now the interpretation candidates can be generated: Each candidate is a set of all the remnants, each coupled with a constituent from the first conjunct[1]. It is made sure that for each interpretation, the remnants are coupled to different constituents. This process results in a list of candidate interpretations.

---

[1] The constituent the remnant is coupled with, is called "antecedent" in the program code - for the sake of conciseness. However, "antecedent" is not an adequate term for what we want to describe, so it will not be used in this text.

The next phase consists of the evaluation of the candidates. Per interpretation, the pattern of constraint violations is determined by the program and added to the interpretation. When all the violation information has been collected, optimization begins. For each evaluation method, the optimal interpretation is printed on the screen. An example of output was already given in section 5.1.1. The program code of the simulation can be found in appendix C.

6 Testing the computer simulation

To test the computer model, a suitable test set had to be found. But first of all, a complete constraint hierarchy had to be established.

6.1 Constraint hierarchy

In section 2.2, a partial constraint hierarchy for gapping was constructed, based on informal acceptability judgements. Construction of a full constraint hierarchy is not possible with this method; to determine relative positions of constraints in the hierarchy, contrasting pairs of violations must be found. It was simply not possible to construct those contrasting pairs for every pair of constraints. For this experiment, a constraint hierarchy was used that corresponds with the partial hierarchy constructed in section 2.2. This partial hierarchy is repeated below:

OvPar > FSP > Simplex > MinDis > SubPred
FSP > ThemFit > Stay > Same
ThemFit > MinDis
MinDis > FeatPar

The hierarchy that will be used in the experiment is the following:

OvPar > FSP > ThemFit > Stay > Same > Simplex > MinDis > SubPred > FeatPar

To arrive at a complete hierarchy from this partial hierarchy, some blanks had to be filled in. However, for testing the hypotheses, this is irrelevant. Majority cumulativity is completely hierarchy independent: It will give the same predictions no matter what the hierarchy. For global restricted cumulativity, hierarchy is also unimportant: A number of constraints, somewhere in the hierarchy, add up to overrule constraints high in the hierarchy. For local restricted cumulativity, there was enough information in the partial hierarchy:

ThemFit > Stay > Same

The only way of violating Stay or Same is by forcing this violation with ThemFit. Stay and Same are only violated if a violation of ThemFit can be avoided by doing so. Below an example in which Stay is violated because of ThemFit:

26)	Cola	drinkt	Grace	en	wijn	Karen.
	Coke	*drinks*	*Grace*	*and*	*wine*	*Karen.*

ThemFit, Stay and Same are the only relevant constraints for Local Restricted Cumulativity and their relative ordering has already been established.

6.2 Test set

To evaluate the computer model, its performance had to be tested on a written language corpus. The Eindhoven corpus (Uit den Boogaart, 1975) was scanned for candidates[2]. This annotated corpus contains 750.000 words. Not all of the 52

[2] With kind help of Martijn Hennink, who scanned the corpus for suitable candidates.

sentences that were initially selected were suitable to be used in this study. A number of the sentences did not qualify as possibly gapped sentences. The sentences that did qualify had to obey some criteria to make sure that they were suitable for both the simulation and test subjects:

1. The sentence must contain exactly two conjuncts;
2. The conjuncts need to be linked by a conjunction element (for example, "en" (*and*), "of" (*or*) or "maar" (*but*);
3. The sentence must only contain one possible gapping construction;
4. Regarding limitations of the simulation parsing system: The (possible) gapping construction must not be embedded and the (possible) gapping construction cannot contain "er" or "daar";
5. The sentence should be understandable by subjects without context.

The sentences that "..must not be embedded" in 4 refers to are sentences such as the following:

"Dat	Grace	Will	sloeg	en	Jack	Karen, beschouwde	Stan	als
That	*Grace*	*Will*	*hit*	*and*	*Jack*	*Karen, considered*	*Stan*	*as*

normaal"
normal.

An example of a construction containing "er" (which could also replaced with "daar") - mentioned under 4 - is the following sentence:

"Hij heeft een huis gekocht en er veel tijd aan besteed."
He bought a house and spent a lot of time on it.

It could certainly be interesting to integrate "er" (and "daar") in a computer simulation. Parsing a (possibly) gapped construction containing "er" and "daar" is quite a challenge. As the focus in this study is on resolving ambiguity rather than on building a parser, this is left for further research.

Eliminating sentences that did not reach these demands, eight sentences turned out to be suitable. From the sentences with more than two conjuncts, two were selected and adapted slightly, so that the number of conjuncts was brought down to two. This brought the total number of corpus sentences to ten. An overview of corpus test sentences is given in appendix A. Sentences 8 and 10 originally consisted of more than two conjuncts. The part between slashes was left out in using the sentences as input for the computer simulation.

6.3 Simulation output

The ten sentences from the Eindhoven corpus were used as input for the computer simulation. The full output of this is given in appendix D.

The program first gives all the possible interpretation options and their constraint violations. Not all of these options will be regarded as different interpretations by humans. The simulation simply uses all the constituents as elements that the remnants can be coupled with. In many of these reconstructions OvPar - which is a high ranked constraint - is violated,.

It turned out that, for all these sentences, the computer simulation produced the same output predictions for all the evaluation methods (strict domination, majority cumulativity, local restricted cumulativity and global restricted cumulativity).

6.3.1 Constraint violations of optimal candidates

It turned out that nine out of ten optimal interpretations violated no constraints. Sentence 8 had an optimal interpretation with a violation of FeatPar. Sentence 8 is repeated below as example 27:

27)	Groep 1	trok	de arm	na vijftien minuten	uit de testkamer,
	Group 1	*pulled*	*the arm*	*after fifteen minutes*	*from the test room*

en	groep 4	na een uur.
and	*group 4*	*after an hour.*

Two possible interpretations of the second conjunct of this sentence are:

27a)	...en	groep 1	trok	groep 4	na een uur	uit de testkamer.
	...and	*group 1*	*pulled*	*group 4*	*after an hour*	*from the test room.*
27b)	...en	groep 4	trok	de arm	na een uur	uit de testkamer.
	...and	*group 4*	*pulled*	*the arm*	*after an hour*	*from the test room.*

The computer simulation chose interpretation 27a. In determining the constraint hierarchy, MinDis was placed higher than FeatPar, based on an example where MinDis competed with FeatPar. Because of this fact, the interpretation in 27b was not chosen as optimal: it violates MinDis. Intuitively, for this sentence, 27b seems to the best interpretation. The experimental data will have to decide whether this presumption is right.

6.3.2 Parsing the sentences

For now, the sentences presented to the simulation as input are manually parsed (with the help of Klein and Van den Toorn, 1991). This parsing process was not as straightforward as it might seem at first sight. Consider the following corpus sentence:

28)	Wij	hebben	nu	twee keer	goed	gespeeld,	maar ook
	We	*have*	*now*	*twice*	*well*	*played*	*but also*

twee keer	verloren.
twice	*lost.*

Two contrasting parts of the sentence are "goed gespeeld" (*well played*) and "lost" (*verloren*). When this sentence is parsed manually, this is quite simple to see. In the computer simulation, a reconstruction of a second conjunct was made by replacing the constituents that the remnants were coupled with by the remnants themselves. If the sentence was parsed automatically, the reconstruction might also be:

28a)	....maar	wij	hebben	ook	twee keer	goed	verloren.
	....but	*we*	*have*	*also*	*twice*	*well*	*lost.*

There seems to be a difference between elements that a verb does subcategorize for and verbs that the verb does not subcategorize for. In section 8.1, this observation will be discussed further.

6.3.3 ADVERBIAL PHRASES

The main focus of the study is on verbal arguments. In the corpus sentences, many of the sentences also contained (several) adverbial phrases. These had to be dealt with as well. Temporal adverbial phrases were treated as a separate constituent type. This meant that coupling a constituent with "adverbial phrase" as a possible role leads to an OvPar violation. Consider this example:

29)	Vrijdag	eet	Grace	in het restaurant	en	Will	maandag.
	Friday	*eats*	*Grace*	*in the restaurant*	*and*	*Will*	*Monday.*

Coupling "maandag" (*Monday*) with "in het restaurant" (*in the restaurant*) leads to an OvPar violation. The computer system predicts this, because the constituent "vrijdag" (*Friday*) is a temporal adverbial phrase, "in het restaurant" (*in the restaurant*) an adverbial phrase and for the lexical entry "maandag" (*Monday*), "temporal adverbial phrase" was entered as a role option and "adverbial phrase" was not.

FeatPar can be applied to (temporal) adverbial phrases as well as to verb arguments. An example is the following sentence, sentence 2 from the corpus set:

30)	Voor de Tarzan-liefhebbers	is	Johnnie	vrijdagavond
	For Tarzan-lovers	*is*	*Johnnie*	*Friday night*
	op de televisie	te zien	in het NOS-programma Premiere	en
	on television	*to see*	*in the NOS-show Premiere*	*and*
	woensdag 1 juli	in de televisiehoofdfilm " Tarzan en de Amazones".		
	Wednesday July 1st	*in the television film "Tarzan and the Amazones".*		

The two (non-temporal) adverbial phrases in the first conjunct could both be coupled with the pp "in de televisiehoofdfilm Tarzan en de Amazones" (*in the main television film "Tarzan and the Amazones"*). However, in features this constituents was similar to "in het NOS-programma Premiere" (*in the NOS show Premiere*) because of its preposition and not to "op de televisie" (*on television*).

Adverbial phrases are included in this study; the subtype "temporal adverbial phrase" is distinguished. If a nontemporal adverbial phrase is coupled with a temporal adverbial phrase, Overt Syntactic Parallelism is violated. Featural Parallelism applies to adverbial phrases as well as to verb arguments.

7 Experiment

In chapter 4, hypotheses about cumulativity were formulated and in chapters 5 and 6, a computer simulation of the interpretation of possibly gapped sentences and the output of this simulation were discussed. The hypotheses about cumulativity and the accuracy of the computer model are evaluated by a small experiment, for which ten subjects were presented with a written questionnaire.

7.1 Stimuli

The purpose of the experiment is clearly twofold, therefore the test corpus consists of two parts: Sentences from a natural language corpus to test whether the computer model is able to yield an output if it is presented with a naturalistic input, and artificially constructed sentences to test the proposed hypotheses about cumulativity.

7.1.1 SENTENCES FROM A NEWSPAPER CORPUS

Ten sentences from the Eindhoven corpus were used as input for the computer simulation. The results that the computer simulation produced, are described in section 6.3 and given in appendix D. In this experiment, it was tested whether the output of the computer simulation matched the responses of the subjects.

7.1.2 ARTIFICIALLY CONSTRUCTED SENTENCES

The second part of the test set consists of sentences that were artificially constructed. For the global restricted cumulativity and the majority cumulativity hypotheses, two types of test sentences were included, involving different combinations of violations of constraints. For global restricted cumulativity, two versions were included: A version with an interpretation where a candidate violation pattern SubPred+MinDis+Simplex competed with an interpretation with only an FSP violation (SubPred+MinDis+Simplex versus FSP) and a version with violation pattern SubPred+MinDis+Simplex competed with an interpretation with only an OvPar violation (SubPred+MinDis+Simplex versus OvPar). For majority cumulativity there were also two versions: A condition where two candidates with only soft constraint violations competed and a condition where a candidate with soft constraint violations competed with a candidate with fewer constraint violations. However, these violations were violations of hard constraints. In chapter 1, Keller's research (2001) on gradient grammatical acceptability and the soft/hard distinction of constraints was discussed. Keller concluded from his research that "the degree of unacceptability increases with the number of violations". He also made a sharp distinction between soft constraints, that cause "mild unacceptability" and hard constraints that cause "strong unacceptability". For the hierarchy used in this study, the soft-hard distinction, based on Keller's criteria, is the following:

OvPar	hard
FSP	hard
ThemFit	hard
Stay	hard
Same	hard
Simplex	hard

MinDis	soft
SubPred	soft
FeatPar	soft

In the experiment, a condition with OvPar versus MinDis+FeatPar (hard vs. soft) and a condition with MinDis versus SubPred+FeatPar (soft vs. soft) was included. This way, it could be tested whether the soft-hard distinction made a difference for the Majority Cumulativity hypotheses.

Apart from the sentences to test the cumulativity hypotheses, translated test sentences from Carlson (2001) were included. Her research was mentioned earlier in section 2.1.5. She conducted two experiments, of which the first one is very similar to the experiment in this study. Carlson's experiment was done in English. It can be interesting to see the results of the Dutch version of her experiment. Among the sentences Carlson used, were these types:

Bake A (object must be inanimate)
Alice bakes cakes for tourists and Caroline for her family.
Take A (object might be either animate or inanimate)
Josh visited the office during the vacation and Sarah during the week.
Introduce (object must be animate)
Dan amazed the judges with his talent and James with his musicality.

In Carlson's experiment, the Bake A sentences yielded the highest number of gapping responses, followed by the Take A and Introduce sentences. The "Bake A", "Take A" and "Introduce" terminology has been borrowed from her research. Below an overview of the types of sentences tested:

Local restricted cumulativity

Will plukt een roos en een tulp Grace.
Will picks a rose and a tulip Grace.

	ThemFit	Stay	Same
een tulp → Will, Grace → een roos	*		
een tulp → een roos, Grace→Will		*	*

Global restricted cumulativity

A

Will vraagt hem om Grace te negeren en Jack om Karen te ontlopen.
Will asks him to ignore Grace and Jack to avoid Karen.

	FSP	Simplex	MinDis	SubPred
Jack → Will		*	*	*
Jack → hem	*			

B

Will	verzoekt	vandaag	om	Grace	te negeren	en	Jack
Will	*requests*	*today*	*to*	*Grace*	*ignore*	*and*	*Jack*

om Karen te ontlopen.
to Karen avoid.

	OvPar	Simplex	MinDis	SubPred	FeatPar
Jack → Will		*	*	*	*
Jack → vandaag	*				

Majority cumulativity

A

Jack wacht met Grace op de ober en Will met de butler.
Jack waits with Grace for the waiter and Will with the butler.

	OvPar	MinDis	FeatPar
Will → Jack, de butler → Grace		*	*
Will → Grace, de butler → de ober	*		

B

De koning belooft het volk te blijven en de prins te vertrekken.
The king promises the people to stay and the prince to leave.

	MinDis	SubPred	FeatPar
de prins → de koning	*		
de prins → het volk		*	*

Carlson's test sentences

Bake A

Grace bakt cakes voor toeristen en Karen voor haar familie.
Grace bakes cakes for tourists and Karen for her family.

	ThemFit	MinDis	FeatPar
Karen → cakes	*		*
Karen → Grace		*	

Take A - verb subcategorizes for both animate and inanimate arguments.

Jack bezocht het kantoor in de vakantie en Karen in het weekend.
Jack visited the office during the vacation and Karen during the weekend.

	MinDis	FeatPar
Karen → het kantoor		*
Karen → Jack	*	

Introduce - verb subcategorizes for only animate arguments.

Stan verbaasde de jury met zijn talent en Jack met zijn muzikaliteit.
Stan surprised the jury with his talent and Jack with his musicality.

	MinDis	FeatPar
Jack → de jury		*
Jack → Stan	*	

Carlson's Take A and Introduce conditions yield the same OT tableau, given the constraints that were assumed. In her experimental research, the Take A condition received a much higher gapping response than the Introduce condition. Carlson's explanation for this was the higher plausibility for the gapping interpretation in the Take A condition in comparison to the Introduce condition. Of each condition, three different sentences were included in the experiment, except for Carlson's Take A and Introduce condition, of which two instances per condition were included.

7.1.3 FILLERS

Eight filler sentences were included. Two types were included:

Rosario speelde backgammon met Karen en Grace ook.
Rosario played backgammon with Karen and Grace too.

Stan eet koekjes en drinkt bier.
Stan eats cookies and drinks beer.

These sentences do contain coordination constructions, but they are not possible gapping sentences. The total number of sentences in the experiment was 40: All of the experimental sentences can be found in appendix A (corpus sentences) and B (constructed sentences).

7.2 Subjects

Ten subjects (aged 21-60) were presented with the questionnaire. They all had Dutch as their native language. Their level of (completed) education ranged from highschool to university.

7.3 Procedure

The experiment consisted of a written questionnaire of 40 sentences in random order. For each sentence, two interpretation options were given, (a) and (b). If the perceived interpretation was not among them, the subject could write his or her own interpretation under (c). After choosing an interpretation, the subject was asked to judge how difficult it was to interpret the sentence. The subject could give ratings between 1 (very easy) and 5 (very difficult). Furthermore, for each sentence, the subjects had to indicate whether they found the sentence ambiguous. The instruction that was given to the subjects (in written form) before participating in the experiment can be found in appendix E.

7.4 Results

An overview of the test results will be given. Firstly, the results of the Eindhoven corpus sentences will be given. After that, the responses to the Dutch versions of the Carlson sentences will be compared to the results presented in Carlson (2001). Then the results of the different test conditions for the hypotheses will be discussed. Finally, some overall figures will be given about the experiment.

7.4.1 RESULTS EINDHOVEN CORPUS

Table 9 gives an overview of the responses to the ten sentences from the Eindhoven corpus. As is shown in the table, in most cases, the subjects agreed with the interpretation the computer model attributed to the sentences. All the results that were deviant compared to the output of the computer model came from one test sentence. This sentence and possible reasons for this outcome will be discussed in section 8.1. The subjects did not have many problems interpreting the sentences and they were generally not seen as ambiguous.

# subjects = 10	% responses corresponding with computer model response	Average difficulty rating (1-5)	Average rating factor "ambiguous" (0 = not ambiguous, 1 = ambiguous)
Corpus sentences	91 (28.46)	1.87 (0.45)	0.11 (0.12)

Table 9: Results of the corpus sentences. Standard deviation between brackets.

The three sentences that - apart from the optimal interpretation - had other interpretations that did not violate Overt Syntactic Parallelism, were considered ambiguous more often (in 20% of the cases) than the other seven sentences that did not (in 7% of the cases these sentences were considered ambiguous).

7.4.2 RESULTS TRANSLATED CARLSON SENTENCES

The results of the translated sentences from Carlson's experiment, in table 10, are given as the number of gapping responses.

# subjects = 10	Number of different sentences included	% gapping responses	Average difficulty rating (1-5)	Average rating factor "ambiguous" (0 = not ambiguous, 1 = ambiguous)
Bake A	3	83 (36)	1.80 (0.69)	0.47 (0.32)
Take A	2	80 (42,16)	2.10 (0.84)	0.75 (0.42)
Introduce	2	35 (33.75)	2.30 (0.79)	0.65 (0.47)

Table 10: Results from the translated Carlson sentences. Standard deviation between brackets.

The Bake A condition yielded the highest number of gapping responses, with 83%. Second came the Take A condition and the Introduce condition yielded the lowest number of gapping responses. A difference between the Take A and Introduce conditions, was that all of the subjects were consistent in responding to the instances of the Take A condition, while only 50% of the subjects was consistent in the Introduce condition. Considering the high overall consistency, this last result is notable. 80% of the subjects that responded inconsistently to the Introduce condition showed the same response pattern. The two Introduce sentences were these:

31) Stan verbaasde de jury met zijn talent en Jack met zijn muzikaliteit.
Stan surprised the jury with his talent and Jack with his musicality.

32) Will complimenteerde de gastheren met de decoratie en Grace met het eten.
Will complimented the hosts on the decoration and Grace on the food.

Of the subjects with an inconsistent response pattern to these sentences, 80% rated sentence 31 as a gapping sentence (in which Jack surprised the jury with his musicality), and 32 as a non-gapping sentence (in which Will complimented Grace on the food).

In table 11, Carlson's results for these conditions in English are given. The design of the experiment in this study is very similar to the design of Carlson's experiment; the results can easily be compared.

# subjects = 68	% gapping responses	Average difficulty rating (1-5)
Carlson Bake A	81 (13)	2.43 (0.71)
Carlson Take A	40 (20)	3.07 (0.62)
Carlson Introduce	21 (18)	2.45 (0.28)

Table 11: Carlson's results for the English test sentences. Standard deviation between brackets.

In both experiments, the Bake A condition yields the highest number of gapping responses, followed by the Take A and Introduce conditions. A striking difference the two experiments is the number of gapping responses to the Take A sentences: In Carlson's experiment, there was a 40% gapping response, against 80% in this study. Generally, Carlson's subjects rated the sentences more difficult than the subjects in this study, the difference being the highest in the Take A condition. This condition was considered the most difficult by the subjects in Carlson's experiment, while in this study, the Introduce condition seemed to be the most difficult.

Of course, in Carlson's study, the number of subjects was far higher than in this study. Furthermore, in this study, only a limited number of different sentences per condition was included. More extensive experimental work would have to confirm the results from this study.

7.4.3 RESULTS CUMULATIVITY HYPOTHESES

In table 12, an overview of the results for the conditions local restricted cumulativity (LocalRes), global restricted cumulativity (GlobalRes) A and B and majority cumulativity A and B (Majority) are given. The A and B conditions were varieties of a condition, involving different constraints. The percentages given are the percentages of responses that fit the hypothesis in question. So, a percentage of 43% for LocalRes means that that percentage of the responses for the LocalRes condition supported that hypotheses.

# subjects = 10	Constraint violations involved	Number of different sentences included	% responses corresponding with hypotheses	Average difficulty rating (1-5)	Average rating factor "ambiguous" (0 = not ambiguous, 1 = ambiguous)
LocalRes	Stay + Same vs. Thematic Fit	3	43 (41.72)	2.57 (1.07)	0.43 (0.50)
GlobalRes A	Simplex + Mindis + SubPred vs. FSP	3	80 (32.20)	2.53 (0.57)	0.50 (0.39)
GlobalRes B	Simplex + Mindis + SubPred vs. OvPar	3	63 (45.68)	2.60 (0.44)	0.53 (0.42)
Majority A	MinDis + FeatPar vs. OvPar	3	0 (0)	2.60 (0.60)	0.40 (0.26)
Majority B	SubPred + FeatPar vs. MinDis	3	3 (10.54)	2.37 (0.37)	0.40 (0.31)

Table 12: Results of the hypotheses test sentences. Standard deviation between brackets.

GlobalRes A has the highest score for cumulativity with 80%. The other global restricted cumulativity condition, GlobalRes B, has a high score as well: 63%.

The subjects whose responses corresponded (partly) with the LocalRes hypotheses, had a lower average gapping response to the Carlson sentences (39%) than subjects whose responses did not correspond with the LocalRes hypotheses (72%). There were great individual differences between the subjects.

7.4.4 RESULTS OF EVALUATION METHODS OVER ALL TEST CONDITIONS

Table 13 on the next page is an overview of the types of test sentences and which percentage of the responses corresponded with strict domination, unrestricted cumulativity, majority cumulativity, global restricted cumulativity and local restricted cumulativity. The rows represent the different test conditions. The conditions were designed to test specific hypotheses (placed in the third column). In this table, percentages are given of how the different hypotheses perform on ALL of the test sentences (measured by responses of test subjects).

# subjects = 10 Constraints involved in competing candidates	Number of items	Hypothesis that condition was designed for	Strict domination[1]	Unrestricted cumulativity[2]	Majority cumulativity[3]	Local restricted cumulativity	Global restricted cumulativity
(1)Same+Stay vs. (2)ThemFit	3	Local restricted	(1) 50% [4]	(2) 43%	(2) 43%	(1) 43%	(1) 50%
(1)SubPred+ Simplex+ MinDis vs. (2)FSP	3	Global restricted	(1) 20%	(2) 80%	(2) 80%	(1) 20%	(2) 80%
(1)SubPred+ Simplex+ MinDis vs. (2)ovPar	3	Global restricted	(1) 37%	(2) 63%	(2) 63%	(1) 37%	(2) 63%
(1)MinDis+ FeatPar vs. (2)ovPar	3	Majority	(1) 93%	(1) 93%	(2) 0%	(1) 93%	(1) 93%
(1)SubPred+ FeatPar vs. (2)minDis	3	Majority	(1) 97 %	(2) 3%	(2) 3%	(1) 97 %	(1) 97 %
(1)ThemFit+ FeatPar vs. (2)minDis	3	Carlson Bake A	(2) 83%	(2) 83%	(2) 83%	(2) 83%	(2) 83%
(1)MinDis vs. (2)FeatPar	2	Carlson Take A	(2) 20 %	(2) 20 %	(2) 20 %	(2) 20 %	(2) 20 %
(1)MinDis vs. (2)FeatPar	2	Carlson Introduce	(2) 65 %	(2) 65 %	(2) 65 %	(2) 65 %	(2) 65 %
		Average:	60%	58%	45%	59%	71%

Table 13: Results of evaluation methods over all test conditions.

1) The hierarchy used in this study was the following:
OvPar > FSP > ThemFit > Stay > Same > Simplex > MinDis > SubPred > FeatPar
2) The variety used here had the following weight distribution:
OvPar=1, FSP=0.9, ThemFit=0.8, Stay=0.7, Same=0.6, Simplex=0.5, MinDis=0.4, SubPred=0.3, FeatPar=0.2
3) This is the form of unrestricted cumulativity where the hierarchy only plays a role when the best candidates have the same amount of constraint violations. The candidate with the fewest constraint violations is optimal according to the majority cumulativity evaluation method.
4) In brackets is the number of the candidate that is optimal according to the hypothesis; the percentage is the percentage of responses that corresponded with that candidate. In a few cases the percentage corresponding with (1) and (2) do not add up to 100%: This is because subjects sometimes chose a deviant interpretation. These were left out.

The sentences were designed to test cumulativity hypotheses, so they are not representative for "normal" potential gapping sentences as they might be encountered in real life. Strict domination and unrestricted cumulativity (with a weight distribution as given under [2]) yield about the same percentage of responses corresponding to the hypothesis in question. In the global restricted conditions, the responses match the unrestricted cumulativity hypothesis better than strict domination. For the majority cumulativity conditions, where the predictions differ, strict domination seems to do better. Unrestricted cumulativity and strict domination predict different interpretations for the local restricted cumulativity conditions. The responses of the test subjects did not point towards either of the hypotheses.

Majority cumulativity does worse than strict domination and unrestricted cumulativity. Its predictions are more extreme (i.e. different from strict domination) than the predictions of unrestricted cumulativity, because the hierarchical relations are completely irrelevant for majority cumulativity. From the test sentences in the table, this difference is reflected in the "MinDis+FeatPar vs. OvPar" condition. For the majority cumulativity conditions, almost all of the responses of the test subjects do not match with the majority cumulativity condition.

Local restricted cumulativity only predicts a different outcome for the condition designed to test it. As was mentioned earlier, the responses of the subjects do not point towards either strict domination or a cumulativity hypothesis.

Of the evaluation methods presented here, global restricted cumulativity has the highest overall "success percentage". The global restricted conditions, in which the predictions differ from strict domination, a majority of the test subjects chose the response that fitted the global restricted cumulativity hypothesis. Therefore, the overall score of this evaluation method is higher for this set of test sentences.

7.4.5 Difficulty and ambiguity

As shown in table 9, perhaps not surprisingly, in only 11% of the cases the Corpus sentences were being considered ambiguous, against an average ambiguity rate of 52% for the constructed sentence types. The mean difficulty rating was higher for the constructed sentences (2,36) than for the corpus sentences (1,87). Because of the small number of subjects and non-normal distribution, statistics cannot be given. Looking at all of the sentences, regardless of their type, the sentences that were judged ambiguous received a higher average difficulty rating than the sentences that were not considered ambiguous: $t(40)=-9.12$, $p<.01$.

7.4.6 Consistency

For each condition of the hypotheses testing sentences, a number of different sentences was included in the experiment. There were eight conditions. In 80 % of the cases, the subjects were constistent in their responses for a certain condition.

7.4.7 Deviant responses

Apart from two interpretation options, subjects were also allowed to give a different interpretation, if the interpretation they perceived was not among the pre-constructed answers. In most of the cases, 99%, subjects were able to choose from the given options. In 1% of the cases (four answers, given by four different subjects), the answer was different from the two given options. This concerned two test sentences:

33)	Will plukt een roos en een tulp Grace.				
	Will picks a rose and a tulip Grace.				
34)	Karen lachte	met de tuinman	om Will	en	
	Karen laughed	*with the gardener*	*at Will*	*and*	
	de glazenwasser	met Ellen.			
	the window cleaner	*with Ellen.*			

In the case of sentence 33, two subjects thought the tulip was called Grace. This is not even such a strange thought, as tulip (varieties) are named after people sometimes. In sentence 34, it was perceived by two subjects that Karen was laughing at the window cleaner, who was with Ellen. "De glazenwasser met Ellen" was seen as a constituent by them.

8 Discussion

This study had two main purposes: Designing and testing a computer simulation of the phenomenon of gapping and testing a number of hypotheses on cumulativity in gapping. For the first goal, a computer simulation was built in Prolog and tested with sentences from the Eindhoven corpus. These sentences were presented to both the computer model and test subjects. For the second goal, sentences were constructed to test the different hypotheses suggested in section 4.3. In this section the implications of the results will be discussed.

8.1 Computer simulation and "real life" input

In most cases, the computer model was accurate in predicting the interpretation subjects gave to the corpus sentences. Now, the "real life" sentences from the Eindhoven corpus turned out to be a lot less difficult and ambiguous than the constructed sentences, which is pretty obvious. The optimal interpretation was either without violations, or violated only Featural Parallelism. There was a very high level of agreement among subjects about the optimal interpretations of the sentences.

For one sentence, the computer simulation gave a different prediction than the subjects. This was for the following sentence:

35)	Groep 1		trok	de arm		na vijftien minuten	uit de testkamer,
	Group 1		*pulled*	*the arm*		*after fifteen minutes*	*from the test room*
	en	groep 4		na een uur.			
	and	*group 4*		*after an hour.*			
35a)	...en	groep 4		trok	de arm	na een uur	uit de testkamer.
	...and	*group 4*		*pulled*	*the arm*	*after an hour*	*from the test room.*
35b)	...en	groep 1		trok	groep 4	na een uur	uit de testkamer.
	...and	*group 1*		*pulled*	*group 4*	*after an hour*	*from the test room.*

This was also the only sentence that had no interpretation option without constraint violations. The presumption that was expressed in section 6.3.1 - that human subjects would interpret this sentence in a different way than the computer model - turns out to be right: Human subjects interpret this sentence in such a way that group 4 pulled the arm from the test room after an hour, the gapping interpretation (as in 35a), while the computer simulation interpreted the sentence such that group 1 pulled group 4 from the test room (as in 35b).

There are several ways to explain the discrepancy. A possibility is that the constraint ranking is not accurate. Perhaps FearPar should be above MinDis in the constraint hierarchy; then the output of the computer simulation would be different for sentence 35. It would then pick 35a as optimal interpretation, because the featural similarity between "groep 1" (*group 1*) and "groep 4" (*group 4*) would be more important than the fact that coupling "groep 4" and "de arm" (*the arm*) prevents a violation of MinDis. However, the relationship between MinDis and FeatPar seems to be a bit more complicated than that FeatPar simply ought to be ranked higher than MinDis: In some cases MinDis seems to be able to overrule FeatPar. Consider a Carlson Introduce-type sentence:

36) Stan verbaasde de jury met zijn talent en Jack met zijn muzikaliteit.
Stan surprised the jury with his talent and Jack with his musicality.

In this type of sentences, MinDis seemed to be able to overrule FeatPar. Only 35% of the subjects picked the gapping interpretation (where MinDis was violated) here. Both "groep 1" (*group* 1) and "groep 4" (*group 2*) in sentence 35 and "Stan" and "Jack" in sentence 36 were labelled featurally parallel in the lexicon. Comparing sentences 35 and 36 raises the question whether FeatPar can be violated in different degrees. This will be discussed further in section 8.3.

For the corpus sentences, the cumulativity hypotheses did not yield different predictions than the strict domination model. It seems that in "real life" if potentially gapped sentences are produced, they are quite easy to interpret.

The way the computer simulation reconstructs sentences is by simply replacing the constituents the remnants are coupled with by the remnants.

37)	Wij	hebben	nu	twee keer	goed	gespeeld,	maar ook
	We	*have*	*now*	*twice*	*well*	*played*	*but also*
	twee keer	verloren.					
	twice	*lost.*					

In this sentence, "goed gespeeld" (*well played*) contrasted with "verloren" (*lost*). In a reconstruction, we do not want "goed" (*well*) to be repeated in the second conjunct. In sentences with more than one verb, when there is a verb in the second conjunct that contrasts with the main verb in the first conjunct, the "obligatory" constituents are borrowed from the main verb in the first conjunct.

The question is whether the information from the first conjunct that the main verb does not subcategorize for, is transmitted to the meaning of the incomplete coordinate structure. Three sentences that are very similar:

38)	Grace	heeft	Will	in de bus	gezien	en	Karen	Stan	gesproken.
	Grace	*has*	*Will*	*on the bus*	*seen*	*and*	*Karen*	*Stan*	*spoken to.*
39)	In de bus		heeft	Grace Will	gezien	en	Karen	Stan	gesproken.
	On the bus		*has*	*Grace Will*	*seen*	*and*	*Karen*	*Stan*	*spoken to.*
40)	Grace	heeft	Will	gisteren	gezien	en	Karen	Stan	gesproken.
	Grace	*has*	*Will*	*yesterday*	*seen*	*and*	*Karen*	*Stan*	*spoken to.*

In sentence 38, Karen did not necessarily speak to Stan on the bus. Sentences 39 and 40 are most likely to be interpreted such that Karen spoke to Stan on the bus/yesterday. In 39 "in de bus" (*on the bus*) is topicalized, which has an effect on the interpretation of the second conjunct. In the case of 40, it could be the temporal aspect that makes a difference in the interpretation of the second conjunct, in comparison to 39. Concluding, some information from the first conjunct is transmitted to the meaning of the second conjunct, but this does not automatically happen to all information in the first conjunct. It is not quite clear what rules this phenomenon is subject to. This could be an interesting issue for further research, but for now, it is of minor relevance.

The computer model could be improved by coupling it with an automatic grammatical parser and a lexicon (see also section 8.3 on gradience). Then all the information needed from the first conjunct to evaluate the constraints could be retrieved automatically. The computer simulation is as yet not capable of processing sentences with more than two conjuncts or sentences that lack a conjunction (words like "en" (*and*), of (*or*) or maar (*but*)). Some sentences that were initially extracted from the Eindhoven corpus contained either more than two conjuncts or lacked a conjunction. These sentences cannot be parsed by the simulation as it is, but could be parsed if the simulation was improved.

8.2 Context and prosody

A topic that cannot be overlooked is the role of context and prosody in interpreting incomplete coordination sentences. In the computer simulation and in the experiment, this factor was not included. But, especially in the cases that were considered highly ambiguous, context and prosody could make a great difference in deciding on an interpretation.

In the local restricted cumulativity condition (Will plukt een roos en een tulp Grace, *Will picks a rose and a tulip Grace*), in about half of the responses the subjects let Grace pick a tulip and in the other half of the cases, subjects let a tulip pick Grace. The group of subjects that did not (consistently) let Grace pick the tulip, also made other choices in the Carlson sentences. In the "tulip-picking" sentences, they chose the syntactically easier interpretation, in which the Stay and Same constraints were not violated, at the expense of the logical meaning of the sentence. In other words, they chose syntactic ease over plausibility. In the Carlson sentences, this group opted for the gapping interpretation much less often than the rest of the subjects. Again, they went for syntactic ease (not violating the Minimal Distance Principle) at the cost of plausibility (in this case Featural Parallelism). A guiding context might change the fifty-fifty division of interpretation choices. Consider the following little story:

"There is a world where the flowers walk around and pick people. Will picks a rose and a tulip Grace."

Admittedly, it is a strange story. But, given this context, subjects might tend to choose the (otherwise) semantically implausible interpretation (where the tulip actually picks Grace) more often.

The global restricted cumulativity A condition (SubPred+MinDis+Simplex vs. FSP) was tested with sentences of the following type:

41) Will vraagt hem om Grace te negeren en Jack om Karen te ontlopen.
Will asks him to ignore Grace and Jack to avoid Karen.
41a) .... en Will vraagt Jack om Karen te ontlopen.
....and Will asks Jack to avoid Karen.
41b) ....en Jack vraagt hem om Karen te ontlopen.
....and Jack asks him to avoid Karen.

In the interpretation of the second conjunct as shown in 41b, FSP is obeyed. FSP was violated in the interpretation in 41a: "Jack" is coupled to "hem" (*him*). "Hem", being a pronoun, is considered old information that should not be coupled with remnants if we want to obey FSP. There was a strong tendency towards the interpretation where FSP was violated (as in 41a). As Kuno (1976) already noted in describing FSP, the use of a pronoun indicates that the information is not new and should not be coupled with remnants. Looking at the example in 41 with this in mind, 41a does violate FSP. However, there are other factors that can indicate newness, perhaps even for pronouns. Contextual information might be able to stress the newness of even a pronoun. Consider the following little story:

"Jack is talking to a man who is in love with Karen. Now, Will asks **him** to ignore Grace and Jack to avoid Karen!"

Especially when, apart from guiding contextual information, guiding prosodic information is added, the fact that a pronoun is used does not seem to be an obstacle to couple it with a remnant anymore. In the experiment, there was no guiding contextual and prosodic information. However, the choice was between an interpretation in which only FSP was violated, or an interpretation in which SubPred, MinDis and Simplex were violated. Even though SubPred, MinDis and Simplex are low in the constraint hierarchy, three is still a high number of violations. Now, the only thing a subject has to do to avoid such an interpretation, is to imagine a context and/or prosody where "him" is marked as new information. By doing this, the interpretation that would otherwise violate FSP, no longer violates any constraints.

Adding prosodic and/or contextual information to ambiguous sentences removes a certain freedom that subjects otherwise have to imagine prosody/context as it suits them best (in terms of the number of constraint violations).

8.3 Gradience

An interesting finding, corresponding with experimental evidence from Carlson (2001), was the difference in gapping response between the Take a and Introduce sentences. Below, an example of each is given:

Take a:

42) Jack bezocht het kantoor in de vakantie en Karen in het weekend.
Jack visited the office during the vacation and Karen during the weekend.

Introduce:

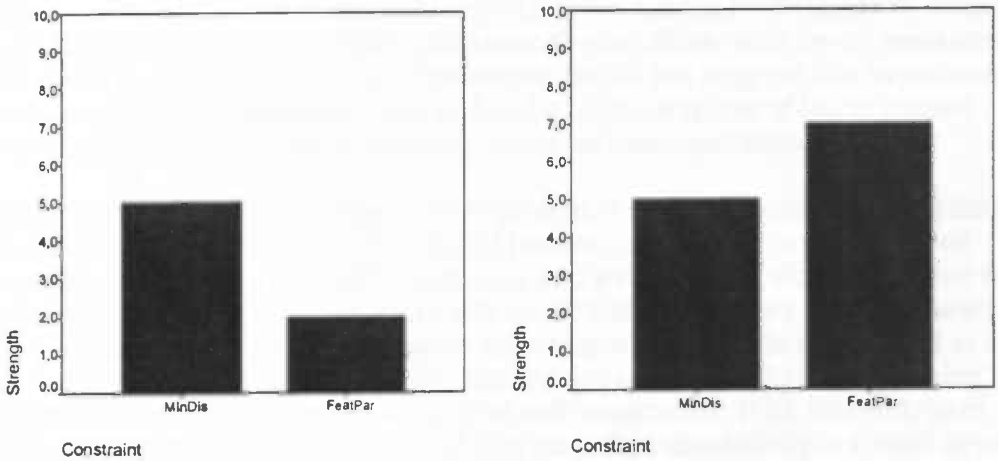
43) Stan verbaasde de jury met zijn talent en Jack met zijn muzikaliteit.
Stan surprised the jury with his talent and Jack with his musicality.

For the computer model, in both sentences the FeatPar constraint is violated in the non-gapping interpretation. The computer model used a binary version of the constraint: Either it was violated, or it was not. However, the interpretation responses of the subjects were very different for the Take a and Introduce sentences. The violation of the FeatPar constraint in the Take a sentences was considered much worse than its violation in the Introduce sentences. The response to the different instances of the Introduce sentences was inconsistent for half of the subjects; 80% of them agreed as to which of the instances was interpreted as a gapping sentence.

These findings suggest that the FeatPar constraint is gradient: It can be violated in a greater or lesser degree. This difference in degree of violation seems to be so strong that it can affect the interpretation of a sentence. In 42, for most people, FeatPar seems to be stronger than MinDis and in 43, for most people, FeatPar is weaker than MinDis (as it was originally scaled in the constraint hierarchy). In the graphs shown below, an graphic example is given of what this could look like. The relative strengths of the relevant constraints MinDis and FeatPar are given for a weak violation of FeatPar and a strong violation of FeatPar. The strength of MinDis remains the same, but the strength of FeatPar varies. In the specific example of sentences 42 and 43, animacy could be a factor that makes a difference in the strength of the violation of

FeatPar. In sentence 42, if "Karen" is coupled with "het kantoor" (*the office*), FeatPar is violated. An feature in which these constituents are different, is animateness. In sentence 43, if "Jack" is coupled with "de jury" (*the jury*), the feature that causes the FeatPar violation is the fact that "Jack" is a proper noun and "de jury" is not. Animateness seems a more important feature than whether a constituent is a proper noun or not.

A gradient constraint poses problems implementing FeatPar in a standard OT model of gapping. When working with weight values, a different weight value might be given to the constraint for different levels of violation. A situation as depicted in the graphs below can be simulated by attributing different weight values in different degrees of violation. But how is the "seriousness" of a FeatPar constraint measured? We know that Featural Parallelism has to do with semantic similarity. In a lexical thesaurus like WordNet, semantic relations between words are stored in a way that corresponds with current theories about psycholinguistic theories. With the help of WordNet, the semantic "distance" of words, i.e. their semantic similarity, can be determined. This could provide a measure for Featural Parallelism. But even if there is such a measure, it is still unknown how exactly the amount of Featural Parallelism relates to other constraints. The implementation of gradient constraints like Featural Parallelism is a challenge for the future.



Weak violation of FeatPar   Strong violation of FeatPar

Relative position of MinDis and FeatPar, left with weak violation of FeatPar and right with strong violation of FeatPar. The y-axis (Strength) values were chosen arbitrarily.

8.4 Cumulativity

An important question this study tried to answer is whether constraint cumulativity can be found in the interpretation of possibly gapped sentences. The different hypotheses that were tested will be discussed here.

8.4.1 LOCAL RESTRICTED CUMULATIVITY

Local restricted cumulativity is the type of cumulativity where two adjacent constraints (Stay and Same in this study) can overrule the constraint just above them in the hierarchy (ThemFit in this study), if they are both violated. The local restricted cumulativity test sentences were considered difficult to interpret. As already described in section 8.2, two groups could be distinguished. One group wanted the sentences to make sense and the other accepted the odd interpretation that nevertheless violated no syntactic constraints. The local restricted cumulativity hypothesis seems to hold for a group of people, but certainly not for everyone - at least not in an experimental setting. An experiment that includes context effects might give interesting results.

8.4.2 MAJORITY CUMULATIVITY

Majority cumulativity (derived from Keller's (2001) research) claims that candidate with fewer constraint violations is always a better candidate than a candidate with more constraint violations, regardless of the hierarchical position of the constraints. For majority cumulativity, no evidence was found in the experiment. Even though in some cases cumulativity effects can arise, the constraints on gapping are clearly hierarchically ordered. Two conditions for the majority cumulativity hypothesis were included; in one condition, a candidate with a soft constraint violation competed a candidate with a higher number of soft constraint violations - these constraints all had a lower rank. In the other condition, a candidate with a number of hard constraint violations competed a candidate with a higher number of soft constraint violations (of which the constraints were all lower ranked). The majority cumulativity hypothesis predicts that in both conditions, the candidate with fewer constraint violations is considered optimal. The responses did not support this hypothesis: In both conditions the candidate that violated a higher number of lower ranked constraints was considered optimal by the experiment participants.

Two conditions of this hypothesis were presented to the subjects. This was done to test whether a difference could be found between a situation in which a hard constraints competes with soft constraints and a situation in which a (higher ranked) soft constraint competes with other soft constraints. In Keller's (2001) research on the production side of gapping constructions, hard constraints are defined as causing "strong unacceptability" and soft constraints as causing "mild unacceptability"; hard constraints receive a higher ranking than soft constraints. With this soft/hard distinction in mind, it could be stated that majority cumulativity is a valid hypothesis, but only for a competition between only hard constraint violations or only soft constraint violations. The results of the experiment in this study do not support majority cumulativity at all; not when a candidate with hard constraint violations competes with a candidate with soft constraint violations and not when a candidate with soft constraint violations compete with a candidate with soft constraint violations. However, it has to be kept in mind that the experiment was small and should be confirmed by larger studies.

8.4.3 GLOBAL RESTRICTED CUMULATIVITY

The results for both global restricted cumulativity conditions seemed to support the hypothesis. In one condition an interpretation with the constraint violation pattern SubPred+MinDis+Simplex competed with an interpretation that violated FSP. In the other condition an interpretation candidate with the violation pattern SubPred+MinDis+Simplex competed with an interpretation candidate that violated only OvPar.

Especially for the version SubPred+MinDis+Simplex vs. FSP a large majority of the subjects chose the interpretation where FSP was violated. In section 8.2 a side note to this result was given. FSP is a complicated constraint; even when a pronoun is used, contextual or prosodic information can mark it as new. If this information is not given, subjects might fill in the contextual and prosodic information in such a way that the interpretation violates as few constraints as possible. If this is indeed the situation, the interpretation that was supposed to violate FSP does no longer violate any constraints.

The other condition, in which an interpretation with the violation combination SubPred+MinDis+Simplex competed with an interpretation that violates OvPar, the majority of subjects responded according to the global restricted cumulativity hypothesis. The sentences were of the following type:

44)	Will	verzoekt	vandaag	om	Grace	te negeren	en	Jack
	Will	*requests*	*today*	*to*	*Grace*	*ignore*	*and*	*Jack*
	om	Karen	te ontlopen.					
	to	*Karen*	*avoid.*					

The two interpretations of this sentence are the following:

44a)	....en	Jack	verzoekt	vandaag		om	Karen te ontlopen.
	....and	*Jack*	*requests*	*today*		*to*	*Karen avoid.*
44b)	....en	Will	verzoekt	Jack	om		Karen te ontlopen.
	....and	*Will*	*requests*	*Jack*	*to*		*Karen avoid.*

In sentence 44b OvPar is violated: "Jack" is coupled to "vandaag" (*today*), but the syntactic characteristics of "Jack" do not allow it to take the place of an adverbial clause. However, 44b is a plausible sentence, as Jack can function as the (optional) indirect object of "verzoekt" (*requests*). In other cases, violating OvPar yields an unacceptable sentence. This is what happens when OvPar is violated in majority cumulativity condition A - where a hard constraint violation (of OvPar) competes with multiple soft constraint violations. Those sentences were of the following type:

45)	Jack	wacht met Grace	op de ober	en	Will	met	de butler.
	Jack	*waits with Grace*	*for the waiter*	*and*	*Will*	*with*	*the butler.*

Two ways of coupling the remnants with constituents in the first conjuncts are the following:

45a)	....en		Jack	wacht	met Will		met de butler.
	....and	*Jack*	*waits*	*with Will*	*with*		*the butler.*
45b)	....en	Will	wacht	met de butler			op de ober.
	....and	*Will*	*waits*	*with the butler*			*for the waiter.*

Note that in Dutch "wachten" (*to wait*) does not mean "to serve". For sentence 45, there is no plausible alternative for interpretation 45b. A violation of OvPar yields an implausible, strangely constructed interpretation. For sentences like 44, where violating OvPar still produces a plausible sentence, the OvPar-violating interpretation was chosen more often than for sentences like 45, where violating OvPar automatically produces an implausible (possibly even ungrammatical) sentence. When OvPar is violated, there is a difference between a situation in which the second conjunct is still plausible or grammatically acceptable and a situation in which this is not the case.

In standard OT, global restricted cumulativity is the least straightforward to implement of all the cumulativity hypotheses. A combination of constraint violations can overrule all the higher constraints in the hierarchy. With weight values this cannot be implemented, as was already demonstrated in section 4.3.3. Jäger and Rosenbach (to appear) mention constraint conjunction as a way to simulate ganging-up cumulativity (the kind of cumulativity this study deals with). Their problem with constraint conjunction is that the number of parameters in the theory (the number of constraints) grows; if there is an alternative theory with fewer parameters, this theory should be favoured, according to Occam's razor. However, for global restricted cumulativity there is no alternative with weight values. A conjunction of SubPred, MinDis and Simplex, placed high in the hierarchy, would produce predictions that correspond with the experiment results. In traditional OT this solution is not standard. However, at this point this is the best way to implement global restricted cumulativity in a incomplete coordination interpretation model.

9 Conclusion

A constraint evaluation system governed by strict domination does not seem to be able to explain all the experimental findings of the interpretation of incomplete coordination structures. Cumulativity of constraints can explain some phenomena that arise in the interpretation of possibly gapped sentences. From the different cumulativity hypotheses, the data seemed to support global restricted cumulativity most. The responses of about half of subjects seemed to support local restricted cumulativity. However, the absence of context and prosody may have influenced these results considerably. Integrating these factors into experimental research could shed more light on this issue.

All the constraints in this study were formulated as binary constraints. Holding on to a system with only binary constraints cannot explain some of the outcomes of the experiment. At least for the Featural Parallelism constraint, there are indications that using gradience could be a way to cover the interpretation phenomena in possibly gapped sentences. However, gradience is difficult to implement in an OT model.

More extensive experimental research on gapping in Dutch could confirm and extend the findings from this study, especially if context and prosody are included.

Bibliography

Abney, S. (1997) "Stochastic attribute-value grammars" in *Computational Linguistics* 23:4, pp. 597-618.

Berger, A., Della Pietra, S. and Della Pietra, V. (1996) "A maximum entropy approach to natural language processing" in *Computational Linguistics* 22:1, pp. 39-71.

Blutner, R., De Hoop, H. and Hendriks, P. *Optimal Communication (pre-final version)* (to appear with CSLI Publications, Stanford).

Boersma, P. and Hayes, B. (2001) "Empirical tests of the Gradual Learning Algorithm" in *Linguistic Inquiry* 32:1, pp. 45-86.

Bouma, G. (2003) "Doing Dutch Pronouns automatically in Optimality Theory" in *Proceedings of the EACL 2003 Workshop on The Computational Treatment of Anaphora*, Budapest.

Bouma, G., Van Noord, G. and Malouf, R. (2001) "Alpino: Wide-coverage computational analysis of dutch" in *Computational Linguistics in The Netherlands 2000*.

Carlson, K. (2001) "The Effects of Parallelism and Prosody in the Processing of Gapping Structures" in *Language and Speech* 44:1, pp. 1-26

Hendriks, P. and de Hoop, H. (2001) "Optimality Theoretic Semantics" in *Linguistics and Philosophy* 24:1, pp. 1-32.

Hoeks, J.C.J., & Hendriks, P. (2005) "Optimality Theory and human sentence processing: the case of coordination" in B.G. Bara, L. Barsalou, & M. Bucciarelli (Eds.), *Proceedings of the Twenty-Seventh Annual Conference of the Cognitive Science Society*.

Jäger, G. and Rosenbach, A. "The winner takes it all - almost. Cumulativity in grammatical variation" (to appear in *Linguistics*).

Keller, F. (1998) "Gradient Grammaticality as an Effect of Selective Constraint Re-ranking" in Gruber, M.C., Higgins, D., Olson K.S. and Wysocki, T. (eds.) *Papers from the 34th Meeting of the Chicago Linguistic Society*. Vol. 2: The Panels, 95-109. Chicago.

Keller, F. (2001) "Experimental Evidence for Constraint Competition in Gapping Constructions" in Müller, G. and Sternefeld, W. (eds.) *Competition in Syntax*, pp. 211-248, Berlin, Mouton de Gruyter.

Klein, M. and Van den Toorn, M.C. (1991) *Praktische cursus zinsontleding*, Groningen, Wolters Noordhoff.

Kuno, S. (1976) "Gapping, a functional analysis" in *Linguistics Inquiry* 7, pp. 300-318.

Legendre, G., Miyata, Y. and Smolensky, P. (1990) "Harmonic Grammar - A formal multi-level theory of linguistic well-formedness: An application" paper presented at the Proceedings of the Twelfth Annual Conference of the Cognitive Science Society.

Legendre, G., Sorace, A. and Smolensky, P. (2005) "The Optimality Theory - Harmonic Grammar connection" in Legendre, G. and Smolensky, P. (2005) *The Harmonic Mind: From Neural Computation to Optimality-Theoretic Grammar*.

Neijt, A. (1979) *Gapping, a contribution to sentence grammar*. Dordrecht, Foris.

Perquin, A. (1999) *Mind the gap*, master thesis, University of Amsterdam.

Prince, A. and Smolensky, P. (2004/1993) *Optimality Theory: Constraint interaction in generative grammar*. Oxford, Blackwell. Previously distributed as Technical Report RuCCSTR-2, New Brunswick NJ, Rutgers Center for Cognitive Science, Rutgers University.

Prüst, H. (1992) *On discourse structuring, VP Anaphora and Gapping*. PhD thesis, University of Amsterdam.

Schwartzschild, R. (1999) "Givenness, Avoid-F and other constraints on the placement of accent" *Natural Language Semantics* 7, pp. 141-177.
Uit den Boogaart, P. C. (1975) *Woordfrequenties in geschreven en gesproken Nederlands*. Werkgroep Frequentie-onderzoek van het Nederlands. Utrecht Oosthoek, Scheltema & Holkema.

Appendix A: Corpus test sentences

Test sentences from the Eindhoven corpus (Uit den Boogaart, 1975)

First the Dutch sentences are given, then a translation of the sentences. Sentences 8 and 10 consisted of more than two conjuncts. The part between slashes was not included in this study.

Dutch:

1) Je bent vijf dagen van de week gevaarlijk en een of twee dagen ongevaarlijk.

2) Voor de Tarzan-liefhebbers is Johnnie vrijdagavond op de televisie te zien in het NOS-programma Premiere en woensdag 1 juli in de televisiehoofdfilm "Tarzan en de Amazones".

3) Hoek's machines daalde acht en centrale suiker zes punten.

4) Intussen zijn de schepen veel groter en sneller geworden en de charterperioden langer.

5) Negentig parlementsleden stemden voor en tien tegen.

6) Ze willen de ronkende vliegtuigen niet en ook de bommen niet.

7) Ook dient volgens de raadscommissie het geneeskundig toezicht uitgebreid te worden en een commissie uit de ouders samengesteld te worden.

8) Groep 1 trok de arm na vijftien minuten uit de testkamer//, groep 2 na dertig minuten, groep 3 na 45 minuten// en groep 4 na een uur.

9) Wij hebben nu twee keer goed gespeeld, maar ook twee keer verloren.

10) De TROS kwam met zes procent op de derde plaats //, de NCRV met vijf procent op de vierde// en de AVRO - met drie procent - op de vijfde plaats.

Translation:

1) You are dangerous five days a week and harmless two days.

2) For Tarzan lovers, Johnny is on television in the NOS show Premiere on Friday and in the main television movie "Tarzan and the Amazones" on Wednesday July 1st.

3) Hoek's machines dropped eight and centrale suiker six points.

4) Meanwhile, the ships have become much larger and faster and the charter periods longer.

5) Ninety members of parliament voted in favour and ten against.

6) They do not want the roaring planes and not the bombs.

7) Also, according to the council committe, medical supervision should be extended and a committe of parents put together.

8) Group 1 pulled their arm from the test room after fifteen minutes//, group 2 after thirty minutes, group 3 after 45 minutes// and group 4 after an hour.

9) We have played well twice now, but also lost twice.

10) The TROS ended third with six percent//, the NCRV fourth with five percent// and the AVRO - with three percent - fifth.

Appendix B: Artificially constructed test sentences

First the Dutch sentences are given, then a translation of the sentences.

Dutch:

Local restricted cumulativity

1a) Will plukt een roos en een tulp Grace.
1b) Jack eet een snoepje en een koekje Karen.
1c) Stan leest een tijdschrift en een boek Ellen.

Global restricted cumulativity

A: SubPred+MinDis+Simplex versus FSP
2a) Will vraagt hem om Grace te negeren en Jack om Karen te ontlopen.
2b) Stan vraagt haar om Ellen te halen en Will om Grace te brengen.
2c) Jack verzoekt hem om Will te slaan en Stan om Rob te meppen.

B: SubPred+MinDis+Simplex versus OvPar
3a) Will verzoekt vandaag om Grace te negeren en Jack om Karen te ontlopen.
3b) Jack verplicht uitdrukkelijk om Will te groeten en Grace om Stan te verwelkomen.
3c) Grace verzoekt nadrukkelijk om Rob uit te nodigen en Karen om Will mee te vragen.

Majority cumulativity

A: Hard constraints versus soft constraints
4a) Jack wacht met Grace op de ober en Will met de butler.
4b) Will gaat met Karen naar oma en Jack met opa.
4c) Karen lachte met de tuinman om Will en de glazenwasser met Ellen.

B: Soft constraints versus soft constraints
5a) De butler belooft Will te bellen en de ober te faxen.
5b) De conducteur belooft Grace te stoppen en de machinist door te gaan.
5c) De koning belooft het volk te blijven en de prins te vertrekken.

Translated sentences from Carlson

Bake A
6a) Grace bakt cakes voor toeristen en Karen voor haar familie.
6b) Jack gaf koekjes aan de kinderen en Will aan de buren.
6c) Stan vertelt grapjes aan volwassenen en Rob aan kinderen.

Take A
7a) Jack bezocht het kantoor in de vakantie en Karen in het weekend.
7b) Will bekritiseerde het eten tijdens het diner en Karen tijdens het feest.

Introduce
8a) Stan verbaasde de jury met zijn talent en Jack met zijn muzikaliteit.
8b) Will complimenteerde de gastheren met de decoratie en Grace met het eten.

Fillers
9a) Rosario speelde backgammon met Karen en Grace ook.
9b) Stan haalt koffie voor Will en Jack ook.
9c) Grace ging winkelen met Jack en Will ook.
9d) Rob verliest niet graag van Grace en Ellen ook niet.

10a) Stan eet koekjes en drinkt bier.
10b) Grace ontwerpt huizen en richt kantoren in.
10c) Karen ontmoet Ellen en mist Rob.
10d) Jack beledigt de agent en grieft de receptioniste.

Translation:

Local restricted cumulativity

1a) Will picks a rose and a tulip Grace.
1b) Jack eats candy and a cookie Karen.
1c) Stan reads a magazine and a book Ellen.

Global restricted cumulativity

A: SubPred+MinDis+Simplex versus FSP
2a) Will asks him to ignore Grace and Jack to avoid Karen.
2b) Stan asks her to get Ellen and Will to bring Grace.
2c) Jack requests him to hit Will and Stan to slap Rob.

B: SubPred+MinDis+Simplex versus OvPar
3a) Today, Will requests to ignore Grace and Jack to avoid Karen.
3b) Jack pointedly commands to greet Will and Grace to welcome Stan.
3c) Grace pointedly requests to invite Rob and Karen to ask Will to come.

Majority cumulativity

A: Hard constraints versus soft constraints
4a) Jack waits with Grace for the waiter and Will with the butler.
4b) Will goes to grandma with Karen and Jack with grandpa.
4c) Karen laughed at Will with the gardener and the window cleaner with Ellen.

B: Soft constraints versus soft constraints
5a) The butler promises Will to phone and the waiter to fax.
5b) The conductor promises Grace to stop and the driver to continue.
5c) The king promises the people to stay and the prince to leave.

Translated sentences from Carlson

Bake A

6a) Grace bakes cakes for tourists and Karen for her family.

6b) Jack gave cookies to the children and Will to the neighbours.
6c) Stan tells jokes to adults and Rob to children.

Take A

7a) Jack visited the office during the vacation and Karen during the weekend.
7b) Will criticized the food at the dinner and Karen at the party.

Introduce

8a) Stan amazed the jury with his talent and Jack with his musicality.
8b) Will complimented the hosts on the decorations and Grace on the food.

Fillers

9a) Rosario played backgammon with Karen and Grace too.
9b) Stan gets coffee for Will and Jack too.
9c) Grace went shopping with Jack and Will too.
9d) Rob does not like to lose to Grace and Ellen neither.

10a) Stan eats cookies and drinks beer.
10b) Grace designs houses and furnishes offices.
10c) Karen meets Ellen and misses Rob.
10d) Jack offends the officer and grieves the receptionist.

Appendix C: Computer simulation code

```
%% GAPPING by Marieke van der Feen

%% Include lexicon and grammar.
:- [lexicon].
:- [grammar].

%% The hierarchy.
%% A list of constraints I want to include: same, simplex, minDis,
%% subPred, fsp, featPar, themFit, ovPar.
constrainthierarchy([subMinSim,ovPar,fsp,themFit,stay,same,simplex,mi
nDis,subPred,featPar]).

%% An example of a weight set that simulates strict domination. It is
%% currently not used in the existing evaluation methods.
cumuweight(ovPar,1).
cumuweight(fsp,0.45).
cumuweight(themFit,0.2).
cumuweight(stay,0.05).
cumuweight(same,0.01).
cumuweight(simplex,0.001).
cumuweight(minDis,0.0001).
cumuweight(subPred,0.00001).
cumuweight(featPar,0.000001).

%% Weight values for a local restricted cumulative model, where only
%% stay & same are cumulative: they overrule featPar
%% together.
stSaweight(ovPar,1).
stSaweight(fsp,0.5).
stSaweight(themFit,0.25).
stSaweight(stay,0.126).
stSaweight(same,0.125).
stSaweight(simplex,0.0004).
stSaweight(minDis,0.0001).
stSaweight(subPred,0.000004).
stSaweight(featPar,0.000001).


%% The methods for evaluating candidates:
%% nonCumu,majorityCumu,localCumu, globalResCumu.
methods([nonCumu,localCumu,globalResCumu,majorityCumu]).

%% THREE THINGS NEED TO BE DONE: COLLECT COUPLINGS, MAKE A LIST OF
CANDIDATES AND THEIR VIOLATIONS, AND OPTIMIZE

%% Analyze the sentence, give the optimal interpretation and a
%% reconstruction of the second conjunct.
analyze(Sentence) :-
      gatherCouplings(Sentence,Couplings),
      listViolations(Sentence,Couplings,Violations),
      optimize(Sentence,Violations).


%% PARSE THE SENTENCE AND COLLECT ALL NP COUPLINGS

%% Collect all NP couplings.
gatherCouplings(Sentence,Interpretations) :-
```

```
    findNps(Sentence,Antecedents,Remnants,1),
    interpretations(Antecedents, Remnants,Interpretations).

%% Extract all NP's from both conjuncts and divide them into Remnants
%% and Antecedents.
findNps([],[],[],_).
findNps([[_,coord|_]|Sentence],Antecedents,Remnants,_):-
    findRemnants(Sentence,Antecedents,Remnants,1).
findNps([[Element,Role|RestInfo]|Sentence],
[[Element,Number,Role|RestInfo]|Antecedents],Remnants,Number):-
    not(Role=vh),
    not(Role=coord),
    NewNumber is Number + 1,
    findNps(Sentence,Antecedents,Remnants,NewNumber).
findNps([[_,vh|_]|Sentence],Antecedents,Remnants,Number) :-
    findNps(Sentence,Antecedents,Remnants,Number).

%% The np's found after the coord element are the remnants.
findRemnants([],[],[],_).
findRemnants([[Element|RestInfo]|Sentence],Antecedents,
[[Element,Number,RoleOptions|RestInfo]|Remnants],Number) :-
    role(Element,RoleOptions),
    NewNumber is Number + 1,
    findRemnants(Sentence,Antecedents,Remnants,NewNumber).

%% After the NP's have been found, generate all couplings and from
%% them, interpretations (sets of couplings).
interpretations(Antecedents,Remnants,Interpretations) :-
    makeCouplings(Antecedents,Remnants,Couplings),
    genInterpretations(Couplings,Interpretations).

%% Couple all of the remnants to all of the antecedent NP's.
makeCouplings(_,[],[]).
makeCouplings(Antecedents,[Firstremnant|Restremnants],
[Morecouplings|Couplings]) :-
    coupleCurrentRemnant(Firstremnant,Antecedents,Morecouplings),
    makeCouplings(Antecedents,Restremnants,Couplings).

%% Couple the current remnant to all of the antecedent NP's.
coupleCurrentRemnant(_,[],[]).
coupleCurrentRemnant(Remnant,[CurrentAntecedent|Antecedents],
[[Remnant,CurrentAntecedent]|Couplings]) :-
    coupleCurrentRemnant(Remnant,Antecedents,Couplings),!.
coupleCurrentRemnant(Remnant,[_|Antecedents],Couplings) :-
    coupleCurrentRemnant(Remnant,Antecedents,Couplings).


%% Couple a element A with all elements of a list [H|Rest].
couple(_,[],[]).
couple([FirstRem,ThisAnt], [[_,ThisAnt]|Rest], List) :-
    couple([FirstRem,ThisAnt], Rest, List),!.
couple([FirstRem,ThisAnt],[[SecondRem,OtherAnt]|Rest],
[[[FirstRem,ThisAnt],[SecondRem,OtherAnt]]|List]) :-
    couple([FirstRem,ThisAnt], Rest, List).

%% Check if Ant is different from all other Ants in a list.
differentAnts([],_).
differentAnts([[_,FirstAnt]|Rest],Ant) :-
    not(Ant = FirstAnt),
    differentAnts(Rest,Ant).
```

```
%% Couple a element A with all elements of a list of lists
%% [A|FirstCoupling].
coupleGroup(_,[],[]).
coupleGroup(PartInter,[[_,ThisAnt]|FirstCoupling],CoupleList) :-
      not(differentAnts(PartInter,ThisAnt)),
      coupleGroup(PartInter,FirstCoupling,CoupleList),!.
coupleGroup(PartInter,[[Rem3,ThisAnt]|FirstCoupling],
[Couple|CoupleList]) :-
      differentAnts(PartInter,ThisAnt),
      append(PartInter,[[Rem3,ThisAnt]],Couple),
      coupleGroup(PartInter,FirstCoupling,CoupleList).

%% Couple a list of lists [FirstCombi|FirstRemCouplings] with all
%% elements of a list ThirdRemCouplings.
coupleGroups([],_,[]).
coupleGroups([FirstCombi|FirstRemCouplings],ThirdRemCouplings,
AllInter) :-
      coupleGroup(FirstCombi,ThirdRemCouplings,MoreInter),
      append(MoreInter,Inter,AllInter),
      coupleGroups(FirstRemCouplings,ThirdRemCouplings, Inter).

%% Generate all possible couplings of elements of two lists
[First|FirstRemCouplings] and SecondRemCouplings.
coupleFirstSecond([],_,[]).
coupleFirstSecond([First|FirstRemCouplings],SecondRemCouplings,
AllInter) :-
    couple(First,SecondRemCouplings,MoreInter),
    coupleFirstSecond(FirstRemCouplings,SecondRemCouplings,Inter),
    append(MoreInter,Inter,AllInter).

%% Generate all possible couplings of elements.
genInterpretations([FirstRemCouplings],FirstRemInterpretations) :-
      genFirstRemInterpretations(FirstRemCouplings,FirstRemInterpreta
      tions).
genInterpretations([FirstRemCouplings,SecondRemCouplings],AllInter):-
      coupleFirstSecond(FirstRemCouplings,SecondRemCouplings,
      AllInter),!.
genInterpretations([FirstRemCouplings,
SecondRemCouplings|RestRemCouplings],AllInter) :-
      coupleFirstSecond(FirstRemCouplings,SecondRemCouplings,
      PartInter),
      genRestInterpretations(PartInter,RestRemCouplings,AllInter).

genRestInterpretations(PartInter,[RestRemCouplings],AllInter) :-
      coupleGroups(PartInter,RestRemCouplings,AllInter).
genRestInterpretations(PartInter,[RemCouplings|RestRemCouplings],
AllInter) :-
      coupleGroups(PartInter,RemCouplings,MoreInter),
      genRestInterpretations(MoreInter,RestRemCouplings,AllInter).

genFirstRemInterpretations([],[]).
genFirstRemInterpretations([First|FirstRemCouplings],
[[First]|FirstRemInterpretations]) :-
      genFirstRemInterpretations(FirstRemCouplings,
      FirstRemInterpretations).

%% EVALUATE CANDIDATES WITH THE CONSTRAINTS

%% Make a list of the violations for every interpretation. Calculate
%% Max, the highest sum of antecedent rank numbers.
listViolations(Sentence,Interpretations,Allviolations) :-
```

```
	maxList(Interpretations,Max),
	listViolations(Sentence,Interpretations,Allviolations,Max).
listViolations(_,[],[],_).
listViolations(Sentence,[CurrentInter|Interpretations],
[[CurrentInter,CurrentVio]|Allviolations],Max) :-
	constrainthierarchy(X),
	listViolationsCurrent(X,Sentence,Max,CurrentInter,CurrentVio),
	nl,
	write('The following interpretation: '),
	nl,
	write(CurrentInter),
	nl,
	reconstruct(Sentence,CurrentInter,_),
	nl,
	write('violates these constraints: '),
	write(CurrentVio),
	nl,
	listViolations(Sentence,Interpretations,Allviolations,Max).

%% Determine the highest value of sums of antecedents (added up per
%% interpretation).
maxList([MaxInter],MaxSum) :- addAnts(MaxInter,MaxSum).
maxList([CurrentInter,NextInter|Interpretations],Max) :-
	addAnts(CurrentInter,CurrentSum),
	maxList([NextInter|Interpretations],MaxRest),
	max(CurrentSum,MaxRest,Max).

%% Determine the maximum of two integers.
max(X,Y,X) :-
	X >= Y.
max(X,Y,Y) :-
	X < Y.

%% Add up the rank numbers of the antecedents of the same structure.
addAnts([],0).
addAnts([[[Rem,_,RoleOptions],[Ant,Rank,Role|_]]|RestCurrentInter],
Sum) :-
	member(Role,RoleOptions),
	agreement(Rem,Ant,Role),
	addAnts(RestCurrentInter,NewSum),
	Sum is Rank+NewSum,!.
addAnts([[[Rem,_,RoleOptions],[Ant,Rank,Role|_]]|RestCurrentInter],
Sum) :-
	member(X,Role),
	member(X,RoleOptions),
	agreement(Rem,Ant,Role),
	addAnts(RestCurrentInter,NewSum),
	Sum is Rank+NewSum,!.
addAnts([_|RestCurrentInter],Sum) :-
	addAnts(RestCurrentInter,Sum).

%% Make a list of all violations of the current interpretation.
listViolationsCurrent([],_,_,_,[]).
listViolationsCurrent([FirstConstraint|RestConstraints],Sentence,Max,
CurrentInter,AllCurrentVio) :-
	checkConstraint(FirstConstraint,Sentence,Max,CurrentInter,
	MoreCurrentVio),
	listViolationsCurrent(RestConstraints,Sentence,Max,
	CurrentInter,CurrentVio),
	append(MoreCurrentVio,CurrentVio,AllCurrentVio),!.
```

```
%% Check for constraint violations with the variables needed.
%% For checking "same", a start value of 0 is given for checking
%% whether the order of remnants is the same as that of the
%% antecedents.
checkConstraint(same,_,_,CurrentInter,SameVio) :-
      checkSame(CurrentInter,0,SameVio).
checkConstraint(minDis,_,Max,CurrentInter,MinDisVio) :-
      checkMinDis(CurrentInter,Max,MinDisVio).
%% Calculate SubPredFitness and check "subpred".
checkConstraint(subPred,Sentence,_,CurrentInter,SubPredVio) :-
      subPredFitness(Sentence,SubPredFitness),
      checkSubPred(CurrentInter,SubPredFitness,SubPredVio).
checkConstraint(simplex,Sentence,_,CurrentInter,SimplexVio) :-
      simplexFitness(Sentence,CurrentInter,SimplexFitness),
      checkSimplex(Sentence,CurrentInter,SimplexFitness,SimplexVio).
checkConstraint(featPar,_,_,CurrentInter,FeatParVio) :-
      checkFeatPar(CurrentInter,FeatParVio).
checkConstraint(themFit,Sentence,_,CurrentInter,ThemFitVio) :-
      checkThemFit(Sentence,Sentence,CurrentInter,ThemFitVio).
checkConstraint(fsp,_,_,CurrentInter,FspVio) :-
      checkFsp(CurrentInter,FspVio).
checkConstraint(stay,_,_,CurrentInter,StayVio) :-
      checkStay(CurrentInter,StayVio).
checkConstraint(ovPar,_,_,CurrentInter,OvParVio) :-
      checkOvPar(CurrentInter,OvParVio).
%% For now, this is a dummy constraint. Later it can function as a
%% metaconstraint
checkConstraint(subMinSim,_,_,_,[]).

%% CONSTRAINT "SUBPRED". It is checked whether the current
%% interpretation follows the subPred tendency.

%% Check for SubPred-fitness, i.e. whether the constraint is
%% applicable.
subPredFitness([],[]).
subPredFitness([[_,coord|_]|Sentence],SubPredFitness) :-
      remFitness(Sentence,SubPredFitness).
subPredFitness([[_,Role|_]|Sentence],SubPredFitness) :-
      not(Role = coord),
      subPredFitness(Sentence,SubPredFitness).

%% If the second conjunct contains an NP and a Pred, the constraint
%% SubPred is applicable. Otherwise it is not applicable.
remFitness(SecondConjunct,applicable) :-
      member([Np|_],SecondConjunct),
      role(Np,NpRole),
      member(subj,NpRole),
      member([Pred|_],SecondConjunct),
      role(Pred,PredRole),
      member(inf,PredRole),!.
remFitness(_,notapplicable).

%% If the constraint is not applicable (or not included in the
%% hierarchy) it is never violated.
%% Otherwise it is always violated except when the remnant is coupled
%% to the subject of the antecedent sentence.
checkSubPred(_,notapplicable,[]).
checkSubPred([[[_,1|_],[_,_,subj|_]]|_],applicable,[]).
checkSubPred([[[_,1|_],[_,_,NotSubj|_]]|_],applicable,[subPred]) :-
      not(NotSubj=subj).
```

```
%% CONSTRAINT "OVPAR" (OVERT SYNTACTIC PARALLELISM)

%% The role of the antecedent should be among the RoleOptions of the
%% remnant.
checkOvPar([],[]).
checkOvPar([[[Rem,_,RoleOptions],[Ant,_,Role|_]]|RestCurrentInter],
OvParVio) :-
      member(Role,RoleOptions),
      agreement(Rem,Ant,Role),
      checkOvPar(RestCurrentInter,OvParVio),!.
checkOvPar([[[Rem,_,RoleOptions],[Ant,_,Role|_]]|RestCurrentInter],
OvParVio) :-
      member(X,Role),
      member(X,RoleOptions),
      agreement(Rem,Ant,Role),
      checkOvPar(RestCurrentInter,OvParVio),!.
checkOvPar([_|_],[ovPar]).

%% Checks whether agreement demands are not violated. Only applicable
%% for remnants coupled with a subject.
agreement(Remnant,Antecedent,subj) :-
      element(Remnant,_,[Agr,_]),
      element(Antecedent,_,[Agr,_]),!.
agreement(_,_,Role) :-
      not(Role=subj).


%% CONSTRAINT "STAY"

%% Check whether the remnants are coupled in canonical order.
checkStay(CurrentInter,[]) :-
      collectAntRoles(CurrentInter,Roles),
      canonical(Order),
      sublist(Roles,Order),!.
checkStay(CurrentInter,[stay]) :-
      collectAntRoles(CurrentInter,Roles),
      canonical(Order),
      not(sublist(Roles,Order)).

%% Collect the roles of the antecedents and store them in a list. For
%% IO's, check whether they are "marked" with "aan" or
%% "voor", because this has implications for word order
%% possibilities.
collectAntRoles([],[]).
collectAntRoles([[[[voor|_]|_],[_,_,io|_]]|RestInter],Roles) :-
      collectAntRoles(RestInter,AllRoles),
      append([markedIo],AllRoles,Roles),!.
collectAntRoles([[[[aan|_]|_],[_,_,io|_]]|RestInter],Roles) :-
      collectAntRoles(RestInter,AllRoles),
      append([markedIo],AllRoles,Roles),!.
collectAntRoles([[_,[_,_,Role|_]]|RestInter],Roles) :-
      canonical(Order),
      member(Role,Order),
      collectAntRoles(RestInter,AllRoles),
      append([Role],AllRoles,Roles),!.
collectAntRoles([_|RestInter],Roles) :-
      collectAntRoles(RestInter,Roles).

%% Xs is a sublist of AsXsBs.
sublist(Xs,AsXsBs) :-
      append(AsXs,_,AsXsBs), append(_,Xs,AsXs).
```

```
%% CONSTRAINT "THEMFIT"

%% First, the main verb is filtered out, and with its properties the
%% constraint is checked.
checkThemFit([],_,_,[]).
checkThemFit([[FirstElement,vh|_]|_],Sentence,CurrentInter,
ThemFitVio) :-
      element(FirstElement,vaux,Arguments,_),
      not(member([_,[_,_,[vc,inf]]],CurrentInter)),
      member([vc],Arguments),
      checkComplexWithArguments(Sentence,CurrentInter,Arguments,
      ThemFitVio),!.

%% In case of a single verb.
checkThemFit([[FirstElement,vh|_]|_],_,CurrentInter,ThemFitVio) :-
      element(FirstElement,vmain,Arguments),
      checkWithArguments(CurrentInter,Arguments,ThemFitVio),!.
checkThemFit([[FirstElement,vh|_]|_],_,CurrentInter,ThemFitVio) :-
      element(FirstElement,vcop,Arguments),
      checkWithArguments(CurrentInter,Arguments,ThemFitVio),!.
checkThemFit([[FirstElement|_]|RestSentence],Sentence,CurrentInter,
ThemFitVio) :-
      not(element(FirstElement,vh,_)),
      checkThemFit(RestSentence,Sentence,CurrentInter,ThemFitVio).

%% In case of than one verb.
checkComplexWithArguments(Sentence,CurrentInter,Arguments,ThemFitVio)
:-
      findVcArgs(Sentence,VcArgs),
      append(Arguments,VcArgs,AllArgs),
      checkWithArguments(CurrentInter,AllArgs,ThemFitVio).

%% Find the required arguments for the verbal complement.
findVcArgs([],[]).
findVcArgs([[VcInf,[vc,inf]|_]|_],VcArgs) :-
      element(VcInf,_,Properties),
      convertVcArgs(Properties,VcArgs),!.
findVcArgs([_|Sentence],VcArgs) :-
      findVcArgs(Sentence,VcArgs).

%% Extract properties of the arguments.
convertVcArgs([],[]).
convertVcArgs([[FirstArg,FirstProps]|Properties],[[[vc,FirstArg],
FirstProps]|VcArgs]) :-
      convertVcArgs(Properties,VcArgs).


%% The remnant Element and the Role of its antecedent are checked
%% with the Arguments of the main verb.
checkWithArguments([],_,[]).
checkWithArguments([[[Element|_],Ant]|_],Arguments,[themFit]) :-
      not(checkCurrentWithArguments(Ant,Element,Arguments)).
checkWithArguments([[[Element|_],Ant]|RestCurrentInter],Arguments,
ThemFit) :-
      checkCurrentWithArguments(Ant,Element,Arguments),
      checkWithArguments(RestCurrentInter,Arguments,ThemFit).

%% Check one couple for argument properties. If this check fails, the
%% constraint is violated.
checkCurrentWithArguments([_,_,subj|_],Element,Arguments) :-
```

```
      element(Element,_,[_,ThemProperties]),
      member([subj,[_,ArgProperties]],Arguments),
      allmembers(ArgProperties,ThemProperties),!.
checkCurrentWithArguments([_,_,do|_],Element,Arguments) :-
      element(Element,_,[_,ThemProperties]),
      member([do,ArgProperties],Arguments),
      allmembers(ArgProperties,ThemProperties).
checkCurrentWithArguments([_,_,io|_],Element,Arguments) :-
      element(Element,_,[_,ThemProperties]),
      member([io,ArgProperties],Arguments),
      allmembers(ArgProperties,ThemProperties).
%% This constraint only goes for arguments of the verb, not other
%% complements
checkCurrentWithArguments([_,_,Role|_],_,_) :-
      not(Role=do),
      not(Role=io).

%% If not all members of [First|Rest] are members of List, this one
%% fails.
allmembers([],_).
allmembers([First|Rest],List) :-
      member(First,List),
      allmembers(Rest,List).

%% CONSTRAINT "SAME". It is checked whether the remnants are in the
%% same order as the antecedents.

%% For every rank number of antecedents, it is checked whether it is
%% bigger than the last one.
checkSame([],_,[]).
checkSame([[_,[_,Rank|_]]|RestCurrentInter],Number,SameVio) :-
      Rank > Number,
      checkSame(RestCurrentInter,Rank,SameVio).
checkSame([[_,[_,Rank|_]]|_],Number,[same]) :-
      Rank =< Number.

%% CONSTRAINT "MINDIS". It is checked whether the last processed ants
are coupled first.

%% Check "minDis". Only when the Sum of antecedent rank numbers
%% reaches Max, the constraint is not violated.
checkMinDis(CurrentInter,Max,[]) :-
      addAnts(CurrentInter,Sum),
      Sum = Max.
checkMinDis(CurrentInter,Max,[minDis]) :-
      addAnts(CurrentInter,Sum),
      Sum < Max.


%% CONSTRAINT "SIMPLEX"

%% Check whether the constraint is applicable. If the aux. verb
%% constitutes a simplex relationship with all elements,
%% it is not applicable, i.e. never violated. The same goes if there
%% is only one remnant. In other cases, it is applicable.
simplexFitness(_,[_],notapplicable).
simplexFitness(_,[_,_|_],notapplicable) :-
      not(word(_,vaux,_,nonsimplex)).
simplexFitness(Sentence,[_,_|_],notapplicable) :-
      word(Element,vaux,_,nonsimplex),
      not(member([Element|_],Sentence)).
```

```
simplexFitness(Sentence,[_,_|_],applicable) :-
      word(Element,vaux,_,nonsimplex),
      member([Element|_],Sentence).

%% Check whether Simplex is violated.
checkSimplex(_,_,notapplicable,[]).
checkSimplex(_,CurrentInter,applicable,[simplex]) :-
      member([_,[_,_,subj|_]],CurrentInter),
      member([_,[_,_,[vc|_]|_]],CurrentInter),!.
checkSimplex(_,_,applicable,[]).

%% CONSTRAINT "FEATPAR"

%% If all the coupled elements of an interpretation have matching
%% properties, the constraint is not violated.
%% Otherwise, it is.
checkFeatPar([],[]).
checkFeatPar([[[Rem|_],[Ant|_]]|RestCurrentInter],FeatParVio) :-
      element(Rem,_,[_,Properties]),
      element(Ant,_,[_,Properties]),
      checkFeatPar(RestCurrentInter,FeatParVio),!.
checkFeatPar([[[Rem|_],[Ant|_]]|_],[featPar]) :-
      element(Rem,_,[_,Properties]),
      element(Ant,_,[_,OtherProperties]),
      not(Properties=OtherProperties),!.
checkFeatPar([_|RestCurrentInter],FeatParVio) :-
      checkFeatPar(RestCurrentInter,FeatParVio).

%% CONSTRAINT "FSP"

%% If the element is marked as given, FSP is violated.
checkFsp([],[]).
checkFsp([[_,[_,_,_,given]]|_],[fsp]) :- !.
checkFsp([[_,[Ant|_]]|_],[fsp]) :-
      element(Ant,pn,_),!.
checkFsp([_|RestCurrentInter],FspVio) :-
      checkFsp(RestCurrentInter,FspVio).

%% FIND OPTIMAL CANDIDATE WITH THE LIST OF VIOLATIONS PER CANDIDATE

%% Get the list of methods and optimize all methods.
optimize(Sentence,Violations) :-
      methods(Methods),
      optimize(Methods,Sentence,Violations),!.

%% For every method, the optimal candidate is found.
optimize([],_,_).
optimize([FirstMethod|RestMethods],Sentence,Violations) :-
      optimizeCurrentMethod(FirstMethod,Violations,MethodOptimal),
      processResult(FirstMethod,Sentence,MethodOptimal),
      optimize(RestMethods,Sentence,Violations).

%% Check whether an optimal solution has been found and act
%% accordingly.
processResult(FirstMethod,_,[]) :-
      nl,
      write('Method '),
      write(FirstMethod),
      write(' does not yield an optimal candidate. '),!.
processResult(FirstMethod,Sentence,Optimal) :-
      rebuild(FirstMethod,Sentence,Optimal).
```

```
%% For every relevant method, the action to take.
optimizeCurrentMethod(nonCumu,Violations,Optimal) :-
      nl,
      nl,
      write('****STRICT DOMINATION****'),
      nl,
      optimizeNonCumu(Violations,Optimal),
      checkSingleOptimal(Violations,Optimal).
optimizeCurrentMethod(localCumu,Violations,Optimal) :-
      nl,
      nl,
      write('****LOCAL RESTRICTED CUMULATIVITY****'),
      nl,
      optimizeLocalCumu(Violations,Optimal,_,stSaSet).
optimizeCurrentMethod(majorityCumu,Violations,Optimal) :-
      nl,
      nl,
      write('****MAJORITY CUMULATIVITY****'),
      nl,
      optimizeMajorityCumu(Violations,[Optimal]).
optimizeCurrentMethod(globalResCumu,Violations,Optimal) :-
      nl,
      nl,
      write('****GLOBAL RESTRICTED CUMULATIVITY****'),
      nl,
      optimizeGlobalResCumu(Violations,Optimal).

%% Check whether there is more than one optimal candidate.
checkSingleOptimal([[_,OptiVios]|Violations],[Opti,OptiVios]) :-
      checkMoreOptimal(Violations,[Opti,OptiVios]),!.
checkSingleOptimal([_|Violations],[Opti,OptiVios]) :-
      checkSingleOptimal(Violations,[Opti,OptiVios]).

checkMoreOptimal([],_).
checkMoreOptimal([[_,OptiVios]|Violations],[Opti,OptiVios]) :-
      write('There are more candidates with this combination of
      violations:'),
      checkRestOptimal(Violations,[Opti,OptiVios]),!.
checkMoreOptimal([_|Violations],[Opti,OptiVios]) :-
      checkMoreOptimal(Violations,[Opti,OptiVios]).

checkRestOptimal([],_).
checkRestOptimal([[Cand,OptiVios]|Violations],[Opti,OptiVios]) :-
      write(Cand),nl,
      checkRestOptimal(Violations,[Opti,OptiVios]),!.
checkRestOptimal([_|Violations],[Opti,OptiVios]) :-
      checkRestOptimal(Violations,[Opti,OptiVios]).

%% METHOD "GLOBALRESCUMU"

optimizeGlobalResCumu(Violations,Optimal) :-
      addMetaConstraint(Violations,MoreViolations),
      optimizeNonCumu(MoreViolations,Optimal).

%% Check whether simplex, subPred and minDis are violated. Then add
%% subMinSim to the violations.
addMetaConstraint([],[]).
addMetaConstraint([[Candidate,Vios]|RestCandidates],
[[Candidate,MoreVios]|OtherCandidates]) :-
```

```
      member(simplex,Vios),
      member(subPred,Vios),
      member(minDis,Vios),
      append([subMinSim],Vios,MoreVios),
      write(Candidate),
      write(' violates metaconstraint subMinSim '),
      write(MoreVios),
      nl,
      addMetaConstraint(RestCandidates,OtherCandidates),!.

addMetaConstraint([[Candidate,Vios]|RestCandidates],
[[Candidate,Vios]|OtherCandidates]) :-
      addMetaConstraint(RestCandidates,OtherCandidates),!.

%% METHOD "MAJORITYCUMU"

%% Initialize listOptimal, and succeed when only one optimal
%% candidate is found.
optimizeMajorityCumu([First|Candidates],Optimal) :-
      numberVios(First,Number),
      listOptimal(Candidates,[First],Optimal,Number),
      length(Optimal,1),!.
optimizeMajorityCumu([First|Candidates],[Optimal]) :-
      nl,
      write('There is more than one candidate with the lowest number
      of violations.'),
      nl,
      write('Therefore, strict domination will determine the optimal
      candidate.'),
      numberVios(First,Number),
      listOptimal(Candidates,[First],ListLowest,Number),
      optimizeNonCumu(ListLowest,Optimal),
      checkSingleOptimal(ListLowest,Optimal).

%% Initialize listOptimal and succeed if more than one candidate is
%% found. From that set of candidates with the lowest number of
%% violations, find the optimal by strict domination (nonCumu)
listOptimal([],M,M,_) :-
      nl,
      write('The candidate with the lowest number of violations has
      '),
      write(Lowest),
      write(' violation(s).'),
      !.
listOptimal([First|Candidates],LowestSoFar,ListLowest,LowestNumber)
:-
      numberVios(First,Number),
      calculateOptimal(First,Number,LowestSoFar,LowestNumber,
      NewListLowest,NewLowestNumber),
      listOptimal(Candidates,NewListLowest,ListLowest,
      NewLowestNumber).

calculateOptimal(First,Number,LowestSoFar,LowestNumber,NewListLowest,
LowestNumber):-
      Number=LowestNumber,
      append([First],LowestSoFar,NewListLowest),!.
calculateOptimal(First,Number,_,LowestNumber,[First],Number) :-
      min(Number,LowestNumber,Number).
calculateOptimal(_,Number,LowestSoFar,LowestNumber,LowestSoFar,Lowest
Number) :-
      min(Number,LowestNumber,LowestNumber).
```

```
%% Calculates the length of the list of violations.
numberVios([_,Vios],Number) :-
	length(Vios,Number).

%% METHOD "LOCALCUMU"

%% The same as the general cumu-method, the only difference being a
%% changed weight set (stSaWeights), in predicate "AddValues".

%% Compare all harmony values and give the lowest as output.
optimizeLocalCumu([First|Violations],LowestName,LowestValue,
WeightSet) :-
	findHarmonyValue(First,FirstValue,WeightSet),
	optimizeLocalCumu(Violations,First,FirstValue,LowestName,
	LowestValue,WeightSet),!.

optimizeLocalCumu([],Lowest,LowestValue,Lowest,LowestValue,_) :-
	nl,
	write('The harmony value of the optimal second conjunct is '),
	write(LowestValue),
	write('.').
optimizeLocalCumu([First|Violations],_,LowestSoFarValue,LowestName,Lo
westValue,WeightSet) :-
	findHarmonyValue(First,FirstValue,WeightSet),
	min(FirstValue,LowestSoFarValue,FirstValue),
	optimizeLocalCumu(Violations,First,FirstValue,LowestName,
	LowestValue,WeightSet),!.
optimizeLocalCumu([First|Violations],LowestSoFar,LowestSoFarValue,
LowestName,LowestValue,WeightSet) :-
	findHarmonyValue(First,FirstValue,WeightSet),
	min(FirstValue,LowestSoFarValue,LowestSoFarValue),
	optimizeLocalCumu(Violations,LowestSoFar,LowestSoFarValue,
	LowestName,LowestValue,WeightSet),!.

%% Find and write the harmony value of a certain interpretation.
findHarmonyValue(First,FirstValue,WeightSet) :-
	harmonyValue(First,FirstValue,WeightSet).

%% Determine the weighted sum of violations.
harmonyValue([_,Vios],Value,WeightSet) :-
	addValues(Vios,Value,WeightSet).

addValues([],0,_).
addValues([FirstVio|RestVios],TotalValue,cumuSet) :-
	cumuweight(FirstVio,Weight),
	addValues(RestVios,Value,cumuSet),
	TotalValue is Value + Weight,!.
addValues([FirstVio|RestVios],TotalValue,stSaSet) :-
	stSaweight(FirstVio,Weight),
	addValues(RestVios,Value,stSaSet),
	TotalValue is Value + Weight,!.

%% Return the lowest value of two integers.
min(X,Y,X) :-
	X =< Y.
min(X,Y,Y) :-
	X > Y.

%% METHOD "NONCUMU"
```

```
optimizeNonCumu([Optimal],Optimal).
optimizeNonCumu([First|Candidates],Optimal) :-
      optimizeNonCumu(Candidates,OptiRest),
      bestCand(First,OptiRest,Optimal).

bestCand(X,Y,Best) :-
    constrainthierarchy(Z),
    bestCand(X,Y,Z,Best).

bestCand(X,_,[],X).
bestCand([_,XVio],[Y,YVio],[Zfirst|_],[Y,YVio]) :-
      member(Zfirst,XVio),
      not(member(Zfirst,YVio)),!.
bestCand([X,XVio],[_,YVio],[Zfirst|_],[X,XVio]) :-
      member(Zfirst,YVio),
      not(member(Zfirst,XVio)),!.
bestCand(X,Y,[_|Zrest],Opti) :-
      bestCand(X,Y,Zrest,Opti).


%% Make a reconstruction of the sentence.
rebuild(FirstMethod,Sentence,[Optimal,Violations]) :-
      nl,
      write('A reconstruction of the optimal second conjunct with the
      method '),
      write(FirstMethod),
      write(' is: '),
      nl,
      reconstruct(Sentence,Optimal,_),
      nl,
      write('with violations: '),
      write(Violations).

reconstruct([],_,[]).
reconstruct([[_,coord|_]|_],_,[]).
reconstruct([[Element|_]|Sentence],Optimal,
[#,Remnant,#|Reconstruction]) :-
      member([[Remnant|_],[Element|_]],Optimal),
      write('#'),
      write(Remnant),
      write('# '),
      reconstruct(Sentence,Optimal,Reconstruction),!.
reconstruct([[Element|_]|Sentence],Optimal,[Element|Reconstruction])
:-
      write(Element),
      write(' '),
      reconstruct(Sentence,Optimal,Reconstruction).
```

Appendix D: Computer simulation results

Sentence 1

Je	**bent**	**vijf dagen van de week**	**gevaarlijk**	**en**
You	***are***	***five days a week***	***dangerous***	***and***

een of twee dagen	**ongevaarlijk.**
one or two days	***harmless.***

```
25 ?- analyze([[je,subj],[bent,vh],[[vijf, dagen, van, de,
week],advPhrTemp],[gevaarlijk,nomPred],[en,coord],[[een, of, twee,
dagen]],[ongevaarlijk]]).

The following interpretation:
[[[[een, of, twee, dagen], 1, [advPhrTemp, subj, do]], [je, 1,
subj]], [[ongevaarlijk, 2, [nomPred]], [[vijf, dagen, van, de, week],
2, advPhrTemp]]]
#[een, of, twee, dagen]# bent #ongevaarlijk# gevaarlijk
violates these constraints: [ovPar, fsp, minDis, featPar]

The following interpretation:
[[[[een, of, twee, dagen], 1, [advPhrTemp, subj, do]], [je, 1,
subj]], [[ongevaarlijk, 2, [nomPred]], [gevaarlijk, 3, nomPred]]]
#[een, of, twee, dagen]# bent [vijf, dagen, van, de, week]
#ongevaarlijk#
violates these constraints: [ovPar, fsp, minDis, featPar]

The following interpretation:
[[[[een, of, twee, dagen], 1, [advPhrTemp, subj, do]], [[vijf, dagen,
van, de, week], 2, advPhrTemp]], [[ongevaarlijk, 2, [nomPred]], [je,
1, subj]]]
#ongevaarlijk# bent #[een, of, twee, dagen]# gevaarlijk
violates these constraints: [ovPar, fsp, same, minDis, featPar]

The following interpretation:
[[[[een, of, twee, dagen], 1, [advPhrTemp, subj, do]], [[vijf, dagen,
van, de, week], 2, advPhrTemp]], [[ongevaarlijk, 2, [nomPred]],
[gevaarlijk, 3, nomPred]]]
je bent #[een, of, twee, dagen]# #ongevaarlijk#
violates these constraints: []

The following interpretation:
[[[[een, of, twee, dagen], 1, [advPhrTemp, subj, do]], [gevaarlijk,
3, nomPred]], [[ongevaarlijk, 2, [nomPred]], [je, 1, subj]]]
#ongevaarlijk# bent [vijf, dagen, van, de, week] #[een, of, twee,
dagen]#
violates these constraints: [ovPar, fsp, same, minDis, featPar]

The following interpretation:
[[[[een, of, twee, dagen], 1, [advPhrTemp, subj, do]], [gevaarlijk,
3, nomPred]], [[ongevaarlijk, 2, [nomPred]], [[vijf, dagen, van, de,
week], 2, advPhrTemp]]]
je bent #ongevaarlijk# #[een, of, twee, dagen]#
violates these constraints: [ovPar, same, featPar]


****STRICT DOMINATION****
```

```
A reconstruction of the optimal second conjunct with the method
nonCumu is:
je bent #[een, of, twee, dagen]# #ongevaarlijk#
with violations: []

****LOCAL RESTRICTED CUMULATIVITY****

The harmony value of the optimal second conjunct is 0.
A reconstruction of the optimal second conjunct with the method
localCumu is:
je bent #[een, of, twee, dagen]# #ongevaarlijk#
with violations: []

****GLOBAL RESTRICTED CUMULATIVITY****

A reconstruction of the optimal second conjunct with the method
globalResCumu is:
je bent #[een, of, twee, dagen]# #ongevaarlijk#
with violations: []

****MAJORITY CUMULATIVITY****

The candidate with the lowest number of violations has 0
violation(s).
A reconstruction of the optimal second conjunct with the method
majorityCumu is:
je bent #[een, of, twee, dagen]# #ongevaarlijk#
with violations: []
```

Sentence 2

Voor de Tarzan-liefhebbers		**is**	**Johnnie**	**vrijdagavond**
For Tarzan-lovers		***is***	***Johnnie***	***Friday night***

op de televisie	**te zien**	**in het NOS-programma Premiere**	**en**
on television	***to see***	***in the NOS show Premiere***	***and***

woensdag 1 juli	**in de televisiehoofdfilm " Tarzan en de**
Wednesday July 1st	***in the main television film "Tarzan and the***

Amazones".
Amazones".

```
analyze([[[voor, de,
tarzanliefhebbers],advPhr],[is,vh],[johnnie,subj],[vrijdagavon
d,advPhrTemp],[[op, de, televisie],advPhr],[[te,
zien],vmain],[[in,het,nosprogrammapremiere],advPhr],[en,coord]
,[woensdageenjuli],[[in, de, televisiehoofdfilm, tarzan, en,
de, amazones]]]).

The following interpretation:
[[[woensdageenjuli, 1, [advPhrTemp, subj, do]], [johnnie, 2,
subj]], [[[in, de, televisiehoofdfilm, tarzan, en, de,
amazones], 2, [advPhr]], [vrijdagavond, 3, advPhrTemp]]]
```

```
[voor, de, tarzanliefhebbers] is #woensdageenjuli# #[in, de,
televisiehoofdfilm, tarzan, en, de, amazones]# [op, de,
televisie] [te, zien] [in, het, nosprogrammapremiere]
violates these constraints: [ovPar, minDis, featPar]

The following interpretation:
[[[woensdageenjuli, 1, [advPhrTemp, subj, do]], [johnnie, 2,
subj]], [[[in, de, televisiehoofdfilm, tarzan, en, de,
amazones], 2, [advPhr]], [[op, de, televisie], 4, advPhr]]]
[voor, de, tarzanliefhebbers] is #woensdageenjuli#
vrijdagavond #[in, de, televisiehoofdfilm, tarzan, en, de,
amazones]# [te, zien] [in, het, nosprogrammapremiere]
violates these constraints: [minDis, featPar]

The following interpretation:
[[[woensdageenjuli, 1, [advPhrTemp, subj, do]], [johnnie, 2,
subj]], [[[in, de, televisiehoofdfilm, tarzan, en, de,
amazones], 2, [advPhr]], [[te, zien], 5, vmain]]]
[voor, de, tarzanliefhebbers] is #woensdageenjuli#
vrijdagavond [op, de, televisie] #[in, de, televisiehoofdfilm,
tarzan, en, de, amazones]# [in, het, nosprogrammapremiere]
violates these constraints: [ovPar, minDis, featPar]

The following interpretation:
[[[woensdageenjuli, 1, [advPhrTemp, subj, do]], [johnnie, 2,
subj]], [[[in, de, televisiehoofdfilm, tarzan, en, de,
amazones], 2, [advPhr]], [[in, het, nosprogrammapremiere], 6,
advPhr]]]
[voor, de, tarzanliefhebbers] is #woensdageenjuli#
vrijdagavond [op, de, televisie] [te, zien] #[in, de,
televisiehoofdfilm, tarzan, en, de, amazones]#
violates these constraints: [minDis, featPar]

The following interpretation:
[[[woensdageenjuli, 1, [advPhrTemp, subj, do]], [vrijdagavond,
3, advPhrTemp]], [[[in, de, televisiehoofdfilm, tarzan, en,
de, amazones], 2, [advPhr]], [[voor, de, tarzanliefhebbers],
1, advPhr]]]
#[in, de, televisiehoofdfilm, tarzan, en, de, amazones]# is
johnnie #woensdageenjuli# [op, de, televisie] [te, zien] [in,
het, nosprogrammapremiere]
violates these constraints: [same, minDis, featPar]

The following interpretation:
[[[woensdageenjuli, 1, [advPhrTemp, subj, do]], [vrijdagavond,
3, advPhrTemp]], [[[in, de, televisiehoofdfilm, tarzan, en,
de, amazones], 2, [advPhr]], [johnnie, 2, subj]]]
[voor, de, tarzanliefhebbers] is #[in, de, televisiehoofdfilm,
tarzan, en, de, amazones]# #woensdageenjuli# [op, de,
televisie] [te, zien] [in, het, nosprogrammapremiere]
violates these constraints: [ovPar, same, minDis, featPar]

The following interpretation:
[[[woensdageenjuli, 1, [advPhrTemp, subj, do]], [vrijdagavond,
3, advPhrTemp]], [[[in, de, televisiehoofdfilm, tarzan, en,
```

de, amazones], 2, [advPhr]], [[op, de, televisie], 4, advPhr]]]
[voor, de, tarzanliefhebbers] is johnnie #woensdageenjuli# #[in, de, televisiehoofdfilm, tarzan, en, de, amazones]# [te, zien] [in, het, nosprogrammapremiere]
violates these constraints: [minDis, featPar]

The following interpretation:
[[[woensdageenjuli, 1, [advPhrTemp, subj, do]], [vrijdagavond, 3, advPhrTemp]], [[[in, de, televisiehoofdfilm, tarzan, en, de, amazones], 2, [advPhr]], [[te, zien], 5, vmain]]]
[voor, de, tarzanliefhebbers] is johnnie #woensdageenjuli# [op, de, televisie] #[in, de, televisiehoofdfilm, tarzan, en, de, amazones]# [in, het, nosprogrammapremiere]
violates these constraints: [ovPar, minDis]

The following interpretation:
[[[woensdageenjuli, 1, [advPhrTemp, subj, do]], [vrijdagavond, 3, advPhrTemp]], [[[in, de, televisiehoofdfilm, tarzan, en, de, amazones], 2, [advPhr]], [[in, het, nosprogrammapremiere], 6, advPhr]]]
[voor, de, tarzanliefhebbers] is johnnie #woensdageenjuli# [op, de, televisie] [te, zien] #[in, de, televisiehoofdfilm, tarzan, en, de, amazones]#
violates these constraints: []

The following interpretation:
[[[woensdageenjuli, 1, [advPhrTemp, subj, do]], [[op, de, televisie], 4, advPhr]], [[[in, de, televisiehoofdfilm, tarzan, en, de, amazones], 2, [advPhr]], [[voor, de, tarzanliefhebbers], 1, advPhr]]]
#[in, de, televisiehoofdfilm, tarzan, en, de, amazones]# is johnnie vrijdagavond #woensdageenjuli# [te, zien] [in, het, nosprogrammapremiere]
violates these constraints: [ovPar, same, minDis, featPar]

The following interpretation:
[[[woensdageenjuli, 1, [advPhrTemp, subj, do]], [[op, de, televisie], 4, advPhr]], [[[in, de, televisiehoofdfilm, tarzan, en, de, amazones], 2, [advPhr]], [johnnie, 2, subj]]]
[voor, de, tarzanliefhebbers] is #[in, de, televisiehoofdfilm, tarzan, en, de, amazones]# vrijdagavond #woensdageenjuli# [te, zien] [in, het, nosprogrammapremiere]
violates these constraints: [ovPar, same, minDis, featPar]

The following interpretation:
[[[woensdageenjuli, 1, [advPhrTemp, subj, do]], [[op, de, televisie], 4, advPhr]], [[[in, de, televisiehoofdfilm, tarzan, en, de, amazones], 2, [advPhr]], [vrijdagavond, 3, advPhrTemp]]]
[voor, de, tarzanliefhebbers] is johnnie #[in, de, televisiehoofdfilm, tarzan, en, de, amazones]# #woensdageenjuli# [te, zien] [in, het, nosprogrammapremiere]
violates these constraints: [ovPar, same, minDis, featPar]

The following interpretation:

[[[woensdageenjuli, 1, [advPhrTemp, subj, do]], [[op, de, televisie], 4, advPhr]], [[[in, de, televisiehoofdfilm, tarzan, en, de, amazones], 2, [advPhr]], [[te, zien], 5, vmain]]]
[voor, de, tarzanliefhebbers] is johnnie vrijdagavond #woensdageenjuli# #[in, de, televisiehoofdfilm, tarzan, en, de, amazones]# [in, het, nosprogrammapremiere]
violates these constraints: [ovPar, minDis, featPar]

The following interpretation:
[[[woensdageenjuli, 1, [advPhrTemp, subj, do]], [[op, de, televisie], 4, advPhr]], [[[in, de, televisiehoofdfilm, tarzan, en, de, amazones], 2, [advPhr]], [[in, het, nosprogrammapremiere], 6, advPhr]]]
[voor, de, tarzanliefhebbers] is johnnie vrijdagavond #woensdageenjuli# [te, zien] #[in, de, televisiehoofdfilm, tarzan, en, de, amazones]#
violates these constraints: [ovPar, minDis, featPar]

The following interpretation:
[[[woensdageenjuli, 1, [advPhrTemp, subj, do]], [[te, zien], 5, vmain]], [[[in, de, televisiehoofdfilm, tarzan, en, de, amazones], 2, [advPhr]], [[voor, de, tarzanliefhebbers], 1, advPhr]]]
#[in, de, televisiehoofdfilm, tarzan, en, de, amazones]# is johnnie vrijdagavond [op, de, televisie] #woensdageenjuli# [in, het, nosprogrammapremiere]
violates these constraints: [ovPar, same, minDis, featPar]

The following interpretation:
[[[woensdageenjuli, 1, [advPhrTemp, subj, do]], [[te, zien], 5, vmain]], [[[in, de, televisiehoofdfilm, tarzan, en, de, amazones], 2, [advPhr]], [johnnie, 2, subj]]]
[voor, de, tarzanliefhebbers] is #[in, de, televisiehoofdfilm, tarzan, en, de, amazones]# vrijdagavond [op, de, televisie] #woensdageenjuli# [in, het, nosprogrammapremiere]
violates these constraints: [ovPar, same, minDis, featPar]

The following interpretation:
[[[woensdageenjuli, 1, [advPhrTemp, subj, do]], [[te, zien], 5, vmain]], [[[in, de, televisiehoofdfilm, tarzan, en, de, amazones], 2, [advPhr]], [vrijdagavond, 3, advPhrTemp]]]
[voor, de, tarzanliefhebbers] is johnnie #[in, de, televisiehoofdfilm, tarzan, en, de, amazones]# [op, de, televisie] #woensdageenjuli# [in, het, nosprogrammapremiere]
violates these constraints: [ovPar, same, minDis, featPar]

The following interpretation:
[[[woensdageenjuli, 1, [advPhrTemp, subj, do]], [[te, zien], 5, vmain]], [[[in, de, televisiehoofdfilm, tarzan, en, de, amazones], 2, [advPhr]], [[op, de, televisie], 4, advPhr]]]
[voor, de, tarzanliefhebbers] is johnnie vrijdagavond #[in, de, televisiehoofdfilm, tarzan, en, de, amazones]# #woensdageenjuli# [in, het, nosprogrammapremiere]
violates these constraints: [ovPar, same, minDis, featPar]

```
The following interpretation:
[[[woensdageenjuli, 1, [advPhrTemp, subj, do]], [[te, zien],
5, vmain]], [[[in, de, televisiehoofdfilm, tarzan, en, de,
amazones], 2, [advPhr]], [[in, het, nosprogrammapremiere], 6,
advPhr]]]
[voor, de, tarzanliefhebbers] is johnnie vrijdagavond [op, de,
televisie] #woensdageenjuli# #[in, de, televisiehoofdfilm,
tarzan, en, de, amazones]#
violates these constraints: [ovPar, minDis]

The following interpretation:
[[[woensdageenjuli, 1, [advPhrTemp, subj, do]], [[in, het,
nosprogrammapremiere], 6, advPhr]], [[[in, de,
televisiehoofdfilm, tarzan, en, de, amazones], 2, [advPhr]],
[[voor, de, tarzanliefhebbers], 1, advPhr]]]
#[in, de, televisiehoofdfilm, tarzan, en, de, amazones]# is
johnnie vrijdagavond [op, de, televisie] [te, zien]
#woensdageenjuli#
violates these constraints: [ovPar, same, minDis, featPar]

The following interpretation:
[[[woensdageenjuli, 1, [advPhrTemp, subj, do]], [[in, het,
nosprogrammapremiere], 6, advPhr]], [[[in, de,
televisiehoofdfilm, tarzan, en, de, amazones], 2, [advPhr]],
[johnnie, 2, subj]]]
[voor, de, tarzanliefhebbers] is #[in, de, televisiehoofdfilm,
tarzan, en, de, amazones]# vrijdagavond [op, de, televisie]
[te, zien] #woensdageenjuli#
violates these constraints: [ovPar, same, minDis, featPar]

The following interpretation:
[[[woensdageenjuli, 1, [advPhrTemp, subj, do]], [[in, het,
nosprogrammapremiere], 6, advPhr]], [[[in, de,
televisiehoofdfilm, tarzan, en, de, amazones], 2, [advPhr]],
[vrijdagavond, 3, advPhrTemp]]]
[voor, de, tarzanliefhebbers] is johnnie #[in, de,
televisiehoofdfilm, tarzan, en, de, amazones]# [op, de,
televisie] [te, zien] #woensdageenjuli#
violates these constraints: [ovPar, same, minDis, featPar]

The following interpretation:
[[[woensdageenjuli, 1, [advPhrTemp, subj, do]], [[in, het,
nosprogrammapremiere], 6, advPhr]], [[[in, de,
televisiehoofdfilm, tarzan, en, de, amazones], 2, [advPhr]],
[[op, de, televisie], 4, advPhr]]]
[voor, de, tarzanliefhebbers] is johnnie vrijdagavond #[in,
de, televisiehoofdfilm, tarzan, en, de, amazones]# [te, zien]
#woensdageenjuli#
violates these constraints: [ovPar, same, minDis, featPar]

The following interpretation:
[[[woensdageenjuli, 1, [advPhrTemp, subj, do]], [[in, het,
nosprogrammapremiere], 6, advPhr]], [[[in, de,
televisiehoofdfilm, tarzan, en, de, amazones], 2, [advPhr]],
[[te, zien], 5, vmain]]]
```

```
[voor, de, tarzanliefhebbers] is johnnie vrijdagavond [op, de,
televisie] #[in, de, televisiehoofdfilm, tarzan, en, de,
amazones]# #woensdageenjuli#
violates these constraints: [ovPar, same, minDis, featPar]

****STRICT DOMINATION****

A reconstruction of the optimal second conjunct with the
method nonCumu is:
[voor, de, tarzanliefhebbers] is johnnie #woensdageenjuli#
[op, de, televisie] [te, zien] #[in, de, televisiehoofdfilm,
tarzan, en, de, amazones]#
with violations: []

****LOCAL RESTRICTED CUMULATIVITY****

The harmony value of the optimal second conjunct is 0.
A reconstruction of the optimal second conjunct with the
method localCumu is:
[voor, de, tarzanliefhebbers] is johnnie #woensdageenjuli#
[op, de, televisie] [te, zien] #[in, de, televisiehoofdfilm,
tarzan, en, de, amazones]#
with violations: []

****GLOBAL RESTRICTED CUMULATIVITY****

A reconstruction of the optimal second conjunct with the
method globalResCumu is:
[voor, de, tarzanliefhebbers] is johnnie #woensdageenjuli#
[op, de, televisie] [te, zien] #[in, de, televisiehoofdfilm,
tarzan, en, de, amazones]#
with violations: []

****MAJORITY CUMULATIVITY****

The candidate with the lowest number of violations has 0
violation(s).
A reconstruction of the optimal second conjunct with the
method majorityCumu is:
[voor, de, tarzanliefhebbers] is johnnie #woensdageenjuli#
[op, de, televisie] [te, zien] #[in, de, televisiehoofdfilm,
tarzan, en, de, amazones]#
with violations: []
```

Sentence 3

Hoek's machines	**daalde**	**acht**	**en**	**centrale suiker**	**zes**
Hoek's machines	***dropped***	***eight***	***and***	***centrale suiker***	***six***

punten.
points.

```
10 ?-
analyze([[hoeksmachines,subj],[daalde,vh],[acht,do],[en,coord],[centr
alesuiker],[[zes, punten]]]).

The following interpretation:
[[[centralesuiker, 1, [subj, do, io]], [hoeksmachines, 1, subj]],
[[[zes, punten], 2, [advPhrNumber, subj, do, io]], [acht, 2, do]]]
#centralesuiker# daalde #[zes, punten]#
violates these constraints: []

The following interpretation:
[[[centralesuiker, 1, [subj, do, io]], [acht, 2, do]], [[[zes,
punten], 2, [advPhrNumber, subj, do, io]], [hoeksmachines, 1, subj]]]
#[zes, punten]# daalde #centralesuiker#
violates these constraints: [ovPar, stay, same]


****STRICT DOMINATION****

A reconstruction of the optimal second conjunct with the method
nonCumu is:
#centralesuiker# daalde #[zes, punten]#
with violations: []

****LOCAL RESTRICTED CUMULATIVITY****

The harmony value of the optimal second conjunct is 0.
A reconstruction of the optimal second conjunct with the method
localCumu is:
#centralesuiker# daalde #[zes, punten]#
with violations: []

****GLOBAL RESTRICTED CUMULATIVITY****

A reconstruction of the optimal second conjunct with the method
globalResCumu is:
#centralesuiker# daalde #[zes, punten]#
with violations: []

****MAJORITY CUMULATIVITY****

The candidate with the lowest number of violations has 0
violation(s).
A reconstruction of the optimal second conjunct with the method
majorityCumu is:
#centralesuiker# daalde #[zes, punten]#
with violations: []
```

Sentence 4

Intussen	**zijn**	**de schepen**	**veel groter en sneller**	**geworden**
Meanwhile	***have***	***the ships***	***much bigger and faster***	***become***

en	**de charterperioden**	**langer.**
and	***the charter periods***	***longer.***

```
21 ?- analyze([[intussen,advPhr],[zijn,vh],[[de,
schepen],subj],[[veel, groter, en,
```

```
sneller],nomPred],[geworden,vmain],[en,coord],[[de,
charterperioden]],[langer]]).

The following interpretation:
[[[[de, charterperioden], 1, [subj, do, io]], [intussen, 1, advPhr]],
[[langer, 2, [nomPred]], [[de, schepen], 2, subj]]]
#[de, charterperioden]# zijn #langer# [veel, groter, en, sneller]
geworden
violates these constraints: [ovPar, minDis, featPar]

The following interpretation:
[[[[de, charterperioden], 1, [subj, do, io]], [intussen, 1, advPhr]],
[[langer, 2, [nomPred]], [[veel, groter, en, sneller], 3, nomPred]]]
#[de, charterperioden]# zijn [de, schepen] #langer# geworden
violates these constraints: [ovPar, minDis]

The following interpretation:
[[[[de, charterperioden], 1, [subj, do, io]], [intussen, 1, advPhr]],
[[langer, 2, [nomPred]], [geworden, 4, vmain]]]
#[de, charterperioden]# zijn [de, schepen] [veel, groter, en,
sneller] #langer#
violates these constraints: [ovPar, minDis]

The following interpretation:
[[[[de, charterperioden], 1, [subj, do, io]], [[de, schepen], 2,
subj]], [[langer, 2, [nomPred]], [intussen, 1, advPhr]]]
#langer# zijn #[de, charterperioden]# [veel, groter, en, sneller]
geworden
violates these constraints: [ovPar, same, minDis]

The following interpretation:
[[[[de, charterperioden], 1, [subj, do, io]], [[de, schepen], 2,
subj]], [[langer, 2, [nomPred]], [[veel, groter, en, sneller], 3,
nomPred]]]
intussen zijn #[de, charterperioden]# #langer# geworden
violates these constraints: []

The following interpretation:
[[[[de, charterperioden], 1, [subj, do, io]], [[de, schepen], 2,
subj]], [[langer, 2, [nomPred]], [geworden, 4, vmain]]]
intussen zijn #[de, charterperioden]# [veel, groter, en, sneller]
#langer#
violates these constraints: [ovPar, minDis]

The following interpretation:
[[[[de, charterperioden], 1, [subj, do, io]], [[veel, groter, en,
sneller], 3, nomPred]], [[langer, 2, [nomPred]], [intussen, 1,
advPhr]]]
#langer# zijn [de, schepen] #[de, charterperioden]# geworden
violates these constraints: [ovPar, same, minDis]

The following interpretation:
[[[[de, charterperioden], 1, [subj, do, io]], [[veel, groter, en,
sneller], 3, nomPred]], [[langer, 2, [nomPred]], [[de, schepen], 2,
subj]]]
intussen zijn #langer# #[de, charterperioden]# geworden
violates these constraints: [ovPar, same, minDis, featPar]

The following interpretation:
```

```
[[[[de, charterperioden], 1, [subj, do, io]], [[veel, groter, en,
sneller], 3, nomPred]], [[langer, 2, [nomPred]], [geworden, 4,
vmain]]]
intussen zijn [de, schepen] #[de, charterperioden]# #langer#
violates these constraints: [ovPar, minDis]

The following interpretation:
[[[[de, charterperioden], 1, [subj, do, io]], [geworden, 4, vmain]],
[[langer, 2, [nomPred]], [intussen, 1, advPhr]]]
#langer# zijn [de, schepen] [veel, groter, en, sneller] #[de,
charterperioden]#
violates these constraints: [ovPar, same, minDis]

The following interpretation:
[[[[de, charterperioden], 1, [subj, do, io]], [geworden, 4, vmain]],
[[langer, 2, [nomPred]], [[de, schepen], 2, subj]]]
intussen zijn #langer# [veel, groter, en, sneller] #[de,
charterperioden]#
violates these constraints: [ovPar, same, minDis, featPar]

The following interpretation:
[[[[de, charterperioden], 1, [subj, do, io]], [geworden, 4, vmain]],
[[langer, 2, [nomPred]], [[veel, groter, en, sneller], 3, nomPred]]]
intussen zijn [de, schepen] #langer# #[de, charterperioden]#
violates these constraints: [ovPar, same, minDis]


****STRICT DOMINATION****

A reconstruction of the optimal second conjunct with the method
nonCumu is:
intussen zijn #[de, charterperioden]# #langer# geworden
with violations: []

****LOCAL RESTRICTED CUMULATIVITY****

The harmony value of the optimal second conjunct is 0.
A reconstruction of the optimal second conjunct with the method
localCumu is:
intussen zijn #[de, charterperioden]# #langer# geworden
with violations: []

****GLOBAL RESTRICTED CUMULATIVITY****

A reconstruction of the optimal second conjunct with the method
globalResCumu is:
intussen zijn #[de, charterperioden]# #langer# geworden
with violations: []

****MAJORITY CUMULATIVITY****

The candidate with the lowest number of violations has 0
violation(s).
A reconstruction of the optimal second conjunct with the method
majorityCumu is:
intussen zijn #[de, charterperioden]# #langer# geworden
with violations: []
```

Sentence 5

Negentig parlementsleden	**stemden**	**voor**	**en**	**tien**
Ninety *Members of Parliament*	***voted***	***in favour***	***and***	***ten***

tegen.
against.

```
analyze([[[negentig,
parlementsleden],subj],[stemden,vh],[voor,advPhr],[en,coord],[tien],[
tegen]]).

The following interpretation:
[[[tien, 1, [subj, do, io]], [[negentig, parlementsleden], 1, subj]],
[[tegen, 2, [advPhr]], [voor, 2, advPhr]]]
#tien# stemden #tegen#
violates these constraints: []

The following interpretation:
[[[tien, 1, [subj, do, io]], [voor, 2, advPhr]], [[tegen, 2,
[advPhr]], [[negentig, parlementsleden], 1, subj]]]
#tegen# stemden #tien#
violates these constraints: [ovPar, same]


****STRICT DOMINATION****

A reconstruction of the optimal second conjunct with the method
nonCumu is:
#tien# stemden #tegen#
with violations: []

****LOCAL RESTRICTED CUMULATIVITY****

The harmony value of the optimal second conjunct is 0.
A reconstruction of the optimal second conjunct with the method
localCumu is:
#tien# stemden #tegen#
with violations: []

****GLOBAL RESTRICTED CUMULATIVITY****

A reconstruction of the optimal second conjunct with the method
globalResCumu is:
#tien# stemden #tegen#
with violations: []

****MAJORITY CUMULATIVITY****

The candidate with the lowest number of violations has 0
violation(s).
A reconstruction of the optimal second conjunct with the method
majorityCumu is:
#tien# stemden #tegen#
with violations: []
```

Sentence 6

Ze	**willen**	**de ronkende vliegtuigen**	**niet**	**en ook**	**de bommen**
They	***did***	***the roaring airplanes***	***not***	***and also***	***the bombs***

niet.
not.

```
36 ?- analyze([[ze,subj],[willen,vh],[[de, ronkende,
vliegtuigen],do],[niet,advPhr],[[en, ook],coord],[[de,
bommen]],[niet]]).

The following interpretation:
[[[[de, bommen], 1, [subj, do, io]], [ze, 1, subj]], [[niet, 2,
[advPhr]], [[de, ronkende, vliegtuigen], 2, do]]]
#[de, bommen]# willen #niet# niet
violates these constraints: [ovPar, fsp, minDis, featPar]

The following interpretation:
[[[[de, bommen], 1, [subj, do, io]], [ze, 1, subj]], [[niet, 2,
[advPhr]], [niet, 3, advPhr]]]
#[de, bommen]# willen [de, ronkende, vliegtuigen] #niet#
violates these constraints: [ovPar, fsp, minDis, featPar]

The following interpretation:
[[[[de, bommen], 1, [subj, do, io]], [[de, ronkende, vliegtuigen], 2,
do]], [[niet, 2, [advPhr]], [ze, 1, subj]]]
#niet# willen #[de, bommen]# niet
violates these constraints: [ovPar, fsp, stay, same, minDis, featPar]

The following interpretation:
[[[[de, bommen], 1, [subj, do, io]], [[de, ronkende, vliegtuigen], 2,
do]], [[niet, 2, [advPhr]], [niet, 3, advPhr]]]
ze willen #[de, bommen]# #niet#
violates these constraints: []

The following interpretation:
[[[[de, bommen], 1, [subj, do, io]], [niet, 3, advPhr]], [[niet, 2,
[advPhr]], [ze, 1, subj]]]
#niet# willen [de, ronkende, vliegtuigen] #[de, bommen]#
violates these constraints: [ovPar, fsp, same, minDis, featPar]

The following interpretation:
[[[[de, bommen], 1, [subj, do, io]], [niet, 3, advPhr]], [[niet, 2,
[advPhr]], [[de, ronkende, vliegtuigen], 2, do]]]
ze willen #niet# #[de, bommen]#
violates these constraints: [ovPar, same, minDis, featPar]


****STRICT DOMINATION****

A reconstruction of the optimal second conjunct with the method
nonCumu is:
ze willen #[de, bommen]# #niet#
with violations: []

****LOCAL RESTRICTED CUMULATIVITY****

The harmony value of the optimal second conjunct is 0.
```

```
A reconstruction of the optimal second conjunct with the method
localCumu is:
ze willen #[de, bommen]# #niet#
with violations: []

****GLOBAL RESTRICTED CUMULATIVITY****

A reconstruction of the optimal second conjunct with the method
globalResCumu is:
ze willen #[de, bommen]# #niet#
with violations: []

****MAJORITY CUMULATIVITY****

The candidate with the lowest number of violations has 0
violation(s).
A reconstruction of the optimal second conjunct with the method
majorityCumu is:
ze willen #[de, bommen]# #niet#
with violations: []
```

Sentence 7

Ook	**dient**	**volgens de raadscommissie**
Also	***should***	***according to the council committee***

het geneeskundig toezicht	**uitgebreid**	**te worden**
the medical supervision	***extended***	***to be***

en	**een commissie uit de ouders**	**samengesteld**	**te worden.**
and	***a committee of parents***	***put together***	***to be.***

```
analyze([[ook,advPhr],[dient,vh],[[volgens, de,
raadscommissie],pp],[[het, geneeskundig,
toezicht],subj],[uitgebreid,vmain],[[te,
worden],vaux],[en,coord],[[een, commissie, uit, de,
ouders]],[samengesteld],[[te, worden]]]).


The following interpretation:
[[[[een, commissie, uit, de, ouders], 1, [subj, do, io]], [[het,
geneeskundig, toezicht], 3, subj]], [[samengesteld, 2, [vmain]],
[uitgebreid, 4, vmain]], [[[te, worden], 3, [vaux]], [[te, worden],
5, vaux]]]
ook dient [volgens, de, raadscommissie] #[een, commissie, uit, de,
ouders]# #samengesteld# #[te, worden]#
violates these constraints: []

The following interpretation:
[[[[een, commissie, uit, de, ouders], 1, [subj, do, io]], [[het,
geneeskundig, toezicht], 3, subj]], [[samengesteld, 2, [vmain]],
[[te, worden], 5, vaux]], [[[te, worden], 3, [vaux]], [ook, 1,
advPhr]]]
#[te, worden]# dient [volgens, de, raadscommissie] #[een, commissie,
uit, de, ouders]# uitgebreid #samengesteld#
violates these constraints: [ovPar, same, minDis, featPar]

The following interpretation:
```

```
[[[[een, commissie, uit, de, ouders], 1, [subj, do, io]], [[het,
geneeskundig, toezicht], 3, subj]], [[samengesteld, 2, [vmain]],
[[te, worden], 5, vaux]], [[[te, worden], 3, [vaux]], [[volgens, de,
raadscommissie], 2, pp]]]
ook dient #[te, worden]# #[een, commissie, uit, de, ouders]#
uitgebreid #samengesteld#
violates these constraints: [ovPar, same, minDis, featPar]

The following interpretation:
[[[[een, commissie, uit, de, ouders], 1, [subj, do, io]], [[het,
geneeskundig, toezicht], 3, subj]], [[samengesteld, 2, [vmain]],
[[te, worden], 5, vaux]], [[[te, worden], 3, [vaux]], [uitgebreid, 4,
vmain]]]
ook dient [volgens, de, raadscommissie] #[een, commissie, uit, de,
ouders]# #[te, worden]# #samengesteld#
violates these constraints: [ovPar, same, minDis, featPar]

The following interpretation:
[[[[een, commissie, uit, de, ouders], 1, [subj, do, io]],
[uitgebreid, 4, vmain]], [[samengesteld, 2, [vmain]], [ook, 1,
advPhr]], [[[te, worden], 3, [vaux]], [[volgens, de, raadscommissie],
2, pp]]]
#samengesteld# dient #[te, worden]# [het, geneeskundig, toezicht]
#[een, commissie, uit, de, ouders]# [te, worden]
violates these constraints: [ovPar, same, minDis, featPar]

The following interpretation:
[[[[een, commissie, uit, de, ouders], 1, [subj, do, io]],
[uitgebreid, 4, vmain]], [[samengesteld, 2, [vmain]], [ook, 1,
advPhr]], [[[te, worden], 3, [vaux]], [[het, geneeskundig, toezicht],
3, subj]]]
#samengesteld# dient [volgens, de, raadscommissie] #[te, worden]#
#[een, commissie, uit, de, ouders]# [te, worden]
violates these constraints: [ovPar, same, minDis, featPar]

The following interpretation:
[[[[een, commissie, uit, de, ouders], 1, [subj, do, io]],
[uitgebreid, 4, vmain]], [[samengesteld, 2, [vmain]], [ook, 1,
advPhr]], [[[te, worden], 3, [vaux]], [[te, worden], 5, vaux]]]
#samengesteld# dient [volgens, de, raadscommissie] [het,
geneeskundig, toezicht] #[een, commissie, uit, de, ouders]# #[te,
worden]#
violates these constraints: [ovPar, same, minDis, featPar]

The following interpretation:
[[[[een, commissie, uit, de, ouders], 1, [subj, do, io]],
[uitgebreid, 4, vmain]], [[samengesteld, 2, [vmain]], [[volgens, de,
raadscommissie], 2, pp]], [[[te, worden], 3, [vaux]], [ook, 1,
advPhr]]]
#[te, worden]# dient #samengesteld# [het, geneeskundig, toezicht]
#[een, commissie, uit, de, ouders]# [te, worden]
violates these constraints: [ovPar, same, minDis, featPar]

The following interpretation:
[[[[een, commissie, uit, de, ouders], 1, [subj, do, io]],
[uitgebreid, 4, vmain]], [[samengesteld, 2, [vmain]], [[volgens, de,
raadscommissie], 2, pp]], [[[te, worden], 3, [vaux]], [[het,
geneeskundig, toezicht], 3, subj]]]
ook dient #samengesteld# #[te, worden]# #[een, commissie, uit, de,
ouders]# [te, worden]
violates these constraints: [ovPar, same, minDis, featPar]
```

The following interpretation:
[[[[een, commissie, uit, de, ouders], 1, [subj, do, io]], [uitgebreid, 4, vmain]], [[samengesteld, 2, [vmain]], [[volgens, de, raadscommissie], 2, pp]], [[[te, worden], 3, [vaux]], [[te, worden], 5, vaux]]]
ook dient #samengesteld# [het, geneeskundig, toezicht] #[een, commissie, uit, de, ouders]# #[te, worden]#
violates these constraints: [ovPar, same, minDis, featPar]

The following interpretation:
[[[[een, commissie, uit, de, ouders], 1, [subj, do, io]], [uitgebreid, 4, vmain]], [[samengesteld, 2, [vmain]], [[het, geneeskundig, toezicht], 3, subj]], [[[te, worden], 3, [vaux]], [ook, 1, advPhr]]]
#[te, worden]# dient [volgens, de, raadscommissie] #samengesteld# #[een, commissie, uit, de, ouders]# [te, worden]
violates these constraints: [ovPar, same, minDis, featPar]

The following interpretation:
[[[[een, commissie, uit, de, ouders], 1, [subj, do, io]], [uitgebreid, 4, vmain]], [[samengesteld, 2, [vmain]], [[het, geneeskundig, toezicht], 3, subj]], [[[te, worden], 3, [vaux]], [[volgens, de, raadscommissie], 2, pp]]]
ook dient #[te, worden]# #samengesteld# #[een, commissie, uit, de, ouders]# [te, worden]
violates these constraints: [ovPar, same, minDis, featPar]

The following interpretation:
[[[[een, commissie, uit, de, ouders], 1, [subj, do, io]], [uitgebreid, 4, vmain]], [[samengesteld, 2, [vmain]], [[het, geneeskundig, toezicht], 3, subj]], [[[te, worden], 3, [vaux]], [[te, worden], 5, vaux]]]
ook dient [volgens, de, raadscommissie] #samengesteld# #[een, commissie, uit, de, ouders]# #[te, worden]#
violates these constraints: [ovPar, same, minDis, featPar]

The following interpretation:
[[[[een, commissie, uit, de, ouders], 1, [subj, do, io]], [uitgebreid, 4, vmain]], [[samengesteld, 2, [vmain]], [[te, worden], 5, vaux]], [[[te, worden], 3, [vaux]], [ook, 1, advPhr]]]
#[te, worden]# dient [volgens, de, raadscommissie] [het, geneeskundig, toezicht] #[een, commissie, uit, de, ouders]# #samengesteld#
violates these constraints: [ovPar, same, minDis, featPar]

The following interpretation:
[[[[een, commissie, uit, de, ouders], 1, [subj, do, io]], [uitgebreid, 4, vmain]], [[samengesteld, 2, [vmain]], [[te, worden], 5, vaux]], [[[te, worden], 3, [vaux]], [[volgens, de, raadscommissie], 2, pp]]]
ook dient #[te, worden]# [het, geneeskundig, toezicht] #[een, commissie, uit, de, ouders]# #samengesteld#
violates these constraints: [ovPar, same, minDis, featPar]

The following interpretation:
[[[[een, commissie, uit, de, ouders], 1, [subj, do, io]], [uitgebreid, 4, vmain]], [[samengesteld, 2, [vmain]], [[te, worden], 5, vaux]], [[[te, worden], 3, [vaux]], [[het, geneeskundig, toezicht], 3, subj]]]

ook dient [volgens, de, raadscommissie] #[te, worden]# #[een, commissie, uit, de, ouders]# #samengesteld#
violates these constraints: [ovPar, same, minDis, featPar]

The following interpretation:
[[[[een, commissie, uit, de, ouders], 1, [subj, do, io]], [[te, worden], 5, vaux]], [[samengesteld, 2, [vmain]], [ook, 1, advPhr]], [[[te, worden], 3, [vaux]], [[volgens, de, raadscommissie], 2, pp]]]
#samengesteld# dient #[te, worden]# [het, geneeskundig, toezicht] uitgebreid #[een, commissie, uit, de, ouders]#
violates these constraints: [ovPar, same, minDis, featPar]

The following interpretation:
[[[[een, commissie, uit, de, ouders], 1, [subj, do, io]], [[te, worden], 5, vaux]], [[samengesteld, 2, [vmain]], [ook, 1, advPhr]], [[[te, worden], 3, [vaux]], [[het, geneeskundig, toezicht], 3, subj]]]
#samengesteld# dient [volgens, de, raadscommissie] #[te, worden]# uitgebreid #[een, commissie, uit, de, ouders]#
violates these constraints: [ovPar, same, minDis, featPar]

The following interpretation:
[[[[een, commissie, uit, de, ouders], 1, [subj, do, io]], [[te, worden], 5, vaux]], [[samengesteld, 2, [vmain]], [ook, 1, advPhr]], [[[te, worden], 3, [vaux]], [uitgebreid, 4, vmain]]]
#samengesteld# dient [volgens, de, raadscommissie] [het, geneeskundig, toezicht] #[te, worden]# #[een, commissie, uit, de, ouders]#
violates these constraints: [ovPar, same, minDis, featPar]

The following interpretation:
[[[[een, commissie, uit, de, ouders], 1, [subj, do, io]], [[te, worden], 5, vaux]], [[samengesteld, 2, [vmain]], [[volgens, de, raadscommissie], 2, pp]], [[[te, worden], 3, [vaux]], [ook, 1, advPhr]]]
#[te, worden]# dient #samengesteld# [het, geneeskundig, toezicht] uitgebreid #[een, commissie, uit, de, ouders]#
violates these constraints: [ovPar, same, minDis, featPar]

The following interpretation:
[[[[een, commissie, uit, de, ouders], 1, [subj, do, io]], [[te, worden], 5, vaux]], [[samengesteld, 2, [vmain]], [[volgens, de, raadscommissie], 2, pp]], [[[te, worden], 3, [vaux]], [[het, geneeskundig, toezicht], 3, subj]]]
ook dient #samengesteld# #[te, worden]# uitgebreid #[een, commissie, uit, de, ouders]#
violates these constraints: [ovPar, same, minDis, featPar]

The following interpretation:
[[[[een, commissie, uit, de, ouders], 1, [subj, do, io]], [[te, worden], 5, vaux]], [[samengesteld, 2, [vmain]], [[volgens, de, raadscommissie], 2, pp]], [[[te, worden], 3, [vaux]], [uitgebreid, 4, vmain]]]
ook dient #samengesteld# [het, geneeskundig, toezicht] #[te, worden]# #[een, commissie, uit, de, ouders]#
violates these constraints: [ovPar, same, minDis, featPar]

The following interpretation:
[[[[een, commissie, uit, de, ouders], 1, [subj, do, io]], [[te, worden], 5, vaux]], [[samengesteld, 2, [vmain]], [[het, geneeskundig, toezicht], 3, subj]], [[[te, worden], 3, [vaux]], [ook, 1, advPhr]]]

```
#[te, worden]# dient [volgens, de, raadscommissie] #samengesteld#
uitgebreid #[een, commissie, uit, de, ouders]#
violates these constraints: [ovPar, same, minDis, featPar]

The following interpretation:
[[[[een, commissie, uit, de, ouders], 1, [subj, do, io]], [[te,
worden], 5, vaux]], [[samengesteld, 2, [vmain]], [[het, geneeskundig,
toezicht], 3, subj]], [[[te, worden], 3, [vaux]], [[volgens, de,
raadscommissie], 2, pp]]]
ook dient #[te, worden]# #samengesteld# uitgebreid #[een, commissie,
uit, de, ouders]#
violates these constraints: [ovPar, same, minDis, featPar]

The following interpretation:
[[[[een, commissie, uit, de, ouders], 1, [subj, do, io]], [[te,
worden], 5, vaux]], [[samengesteld, 2, [vmain]], [[het, geneeskundig,
toezicht], 3, subj]], [[[te, worden], 3, [vaux]], [uitgebreid, 4,
vmain]]]
ook dient [volgens, de, raadscommissie] #samengesteld# #[te, worden]#
#[een, commissie, uit, de, ouders]#
violates these constraints: [ovPar, same, minDis, featPar]

The following interpretation:
[[[[een, commissie, uit, de, ouders], 1, [subj, do, io]], [[te,
worden], 5, vaux]], [[samengesteld, 2, [vmain]], [uitgebreid, 4,
vmain]], [[[te, worden], 3, [vaux]], [ook, 1, advPhr]]]
#[te, worden]# dient [volgens, de, raadscommissie] [het,
geneeskundig, toezicht] #samengesteld# #[een, commissie, uit, de,
ouders]#
violates these constraints: [ovPar, same, minDis, featPar]

The following interpretation:
[[[[een, commissie, uit, de, ouders], 1, [subj, do, io]], [[te,
worden], 5, vaux]], [[samengesteld, 2, [vmain]], [uitgebreid, 4,
vmain]], [[[te, worden], 3, [vaux]], [[volgens, de, raadscommissie],
2, pp]]]
ook dient #[te, worden]# [het, geneeskundig, toezicht] #samengesteld#
#[een, commissie, uit, de, ouders]#
violates these constraints: [ovPar, same, minDis]

The following interpretation:
[[[[een, commissie, uit, de, ouders], 1, [subj, do, io]], [[te,
worden], 5, vaux]], [[samengesteld, 2, [vmain]], [uitgebreid, 4,
vmain]], [[[te, worden], 3, [vaux]], [[het, geneeskundig, toezicht],
3, subj]]]
ook dient [volgens, de, raadscommissie] #[te, worden]# #samengesteld#
#[een, commissie, uit, de, ouders]#
violates these constraints: [ovPar, same, minDis]


****STRICT DOMINATION****

A reconstruction of the optimal second conjunct with the method
nonCumu is:
ook dient [volgens, de, raadscommissie] #[een, commissie, uit, de,
ouders]# #samengesteld# #[te, worden]#
with violations: []

****LOCAL RESTRICTED CUMULATIVITY****

The harmony value of the optimal second conjunct is 0.
```

```
A reconstruction of the optimal second conjunct with the method
localCumu is:
ook dient [volgens, de, raadscommissie] #[een, commissie, uit, de,
ouders]# #samengesteld# #[te, worden]#
with violations: []

****GLOBAL RESTRICTED CUMULATIVITY****

A reconstruction of the optimal second conjunct with the method
globalResCumu is:
ook dient [volgens, de, raadscommissie] #[een, commissie, uit, de,
ouders]# #samengesteld# #[te, worden]#
with violations: []

****MAJORITY CUMULATIVITY****

The candidate with the lowest number of violations has 0
violation(s).
A reconstruction of the optimal second conjunct with the method
majorityCumu is:
ook dient [volgens, de, raadscommissie] #[een, commissie, uit, de,
ouders]# #samengesteld# #[te, worden]#
with violations: []
```

Sentence 8

Groep 1	**trok**	**de arm**	**na vijftien minuten**
Group 1	***pulled***	***the arm***	***after fifteen minutes***

uit de testkamer//,	**groep 2**	**na dertig minuten,**
from the test room	***group 2***	***after thirty minutes***

groep 3	**na 45 minuten// en**	**groep 4**	**na een uur.**
group 3	***after 45 minutes***	***and group 4***	***after an hour.***

```
28 ?- analyze([[groep1,subj],[trok,vh],[[de, arm],do],[[na, vijftien,
minuten],advPhrTemp],[[uit, de,
testkamer],advPhr],[en,coord],[groep4],[[na, een, uur]]]).

The following interpretation:
[[[groep4, 1, [subj, do, io]], [groep1, 1, subj]], [[[na, een, uur],
2, [advPhrTemp]], [[de, arm], 2, do]]]
#groep4# trok #[na, een, uur]# [na, vijftien, minuten] [uit, de,
testkamer]
violates these constraints: [ovPar, minDis, featPar]

The following interpretation:
[[[groep4, 1, [subj, do, io]], [groep1, 1, subj]], [[[na, een, uur],
2, [advPhrTemp]], [[na, vijftien, minuten], 3, advPhrTemp]]]
#groep4# trok [de, arm] #[na, een, uur]# [uit, de, testkamer]
violates these constraints: [minDis]

The following interpretation:
[[[groep4, 1, [subj, do, io]], [groep1, 1, subj]], [[[na, een, uur],
2, [advPhrTemp]], [[uit, de, testkamer], 4, advPhr]]]
#groep4# trok [de, arm] [na, vijftien, minuten] #[na, een, uur]#
violates these constraints: [ovPar, minDis, featPar]
```

```
The following interpretation:
[[[groep4, 1, [subj, do, io]], [[de, arm], 2, do]], [[[na, een, uur],
2, [advPhrTemp]], [groep1, 1, subj]]]
#[na, een, uur]# trok #groep4# [na, vijftien, minuten] [uit, de,
testkamer]
violates these constraints: [ovPar, stay, same, minDis, featPar]

The following interpretation:
[[[groep4, 1, [subj, do, io]], [[de, arm], 2, do]], [[[na, een, uur],
2, [advPhrTemp]], [[na, vijftien, minuten], 3, advPhrTemp]]]
groep1 trok #groep4# #[na, een, uur]# [uit, de, testkamer]
violates these constraints: [featPar]

The following interpretation:
[[[groep4, 1, [subj, do, io]], [[de, arm], 2, do]], [[[na, een, uur],
2, [advPhrTemp]], [[uit, de, testkamer], 4, advPhr]]]
groep1 trok #groep4# [na, vijftien, minuten] #[na, een, uur]#
violates these constraints: [ovPar, minDis, featPar]

The following interpretation:
[[[groep4, 1, [subj, do, io]], [[na, vijftien, minuten], 3,
advPhrTemp]], [[[na, een, uur], 2, [advPhrTemp]], [groep1, 1, subj]]]
#[na, een, uur]# trok [de, arm] #groep4# [uit, de, testkamer]
violates these constraints: [ovPar, same, minDis, featPar]

The following interpretation:
[[[groep4, 1, [subj, do, io]], [[na, vijftien, minuten], 3,
advPhrTemp]], [[[na, een, uur], 2, [advPhrTemp]], [[de, arm], 2,
do]]]
groep1 trok #[na, een, uur]# #groep4# [uit, de, testkamer]
violates these constraints: [ovPar, same, minDis, featPar]

The following interpretation:
[[[groep4, 1, [subj, do, io]], [[na, vijftien, minuten], 3,
advPhrTemp]], [[[na, een, uur], 2, [advPhrTemp]], [[uit, de,
testkamer], 4, advPhr]]]
groep1 trok [de, arm] #groep4# #[na, een, uur]#
violates these constraints: [ovPar, minDis, featPar]

The following interpretation:
[[[groep4, 1, [subj, do, io]], [[uit, de, testkamer], 4, advPhr]],
[[[na, een, uur], 2, [advPhrTemp]], [groep1, 1, subj]]]
#[na, een, uur]# trok [de, arm] [na, vijftien, minuten] #groep4#
violates these constraints: [ovPar, same, minDis, featPar]

The following interpretation:
[[[groep4, 1, [subj, do, io]], [[uit, de, testkamer], 4, advPhr]],
[[[na, een, uur], 2, [advPhrTemp]], [[de, arm], 2, do]]]
groep1 trok #[na, een, uur]# [na, vijftien, minuten] #groep4#
violates these constraints: [ovPar, same, minDis, featPar]

The following interpretation:
[[[groep4, 1, [subj, do, io]], [[uit, de, testkamer], 4, advPhr]],
[[[na, een, uur], 2, [advPhrTemp]], [[na, vijftien, minuten], 3,
advPhrTemp]]]
groep1 trok [de, arm] #[na, een, uur]# #groep4#
violates these constraints: [ovPar, same, minDis, featPar]


****STRICT DOMINATION****
```

```
A reconstruction of the optimal second conjunct with the method
nonCumu is:
groep1 trok #groep4# #[na, een, uur]# [uit, de, testkamer]
with violations: [featPar]

****LOCAL RESTRICTED CUMULATIVITY****

The harmony value of the optimal second conjunct is 1e-006.
A reconstruction of the optimal second conjunct with the method
localCumu is:
groep1 trok #groep4# #[na, een, uur]# [uit, de, testkamer]
with violations: [featPar]

****GLOBAL RESTRICTED CUMULATIVITY****

A reconstruction of the optimal second conjunct with the method
globalResCumu is:
groep1 trok #groep4# #[na, een, uur]# [uit, de, testkamer]
with violations: [featPar]

****MAJORITY CUMULATIVITY****

There is more than one candidate with the lowest number of
violations.
Therefore, strict domination will determine the optimal candidate.
The candidate with the lowest number of violations has 1
violation(s).
A reconstruction of the optimal second conjunct with the method
majorityCumu is:
groep1 trok #groep4# #[na, een, uur]# [uit, de, testkamer]
with violations: [featPar]
```

Sentence 9

Wij	**hebben**	**nu**	**twee keer**	**goed**	**gespeeld,**	**maar ook**
We	***have***	***now***	***twice***	***well***	***played***	***but also***

twee keer	**verloren.**
twice	***lost.***

```
27 ?- analyze([[wij,subj],[hebben,vh],[nu,advPhrTemp],[[twee,
keer],advPhrNumber], [[goed,gespeeld],vmain],[[maar,
ook],coord],[[twee, keer]],[verloren]]).

The following interpretation:
[[[[twee, keer], 1, [advPhrNumber, advPhrTemp, subj, do, io]], [wij,
1, subj]], [[verloren, 2, [vmain]], [nu, 2, advPhrTemp]]]
#[twee, keer]# hebben #verloren# [twee, keer] [goed, gespeeld]
violates these constraints: [ovPar, fsp, minDis, featPar]

The following interpretation:
[[[[twee, keer], 1, [advPhrNumber, advPhrTemp, subj, do, io]], [wij,
1, subj]], [[verloren, 2, [vmain]], [[twee, keer], 3, advPhrNumber]]]
#[twee, keer]# hebben nu #verloren# [goed, gespeeld]
violates these constraints: [ovPar, fsp, minDis, featPar]

The following interpretation:
```

```
[[[[twee, keer], 1, [advPhrNumber, advPhrTemp, subj, do, io]], [wij,
1, subj]], [[verloren, 2, [vmain]], [[goed, gespeeld], 4, vmain]]]
#[twee, keer]# hebben nu [twee, keer] #verloren#
violates these constraints: [ovPar, fsp, minDis, featPar]

The following interpretation:
[[[[twee, keer], 1, [advPhrNumber, advPhrTemp, subj, do, io]], [nu,
2, advPhrTemp]], [[verloren, 2, [vmain]], [wij, 1, subj]]]
#verloren# hebben #[twee, keer]# [twee, keer] [goed, gespeeld]
violates these constraints: [ovPar, fsp, same, minDis, featPar]

The following interpretation:
[[[[twee, keer], 1, [advPhrNumber, advPhrTemp, subj, do, io]], [nu,
2, advPhrTemp]], [[verloren, 2, [vmain]], [[twee, keer], 3,
advPhrNumber]]]
wij hebben #[twee, keer]# #verloren# [goed, gespeeld]
violates these constraints: [ovPar, minDis, featPar]

The following interpretation:
[[[[twee, keer], 1, [advPhrNumber, advPhrTemp, subj, do, io]], [nu,
2, advPhrTemp]], [[verloren, 2, [vmain]], [[goed, gespeeld], 4,
vmain]]]
wij hebben #[twee, keer]# [twee, keer] #verloren#
violates these constraints: [minDis, featPar]

The following interpretation:
[[[[twee, keer], 1, [advPhrNumber, advPhrTemp, subj, do, io]],
[[twee, keer], 3, advPhrNumber]], [[verloren, 2, [vmain]], [wij, 1,
subj]]]
#verloren# hebben nu #[twee, keer]# [goed, gespeeld]
violates these constraints: [ovPar, fsp, same, minDis]

The following interpretation:
[[[[twee, keer], 1, [advPhrNumber, advPhrTemp, subj, do, io]],
[[twee, keer], 3, advPhrNumber]], [[verloren, 2, [vmain]], [nu, 2,
advPhrTemp]]]
wij hebben #verloren# #[twee, keer]# [goed, gespeeld]
violates these constraints: [ovPar, same, minDis]

The following interpretation:
[[[[twee, keer], 1, [advPhrNumber, advPhrTemp, subj, do, io]],
[[twee, keer], 3, advPhrNumber]], [[verloren, 2, [vmain]], [[goed,
gespeeld], 4, vmain]]]
wij hebben nu #[twee, keer]# #verloren#
violates these constraints: []

The following interpretation:
[[[[twee, keer], 1, [advPhrNumber, advPhrTemp, subj, do, io]],
[[goed, gespeeld], 4, vmain]], [[verloren, 2, [vmain]], [wij, 1,
subj]]]
#verloren# hebben nu [twee, keer] #[twee, keer]#
violates these constraints: [ovPar, fsp, same, minDis]

The following interpretation:
[[[[twee, keer], 1, [advPhrNumber, advPhrTemp, subj, do, io]],
[[goed, gespeeld], 4, vmain]], [[verloren, 2, [vmain]], [nu, 2,
advPhrTemp]]]
wij hebben #verloren# [twee, keer] #[twee, keer]#
violates these constraints: [ovPar, same, minDis]

The following interpretation:
```

```
[[[[twee, keer], 1, [advPhrNumber, advPhrTemp, subj, do, io]],
[[goed, gespeeld], 4, vmain]], [[verloren, 2, [vmain]], [[twee,
keer], 3, advPhrNumber]]]
wij hebben nu #verloren# #[twee, keer]#
violates these constraints: [ovPar, same, minDis]


****STRICT DOMINATION****

A reconstruction of the optimal second conjunct with the method
nonCumu is:
wij hebben nu #[twee, keer]# #verloren#
with violations: []

****LOCAL RESTRICTED CUMULATIVITY****

The harmony value of the optimal second conjunct is 0.
A reconstruction of the optimal second conjunct with the method
localCumu is:
wij hebben nu #[twee, keer]# #verloren#
with violations: []

****GLOBAL RESTRICTED CUMULATIVITY****

A reconstruction of the optimal second conjunct with the method
globalResCumu is:
wij hebben nu #[twee, keer]# #verloren#
with violations: []

****MAJORITY CUMULATIVITY****

The candidate with the lowest number of violations has 0
violation(s).
A reconstruction of the optimal second conjunct with the method
majorityCumu is:
wij hebben nu #[twee, keer]# #verloren#
with violations: []
```

Sentence 10

De TROS **kwam** **met zes procent** **op de derde plaats**
The TROS ***came*** ***with six percent*** ***on the third place***

//, de NCRV **met vijf procent** **op de vierde//**
the NCRV ***with five percent*** ***on the fourth***

en de AVRO **- met drie procent -** **op de vijfde plaats.**
and the AVRO ***-with three percent-*** ***on the fifth place***

```
82 ?- analyze([[[de, tros],subj],[kwam,vh],[[met, zes,
procent],advPhr],[[op, de, derde, plaats],advPhr],[en,coord],[[de,
avro]],[[met, drie, procent]],[[op, de, vijfde, plaats]]]).

The following interpretation:
[[[[de, avro], 1, [subj, do, io]], [[de, tros], 1, subj]], [[[met,
drie, procent], 2, [advPhr]], [[met, zes, procent], 2, advPhr]],
[[[op, de, vijfde, plaats], 3, [advPhr]], [[op, de, derde, plaats],
3, advPhr]]]
#[de, avro]# kwam #[met, drie, procent]# #[op, de, vijfde, plaats]#
```

violates these constraints: []

The following interpretation:
[[[[de, avro], 1, [subj, do, io]], [[de, tros], 1, subj]], [[[met, drie, procent], 2, [advPhr]], [[op, de, derde, plaats], 3, advPhr]], [[[op, de, vijfde, plaats], 3, [advPhr]], [[met, zes, procent], 2, advPhr]]]
#[de, avro]# kwam #[op, de, vijfde, plaats]# #[met, drie, procent]#
violates these constraints: [same]

The following interpretation:
[[[[de, avro], 1, [subj, do, io]], [[met, zes, procent], 2, advPhr]], [[[met, drie, procent], 2, [advPhr]], [[de, tros], 1, subj]], [[[op, de, vijfde, plaats], 3, [advPhr]], [[op, de, derde, plaats], 3, advPhr]]]
#[met, drie, procent]# kwam #[de, avro]# #[op, de, vijfde, plaats]#
violates these constraints: [ovPar, same, minDis, featPar]

The following interpretation:
[[[[de, avro], 1, [subj, do, io]], [[met, zes, procent], 2, advPhr]], [[[met, drie, procent], 2, [advPhr]], [[op, de, derde, plaats], 3, advPhr]], [[[op, de, vijfde, plaats], 3, [advPhr]], [[de, tros], 1, subj]]]
#[op, de, vijfde, plaats]# kwam #[de, avro]# #[met, drie, procent]#
violates these constraints: [ovPar, same, minDis, featPar]

The following interpretation:
[[[[de, avro], 1, [subj, do, io]], [[op, de, derde, plaats], 3, advPhr]], [[[met, drie, procent], 2, [advPhr]], [[de, tros], 1, subj]], [[[op, de, vijfde, plaats], 3, [advPhr]], [[met, zes, procent], 2, advPhr]]]
#[met, drie, procent]# kwam #[op, de, vijfde, plaats]# #[de, avro]#
violates these constraints: [ovPar, same, minDis, featPar]

The following interpretation:
[[[[de, avro], 1, [subj, do, io]], [[op, de, derde, plaats], 3, advPhr]], [[[met, drie, procent], 2, [advPhr]], [[met, zes, procent], 2, advPhr]], [[[op, de, vijfde, plaats], 3, [advPhr]], [[de, tros], 1, subj]]]
#[op, de, vijfde, plaats]# kwam #[met, drie, procent]# #[de, avro]#
violates these constraints: [ovPar, same, minDis, featPar]

****STRICT DOMINATION****

A reconstruction of the optimal second conjunct with the method nonCumu is:
#[de, avro]# kwam #[met, drie, procent]# #[op, de, vijfde, plaats]#
with violations: []

****LOCAL RESTRICTED CUMULATIVITY****

The harmony value of the optimal second conjunct is 0.
A reconstruction of the optimal second conjunct with the method localCumu is:
#[de, avro]# kwam #[met, drie, procent]# #[op, de, vijfde, plaats]#
with violations: []

****GLOBAL RESTRICTED CUMULATIVITY****

```
A reconstruction of the optimal second conjunct with the method
globalResCumu is:
#[de, avro]# kwam #[met, drie, procent]# #[op, de, vijfde, plaats]#
with violations: []

****MAJORITY CUMULATIVITY****
The candidate with the lowest number of violations has 0
violation(s).
A reconstruction of the optimal second conjunct with the method
majorityCumu is:
#[de, avro]# kwam #[met, drie, procent]# #[op, de, vijfde, plaats]#
with violations: []
```

Appendix E: Experiment

Experiment zinsinterpretatie

In dit experiment wordt je van een aantal zinnen gevraagd hoe je ze zou interpreteren. Je krijgt daarvoor een aantal opties, waarvan je er één mag kiezen. Als jouw interpretatie niet bij (a) of (b) zit, kun je zelf een interpretatie invullen bij (c). Na elke zin wordt je ook gevraagd hoe moeilijk de zin was om te interpreteren; je kunt de zin een cijfer geven tussen 1 (erg makkelijk) en 5 (erg moeilijk). Verder wordt gevraagd of jij de zin op meerdere manieren kunt interpreteren. Omcirkel bij de vragen het antwoord van je keuze.

Laat je bij het evalueren van de moeilijkheid van de zinnen niet afleiden door de - soms onlogische - keuzemogelijkheden. Kijk gewoon of je de zin meteen (op welke manier dan ook) begrijpt na het lezen.

Dit experiment is ontworpen om mijn computermodel te testen. Het gaat dus niet om goede of foute antwoorden, maar om hoe mensen de zinnen beoordelen en of de computer dat goed kan simuleren. Het experiment duurt ongeveer 20 minuten.

Alvast bedankt voor je deelname!

Sentence interpretation experiment

In this experiment, a number of sentences will be presented to you and it will be asked how you would interpret them. A number of options will be given, of which you can pick one. If your interpretation is not given under (a) or (b), you can write down your own interpretation under (c). After every sentence, it will be asked how difficult it was to interpret that sentence; you can rate the sentence 1 (very easy) to 5 (very difficult). Then it is asked whether you can interpret the sentence in more ways. Circle the answer of your choice.

Do not let the - sometimes illogical - answer options distract you. Just evaluate whether you understand the sentence (in whatever way) immediately after reading.

This experiment was designed to test my computer model. It is nog about right or wrong anwers, but about how people judge the sentences and whether the computer is capable of simulating that. The experiment takes about 20 minutes.

Thanks in advance for your participation!

Test sentences in the order presented to subjects

1) Je bent vijf dagen van de week gevaarlijk en een of twee dagen ongevaarlijk.
2) Will plukt een roos en een tulp Grace.
3) Jack wacht met Grace op de ober en Will met de butler.
4) Will verzoekt vandaag om Grace te negeren en Jack om Karen te ontlopen.
5) Jack eet een snoepje en een koekje Karen.
6) Wij hebben nu twee keer goed gespeeld , maar ook twee keer verloren.
7) Jack bezocht het kantoor in de vakantie en Karen in het weekend.
8) Rosario speelde backgammon met Karen en Grace ook.
9) Will bekritiseerde het eten tijdens het diner en Karen tijdens het feest.
10) Voor de Tarzan-liefhebbers is Johnnie vrijdagavond op de televisie te zien in het NOS-programma Premiere en woensdag 1 juli in de televisiehoofdfilm "Tarzan en de Amazones".
11) Will gaat met Karen naar oma en Jack met opa.
12) Hoek's machines daalde 8 en Centrale Suiker 6 punten.
13) Jack verplicht uitdrukkelijk om Will te groeten en Grace om Stan te verwelkomen.
14) Grace bakt cakes voor toeristen en Karen voor haar familie.
15) De butler belooft Will te bellen en de ober te faxen.
16) Grace verzoekt nadrukkelijk om Rob uit te nodigen en Karen om Will mee te vragen.
17) Intussen zijn de schepen veel groter en sneller geworden en de charterperioden langer.
18) Jack beledigt de agent en grieft de receptioniste.
19) Will vraagt hem om Grace te negeren en Jack om Karen te ontlopen.
20) Stan verbaasde de jury met zijn talent en Jack met zijn muzikaliteit.
21) Stan haalt koffie voor Will en Jack ook.
22) De conducteur belooft Grace te stoppen en de machinist door te gaan.
23) Grace ging winkelen met Jack en Will ook.
24) Stan leest een tijdschrift en een boek Ellen.
25) Will complimenteerde de gastheren met de decoratie en Grace met het eten.
26) Stan vraagt haar om Ellen te halen en Will om Grace te brengen.
27) Karen ontmoet Ellen en mist Rob.
28) Rob verliest niet graag van Grace en Ellen ook niet.
29) De koning belooft het volk te blijven en de prins te vertrekken.
30) De TROS kwam met zes procent op de derde plaats en de AVRO - met drie procent - op de vijfde plaats.
31) Negentig parlementsleden stemden voor en tien tegen.
32) Jack gaf koekjes aan de kinderen en Will aan de buren.
33) Stan vertelt grapjes aan volwassenen en Rob aan kinderen.
34) Stan eet koekjes en drinkt bier.
35) Ze willen de ronkende vliegtuigen niet en ook de bommen niet.
36) Grace ontwerpt huizen en richt kantoren in.
37) Ook dient volgens de raadscommissie het geneeskundig toezicht uitgebreid te worden en een commissie uit de ouders samengesteld te worden.
38) Groep 1 trok de arm na vijftien minuten uit de testkamer en groep 4 na een uur.
39) Karen lachte met de tuinman om Will en de glazenwasser met Ellen.
40) Jack verzoekt hem om Will te slaan en Stan om Rob te meppen.