# Case Constraints and Empty Categories in Optimality Theory Parsing*

Ron Artstein

April 28, 2000

**Abstract**

Double object questions are harder to process than the corresponding questions with prepositional objects (Boland 1997). When given the choice between the two, a parser prefers to wait and not associate a filler with a gap in a double object construction, contrary to the prediction of the Active Filler Hypothesis (Frazier 1987). I propose that the reason the parser holds back is that it wants to avoid a structure that has dative Case; this pressure is stronger than the pressure to associate a filler with a gap, which comes from the thematic requirements of elements in the sentence (and not from the cost of holding the filler in memory). The results are formulated in an extension to Stevenson and Smolensky's (1997) Optimality Theory parser, as tension between constraints that mark Case and argument structure.

## 1 Introduction

This paper extends the work of Stevenson and Smolensky (1997) on an Optimality Theory parsing mechanism. Stevenson and Smolensky developed a parser that dealt specifically with attachment ambiguities; these ambiguities arise when a constituent can be attached to more than one site in the parse tree.

*This paper was written in partial fulfillment of the requirements for a Cognitive Science Certificate at the Rutgers Center for Cognitive Science (RuCCS), under the supervision of Suzanne Stevenson; most of the research was done during the summer and fall of 1998 and spring of 1999. The results of this research have been presented at Rutgers University, the University of Maryland, the University of Pennsylvania, and at AMLaP-99 (Architectures and Mechanisms for Language Processing) at the University of Edinburgh; I thank the audiences for their comments and suggestions. I owe particular thanks to Julie Boland, Ted Gibson, Jane Grimshaw, Paul Smolensky, Susanne Stevenson, and Bruce Tesar. This research was supported by a grant from Johns Hopkins University #8804-27486, "Johns Hopkins University Learning Systems." (Principal investigator: Paul Smolensky; local supervisors: Alan Prince, Suzanne Stevenson; funding from NSF).

(1) John saw the man with the binoculars.

The PP *with the binoculars* in the above example can either modify the verb *saw* or the noun *man*; this leads to two syntactic representations, where the PP is attached to the VP (2a) or to the DP (2b).

(2) a. John saw [the man] [with the binoculars]

b. John saw [the man with the binoculars]

We say that sentence (1) is *globally* ambiguous, because the ambiguity persists all the way to the end of the sentence. Sentences may also have *local* ambiguities, which are resolved later on. The following sentence is unambiguous.

(3) Jill put the candy on the table.

The only possible interpretation is that the PP *on the table* modifies the verb *put* and is interpreted as a locative or goal argument. However, when the word *on* is read (or heard, if the utterance is spoken), it has two possible attachment sites: it can modify either the verb or the noun (as in the sentence *The candy on the table is sweet*). At the point of the word *on*, we do not know which is the correct attachment site, because we do not know how the sentence will continue (an attachment opposite from the one in (3) is required in the sentence *Jill put the candy on the table into her mouth*). We say the ambiguity is local because when we look at the sentence as a whole, the attachment site is clear.

The human sentence processing mechanism does not wait until the end of a sentence in order to resolve ambiguities; it makes the decisions at an earlier stage. Sometimes these decision turn out to be wrong, and the result is a "garden path" sentence which is difficult to process; such processing difficulties provide a clue as to what parsing decisions are made in the course of processing a sentence (I use the inverted question mark (¿) to show garden path sentences).

(4) ¿While Mary was mending the socks fell.

In the correct structure of the above sentence, which is grammatical, the adverbial clause ends with the verb *mending*, and the DP *the socks* is the subject of the main clause. This can be indicated in speech with a prosodic boundary, or in writing via a comma.

(5) While Mary was mending, the socks fell.

In the absence of such cues, the human parser tends to interpret the DP *the socks* as the object of the verb *mending*, and is at loss when the matrix verb *fell* appears.

A successful model of a human parser will predict the attachments that humans actually make, including the wrong ones.

The attachment ambiguities that Stevenson and Smolensky (1997) looked at motivated a specific set of ranked Optimality Theory constraints that make up the parser. I extend their work to deal with filler-gap ambiguities. These ambiguities arise when the parser identifies a displaced element ("filler"), and has to decide where to put a corresponding gap (trace).

(6) a. Which patient$_i$ did the nurse bring the doctor $e_i$?

b. Which patient$_i$ did the nurse bring $e_i$ the doctor?

(7) a. Who$_i$ did Fred tell Mary [$e_i$ left the country]?

b. Who$_i$ did Fred tell $e_i$ [Mary left the country]?

(Clifton and Frazier 1989)

The examples above show a global ambiguity—each sentence has two interpretations. Filler-gap ambiguities, just like attachment ambiguities, also have a local variant.

(8) Who did John see Mary with?

There is only one possible site for the trace of the interrogative DP *who*, namely the complement position of the PP *with*.

(9) Who$_i$ did John see Mary with $e_i$?

However, an alternative site is available on the way: after the word *see* is read, the parser could associate the filler (*who*) with the object position of the verb *see*. This turns out to be the wrong site for sentence (8) (but it would be the correct site for a sentence like *Who did John see with Mary?*).

The Active Filler Hypothesis (Frazier 1987) states that the when parser identifies a filler, it tries to find a site to associate it with as soon as possible (Crocker 1993 has an even stronger version of this strategy). This would entail that in sentence (8) the parser in fact does associate the filler with a gap in the object position of *see*; this analysis is then revised when the rest of the sentence turns out to be incompatible (a reanalysis that in this case does not lead to processing difficulty). While I agree that this is the case in this example, I claim that the active filler hypothesis is too strong, and that there are cases where the parser prefers to hold back and not put a gap even when it has an eligible site. Evidence I present in section 4 shows that this is dependent on grammatical Case: the parser will propose a gap when it results in accusative Case, but will hold back if the gap will force an argument to have dative Case.

I capture these facts by adding Case constraints to Stevenson and Smolensky's parser. In a similar fashion to an Optimality Theory grammar, the operation of an Optimality Theory parser is determined by the ranking of the constraints. The pressure to avoid a dative argument, represented by the constraint DAT, is stronger than the need to associate a filler with a gap which comes from the constraint ASS-Θ. Consequently, DAT is ranked higher than ASS-Θ. The constraint ASS-Θ, in turn, is ranked higher than ACC, which works against assigning accusative Case. The rest of the paper is organized as follows: I start with an overview of Optimality Theory (section 2) and Stevenson and Smolensky's Optimality Theory parser (section 3). I then discuss my data regarding filler-gap ambiguities and propose an extension to the parser (sections 4–6). The end of the paper points out additional predictions as well as cases that my model does not capture correctly (sections 7 and 8), and closes with a speculative extension of the theory that would deal with predicting actual processing difficulty and not just initial parsing preferences (section 9).

## 2 Optimality Theory

Optimality Theory (Prince and Smolensky 1993) is an architecture for grammar that characterizes the relation between underlying representation and surface realization in terms of constraints. A universal generation function (GEN) creates a set of candidate surface forms ("outputs") for each underlying representation ("input"); these candidates are then evaluated by the evaluation function (EVAL), which is made of the individual constraints. The candidate that fares best on the evaluation function is the optimal candidate, and Optimality Theory states that the optimal candidate for an underlying form will be the surface representation of that form.

The constraints of Optimality Theory assess each candidate surface form by assigning it a number of violations: the more violations a candidate has on a specific constraint, the worse it fares with respect to that constraint. In an Optimality Theory grammar the constraints are ranked in a strict dominance order. The evaluation function evaluates the candidate surface forms by checking the candidates against the constraints along the order in which the constraints are ranked, starting with the highest. At every stage, the candidates that have the least number of violations are selected and the rest are eliminated. The process then iterates with the next constraint, making further selection among the surviving candidates. When there remains only a single candidate it is declared optimal.

The following example demonstrates how the process works. Suppose for an underlying form $\alpha$ there are three candidate surface forms *a*, *b* and *c*, and four constraints *K*, *L*, *M* and *N*. The constraints assess the candidates as follows.

(10) a. Constraint *K*: candidate *a* 0 violations, *b* 0 violations, *c* 1 violation.
b. Constraint *L*: candidate *a* 1 violation, *b* 1 violation, *c* 0 violations.
c. Constraint *M*: candidate *a* 2 violations, *b* 1 violation, *c* 1 violation.
d. Constraint *N*: candidate *a* 0 violations, *b* 2 violations, *c* 0 violations.

Suppose the constraints are ranked $K \gg L \gg M \gg N$ (the notation $X \gg Y$ indicates that *X* is ranked higher than *Y*). The evaluation process can be read off the following tableau.

(11)

| | | *K* | *L* | *M* | *N* |
|---|---|---|---|---|---|
| | *a* | | * | **! | |
| ☞ | *b* | | * | * | ** |
| | *c* | *! | | * | |

The rows in the tableau represent the candidates and the columns represent the constraints. Each cell shows the number of violations a constraint assigns to a candidate; these are represented with one asterisk for each violation. The constraints are shown in the tableau in the order of their ranking. The evaluation process starts with the highest ranked constraint *K*. The candidates that have the least number of violations with respect to this constraint are candidates *a* and *b*, which both have no violations, so they are selected; candidate *c* has one violation so it is eliminated (the exclamation point in the tableau shows that this violation is crucial and causes the elimination of the candidate). Next comes constraint *L*. The candidates with the least number of violations incurred by this constraint are candidates *a* and *b*, which both have one violation; candidate *c* has already been eliminated, so the number of violations it has is not considered. Candidates *a* and *b* are both selected to continue, so they are evaluated by constraint *M*. Now the candidate with the least number of violations is candidate *a* which has one violation; candidate *b* has two violations so it is eliminated (the exclamation point shows that the second violation is crucial). Since candidate *a* is the only candidate remaining it is declared optimal; this is indicated in the tableau by placing a pointing hand next to it. The lowest ranked constraint *N* does not have an effect in this case.

An important point that can be seen through the example above is that the constraints are violable: the optimal candidate does violate constraints. The optimal candidate is chosen because at every stage in the evaluation process, no competing candidate has more violations. If we think of constraints as evaluating how well formed a candidate is with respect to a certain grammatical principle, then the optimal candidate is the one which is least offensive—it may violate some grammatical principles, but any other candidate would cause worse violations. Optimality Theory states that occurring surface forms are not perfect, but they are the best that can be done.

Another crucial point to note is that constraints are evaluated with respect to the ranking order. In our example, the optimal candidate violates more constraints than any other candidate, and causes more violations overall than any other candidate. But since the constraints are ranked in a strict order, the overall constraint violation profile of the optimal candidate is better than that of the other candidates. If the constraints were ranked differently, a different candidate may be chosen, as demonstrated in the following tableaus.

(12) a.

| | *L* | *M* | *N* | *K* |
|---|---|---|---|---|
| *a* | *! | ** | | |
| *b* | *! | * | ** | |
| ☞ *c* | | * | | * |

b.

| | *K* | *N* | *L* | *M* |
|---|---|---|---|---|
| ☞ *a* | | | * | ** |
| *b* | | *!* | * | * |
| *c* | *! | | | * |

By hypothesis, the constraints of Optimality Theory are universal, and all the constraints apply in every human language. Differences between languages result from different rankings of the constraints.

## 3 Parsing with Optimality Theory

An Optimality Theory grammar, like any other generative grammar, has as its input underlying grammatical representations, and its output is actual surface forms. A parser should take as its input an occuring surface string and return a linguistic representation of that string; in the case of a syntactic parser, the input is a string of words and the output is a syntactic structure.

Ideally, a parser should be related to a grammar—linguistic knowledge should manifest itself in the parsing mechanism somehow. An Optimality Theory parser is naturally associated with an Optimality Theory grammar. The grammar evaluates complete structural representations; parsing however works in real time, which means it has to be done incrementally, taking each word in its turn and incorporating it into the syntactic tree. Optimality Theory is suited for this purpose, since optimization can apply to any structure, including incomplete structures, and decide which of the candidate structures is the best. Partial structures may sometimes be ungrammatical; constraints in Optimality Theory are violable, hence they are able to accommodate temporary ungrammaticalities, which will be resolved with additional input. It seems reasonable, then, that a parser can be designed on the principles of Optimality Theory. If the constraints used by such a parser are taken from the grammar, then we will have a parsing model that gives a tight connection between language knowledge and use.

Stevenson and Smolensky (1997) propose an Optimality Theory parser that accounts for attachment ambiguities. The parser builds syntactic structure incrementally, attaching one word at a time and evaluating the resulting structure with respect to the constraints. The parser has to integrate every word into the structure—it is not allowed to leave two unconnected constituents. The model only predicts initial parsing preferences; in some cases, a continuation that is not compatible with this preference leads to a garden path sentence, which is extremely hard to process; in other cases it leads to a possible reanalysis. The model does not provide a theory of when reanalysis is easy and when it is difficult.

The above model explains a set of attachment ambiguities by appealing to the following four constraints.

(13) OB-HD: A projection (constituent) has a head.

ASS-Θ: A verb assigns all of its thematic roles.

RECENCY: A new word attaches to the most recent site.

UNDO: The structure of the previous partial parse is maintained.

(I have renamed two of Stevenson and Smolensky's constraints, in order to avoid confusing them with familiar grammatical constraints. In the original work OB-HD was called FILL, and ASS-Θ was called ARG.)

These constraints are ranked in the following order.

(14) OB-HD ≫ ASS-Θ ≫ RECENCY ≫ UNDO

Two of the above constraints, OB-HD and ASS-Θ, are taken from the grammar: OB-HD is the same as a grammatical constraint that has the same effect (Grimshaw 1997); ASS-Θ corresponds to the Theta Criterion, more precisely it is the clause in the Theta Criterion that requires the thematic grid of the verb to be satisfied. The other two constraints are not clearly grammatical, they look more like parsing heuristics.

Stevenson and Smolensky support the constraint ranking above by making a number of ranking arguments based on actual parsing preferences. The following parsing preference provides evidence for the ranking of ASS-Θ above RECENCY. The word that is being attached at this point is shown in boldface; the hand (☞) points at the preferred structure, while the cross mark (✗) shows the other possible attachment site; the words in italics following the sentences serve to illustrate the structure by disambiguating it with a possible continuation.

(15) ☞ Jill put [the candy] [**on** … *(the table.)*

✗ Jill put [the candy [**on** … *(the table into her mouth.)*

(16)

| | | ASS-Θ | RECENCY |
|---|---|---|---|
| ☞ | Jill put [the candy] [**on** | | * |
| | Jill put [the candy [**on** | * | |

The preferred parse is the one where the new word *on* is attached to the to the verb, despite the fact that this attachment is at the higher site (violation of RECENCY). This is done in order to avoid leaving a verbal argument open (violating ASS-Θ). The constraints OB-HD and UNDO are not violated by either of these structures, so they do not bear upon the argument.

The next contrast establishes that the constraint OB-HD is ranked above the constraint ASS-Θ.

(17) ☞ I told [the department **committees**] … *(about the project.)*

✗ I told [the department] [**committees** … *(will be formed.)*

(18)

| | | OB-HD | ASS-Θ | RECENCY |
|---|---|---|---|---|
| ☞ | I told [the dept. **committees** | | * | |
| | I told [the dept.] [**committees** | * | | * |

The preference is to attach the new word *committees* to the noun, violating ASS-Θ (since the verb *told* is waiting for its second argument); this avoids opening a new sentence constituent that lacks a head, in violation of OB-HD. This preference cannot be the result of RECENCY, because the previous example shows that RECENCY is ranked below ASS-Θ. The constraint UNDO is not shown in the tableau above. It is violated by the winning candidate but not by the loser, so we know it must be ranked below OB-HD; this is consistent with the ranking in (14), but we need more evidence to find the exact placement of UNDO in the ranking.

A third contrast completes the ranking in (14) by showing that RECENCY is ranked above UNDO.

(19) ☞ The man who [knew [the countess **killed** … *(herself…)*

✗ The man who [knew the countess] **killed** … *(himself.)*

(20)

| | | RECENCY | UNDO |
|---|---|---|---|
| ☞ | The man who [knew [the countess **killed** | | * |
| | The man who [knew the countess] **killed** | * | |

The partial parse tree before the word *countess* was read had to be *The man who [knew the countess]*, because the alternative would violate OB-HD. We see that the preference is to revise the previous structure, violating UNDO, rather than attaching the new word at a higher point on the tree, in violation of RECENCY. The constraints OB-HD and ASS-Θ are not relevant at this stage: in both of the parses in (19) the verb *knew* has both a subject and an object and the verb *killed* has a subject only, so the violations of ASS-Θ are identical. As for OB-HD, the assumption is that structure is only created when needed, so there exists no sentence projection at the time a matrix subject is being processed; this is in contrast to an embedded subject, where a headless sentence projection is necessary in order to attach the embedded subject to the rest of the sentence. Thus, neither of the sentences in (19) violates OB-HD.

Stevenson and Smolensky show parsing preferences from additional sentences with attachment ambiguities that confirm the full constraint ranking in (14). Their additional examples show that the ranking is consistent, that is all the data can be explained by the same ranking of the constraints. The four constraints give the desired initial attachment preferences in all the examples discussed by Stevenson and Smolensky.

It is interesting to note the position of the constraint UNDO at the bottom of the hierarchy: revision of the previous parse is preferred over violation of grammatical constraints or attachment heuristics such as RECENCY. This may sound surprising, given the wealth of known garden path sentences where difficult reanalysis leads to a problem with parsing. However, the current theory makes no claims as to what kind of reanalysis is possible and what is not; consequently the theory does not predict which sentences will turn out to be garden paths. The theory assumes an underlying grammar which determines the possible candidates to be evaluated by the constraints. The ranking of UNDO below other constraints shows that reanalysis, when possible, is better than violating the higher ranked constraints; it does not determine when such reanalysis is possible.

## 4 Filler-gap ambiguities

When processing a structure with overt movement, the parser identifies a displaced element, the "filler". The parser has to find a location in the parse tree that corresponds to the gap, or trace, left by the moved element. The Active Filler Hypothesis (Frazier 1987) says that the parser tries to find such a site as soon as possible; this is because holding the filler is somehow costly to the parser. An alternative interpretation is that finding a site for the filler to bind is required because of other reasons, for instance in order to satisfy a verb's argument structure. I argue for this option: the main pressure to discharge a filler comes from the thematic requirements of other elements in the sentence, specifically from the constraint ASS-Θ, which requires that a head assign all its thematic roles. Furthermore, this depends on the Case that is to be assigned to the gap: if the gap is to be assigned dative Case, the parsing preference is actually not to associate the filler with a gap.

Evidence comes from the following contrast. Sentence (21b) seems more natural than sentence (21a), and appears to be easier to process (cf. Boland 1997, Experiment 2).

(21) a. Which $dog_i$ did John give $e_i$ a bone?

b. Which $dog_i$ did John give a bone to $e_i$?

It appears that after the verb *give*, the human parsing mechanism prefers to leave the filler unresolved, even though there is a possible site for a gap to be bound by the filler. I propose that the reason is the relative markedness of dative Case, compared to accusative Case (see also Scheepers *et al.* 1998 for the relative markedness of dative gaps).

In Stevenson and Smolensky's system, the pressure to have a gap bound by the filler comes from the constraint ASS-Θ, which penalizes structures with unassigned thematic roles. The new constraints that we need to add to the system in order to account for filler-gap parsing preferences are grammatical constraints which mark Case. This is a welcome result, since we want the parser to be based as much as possible on grammatical knowledge and not on parsing heuristics. Markedness relations between Cases are universal, hence the ranking between these constraints is universally fixed (cf. the Case Preference Principle of Meng and Bader 1997).

(22) NOM: No element is assigned nominative Case.

ACC: No element is assigned accusative Case.

DAT: No element is assigned dative Case.

(23) DAT ≫ ACC ≫ NOM

Of course, these constraints will be routinely violated, since the Case Filter requires overt DPs as well as certain empty categories to have Case. Grimshaw (1997) proposes a grammatical constraint CASE that has this effect. It would make sense to have such a constraint in the parser as well, but this would entail that under certain configurations this constraint is violated; I do not know of examples where this happens, so for simplicity I will assume that the Case Filter is built into the underlying grammar, that is the parser will not consider candidates which violate the Case Filter. When an overt DP is read, the parser will have to assign it Case; when the parser has the option of deciding on covert DPs the Case constraints will come into play (interestingly enough, in Grimshaw's system the constraint CASE is undominated; it is sometimes violated by a winner candidate, but then all the competitors have at least the same number of violations).

We now have to determine the ranking of all the other constraints relative to this hierarchy. ASS-Θ is ranked between DAT and ACC.

(24) DAT ≫ ASS-Θ ≫ ACC

So the parser will prefer to have an accusative argument over leaving a thematic role open, but if it only has a choice between assigning a role to a dative argument or not assigning it at all, it will rather keep the role unassigned and wait for a better option to come along.

The parsing of the sentences in (21) thus proceeds in the following manner: when the verb *give* is read, the parser hypothesizes a gap which is interpreted as the accusative object of the verb (a reading that could be continued, for instance, into the sentence *Which dog*$_i$ *did John give e*$_i$ *to Mary?*).

(25)

| | | DAT | ASS-Θ | ACC | UNDO |
|---|---|---|---|---|---|
| ☞ | Which dog$_i$ did J **give** $e_i$(acc) $e$ | | * | * | |
| | Which dog$_i$ did J **give** $e_i$(dat) $e$ | *! | * | | |
| | Which dog$_i$ did J **give** $e$ $e$ | | **! | | |

(In the tableau, an empty category with an index ($e_i$) designates a gap associated with a filler, while an empty category without an index ($e$) is just a place holder for an argument, but not an associated gap.)

When the DP *a bone* is read following the verb, the analysis has to be revised, since there is no grammatical analysis that is compatible with the previous parse tree. In order to avoid a dative argument, the parser interprets the DP as the accusative argument; the filler is now left unresolved, violating ASS-Θ, but satisfying the higher ranked DAT (in order to make the examples more readable, I consider the DP *a bone* as a unit; the parsing decision is actually made as soon as the determiner *a* is read).

(26)

| | | DAT | ASS-Θ | ACC |
|---|---|---|---|---|
| | Which dog$_i$ did J give $e_i$(dat) **a bone**(acc) | *! | | * |
| | Which dog$_i$ did J give **a bone**(dat) $e_i$(acc) | *! | | * |
| | Which dog$_i$ did J give **a bone**(dat) $e$ | *! | * | |
| ☞ | Which dog$_i$ did J give **a bone**(acc) $e$ | | * | * |

Thus at this point the parser is expecting a continuation for the sentence, for instance with the preposition *to*. If it turns out that the sentence has ended then a further revision is necessary, which accounts for the awkwardness of (21b) as compared to (21a).

Note that the parsing preference above is independent of the plausibility of the interpretation. In sentence (21a) it is likely that *a bone* will be the accusative object. A sentence such as (27) is ambiguous between a reading that asks which dog was given to Mary and which dog Mary was given to, with the first being more plausible.

(27) Which dog did John give Mary?

Yet the analysis above predicts that at the point the DP *Mary* is read the parser will interpret it as accusative, and anticipate a continuation with the preposition *to*, despite the fact that such an interpretation is less plausible than the one where *Mary* is interpreted as dative.

Further evidence that it is Case that makes sentence (21a) difficult comes from comparing it to the seemingly similar sentence (28a): in the latter A-bar movement has been replaced by A movement, and there is no processing complexity. Note that the Case configuration here is different, since the gap is Caseless. In addition, the competing structure is ungrammatical.

(28) a. The $\text{dog}_i$ was given $e_i$ a bone.

b. *The $\text{dog}_i$ was given a bone to $e_i$.

The parser knows that any sentence without a gap immediately following the passive verb *given* will be ungrammatical (a violation of the Case Filter). Therefore, when the DP *a bone* is read, there is no competition to the winning candidate; there is no need for reanalysis, and no additional parsing complexity.

(29)

| | | DAT | ASS-Θ | ACC |
|---|---|---|---|---|
| ☞ | The $\text{dog}_i$ was **given** $e_i$(—) | | | |
| | The $\text{dog}_i$ was **given** $e$(—) | | *! | |

(30)

| | | DAT | ASS-Θ | ACC |
|---|---|---|---|---|
| ☞ | The $\text{dog}_i$ was given $e_i$(—) **a bone**(acc) | | | * |

Comparing the preferred structures in (29) and (30), we note that the thematic role assigned to the argument *the dog* may well have changed: in the first stage it may be interpreted as the theme, while in the second stage it has to be interpreted as the recipient, or goal. But this does not correspond to a difference in Case. The contrast between (21a) and (28a) is explained because the constraints look at the formal property of Case.

Introducing the constraint DAT, which is ranked above ASS-Θ, does make some predictions that are different from the model proposed by Stevenson and Smolensky. In the Double Object Verb/Compound Noun ambiguity the revised model predicts a preference for the compound noun interpretation.

(31) ?☞ I gave [the institute] [**animals**] … *(as a tax write-off.)*

?☞ I gave [the institute **animals**] … *(to the zoo.)*

(32)

| | | DAT | ASS-Θ | ACC | RECENCY |
|---|---|---|---|---|---|
| | I gave [the inst.] [**animals**] | *! | | * | * |
| ☞ | I gave [the inst. **animals**] | | * | * | |

The Stevenson and Smolensky model makes the opposite prediction, as can be easily verified by removing the DAT constraint from the tableau in (32). Since both parses can be continued without causing a noticeable processing difficulty, it is hard to determine which is the actual initial assignment preference, and consequently it is hard to evaluate whether the addition of the Case constraints improves the empirical coverage of the model or makes it worse.

This is the only construction where my predictions differ from those of Stevenson and Smolensky. The reason why the new constraint DAT has such little effect on attachment ambiguities is that the parser has to incorporate every word that it reads immediately into the parse tree. When a new DP is read, a conflict between DAT and ASS-Θ can only arise if the parser has the choice between integrating it as a dative argument to some head (violating DAT) or as part of another DP, leaving the dative position unfilled and thus violating ASS-Θ. In other cases where a dative argument is expected the parser will not have a choice but to integrate the new word into the parse tree, so there will be no option of leaving an empty dative position.

## 5 Fitting the constraints into one ranking

So far We have shown that the constraint ASS-Θ is ranked between DAT and ACC. We now proceed to determine the ranking of the Case constraints with respect to the rest of the hierarchy. The constraint DAT mandated that an argument should not be assigned dative Case; the following sentence, however, is hard to process precisely because of the tendency to interpret *the dog* as a second argument, which forces the first argument (*the boy*) to be dative.

(33) ¿I gave the boy the dog bit a bandaid.

In the correct analysis, the string *the dog bit* is a reduced relative clause modifying the DP *the boy*. There is a local ambiguity at the DP *the dog*, and the preference is to parse it as a second argument of *gave*.

(34) ☞ I gave [the boy] [**the dog**] … *(.)*

✗ I gave [the boy [**the dog** … *(bit to his mother.)*

The losing candidate violates both ASS-Θ and OB-HD, while the winning candidate violates DAT. Since we already know that DAT is ranked above ASS-Θ, it must be the higher ranked OB-HD that causes the candidate to lose. We conclude that OB-HD ≫ DAT.

(35)

| | | OB-HD | DAT | ASS-Θ | ACC |
|---|---|---|---|---|---|
| ☞ | I gave [the boy] [the dog] | | * | | * |
| | I gave [the boy [the dog | *! | | * | * |

The above ranking is consistent with Stevenson and Smolensky's system, where the constraint OB-HD is ranked above ASS-Θ; the constraint DAT fits in the hierarchy right between these two constraints.

We have already seen that ACC is ranked below ASS-Θ; we now have to determine the ranking of ACC with respect to the lower ranked constraints. The following example shows the relative ranking of ACC and UNDO. Consider the question (36), which is globally ambiguous between the two structures (36a) and (36b) (the answers in parentheses serve to illustrate the intended meaning).

(36) What do you have to write with?

a. What$_i$ do you$_j$ have PRO$_j$ to write with $e_i$? (I have to write with a pen.)
b. What$_i$ do you$_j$ have $e_i$ PRO to write with $e_i$? (I have a pen to write with.)

Intuitively it seems that (36a) is the preferred interpretation. The verb *have* is ambiguous, and has different argument structures in the two constructions: in (36a) it has a modal meaning like "must", and it takes a complement clause; in (36b) it has a possessive meaning and takes a DP complement, while the clause is an adjunct. Assuming that the parser works word by word, the preference will be to discharge the filler at the verb *have*, since on either interpretation it requires an argument; the verb *have* gets a possessive interpretation.

(37)

| | | ASS-Θ | ACC | UNDO |
|---|---|---|---|---|
| | What$_i$ do you$_j$ **have** *e* | *! | | |
| ☞ | What$_i$ do you$_j$ **have** $e_i$ | | * | |

When the word *to* is read, the preference is reversed, and *have* gets a modal interpretation. This requires breaking the association of the filler with the gap. Since now both structures fare equally well on the constraint ASS-Θ (the thematic grid of *have* is saturated), this has to be the result of the superfluous empty category. I suggest that this is because the empty category has Case. Hence we arrive at the ranking ACC ≫ UNDO.

(38)

| | | Ass-Θ | Acc | Undo |
|---|---|---|---|---|
| ☞ | What$_i$ do you$_j$ have [PRO$_j$ **to** | | | * |
| | What$_i$ do you$_j$ have $e_i$ [PRO **to** | | *! | |

We see that in this case the parser not only prefers to hold the filler in order to avoid assigning Case, it actually “takes back” a filler that has been discharged earlier in the parsing process. This gives further support to the claim that the main force that drives resolution of fillers is the requirement of the parse tree to have its argument structure satisfied, not a cost associated with holding a filler in memory.

The Case constraints thus have an effect of reducing the number of arguments in the sentence, by creating a pressure against empty categories. In Japanese, which is a head final language, the head of a relative clause appears after the relative clause itself; the pressure to reduce the number of empty categories will force the incorporation of as much overt material as possible into the relative clause.

The following sentences (Mazuka and Itoh 1995) are identical until the head of the relative clause appears; sentence (39) is garden-path sentence, while (40) causes no processing difficulty.

(39)

| | | | | | |
|---|---|---|---|---|---|
| ¿*Yakuza-no* | *kanbu-ga* | *wakai* | *kobun-o* | *sagasi-dasita* | *kenzyuu-de* |
| gang-gen | leader-nom | young | member-acc | found | gun-with |

*utikorosite simatta.*
shot to death

‘The leader of the gang shot the young member to death with the gun.’

(40)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| *Yakuza-no* | *kanbu-ga* | *wakai* | *kobun-o* | *sagasi-dasita* | *otoko-ni* | *rei-o* | *itta.* |
| gang-gen | leader-nom | young | member-acc | found | man-dat | thanked | |

‘The leader of the gang thanked the man who found the young member.’

Until the head noun of the relative clause is read, both sentences can be analyzed as a matrix clause (“the gang leader found the young member”). When an extra noun (“gun”/“man”) is encountered, the parser has to decide what the scope of the relative clause is. The correct structures for (39)–(40) are shown below; in (39′) the relative clause includes two null arguments, while in (40′) it includes only one.

(39′) gang-leader$_i$(nom) young-member(acc) [*pro*$_i$(nom) $e_j$(acc) found] gun$_j$

(40′) gang-leader(nom) [$e_j$(nom) young-member(acc) found] man$_j$

The preferred analysis is the one in (40′); this is predicted by our constraints, since (39′) has an extra argument, marked for accusative Case. This yields the preferred reading “the man/gun who found the young member” (note that this is the preferred structure despite the fact that in (39) it yields a nonsensical reading).

(41)

| | | Dat | Ass-Θ | Acc |
|---|---|---|---|---|
| | g.leader$_i$ y.member [*pro*$_i$ $e_j$ found] gun$_j$ "the gun he found" | | | **! |
| ☞ | g.leader [$e_j$ y.member found] gun$_j$ "the gun that found the young member" | | | * |

The model predicts that the initial reanalysis of the matrix clause reading will be one that puts as much overt material into the relative clause. Subsequent reanalysis of the relative clause turns out to be costly, and leads to a garden path effect. (Hirose *et al.* (1999) show that this preference can be manipulated: the preference to incorporate as much material as possible into the relative clause is strongest when the subject is long, for instance "gang leader" in the examples above. When the subject is short, semantic plausibility can result in a preference for a relative clause with additional null arguments; semantic plausibility has no effect when the subject is long. Hirose *et al.* attribute this effect to prosody—a long subject has a prosodic break after it, and readers want to preserve this break when they form the relative clause.)

## 6 The cost of holding a filler

So far I have made the assumption that the only pressure to associate a filler with a gap comes from the thematic requirements of the verbs in the sentence. We have already seen a case where a gap that had been associated with a filler was removed in order to reduce the number of Case marked arguments. A different construction that shows the same effect appears in Frazier, Clifton and Randall (1983). They say that (42a) is easier to process than (42b), and attribute this to the Recent Filler Preference, which favors indexing an empty category with the binder that is closest to it:

(42) a. Mary is one student who$_i$ the teacher$_j$ wanted PRO$_j$ to talk to the principal about $e_i$.

b. Mary is one student who$_i$ the teacher$_j$ wanted $e_i$ to talk to the principal.

Frazier *et al.* do not consider the different grammatical status of the empty categories involved. In our theory, the preference follows from the difference between *PRO* and trace: when the subject of the complement of *want* is a trace (42b) we have an Exceptional Case Marking (ECM) construction, so the trace receives accusative Case and incurs a violation of Acc; *PRO* (42a) on the other hand is by assumption ungoverned, so it does not receive Case at all. A local ambiguity occurs at the word *wanted* and persists through the word *to*.

(43)

| | | OB-HD | ASS-Θ | ACC |
|---|---|---|---|---|
| | … who$_i$ the teacher$_j$ **wanted** *e* | | *! | |
| ☞ | … who$_i$ the teacher$_j$ **wanted** $e_i$ | | | * |
| | … who$_i$ the teacher$_j$ **wanted** [PRO$_j$ | *! | | |
| | … who$_i$ the teacher$_j$ **wanted** [$e_i$ | *! | | * |

(44)

| | | ACC | UNDO |
|---|---|---|---|
| ☞ | … who$_i$ the teacher$_j$ wanted [PRO$_j$ **to** | | * |
| | … who$_i$ the teacher$_j$ wanted [$e_i$ **to** | *! | * |

The preferred interpretation after the word *wanted* has the filler associated with a gap. When the word *to* is read, the gap in the object position of *want* has to be removed in order to allow a sentential complement (forcing a violation of *Undo*). The sentential complement satisfies the thematic requirements of *want*, and there is no other pressure to restore the gap. The preferred reading, which persists through the rest of the sentence, has *PRO* as the subject of the infinitival clause, avoiding a Case marked trace.

What if there was some pressure against holding a filler that is not associated with a gap? If there is such pressure, it must be very weak. Let's assume a constraint FILLER that requires that every filler be associated with a gap. The above example shows that this constraint must be ranked below ACC.

(45)

| | | ACC | FILLER |
|---|---|---|---|
| ☞ | … who$_i$ the teacher$_j$ wanted [PRO$_j$ **to** | | * |
| | … who$_i$ the teacher$_j$ wanted [$e_i$ **to** | *! | |

If the constraint FILLER exists, it is ranked so low that so far we have no evidence for it.

There is further reason to doubt that constraint like FILLER exists in the grammar. Adverbial adjuncts are not subject to argument structure or Case requirements; if a constraint like FILLER is active in the grammar, we would expect to see its effect where the higher ranked constraints have no say. The following sentences show that an adverb can only associate with the verb in the clause in which it appears.

(46)
a. I said that Mary left yesterday. (*said yesterday; left yesterday)
b. I said that yesterday Mary left. (*said yesterday; left yesterday)
c. I said yesterday that Mary left. (said yesterday; *left yesterday)
d. Yesterday I said that Mary left. (said yesterday; *left yesterday)

When the adverb is a question particle it is moved to the left, and then the sentence is ambiguous.

(47) When did you say that Mary left? (said when; left when)

There does not appear to be a difference in processing difficulty between the two readings of (47); both seem very easy. This may suggest that there is no pressure to associate the filler as soon as possible. It is possible, of course, that the adjunct is first associated with the matrix verb, and reanalysis is easy. Either way, we do not have evidence for a pressure to associate a filler with a gap.

# 7 The filled gap effect

The predictions of our model correlate nicely with the findings of Stowe (1986) on places where a parser expects to find a gap. Stowe reports an increase in reading times of certain words where the parser expects a gap (the "filled gap effect"). She compares the following set of sentences.

(48) a. My brother wanted to know if Ruth will bring us home to mom at Christmas.

b. My brother wanted to know who$_i$ $e_i$ will bring us home to mom at Christmas.

c. My brother wanted to know who$_i$ Ruth will bring $e_i$ home to mom at Christmas.

d. My brother wanted to know who$_i$ Ruth will bring us home to $e_i$ at Christmas.

The effect reported by Stowe is an increase in the reading time of the word *us* in sentence (48d), compared to the control sentence (48a). Her explanation is that in (48d) the parser predicts a gap at the position of *us*, as in (48c), and therefore it takes it longer to integrate the word in place of the gap.

Our model also predicts a gap in that position, because it is better to assume a gap (violation of ACC) than to leave the slot empty (violation of ASS-Θ). We thus have the same initial preference. The underlying parser will translate this into an increase in attachment time—this is not a direct result of the Optimality Theory system.

Stowe reports no corresponding increase in the reading time of *Ruth* in (48c) and (48d) or *mom* in (48c), compared to the control. Our parser will not predict a gap in these positions: in the first case, having a gap in subject position will add a violation of OB-HD, and in the latter there is no available filler to bind a gap. So our model's predictions are in accord with Stowe's results.

A fourth position, for which Stowe does not report any data, is the position of the word *to* in (48d). Our parser does not predict a gap before *to*, because such a gap would force the word *us* to have dative Case, and this would cause a stronger violation than leaving the filler unassociated (DAT $\gg$ ASS-Θ).

(49)

| | | DAT | ASS-Θ | ACC |
|---|---|---|---|---|
| ☞ | … $\text{who}_i$ Ruth will bring us **home** *e* | | * | * |
| | … $\text{who}_i$ Ruth will bring us $e_i$ **home** | *! | | * |

So our model does not predict an increased reading time for *to* in (48d), and in this it differs from a model that assumed a gap wherever one is possible. The tableau in (49) also shows that our parser does predict an increased reading time if the next word happens to be *at*, as in sentence (48e) below, because such a continuation will have to lead to a reanalysis of the preferred parse in (49). Again, there is no experimental data with regard to this point.

(48) (e) My brother wanted to know $\text{who}_i$ Ruth will bring us $e_i$ home at Christmas.

This predicted effect is similar to the contrast noted in (21) above, in that the parser prefers to leave an argument slot empty rather than to have a dative argument (the difference is that in (21) it was the gap that was dative, while here the gap is accusative but it forces the pronoun *us* to be dative).

## 8 Additional constraints

The Case constraints introduced in section 4 are not enough to deal with all cases of ambiguities in filler-gap constructions. In the following contrast, the sentence is globally ambiguous.

(50) a. $\text{Who}_i$ did Fred tell Mary [$e_i$ left the country]?

b. $\text{Who}_i$ did Fred tell $e_i$ [Mary left the country]?

Clifton and Frazier (1989) state that intuitively (50b) is the preferred interpretation, and Crocker (1993) cites them and reports this as a "strong preference"; I will accept this judgment as correct. This is not predicted by our model. When the word *Mary* is read it must be attached somehow; the best is to have it as the object of *tell*, keeping the filler unassociated, since this is the only attachment that does not violate OB-HD.

(51)

| | | OB-HD | DAT | ASS-Θ |
|---|---|---|---|---|
| ☞ | Who$_i$ did Fred tell **Mary** *e* | | * | * |
| | Who$_i$ did Fred tell **Mary** [$e_i$ … | *! | * | |
| | Who$_i$ did Fred tell $e_i$ [**Mary** … | *! | * | |

This prediction fits well with the intuition that both of the continuations in (52) below are easier to process than either of the alternatives in (50).

(52) a. Who$_i$ did Fred tell Mary [Bill saw $e_i$]?

b. Who$_i$ did Fred tell Mary about $e_i$?

However, in sentence (50) when the word *left* is read, somehow *Mary* is interpreted as the subject of *left*, and a gap is inserted as the object of *tell*. This constitutes a violation of UNDO, since *Mary* is moved from its initial attachment site. In order to get the data right, we must have a new constraint that dominates UNDO; the existing constraints will not make the right prediction.

(53)

| | | DAT | ASS-Θ | ???? | UNDO |
|---|---|---|---|---|---|
| | Who$_i$ did F tell M [$e_i$ left *e* | * | * | *! | |
| ☞ | Who$_i$ did F tell $e_i$ [M left *e* | * | * | | * |

It appears that *Mary* is moved to the subject position of the embedded clause because of a requirement in this clause. My hypothesis is that the parser is trying to avoid having a trace in this position; I will propose a specific constraint to this effect, call it SUBJ-T. The crucial ranking is SUBJ-T ≫ UNDO. The constraint SUBJ-T is probably related to Grimshaw's (1997) constraint T-GOV, which requires that a trace be governed (however it can not be exactly the same, since under Grimshaw's assumptions the verb *tell* lexically governs the subject position of the complement clause). Our new constraint might also play a role in the examples in (42) above, depending on how exactly this constraint is formulated and what the government relations are in ECM constructions. Notice that the parser cannot avoid having a dative argument following the verb *tell*, since the verb does not allow a sentential complement without an explicit dative argument (in contrast to the case of a DP object as in *tell a story*).

Another instance where the Case constraints make a wrong prediction is the phenomenon of Case matching which occurs in languages with overt Case marking such as German. Meng and Bader (1997) show instances where an ambiguously Case-marked DP that heads a relative clause tends to be interpreted as matching the Case of the relative pronoun. The following sentence, for instance, causes a garden path effect because the DP *Maria*, which needs accusative Case, is initially interpreted as dative. This is the result of the adjacent dative relative pronoun, and despite the fact that normally in German the preference is to interpret ambiguous arguments as accusative rather than dative.

(54) *Ich glaube daß Maria, der ich vorhin begegnet bin, das Buch geliefert hat.*
I believe that Maria $who_{dat}$ I just met am the book delivered has
'I believe that Maria, who I have just met, has delivered the book.'

The preference of a dative interpretation of the head noun must be due to a constraint that is ranked higher than DAT. A simple parsing constraint that prefers matching the Case of a head noun and a relative pronoun will do the trick (the relative pronoun *der* is unambiguously marked for dative by the morphology).

(55)

| | | MATCH | DAT | ACC | NOM |
|---|---|---|---|---|---|
| | Ich glaube daß $Maria_{nom}$ der ... | *! | | | * |
| | Ich glaube daß $Maria_{acc}$ der ... | *! | | * | |
| ☞ | Ich glaube daß $Maria_{dat}$ der ... | | * | | |

However, it is not clear what the grammatical status of a constraint like MATCH is—whether it has an effect in the grammar or if it is just a parsing heuristic.

# 9 Garden paths as an increase in markedness

The final section of this paper is rather speculative. The model so far only predicts initial attachment preferences; it makes no claims about reanalysis. This section is an attempt to characterize reanalysis in terms of the Optimality Theory parser.

In an Optimality Theory grammar, markedness is always relative: a candidate does not have an absolute markedness value, its markedness can only be compared to other candidates in the set against which it is evaluated. A form can be optimal even if it has a lot of violations, provided that the competing candidates fare even worse. Stevenson and Smolensky (1997) tried to characterize absolute markedness (easy and difficult sentences) by comparing violation patterns of optimal forms from different candidate sets, in effect creating an absolute measure of markedness. I will be more conservative: my suggestion is still comparative, but it compares violation profiles at different stages in the parse process. I propose that processing difficulty arises when the markedness of the parse increases from one stage to another.

If indeed an increase in markedness results in processing difficulty, we may expect that a rise in the constraint violation pattern should create a garden path sentence. This can not be outright true: for instance, whenever a new clause begins we immediately get an OB-HD violation, yet this in itself does not make a sentence unprocessable despite the high ranking of OB-HD. I will therefore suggest that the above claim only applies to reanalysis. When a reanalysis results in an increase in markedness, this leads to difficult processing.

Consider the difference in processing difficulty between the following two sentences.

(56) a. ¿John warned Mary left.

b. John knows Mary left.

Pritchett (1992) characterizes the difference in structural terms (the On Line Locality Constraint): in order to get the correct interpretation of (56a) the parser has to perform an operation that's not available to it, namely moving *Mary* from one position to another that is not dominated by the original position. In contrast, the new position of *Mary* in (56b) is dominated by its original position, a move that is possible for the parser.

In our model there is no inherent difference between the two operations: both break an attachment and create a new one (in violation of UNDO). The only way to get Pritchett's structural explanation would be to put it into the parser as a high ranking constraint (or incorporate it into the underlying grammar), a move that does not give a new explanatory insight. But if we accept Pritchett's structural analysis, we see that in (57) there is an increase in the markedness of the optimal candidate from one stage of the parse to the next, while in (58) there is a decrease in markedness.

(57)

| | | OB-HD | ASS-Θ | ACC | UNDO |
|---|---|---|---|---|---|
| ☞ | John warned **Mary** | | **?** | * | |
| | John warned *e* [**Mary** ... | *! | * | | |
| ☞ | John warned *e* [Mary **left**] | | * | | * |

(58)

| | | OB-HD | ASS-Θ | ACC | UNDO |
|---|---|---|---|---|---|
| ☞ | John knows **Mary** | | | * | |
| | John knows [**Mary** ... | *! | | | |
| ☞ | John knows [Mary **left**] | | | | * |

(in order for (57) to work we must assume that the absence of a sentential complement for *warn* does not create an ASS-Θ violation, that is the sentential complement is not an argument).

Because of the increase in markedness, we expect that the transition (reanalysis) in (57) should be more difficult than the transition in (58), though we still have no account of why the harder one is so difficult that it constitutes a garden path. A similar increase in markedness can be seen in the following example.

(59) ¿While Mary was mending the socks fell.

(60)

| | | Ob-Hd | Ass-Θ | Acc | Undo |
|---|---|---|---|---|---|
| ☞ | While M was [mending **the socks**] | | | * | |
| | While M was mending *e* [**the socks** . . . | *! | * | | |
| | | | | | |
| ☞ | While M was mending *e* [the socks **fell**] | | * | | * |

A common pattern emerges in the tableaus in (57) and (60): when the new verb "steals" the object of the first verb, and the verb is still looking for that argument in the resulting sentence, then an extra Ass-Θ violation ensues; this causes the markedness to increase, and renders the reanalysis difficult.

Unfortunately, this cannot be the whole story. An extra violation of Ass-Θ also arises when the complement of the subordinate verb *knows* is a transitive verb.

(61) John knows Mary kicked Bill.

(62)

| | | Ob-Hd | Ass-Θ | Acc | Undo |
|---|---|---|---|---|---|
| ☞ | John knows **Mary** | | | * | |
| | John knows [**Mary** . . . | *! | | | |
| | | | | | |
| ☞ | John knows [Mary **kicked** *e*] | | * | | * |

If we want to maintain that the difficulty in sentences (56a) and (59) comes from the thematic structure of the first verb, we will have to acknowledge a difference between an Ass-Θ violation where an argument has just been predicted but hasn't been read yet, and Ass-Θ violations where there is a competing candidate with the argument position filled, or there has been such a candidate at a previous stage in the parsing process. I leave this as an open problem.

## References


Boland, J. E. (1997). "The Relationship between Syntactic and Semantic Processes in Sentence Comprehension." *Language and Cognitive Processes* 12(4), pp. 423–484.

Clifton, C. and L. Frazier (1989). "Comprehending Sentences with Long-Distance Dependencies." In G. N. Carlson and M. K. Tanenhaus (eds.), *Linguistic Structure in Language Processing*, pp. 273–317. Dordrecht: Kluwer.

Crocker, M. W. (1993). “Properties of the Principle-Based Sentence Processor”. In *Proceedings of the 15th Annual Cognitive Science Society Conference*, Boulder, CO, June 1993, pp. 371–376.

Frazier, L. (1987). “Syntactic Processing: Evidence from Dutch.” *Natural Language and Linguistic Theory* 5(4), pp. 519–559.

Frazier, L., C. Clifton, and J. Randall (1983). “Filling Gaps: Decision Principles and Structure in Sentence Comprehension.” *Cognition* 13, pp. 187–222.

Grimshaw, J. (1997). “Projection, Heads, and Optimality.” *Linguistic Inquiry* 28(3), pp. 373–422.

Hirose, Y., A. Inoue, J. D. Fodor, and K. Kakehi (1999). “Evidence for an Effect of Implicit Prosodic Phrasing on Ambiguity Resolution in Reanalysis.” Talk given at CUNY-99.

Mazuka, R. and K. Itoh (1995). “Can Japanese Speakers be Led down the Garden Path?” In R. Mazuka and N. Nagai (eds.), *Japanese Sentence Processing*, pp. 295–329. Hillsdale, NJ: Lawrence Erlbaum Associates.

Meng, M. and M. Bader (1997). “Syntax and Morphology in Sentence Processing.” Manuscript, University of Jena.

Prince, A. and P. Smolensky (1993). *Optimality Theory: Constraint Interaction in Generative Grammar*. Technical Report 2 of the Rutgers Center for Cognitive Science, Rutgers University.

Pritchett, B. L. (1992). *Grammatical Competence and Parsing Performance*. Chicago: University of Chicago Press.

Scheepers, C., B. Hemforth, and L. Konieczny (1998). “The Impact of Constituent Ordering on Case Assignment in German.” Poster presented at AMLaP-98.

Stevenson, S. and P. Smolensky (1997). “Optimal Sentence Processing.” Conference on Computational Psycholinguistics (CPL ’97).

Stowe, L. A. (1986). “Parsing WH-constructions: Evidence for On-line Gap Location.” *Language and Cognitive Processes* 1, pp. 227–245.