

# A Refined Notion of Memory Usage for Minimalist Parsing

Thomas Graf    Brigitta Fodor    James Monette  
Gianpaul Rachiele    Aunika Warren    Chong Zhang

Stony Brook University, Department of Linguistics

mail@thomasgraf.net, {firstname.lastname}@stonybrook.edu

## Abstract

Recently there has been a lot of interest in testing the processing predictions of a specific top-down parser for Minimalist grammars (Stabler, 2013). Most of this work relies on memory-based difficulty metrics that relate the shape of the parse tree to processing behavior. We show that none of the difficulty metrics proposed so far can explain why subject relative clauses are more easily processed than object relative clauses in Chinese, Korean, and Japanese. However, a minor tweak to how memory load is determined is sufficient to fully capture the data. This result thus lends further support to the hypothesis that very simple notions of resource usage are powerful enough to explain a variety of processing phenomena.

## 1 Introduction

One of the great advantages of mathematical linguistics is that its formal rigor allows for the exploration of ideas and questions that could not even be precisely formulated otherwise. A promising project along these lines is the investigation of syntactic processing from a computationally informed perspective (Joshi, 1990; Rambow and Joshi, 1995; Steedman, 2001; Hale, 2011; Yun et al., 2014). This requires I) an articulated theory of syntax that has sufficient empirical coverage to be applicable to a wide range of constructions, II) a sound and complete parser for the syntactic formalism, and III) a linking theory that derives psycholinguistic predictions from these two components. A successful

model along these lines provides profound insights into the mechanisms of linguistic performance, and it can also rule out certain syntactic proposals as psycholinguistically inadequate. Unfortunately there are multiple choices for each one of the three components, which raises the question of which combinations are empirically adequate.

This paper explores this issue for Minimalist grammars (MGs), a formalization of the Chomskyan variety of generative grammar that informs a lot of psycholinguistic research nowadays. Taking as our vantage point Kobele *et al.* (2012; henceforth KGH) and their method for deriving structure-sensitive processing predictions from Stabler’s (2013) MG top-down parser, we evaluate how well the parser captures the processing difficulty of relative clauses in Chinese, Japanese, and Korean — a phenomenon that escapes many processing models in the literature. By carefully modulating the set of syntactic assumptions as well as the linking hypotheses, we show that none of the memory-based proposals in the tradition of KGH yield the right predictions. The correct results are obtained, however, if the size of parse items also counts towards their memory usage. Our paper thus serves a dual purpose: it provides a positive result in the form of a more refined notion of memory usage that explains the observed processing behavior, and a negative one by eliminating many combinations of the three factors listed above.

Our discussion starts with two introductory sections that familiarize the reader with the research this paper follows up on. We first discuss MGs, the MG top-down parser, and how this parser has been used to model processing phenomena in re-

cent years. This is followed by a brief review of a long standing problem in syntactic processing: the preference for subject relative clauses (SRCs) over object relative clauses (ORCs) irrespective of cross-linguistic word order differences. We present two prominent relative clause analyses from the syntactic literature, and we discuss why the preference for SRCs over ORCs is surprising given current psycholinguistic models. In Sec. 4 we finally demonstrate that the MG parser cannot make the right predictions with any of the proposed metrics unless one refines their conception of memory load.

## 2 Minimalist Grammars for Processing

### 2.1 Minimalist Grammars

MGs (Stabler, 1997) are a formalization of the most recent iteration of transformational grammar, known as Minimalism. Since they formalize ideas that form the underpinning for the majority of contemporary research in theoretical syntax and syntactic processing, they act as a form of glue that makes these ideas amenable to more rigorous study. The main purpose of MGs in this paper is to provide a specific type of structure for the parser to operate on — derivation trees. Consequently the technical machinery is of interest only to the extent that it illuminates the connection between derivations and MG parsing, and we thus omit formal details where possible.

An MG is a finite set of lexical items (LIs), where every LI consists of a *phonetic exponent* and a finite, non-empty string of *features*. Each feature has a *positive* or *negative* polarity, and it is either a *Merge* feature (written in upper caps) or a *Move* feature (written in lower caps). MGs assemble LIs into trees via the structure-building operations *Merge* and *Move* according to the feature specifications of the LIs. Intuitively, *Merge* may combine two LIs if their respective first unchecked features are *Merge* features and differ only in their polarity. The LI with the positive polarity feature acts as the head of the assembled phrase. *Move*, on the other hand, removes a phrase from an already assembled tree and puts it in a different position; see Stabler (2011) for a formal definition. Figure 1 shows a simplified tree for *John, the girl likes*, with dashed lines indicating which positions certain phrases were displaced from.

The structure of a sentence is also fully encoded

by its derivation tree, i.e. the record of how its phrase structure tree was assembled from the LIs via applications of *Move* and *Merge*. Every derivation tree corresponds to exactly one phrase structure tree, but the reverse does not necessarily hold. The main difference between the two types of tree is that moving phrases remain in their base position in the derivation tree — compare, for instance, the positions of *John* and *the girl* in the two trees in Fig. 1 (for the sake of clarity interior nodes have the same label as their counterpart in the phrase structure tree). As a result, derivation trees do not directly reflect the word order of a sentence, which must be derived by carrying out the movement steps.

In addition, an MG’s set of well-formed derivation trees forms a regular tree language thanks to a specific restriction on *Move* that is known as the Shortest Move Constraint (Michaelis, 2001; Koble et al., 2007; Salvati, 2011; Graf, 2012). The set of well-formed phrase structure trees, on the other hand, is supra-regular — a corollary of MGs’ weak equivalence to MCFGs (Harkema, 2001; Michaelis, 2001). The fact that derivation trees do not need to directly encode linear order thus reduces their complexity significantly in comparison to phrase structure trees. Since derivation trees offer a complete regular description of the structure of a sentence, and because regular tree languages can be viewed as context-free grammars (CFGs) with an ancillary hidden alphabet (Thatcher, 1967), MGs turn out to be close relatives of CFGs with a more complex mapping from trees to strings. It is this close connection to CFGs that forms the foundation of Stabler (2013)’s top-down parser.

### 2.2 MG Parsing as Tree Traversal

Stabler (2013)’s parser for MGs builds on standard depth-first, top-down parsing strategies for CFGs but modifies them in three important respects: I) the parser is equipped with a search beam that discards the most unlikely analyses, thus avoiding the usual problems with left recursion, II) the parser constructs derivation trees rather than phrase structure trees, and III) since derivation trees do not directly reflect linear order, the parser moves through them in a particular fashion that would approximate a left-most, depth-first search in the corresponding phrase structure trees.

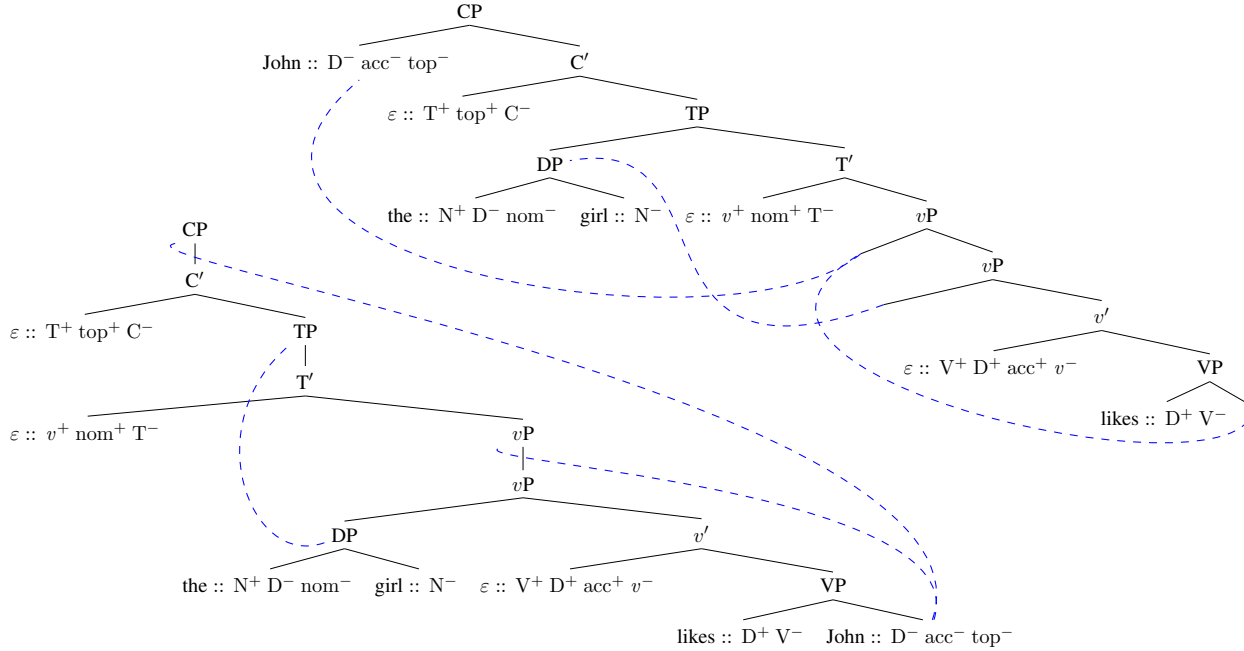


Figure 1: MG phrase structure tree and derivation tree for *John, the girl likes*; dashed branches indicate movement

We completely ignore the beam in this paper and instead adopt KGH’s assumption that the parser is equipped with a perfect oracle so that it never makes any wrong guesses during the construction of the derivation tree. While psychologically implausible, this idealization is meant to stake out a specific research goal: processing effects must be explained purely in terms of the syntactic complexity of the involved structures, rather than the difficulty of finding these structures in a large space of alternatives. More pointedly, we assume that parsing difficulty *modulo* non-determinism is sufficient to account for the processing phenomena under discussion.

With non-determinism completely eliminated from the picture, the parse of some sentence  $s$  reduces to a specific traversal of the derivation tree of  $s$ . In general, the parser follows a left-most, depth-first strategy, where a node is left-most if it is a specifier or if it is a head with a complement. However, when a Move node is encountered, two things can happen, depending on whether the Move node is an *intermediary* landing site or a *final* one. Let  $p$  be a moving phrase and  $m_1, \dots, m_n$  the Move nodes that denote an instance of Move displacing  $p$ . Then  $m_i$  is a final landing site (or simply *final*) iff there is

no  $m_j$ ,  $1 \leq j \leq n$ , that properly dominates  $m_i$  in the derivation tree. A Move node is an intermediary landing site (or *intermediary*) iff there is no phrase in the derivation tree for which it is a final landing site. An intermediary Move node does not affect the parser’s tree traversal strategy. A final Move node, on the other hand, causes the parser to take the shortest path to the phrase that will be displaced by this instance of Move. Once the root of that phrase has been reached, the parser traverses its subtree in the usual fashion and then returns to the point where it veered off the standard path.

The traversal is made fully explicit via a notation adopted from KGH where each node in the derivation tree has a superscripted *index* and a subscripted *outdex*. The index lists the point at which the parse item corresponding to the node is inserted into the parser’s memory queue, whereas the outdex gives the point at which said parse item is removed from the queue. Both values can be computed in a purely tree-geometric fashion. Let  $s[\text{surface}]\text{-precedence}$  be the relation that holds between nodes  $m$  and  $n$  in a derivation tree iff their counterparts  $m'$  and  $n'$  in the corresponding phrase structure tree stand in the precedence relation (if  $m$  undergoes movement, its

counterpart  $m'$  is the final landing site rather than its base position). Then indices and outdices can be inferred without knowledge of the parser by the following procedure (cf. Fig. 2 on page 7):

- The index of the root is 1. For every other node, its index is identical to the outdex of its mother.
- If nodes  $n$  and  $n'$  are distinct nodes with index  $i$ , and  $n$  reflexively dominates a node that is not s-preceded by any node reflexively dominated by  $n'$ , then  $n$  has outdex  $i + 1$ .
- Otherwise, the outdex of node  $n$  with index  $i$  is  $\max(i + 1, j + 1)$ , where  $j \geq 0$  is greatest among the outdices of all nodes that s-precede  $n$  but are not reflexively dominated by  $n$ .

### 2.3 Parsing Metrics

In order to allow for psycholinguistic predictions, the behavior of the parser must be related to processing difficulty via a parsing metric. There is no *a priori* limit on the complexity of metrics one may entertain, but the methodologically soundest position is to explore simple metrics before moving on to more complicated ones.

Extending KGH, Graf and Marcinek (2014; henceforth GM) evaluate a variety of memory-based metrics that measure I) how long a node is kept in memory (*tenure*), or II) how many nodes must be kept in memory (*payload*), or III) specific combinations of these two factors. Tenure and payload are easily defined using the node indexation scheme. A node's tenure is the difference between its index and outdex, and the payload of the derivation tree is equal to the number of nodes with a tenure strictly greater than 2 (in the derivation trees in Figs. 2–5, these nodes are boxed to highlight their contribution to the payload).

GM define three metrics, the first of which is adopted directly from KGH. Depending on the metric, the difficulty of a parse is given by

$$\mathbf{Max} \max(\{t \mid t \text{ is the tenure of some node } n\})$$

$$\mathbf{Box} |\{n \mid n \text{ is a node with tenure } > 2\}|$$

$$\mathbf{Sum} \sum_{n \text{ has tenure } > 2} \text{tenure-of}(n)$$

GM define an additional six variants by restricting the choice of nodes  $n$  to LIs and pronounced LIs, respectively. They then compare the predictions of these nine metrics with respect to right embedding VS center embedding, and nested dependencies VS crossing dependencies (both of which were originally analyzed in KGH), as well as two phenomena involving relative clauses: I) sentential complements containing a relative clause VS a relative clause containing a sentential complement, and II) the preference for subject relative clauses (SRCs) over object relative clauses (ORCs) in English. They conclude that the only metric that makes the right predictions in all four constructions is **Max** restricted to pronounced LIs.

Irrespective of the choice of metric, though, the psycholinguistic predictions of the MG parser vary with the choice of syntactic analysis. KGH use this fact for a persuasive demonstration of how processing data can be brought to bear on the distinction between so-called phrasal movement and head movement. It is unclear, however, whether this should be interpreted as support for a specific movement analysis or as evidence against the assumed difficulty metric. GM's comparison sheds little light on this because it presupposes a specific syntactic analysis for each phenomenon. A more elaborate comparison is required that varies both the parsing metric and the choice of syntactic analysis, ideally resulting in only a few empirically adequate combinations. The processing contrast between *prenominal* SRCs and ORCs is exactly such a case.

## 3 Surveying Relative Clauses

### 3.1 Syntax

The main idea of this paper is that the space of possible combinations of syntactic analyses and parsing metrics can be narrowed down quite significantly by looking at processing phenomena that have proven difficult to account for. As we will see next, the fact that SRCs are easier to parse than ORCs in Chinese, Korean, and Japanese constitutes such a problem. We first discuss how the two have been analyzed in the syntactic literature, while the next section explains why many well-known processing models have a hard time capturing the data.

Relative clauses (RCs) can be categorized accord-

ing to two parameters. First, the *head noun*, i.e. the noun modified by the RC, may be the subject or the object of the RC, in which case we speak of an SRC and an ORC, respectively. Second, an RC is *postnominal* if it is linearly preceded by its head noun, and *prenominal* otherwise. Note that in prenominal languages the complementizer (if it is realized overtly) usually occurs at the right edge of the RC rather than the left edge. Whether RCs have such an overt complementizer is an ancillary parameter.

Most analyses of RCs were developed for languages like English, French, and German, where RCs are postnominal and have overt complementizers (which might be optional). The general template is [DP Det head-noun [RC complementizer subject verb object]], with either the subject or the object unrealized and the position of the verb depending on language-specific word order constraints.

- (1) a. [DP The mayor [RC who \_ invited the tycoon]] likes wine.  
 b. [DP The mayor [RC who the tycoon invited \_]] likes wine.

The canonical account is the *wh-movement* analysis, according to which the complementizer fills the subject or object position, depending on the type of RC, and then moves into Spec,CP (Chomsky, 1965; Heim and Kratzer, 1998). Alternatively, the complementizer starts out as the C-head and instead a silent operator undergoes movement from the base position to Spec,CP. For the purposes of this paper the two variants of the *wh-movement* analysis are fully equivalent.

The *promotion* analysis is a well-known competing proposal (Vergnaud, 1974; Kayne, 1994). It combines the ideas above and posits that the complementizer starts out as the C-head, but instead of a silent operator it is the head noun that moves from the embedded subject/object position into Spec,CP. In contrast to the *wh-movement* analysis, the head noun is thus part of the RC. Crucially, though, all three proposals involve an element that fills the seemingly empty argument position of the verb and subsequently moves to Spec,CP.

Languages with prenominal RCs, such as Chinese, Japanese, and Korean, can be analyzed along these lines, but differences in word order lead to a significant increase in analytic complexity. Below is

an example of the English sentence in (1) with Chinese word order.

- (2) a. [DP [RC \_ invited the tycoon who] the mayor] likes wine.  
 b. [DP [RC the tycoon invited \_ who] the mayor] likes wine.

On a theoretical level, there are two major complications. First, while Chinese is an SVO language like English, Japanese and Korean are SOV languages, which requires movement of the object to Spec,vP, thereby adding at least one more movement step within each RC in these two languages. More importantly, the prenominal word order must be derived from the postnominal one via movement, which causes the *wh-movement* analysis and the *promotion* analysis to diverge more noticeably.

In the *promotion* analysis, the RC is no longer a CP, but rather a RelP that contains a CP (see also Yun et al., 2014). The head noun still moves from within the RC to Spec,CP, but this is followed by the TP moving to Spec,RelP so that one gets the desired word order with the complementizer between the rest of the RC in Spec,RelP and the head noun in Spec,CP. In the *wh-movement* analysis, the head noun is once again outside the RC, which is just a CP instead of a RelP. The complementizer starts out in subject or object position depending on the type of RC, and then moves into a right specifier of the CP. The CP subsequently moves to the specifier of the DP of the head noun, once again yielding the desired word order with the complementizer between the RC and the head noun.

In sum, the *promotion* analysis needs to posit a new phrase RelP but all movement is leftward and takes place within this phrase, whereas the *wh-movement* analysis sticks with a single CP but invokes one instance of rightward movement and moves the RC into Spec,DP, a higher position than Spec,RelP. Both accounts are fairly complicated due to the sheer number and intricate timing of movement steps — the reader is advised to carefully study the derivations in Figures 2 through 5.

Involved as they might be, both the *promotion* analysis and the *wh-movement* analysis are workable solutions for the kind of prenominal SRCs and ORCs found in Chinese, Korean, and Japanese. The latter two only add an additional movement step for

each object to Spec,*v*P, and Japanese differs from Chinese and Korean in that the RC complementizer is never pronounced.

### 3.2 Psycholinguistics

SRCs and ORCs have been the subject of extensive psycholinguistic research, with overwhelming evidence pointing towards SRCs being easier to process than ORCs irrespective of whether RCs are prenominal or postnominal in a given language (Mecklinger et al., 1995; Gibson and Pearlmuter, 1998; Mak et al., 2002; Miyamoto and Nakamura, 2003; Gordon et al., 2006; Kwon et al., 2006; Mak et al., 2006; Ueno and Garnsey, 2008; Kwon et al., 2010; Miyamoto and Nakamura, 2013). The data is less clear-cut in Chinese (Lin and Bever, 2006), but it has recently been argued that this is only because of certain structural ambiguities (Gibson and Wu, 2013). Yun et al. (2014) even show how such an ambiguity-based account can be formalized via the MG parser. Recall, though, that we deliberately ignore ambiguities in this paper in an effort to find the simplest empirically adequate linking between derivations and processing behavior. For this reason, we assume that Chinese would also exhibit a uniform preference for SRCs over ORCs if it were not for the confound of structural ambiguity.

That language-specific differences in word order have no effect on the difficulty of SRCs relative to ORCs is unexpected under a variety of psycholinguistic models. Dependency Locality Theory (Gibson, 1998) and the Active-Filler strategy (Frazier, 1987), for example, contend that parsing difficulty increases with the distance between a filler and its gap due to a concomitant increase in memory load — an idea that is also implicit in KGH’s **Max** metric. However, both models calculate distance over strings rather than trees. Since prenominal RCs put the object position (i.e. the gap) linearly closer to the head noun (the filler), while the subject is farther away, ORCs should be easier than SRCs.

The failure of string-based memory load models can be remedied in two ways. One is to abandon the notion that the SRC-ORC asymmetry derives from structural factors, replacing it by functional concepts such as Keenan and Comrie’s (1977) accessibility hierarchy, which claims that objects are harder to manipulate than subjects irrespective of the con-

struction involved. While certainly a valid hypothesis, a computationally informed perspective has little light to shed on it. We thus discard this option and focus instead on how a more elaborate concept of sentence structure may interact with memory-based concepts of parsing difficulty. More precisely: can the MG parser, when coupled with a suitable RC analysis and one of the metrics discussed in Sec. 2.3, explain why SRCs are easier to parse than ORCs?

## 4 Parser Predictions

### 4.1 Overview of Data

The annotated derivation trees for Chinese and Korean RCs are given in Figures 2 through 5. Japanese is omitted since it has exactly the same analysis as Korean except that the RC complementizer remains unpronounced. Interior nodes are labeled with projections instead of Merge and Move for the sake of increased readability, and a dashed branch spanning from node  $m$  to node  $n$  indicates movement of the whole subtree rooted in  $m$  to the specifier of  $n$ . For the wh-analysis, we use a dotted line instead of a dashed one if movement is to a right specifier rather than a left one. Since these notational devices make features redundant, they are omitted completely.

The tenure values for Chinese and Korean are summarized in Tables 1 and 2, respectively. The table subgroups nodes according to whether they are pronounced LIs, unpronounced LIs, or interior nodes. It also includes the summed tenure values for, respectively, pronounced LIs, all LIs, and all listed nodes. Once again we omit Japanese since it shows exactly the same behavior as Korean, except that the complementizer would be grouped under “lexical” and not “pronounced”.

### 4.2 Evaluation of Metrics

All the metrics discussed in Sec. 2.3 fail insofar as they do not predict a consistent preference for SRC over ORC. On the other hand, some metrics fare worse than others because they predict the very opposite, ORC being easier than SRC. This is the case for **Sum**, which adds the tenure of all nodes that contribute to the derivation’s payload. The problem is that ORCs have a smaller total tenure than SRCs in Korean and Japanese irrespective of the choice of analysis. Furthermore, if the tenure of phrasal nodes

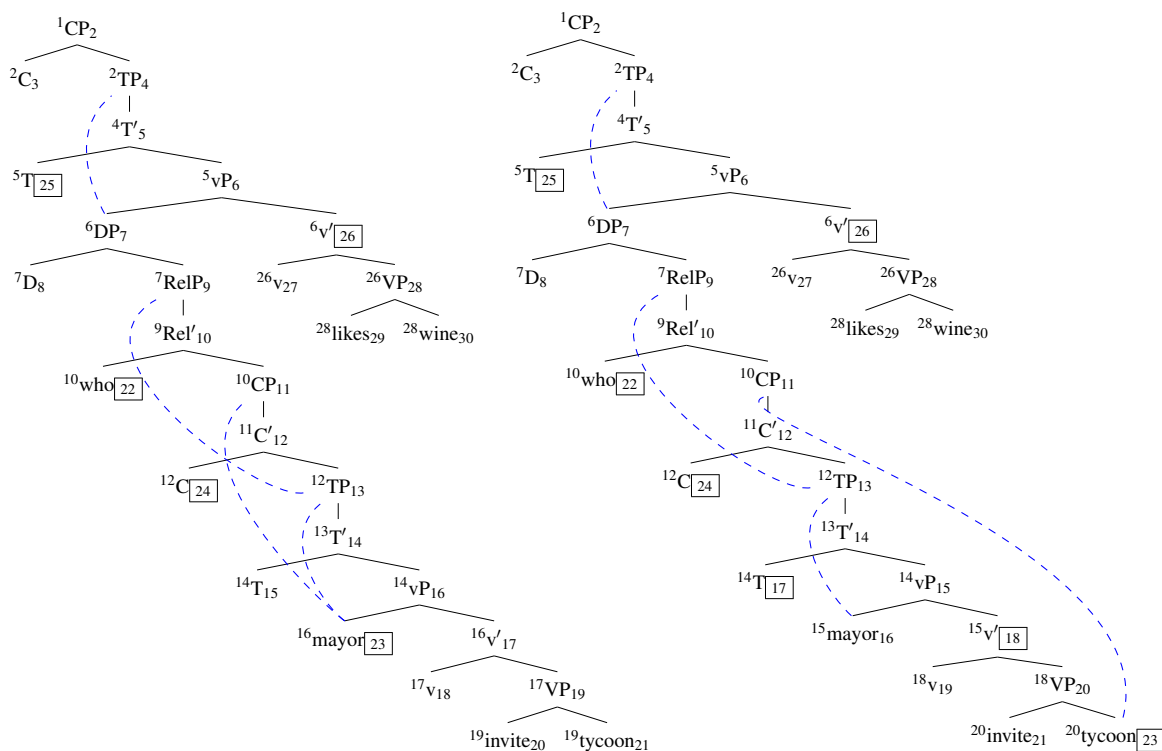


Figure 2: SRC and ORC in Chinese, promotion analysis

is ignored, then **Sum** also makes the wrong predictions for Chinese. This shows that all variants of **Sum** are completely unsuitable to account for the observed processing differences, corroborating previous findings by GM.

A more complicated picture emerges with pure payload, formalized as **Box**. Depending on the choice of analysis and which nodes count towards payload, **Box** predicts a preference for SRC, for ORC, or a tie. The unwanted preference for ORCs emerges I) with the wh-movement analysis in Korean if all nodes are taken into consideration, II) with both analyses in Korean if only LIs matter, and III) with both analyses in Korean and the wh-movement analysis in Chinese if only pronounced LIs are taken into account. The only defensible variant of **Box**, then, is the one that considers the full payload rather than its restriction to lexical or pronounced nodes. In combination with the promotion analysis, this predicts an SRC preference in Chinese and a tie in Korean.

Unfortunately, it has been shown by GM that **Box** fails to make a distinction in processing difficulty

for crossing and nested dependencies, the latter of which are harder to parse despite their reduced computational complexity (Bach et al., 1986). Unless the relative ease of crossing dependencies can be explained by some other mechanism, an MG parser with **Box** cannot model all the phenomena that were already accounted for in KGH and GM.

Crossing dependencies were actually one of KGH’s main arguments in support of **Max**— the maximum tenure among all nodes determines overall parsing difficulty — so if this metric fares just as well as **Box** for SRCs and ORCs, it is the preferable choice. Unfortunately, **Max** is ill-suited for the problem at hand. If one simply looks at the highest tenure value, **Max** predicts ties for SRCs and ORCs no matter which analysis or type of node is considered. If the metric is applied recursively such that derivation  $d$  is easier than  $d'$  iff they agree on the  $n$  highest tenure values and the  $n + 1$ -th value of  $d$  is lower than the  $n + 1$ -th value of  $d'$ , then **Max** predicts ORC preferences under all combinations. So recursive application of **Max** leads from universal ties to a universal ORC preference.

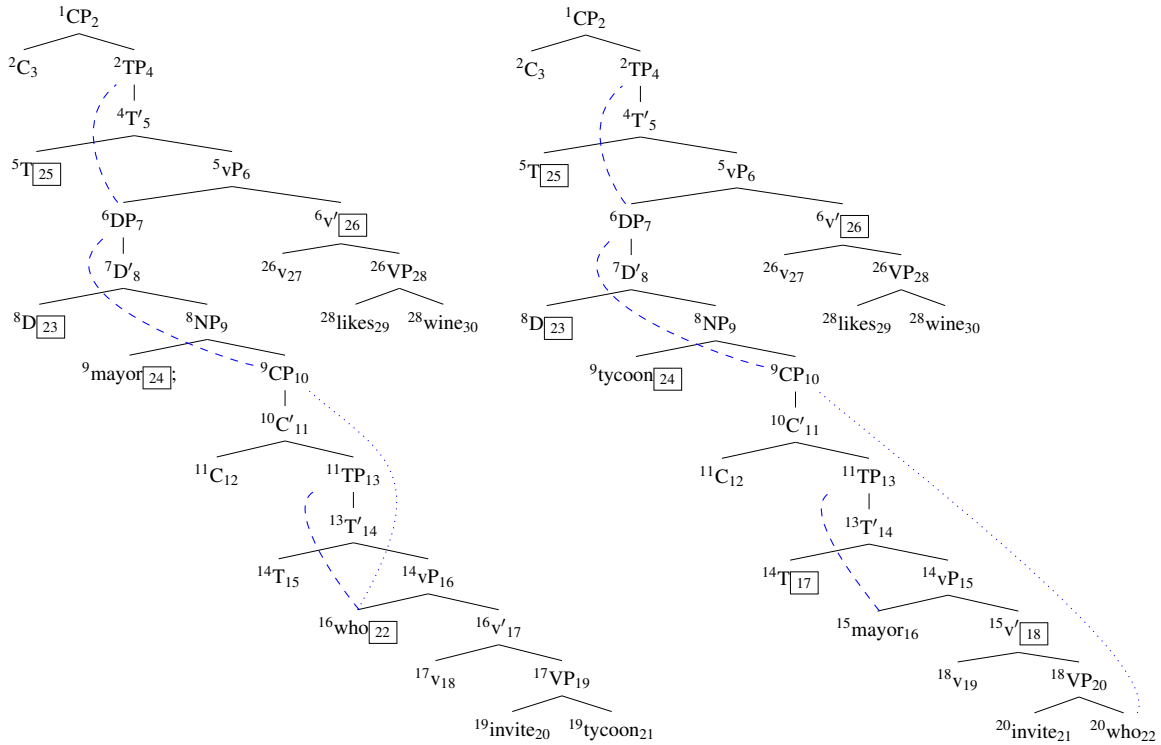


Figure 3: SRC and ORC in Chinese, wh-movement analysis

It seems, then, that we have a choice between an unrestricted version of **Box**, which works only with the promotion analysis and treats crossing and nested dependencies the same, and a non-recursive unrestricted version of **Max**, which treats prenominal SRCs and ORCs the same irrespective of the chosen analysis. Either metric needs to be supplemented by some additional principle to handle these cases. Recall, though, that **Box** predicts a tie for Korean under the promotion analysis. Furthermore, GM showed that the non-recursive version of **Max** is also unsuitable for postnominal RCs and fails to make a clear distinction between the easy case of a sentential complement containing an RC and the much harder case of an RC containing a sentential complement. So whatever additional principle one might propose, it must establish parsing preferences for a diverse range of phenomena.

### 4.3 A Refined Tenure Metric

On an intuitive level it is rather surprising that no metric grants a clear advantage to SRCs across the board. After all, SRC and ORC derivations differ only in the movement branch to the CP, which is

much longer for ORCs than for SRCs as subjects occupy a higher structural position than objects. Since all the metrics home in on some aspect of memory load, one would expect at least one of them to pick up on this difference. That this does not happen is due to the very nature of tenure.

A node has high tenure if its corresponding parse item enters the parser’s queue early but cannot be worked on for a long time. In the case of RCs, the complementizer (or alternatively the head noun in the wh-movement analysis) occupies a very high structural position, so that it is encountered early during the construction of the RC. At the same time, it cannot be removed from the queue until the full RC has been constructed, which means that the parser has to move all the way down to the verb and the object. But as long as the complementizer has not been removed from the queue, none of the nodes following it can be removed, either. The result is a “parsing bottle neck” that leads to high tenure on a large number of nodes. The difference between SRCs and ORCs has no effect because it does not change the need for the parser to build the entire RC



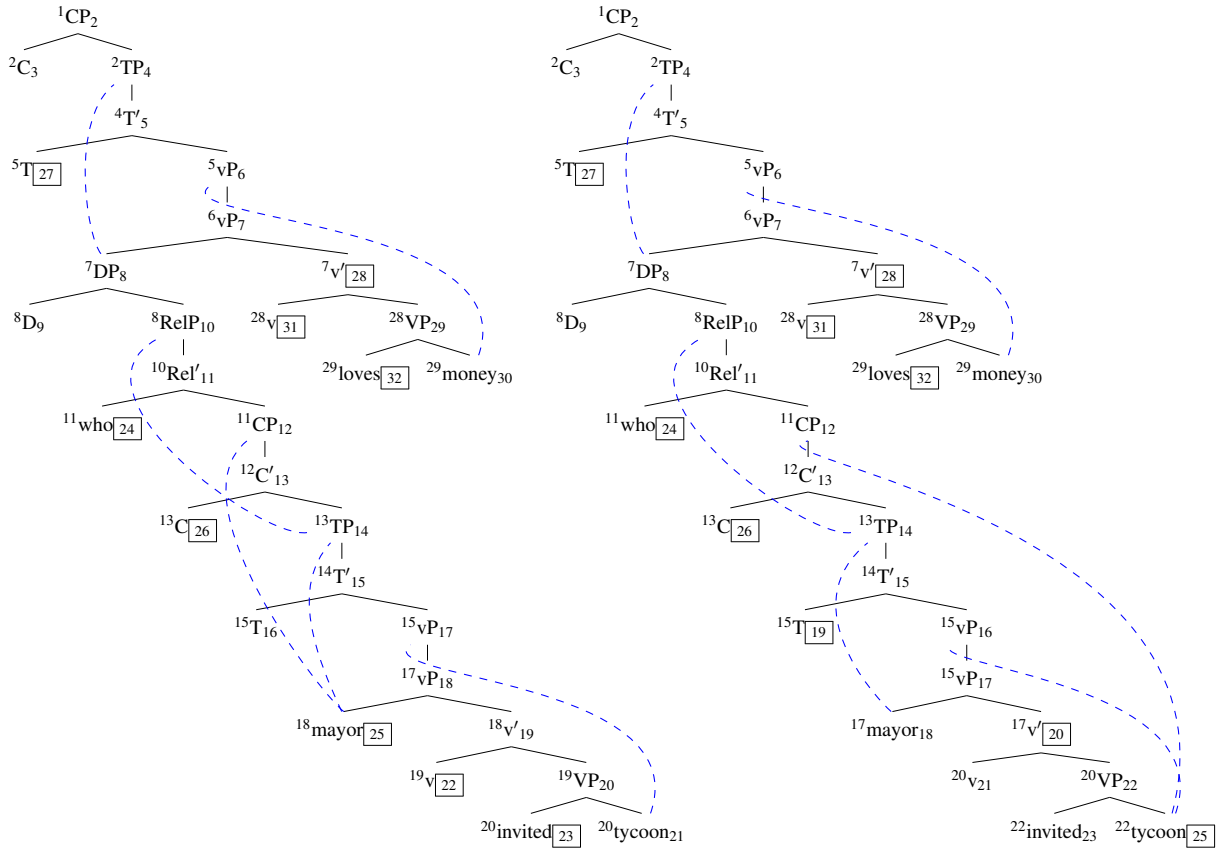


Figure 4: SRC and ORC in Korean, promotion analysis

before it can work on the complementizer, which is the actual cause for the bottle neck.

The central problem, then, is that the structural differences between SRC and ORC are too marginal to outweigh the effects of their shared structure on tenure. There are many conceivable ways around this, e.g. by combining payload and tenure so that each node's tenure from steps  $i$  to  $j$  is scaled relative to the overall payload from  $i$  to  $j$ . The most natural idea of multiplying tenure and payload leads to an ORC preference, but division seems to produce the correct results, even for the phenomena discussed in KGH and GM. However, such a step would take us away from the ideal of a simple metric. A less involved solution is to refine the granularity of tenure in a particular way.

Tenure measures how long a parse items remains in memory, but it does not take into account how much memory a given parse item consumes. Con-

sider the parse item corresponding to the embedded CP of the SRC derivation in Fig. 2 on page 7. The step from CP to  $C'$  corresponds to a specific inference rule in the parser that constructs the  $C'$  parse item from the one for CP by adding a movement feature  $f^-$  to the list of movers that still need to be found. From here on out,  $f^-$  has to be passed around from parse item to parse item until it is finally instantiated on the object. All the parse items along this path would have been smaller if they did not have to carry along  $f^-$  in the list of movers. Therefore movement dependencies increase memory load to the extent that they increase the size of parse items (and thus the number of bits that are required for the encoding of said items).

From this perspective, the processing difference between SRCs and ORCs is due to the fact that the longer movement branch in ORCs means that some parse items are bigger in the ORC than their SRC

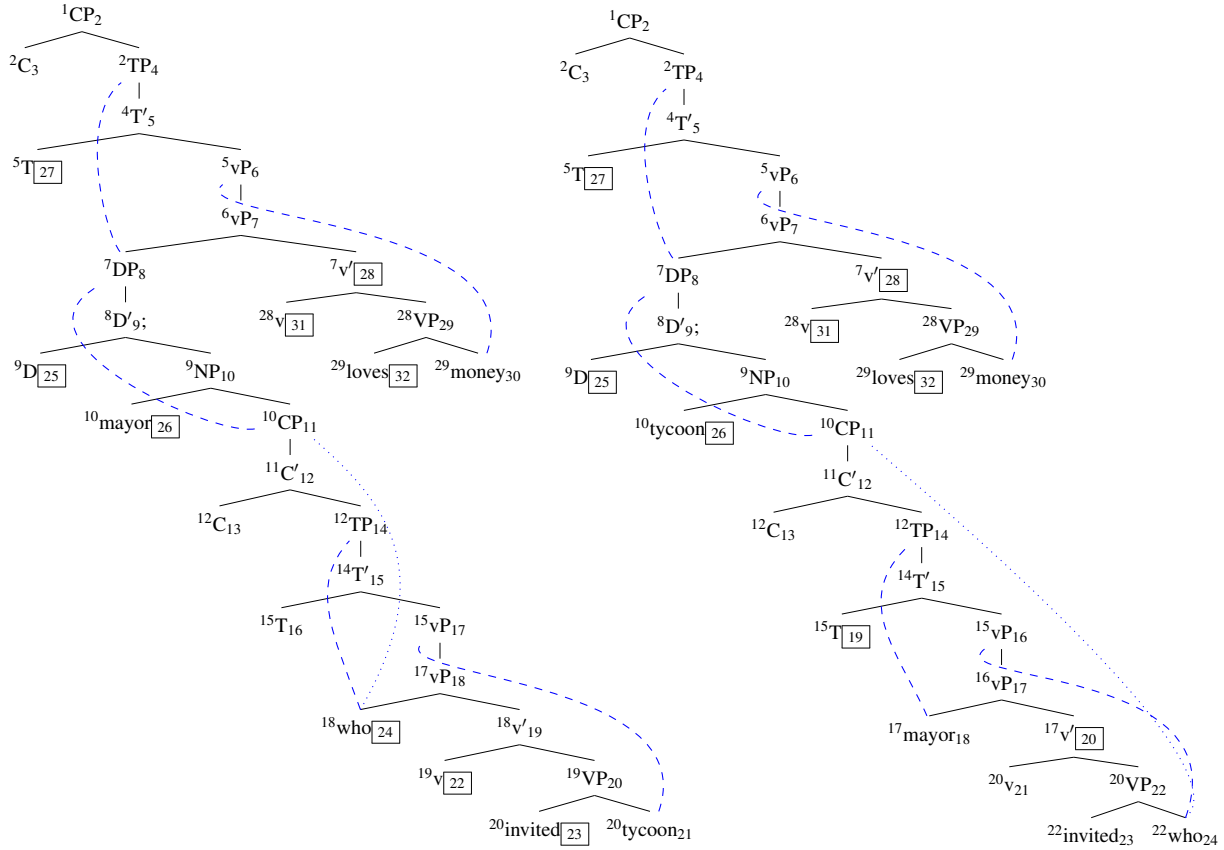


Figure 5: SRC and ORC in Korean, wh-movement analysis

counterparts. One must be careful, though, because only the features of final landing sites are passed along in this fashion — as defined in Stabler (2013), the parser handles the features of intermediary landing sites without increased memory usage. Once one controls for the fact that some final landing sites in the SRC are intermediate in the ORC, and the other way round, there still remains a small advantage for the SRC even in Korean. In both the SRC and the ORC in Figs. 4 and 5, all the interior nodes inside the embedded CP have to pass along at least one feature. More precisely,  $C'$ , TP,  $v'$  and VP pass along exactly one feature, while both  $v$ Ps carry exactly two features. Only  $T'$  shows a difference: in the SRC it hosts only the negative feature that triggers movement of the subject, whereas in the ORC it must also pass along the feature for the object.

This comparison is rather involved, but it can be approximated via the index-based metric **Gap** (in-

spired by filler-gap dependencies), where  $i_p$  is the index of moving phrase  $p$  and  $f_p$  the index of the final landing site:

$$\mathbf{Gap} = \sum_p \text{a moving phrase } f_p - i_p$$

Both **Box** and non-recursive **Max** as discussed above now make the right predictions in conjunction with **Gap** as a secondary metric to resolve ties (this includes also the constructions investigated in KGH and GM). Such a system will grant an advantage to SRCs as long as subjects occur in a higher position than objects. Consequently, it argues against proposals where subjects start out lower than objects (Sigurðsson, 2006). **Box** furthermore favors the promotion analysis over wh-movement, while **Max** remains agnostic.

## Conclusion

We showed that the MG parser does not make the right predictions for prenominal SRCs and ORCs

Language	Analysis	RC Type	Node Type	Node	Index	Outdex	Tenure		
Chinese	Promotion	SRC	pronounced	who	10	22	12		
				mayor	16	23	7		
			lexical	matrix T	5	25	<i>20</i>		
				C	12	24	12		
			interior	matrix $v'$	6	26	<i>20</i>		
	<i>Summed tenure:</i>					19	51	71	
		ORC		pronounced	who	10	22	12	
					mayor	20	23	3	
				lexical	matrix T	5	25	<i>20</i>	
					C	12	24	12	
				interior	embedded T	14	17	3	
		matrix $v'$	6		26	<i>20</i>			
		embedded $v'$	15		18	3			
		<i>Summed tenure:</i>					15	50	73
			Wh	SRC	pronounced	mayor	9	24	15
who						16	22	6	
lexical	matrix T				5	25	<i>20</i>		
	D				8	23	15		
interior	matrix $v'$				5	25	<i>20</i>		
<i>Summed tenure:</i>					21	56	76		
	ORC			pronounced	mayor	9	24	15	
					matrix T	5	25	<i>20</i>	
				lexical	D	8	23	15	
					embedded T	14	17	3	
		interior		matrix $v'$	6	26	<i>20</i>		
	embedded $v'$		15	18	3				
	<i>Summed tenure:</i>					15	53	76	

Table 1: Tenure of nodes for Chinese, grouped by analysis; maximum tenure values are in *italics*

under any of the tree-geometric metrics that have been proposed in the literature so far. However, the observed processing effects can be explained if one also take the memory requirements of movement dependencies into account, formalized via the metric **Gap**. The next step will be to test this hypothesis against recent data from Basque (Carreiras et al., 2010), where a uniform preference for ORCs has been observed. Basque is an ergative language, for which it has been argued that subject and object might occur in different positions. If so, the observed behavior may fall out naturally from slightly different movement patterns and their effect on the size of parse items.

A more pressing concern, though, is the mathematical investigation of the parser — a sentiment that is also expressed by KGH. The current method of testing various metrics against numerous con-

structions is essential for mapping out the space of empirically pertinent alternatives, but it is needlessly labor intensive due to the usual pitfalls of combinatorial explosion. Nor does it enjoy the elegance and generality of a proof-based approach. We believe that true progress in this area hinges on a sophisticated understanding of the tree traversal algorithm instantiated by the parser and how exactly this tree traversal interacts with specific metrics to prefer particular tree shapes over others. Our insistence on simple metrics, free from complicating aspects like probabilities, stems from this desire to keep the parser as open to future mathematical inquiry as possible.

## Acknowledgments

We are greatly indebted to John Drury, Jiwon Yun, and the three anonymous reviewers for their com-

Language	Analysis	RC Type	Node Type	Node	Index	Outdex	Tenure			
Korean	Promotion	SRC	pronounced	who	11	24	13			
				tycoon	18	25	7			
				invited	20	23	3			
				loves	29	32	3			
			lexical	matrix T	5	27	22			
				C	13	26	13			
				embedded $v$	19	22	3			
				matrix $v$	28	31	3			
			interior	matrix $v'$	7	28	21			
				<i>Summed tenure:</i>			26	67	88	
					ORC	pronounced	who	11	24	13
							mayor	22	25	3
							loves	29	32	3
						lexical	matrix T	5	27	22
C	13	26					13			
embedded T	15	19					4			
matrix $v$	28	31					3			
interior	embedded $v'$	17				20	3			
	matrix $v'$	7				28	21			
	<i>Summed tenure:</i>					19	61	85		
Wh	SRC					pronounced	tycoon	10	26	16
							who	18	24	6
							loves	28	31	3
							invited	20	23	3
			lexical	matrix T	5	27	22			
				D	9	25	16			
				embedded $v$	19	22	3			
				matrix $v$	27	30	3			
			interior	matrix $v'$	7	28	21			
				<i>Summed tenure:</i>			28	72	93	
					ORC	pronounced	tycoon	10	26	16
							loves	29	32	3
							matrix T	5	27	22
						lexical	D	9	25	16
embedded T	15	19					4			
matrix $v$	28	31					3			
interior	embedded $v'$	17					20	3		
interior	matrix $v'$	7				28	21			
	<i>Summed tenure:</i>					19	64	88		

Table 2: Tenure of nodes for Korean, grouped by analysis; maximum tenure values are in *italics*

ments and remarks that allowed us to streamline essential parts of this work and improve the presentation of the material.

## References

- Emmon Bach, Colin Brown, and William Marslen-Wilson. 1986. Crossed and nested dependencies in German and Dutch: A psycholinguistic study. *Language and Cognitive Processes*, 1:249–262.
- Manuel Carreiras, Jon Andoni Duñabeitia, Marta Vergara, Irene de la Cruz-Pavía, and Itziar Laka. 2010. Subject relative clauses are not universally easier to process: Evidence from Basque. *Cognition*, 115:79–92.
- Noam Chomsky. 1965. *Aspects of the Theory of Syntax*. MIT Press, Cambridge, Mass.
- Lyn Frazier. 1987. Sentence processing: A tutorial review.
- Edward Gibson and Neal J. Pearlmutter. 1998. Constraints on sentence comprehension. *Trends in Cognitive Sciences*, 2(7):262–268.
- Edward Gibson and H.-H. Iris Wu. 2013. Processing Chinese relative clauses in context. *Language and Cognitive Processes*, 28(1-2):125–155.
- Edward Gibson. 1998. Linguistic complexity: Locality of syntactic dependencies. *Cognition*, 68:1–76.
- Peter C. Gordon, Randall Hendrick, Marcus Johnson, and Yoonhyoung Lee. 2006. Similarity-based interference during language comprehension: Evidence from eye tracking during reading. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 32(6):1304.
- Thomas Graf and Bradley Marcinek. 2014. Evaluating evaluation metrics for minimalist parsing. In *Proceedings of the 2014 ACL Workshop on Cognitive Modeling and Computational Linguistics*, pages 28–36.
- Thomas Graf. 2012. Locality and the complexity of minimalist derivation tree languages. In Philippe de Groot and Mark-Jan Nederhof, editors, *Formal Grammar 2010/2011*, volume 7395 of *Lecture Notes in Computer Science*, pages 208–227, Heidelberg. Springer.
- John Hale. 2011. What a rational parser would do. *Cognitive Science*, 35:399–443.
- Henk Harkema. 2001. A characterization of minimalist languages. In Philippe de Groote, Glyn Morrill, and Christian Retoré, editors, *Logical Aspects of Computational Linguistics (LACL'01)*, volume 2099 of *Lecture Notes in Artificial Intelligence*, pages 193–211. Springer, Berlin.
- Irene Heim and Angelika Kratzer. 1998. *Semantics in Generative Grammar*. Blackwell, Oxford.
- Aravind Joshi. 1990. Processing crossed and nested dependencies: An automaton perspective on the psycholinguistic results. *Language and Cognitive Processes*, 5:1–27.
- Richard S. Kayne. 1994. *The Antisymmetry of Syntax*. MIT Press, Cambridge, Mass.
- Edward L. Keenan and Bernard Comrie. 1977. Noun phrase accessibility and universal grammar. *Linguistic Inquiry*, 8:63–99.
- Gregory M. Kobele, Christian Retoré, and Sylvain Salvati. 2007. An automata-theoretic approach to minimalism. In James Rogers and Stephan Kepser, editors, *Model Theoretic Syntax at 10*, pages 71–80.
- Gregory M. Kobele, Sabrina Gerth, and John T. Hale. 2012. Memory resource allocation in top-down minimalist parsing. In *Proceedings of Formal Grammar 2012*.
- Nayoung Kwon, Maria Polinsky, and Robert Kluender. 2006. Subject preference in Korean. In *Proceedings of the 25th West Coast Conference on Formal Linguistics*, pages 1–14. Cascadilla Proceedings Project Somerville, MA.
- Nayoung Kwon, Peter C. Gordon, Yoonhyoung Lee, Robert Kluender, and Maria Polinsky. 2010. Cognitive and linguistic factors affecting subject/object asymmetry: An eye-tracking study of prenominal relative clauses in Korean. *Language*, 86(3):546–582.
- Chien-Jer Charles Lin and Thomas G. Bever. 2006. Subject preference in the processing of relative clauses in Chinese. In *Proceedings of the 25th West Coast Conference on Formal Linguistics*, pages 254–260. Cascadilla Proceedings Project Somerville, MA.
- Willem M. Mak, Wietske Vonk, and Herbert Schriefers. 2002. The influence of animacy on relative clause processing. *Journal of Memory and Language*, 47(1):50–68.
- Willem M. Mak, Wietske Vonk, and Herbert Schriefers. 2006. Animacy in processing relative clauses: The hikers that rocks crush. *Journal of Memory and Language*, 54(4):466–490.
- Axel Mecklinger, Herbert Schriefers, Karsten Steinhauer, and Angela D. Friederici. 1995. Processing relative clauses varying on syntactic and semantic dimensions: An analysis with event-related potentials. *Memory & Cognition*, 23(4):477–494.
- Jens Michaelis. 2001. Transforming linear context-free rewriting systems into minimalist grammars. *Lecture Notes in Artificial Intelligence*, 2099:228–244.
- Edson T. Miyamoto and Michiko Nakamura. 2003. Subject/object asymmetries in the processing of relative clauses in Japanese. In *Proceedings of WCCFL*, volume 22, pages 342–355.
- Edson T. Miyamoto and Michiko Nakamura. 2013. Unmet expectations in the comprehension of relative clauses in Japanese. In *Proceedings of the 35th Annual Meeting of the Cognitive Science Society*.
- Owen Rambow and Aravind Joshi. 1995. A processing model for free word order languages. Technical Report IRCS-95-13, University of Pennsylvania.

- Sylvain Salvati. 2011. Minimalist grammars in the light of logic. In Sylvain Pogodalla, Myriam Quatrini, and Christian Retoré, editors, *Logic and Grammar — Essays Dedicated to Alain Lecomte on the Occasion of His 60th Birthday*, number 6700 in Lecture Notes in Computer Science, pages 81–117. Springer, Berlin.
- Halldór Ármann Sigurðsson. 2006. The nominative puzzle and the low nominative hypothesis. *Linguistic Inquiry*, 37:289–308.
- Edward P. Stabler. 1997. Derivational minimalism. In Christian Retoré, editor, *Logical Aspects of Computational Linguistics*, volume 1328 of *Lecture Notes in Computer Science*, pages 68–95. Springer, Berlin.
- Edward P. Stabler. 2011. Computational perspectives on minimalism. In Cedric Boeckx, editor, *Oxford Handbook of Linguistic Minimalism*, pages 617–643. Oxford University Press, Oxford.
- Edward P. Stabler. 2013. Two models of minimalist, incremental syntactic analysis. *Topics in Cognitive Science*, 5:611–633.
- Mark Steedman. 2001. *The Syntactic Process*. MIT Press, Cambridge, Mass.
- James W. Thatcher. 1967. Characterizing derivation trees for context-free grammars through a generalization of finite automata theory. *Journal of Computer and System Sciences*, 1:317–322.
- Mieko Ueno and Susan M Garnsey. 2008. An ERP study of the processing of subject and object relative clauses in Japanese. *Language and Cognitive Processes*, 23(5):646–688.
- Jean-Roger Vergnaud. 1974. *French Relative Clauses*. Ph.D. thesis, MIT.
- Jiwon Yun, Zhong Chen, Tim Hunter, John Whitman, and John Hale. 2014. Uncertainty in processing relative clauses across East Asian languages. *Journal of East Asian Linguistics*, pages 1–36.