WORD ORDER SCRAMBLING AS A CONSEQUENCE OF INCREMENTAL SENTENCE PRODUCTION

Gerard Kempen
Psychology Department, Leiden University
Max-Planck-Institute for Psycholinguistics, Nijmegen
The Netherlands

&

Karin Harbusch
Computer Science Department
University of Koblenz-Landau
Germany

October 2002
To be published in Haertl, H., & Tappe, H. (Eds.), Mediating between concepts and language — Processing structures. Berlin: Mouton De Gruyter.

1. Introduction

The incremental nature of human grammatical encoding is an important source of word order variation even in languages with relatively strict word order. A well-known example is the ‘heavy constituent shift’ that moves a long phrase, wholesale or in part, to a clause-final position. It seems reasonable to assume that longer constituents on average need more assembly time than shorter ones. An incremental grammatical encoder can make for greater output fluency by assigning a heavy exemplar of such a constituent a late position in the utterance. Conversely, output fluency is promoted also when the encoder allows a light constituent to occupy an early position, e.g. a pronoun referring to the discourse topic. Stated more generally: Given a language with some word order flexibility and an incremental grammatical encoder, constituents whose shape is determined at an earlier point in time will tend to precede constituents that ‘arrive’ later. And the correlation between the arrival time of a constituent (the moment its shape is fixated) and its linear position will tend to be stronger the greater the amount of flexibility. Arrival time may depend not only on syntactic factors such as the complexity of the syntactic assembly process but also on lexical factors (e.g., different retrieval times for low- and high-frequency words) and on semantic factors (e.g., salient fragments of the to-be-expressed meaning being conceptualized prior to less salient ones; see Yamashita & Chang (2001)).

Now suppose that, for some pattern of word order variation in a (semi-)free word order language, there is independent evidence that the actual order of constituents covaries directly with their arrival times. This would eliminate the need for syntactic rules that explicitly control the order of the constituents involved — thereby benefiting theoretical parsimony.

In this chapter we test the viability of this approach by developing a statistical model that generates a detailed pattern of constituent order variation on the basis of the hypothetical arrival times of the constituents. Our test case is the well-known ‘scrambling’ phenomenon of Subject (S), Indirect Object (I) and Direct Object NP (O) in the ‘Mittelfeld’ (Midfield) of German clauses. None of the six possible permuta-
tions of these NPs are definitely ruled out, although the grammaticality (acceptability) ratings they elicit tend to vary widely. We show that simple model assumptions about the typical order of arrival of S, I and O yield accurate estimates of the acceptability of the permutations. Actually, the predictions of our model are at least as accurate as those derived from models based on ranked or weighted ordering constraints, e.g. within an Optimality Theoretical framework. We conclude that our incrementality-based approach presents a viable alternative to current approaches in terms of a hierarchy of ordering constraints, and that satisfactory accounts of the scrambling phenomenon under scrutiny may well be found outside the domain of explicit syntactic ordering rules.

Modeling ‘word order freedom in restraint’ is relevant not only to psycholinguistic theories of grammatical encoding but also to the design of computational sentence generators striving for natural and varied output (e.g., in a computer-supported language training environment). Such systems should neither select the same permutation at all times, nor produce the various grammatical permutations on a strictly random basis. Instead, they must be sensitive to empirical data that reflect speaker ordering habits and preferences, and select permutations accordingly.

In the grammatical encoding model presented below, we apply Performance Grammar (PG), a psycholinguistically motivated grammar formalism which contains separate components generating, respectively, the hierarchical and the linear structure of sentences (Kempen & Harbusch 2002; Harbusch & Kempen, 2002). In Section 2 we outline the psycholinguistic motivation underlying this distinction. Sections 3 and 4 present the essentials of PG’s hierarchical and linearization components. Section 5 introduces the experimentally obtained acceptability ratings for permutations of Subject, Direct Object and Indirect Object NPs in the Midfield of German clauses. In Section 6 we account for these ratings in terms of a probabilistic model. Section 7, finally, summarizes our approach and the conclusions we reached.

2. A psycholinguistic argument for topology-based linearization

In an incremental grammatical encoder, word order is affected by the ‘order of arrival’ of the various syntactic constituents. Constituents that become available for being ordered at an earlier point in time, can occupy an earlier position in the surface string than constituents emerging later — as long as grammar rules do not intervene. Order
of arrival is controlled by several groups of factors: by pragmatic and conceptual factors residing in a ‘conceptualizer’ component and/or in the ‘semantic-syntactic’ interface; and by lexical and syntactic factors in the grammatical encoder (‘formulator’) itself. For instance, ‘old’ information can be conceptualized more easily than ‘new’ information and therefore tends be available for being linearized earlier. A word-finding problem may delay the formulation of a conceptual fragment; ‘heavy’ constituents take more assembly time than ‘light’ constituents. Linearization methods for (semi-)free word order languages should be sensitive to order of arrival of syntactic constituents.

Furthermore, we consider that linearization methods should comply with general psychological properties of sequence generation processes. Models of serial order are conveniently divided into two types depending on whether or not they aim at generating novel sequences that have never been seen before (Dell, Burger & Svec (1997)). ‘Closed’ models only deal with a restricted set of sequences. ‘Open’ models can handle potentially unlimited sets of sequences, including novel ones. In open models, the representation of a to-be-generated sequence is typically split into two parts: (1) a one-dimensional array containing a number of empty slots, and (2) the set of items to be inserted into the slots. Items are ordered by binding them to a slot. Generating a sequence involves traversing the array from beginning to end, at each slot reading out any item(s) bound to it.

Open sequence generation models are informally called slot-and-filler models. Already in 1975, Garrett proposed a slot-and-filler model for syntactic speech errors that involve misorderings of words or phrases (as in Although murder is a form of suicide). He had discovered that the likelihood of such errors is independent of the distance between the permuted elements in the surface string, whereas other types of exchanges (e.g. between phonemes) occur predominantly between elements in neighboring positions. He took this contrast as evidence for two distinct levels of processing: a ‘functional’ level where grammatical relationships between constituents are established, vs. a ‘positional’ level where constituents are ordered from left to right. This distinction has been adopted, in one way or another, by many students of grammatical encoding (e.g. Kempen & Hoenkamp (1987), Levelt (1989), Bock

---

1 For recent applications of slot-and-filler models to the linear ordering of, respectively, phonemes in syllables and morphemes in words, see Hartsuiker (2002) and Janssen, Roelofs & Levelt (2002).
Levelt (1994) and Kempen & Harbusch (2002)). In Kempen & Hoenkamp’s version, the constituents of unordered functional structures acquired their linear position by binding themselves to a slot in a one-dimensional array called ‘holder’. In certain linguistic frameworks, such holders are called topologies (see Kathol (2000, Ch. 3) for the history of this term).

Before turning to our proposal for a topology-based linearization model, we need to explain the essential features of PG’s hierarchical grammatical structures.

3. Hierarchical structures in Performance Grammar

PG’s hierarchical component generates unordered trees by combining 3-tiered ‘mobiles’ called lexical frames. Figure 1 shows the eight lexical frames corresponding to the words of example (1).

(1) Denkst du, dass er das Auto repariert hat?

Think you that he the car repaired has

‘Do you think that he has repaired the car?’

Figure 1. Simplified lexical frames underlying the eight words of sentence (1). Left-to-right order of branches is arbitrary. The basic shape of lexical frames is retrievable from the Mental Lexicon in response to contents of the to-be-expressed conceptual message; however, certain branches (e.g. CMPR) are added as a consequence of local syntactic constraints (CMPR = CoMPlementizeR; CP = Complementizer Phrase). Note that auxiliary verbs have their own verb frame.

The top layer of a frame consists of a single phrasal node (the ‘root’; e.g. S, NP, DP, CP), which is connected to one or more functional nodes in the second layer (e.g., Subject, Head, Direct Object, CoMplement, MODifier). At most one exemplar of a functional node is allowed in the same frame, except for MOD nodes, which may occur several times (including zero). Every functional node dominates exactly one phrasal
node (‘foot’) in the third layer, except for HeaD which immediately dominates a lexical (part of speech) node. Each lexical frame is ‘anchored’ to exactly one lexical item (printed below the lexical node serving as the frame’s HeaD).

Categorial nodes (i.e. lexical and phrasal nodes in the first and third layers of a lexical frame) have associated with them a feature matrix, i.e., a list of pairs that consist of an attribute and a finite set of values. Features are instantiated with a finite, non-empty value set. An attribute is a character string (e.g., "gender", "person", "number"). A value set contains a finite, non-zero number of character strings (e.g., \{sing\}, \{1\textsuperscript{st}, 2\textsuperscript{nd}, 3\textsuperscript{rd}\}), each representing a possible value of the attribute (disjunctive value sets). Lexical frames are combined to form larger mobiles by a non-recursive form of unification, called feature unification. This operation yields a merger of the root node of one frame with one foot node of another frame. As illustrated in Figure 2, unification also serves to select the value of agreement features (e.g. entailing second person singular \textit{denkst} of the verb \textit{denken}).

![Figure 2. Verb frame hierarchy underlying example (1). The root S-node of the verb frame associated with \textit{hat} and the CoMPlEment S-node of \textit{denkst} have merged as a result of unification, and so have the S-CMP of \textit{hat} and the root S-node dominating \textit{repariert}. (In this paper we do not discuss unification of nodes other than S.) Left-to-right order of branches is arbitrary. Feature matrices associated with categorial nodes have been suppressed for reasons of space.](image)

4. Linear structure in PG

In order to assign a left-to-right position to the branches of lexical frames, we introduce an additional type of data structure. Associated with every lexical frame is a one-dimensional topology specifying a fixed number of positions (or slots, landing sites) for its constituents. The topology of a verb frame (i.e., of a finite or non-finite
clause) allocates space for each of various grammatical functions that can be fulfilled by its constituents, e.g., to the HeaD verb, to the SUBject NP, the Direct OBJect NP, etc. The topology that we use for German clauses specifies nine different slots, labeled as indicated in Figure 3.

Constituents may be assigned different positions depending on their shape. For instance, if the Direct OBJect role is fulfilled by a Wh-phrase, it will end up in the ‘Forefield’ of the clause rather than in the ‘Midfield’. On the basis of empirical-linguistic arguments (not to be discuss here), we propose that S-topologies of German contain exactly nine slots, as depicted in Figure 3.

Table 1 illustrates which clause constituents select which slots as their ‘landing site’. Notice, in particular, that the placement conditions refer not only to the grammatical function fulfilled by a constituent but also to its shape. For instance, a CoMPlement clause will land in slot E2 if it is finite; non-finite CoMPlements have several additional placement options.
Table 1. Examples of topology slot fillers (German clauses). Precedence between constituents landing in the same slot is marked by "<<". The selection of fillers for slots M2, M3 and M4 is discussed in Section 5. For details of the landing sites of CoMPlEment clauses, see Kempen & Harbusch (forthcoming).

<table>
<thead>
<tr>
<th>Slot</th>
<th>Filler</th>
</tr>
</thead>
</table>
| F1   | *Declarative main clause*: Subject, Topic or Focus (one constituent only)  
*Interrogative clause*: Wh-constituent, including ob 'whether'  
*Complement clause*: Wh-constituent |
| M1   | *Main clause*: Head verb  
*Complement clause*: CoMPlEmentizeR dass 'that' |
| M2   | One of: Subject (S), Indirect Object (I), Direct Object (O) NP (iff non-Wh) |
| M3   | One of S, I, O (iff non-Wh) |
| M4   | One of S, I, O (iff non-Wh) |
| M5   | Non-finite CoMPlEment of Verb Raiser |
| M6   | *Subordinate clause*: PaRticle << Pre-Infinitive zu 'to' << Head verb |
| E1   | Non-finite CoMPlEment of Verb Raiser |
| E2   | Non-finite CoMPlEment of VP Extraposition verb  
Finite Complement clause |

We show linearization at work on an abbreviated version of example (1). As we assume that every lexical item launches its own lexical frame and topology, a sentence containing more than one verb instantiates several clausal topologies. This applies to verbs of any type, whether main, auxiliary or copula. It follows that sentence (1’) needs two topologies (Figure 4).

(1’) *Denkst du, dass er das Auto repariert?*
*Do you think that he repairs the car?*

<table>
<thead>
<tr>
<th>F1</th>
<th>M1</th>
<th>M2</th>
<th>M3</th>
<th>M4</th>
<th>M5</th>
<th>M6</th>
<th>B1</th>
<th>B2</th>
</tr>
</thead>
<tbody>
<tr>
<td>denkst</td>
<td>du</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>●</td>
<td></td>
</tr>
</tbody>
</table>

Figure 4. Linearization of example (1’).
Slot F1 of the interrogative main clause remains empty in the absence of a Wh-
constituent. The pronominal Subject NPs of the main and the subordinate clause go
to slot M2 of their respective topologies for reasons to be explained in Section 5. The
HeaD verb selects slot M1 in the main clause, and M6 in the CoMplement. The CoM-
PlementizeR dass goes to M1 of the CoMplement’s topology. Finally, the root S-node
dominating the finite CoMplement clause lands in E2, as indicated by the bullet and
upward double arrow.

In case of more than one verb, the topologies associated with them are allowed to
share certain identically labeled slots, conditionally upon several restrictions. After
two slots have been shared, they are no longer distinguishable; in fact, they are the same ob-
ject. In example (1'), the embedded topology shares its F1 slot with the F1 slot of the
matrix. This is indicated by the dashed border of the bottom F1 slot.

Now consider example (2), where the Direct OBJect of repariert is a Wh-phrase
(welches Auto ‘which car’) that has landed in the shared F1 slot (Figure 5).

(2) Welches Auto denkst du, dass er repariert?

‘Which car do you think that he repairs?’

As a consequence, the Wh-constituent gets ‘fronted’ and seems to have been ‘ex-
tracted’ from the complement. Topology sharing manifests itself as upward movement
of constituents in shared slots (single arrow). We call this effect promotion.

Example (1) embodies a more radical case of topology sharing. The topologies as-
associated with hat and repariert share the entire region extending from F1 through M4.
The result is promotion of Direct OBJect das Auto, as shown in Figure 6. We refer to
Harbusch & Kempen (2002) for a detailed description of a variety of German word
order phenomena that can be explained by ‘left-peripheral topology sharing’.
During sentence generation, the overt constituent order of a sentence is determined by a Read-out module that traverses the hierarchy of topologies in depth-first, left-to-right manner, scanning shared slots only once. Any lexical item it ‘sees’ in a slot, is appended to the output string. E.g., welches Auto in example (2) is seen while the Reader scans the matrix topology rather than during its traversal of the embedded topology.

Now that the essentials of PG’s linear ordering component are in place, we can return to the word order phenomena sketched in the Introduction.

5. Semi-free word order in the Midfield of German clauses

Several experimental investigations into the acceptability of word order variation in the Midfield of subordinate clauses of German have recently been published in the psycholinguistic literature (Pechmann, Uszkoreit, Engelkamp & Zerbst (1994, 1996); Rösler, Pechmann, Streb, Röder & Hennighausen (2000) and Keller (2000a, b)). The data patterns emerging from these experiments are very similar. While none of the six possible permutations of Subject (S), Indirect Object (I) and Direct Object (O) are definitely ruled out, some are judged considerably more acceptable than others. Furthermore, the acceptability varies in function of whether the constituents are full NPs or pronominal NPs.

Theoretical accounts for the obtained data patterns usually employ a ranked or weighted set of Linear Precedence (LP) constraints (see Pechmann et al. (1994, 1996), Müller (1999) and Keller (2000a, b)). A typical example is given in (3). The symbol "<<" denotes linear precedence, and the constraints are listed in order of decreasing rank/weight.
Keller’s experiments yield numerical weight values for the constraints. These confirm, in line with the Optimality Theoretical framework\textsuperscript{2} he adopts, that

- violation of a higher ranked/weighted constraint reduces the acceptability ratings more seriously than violation of lower ones, and
- multiple violations affect the ratings additively.

Although models based on ranked/weighted LP constraints are descriptively adequate, they are not necessarily psychologically plausible. There is no evidence that the linearization system actually applies constraints like those in (3) at all. Actually, the phenomenon of incremental grammatical encoding suggests an alternative without explicit syntactic ordering rules for constituents whose position is flexible.

Consider the slots M2, M3 and M4 in Table 1. Each of these accepts one constituent of type S, I and O. We assume their surface order depends on their order of arrival, that is, their position in the queue of constituents ready to be ordered. Now let us assume that — due to the combined effect of pragmatic, conceptual, lexical, and syntactic processing factors — Subject NPs are likely to precede Indirect Objects in the queue and, likewise, that I usually precedes O. That is:

\[ p(S<<I) > .5 \quad \text{and} \quad p(I<<O) > .5 \]

implying that the surface order SIO will occur much more frequently than any other order. This suggests we can predict the actual probability of occurrence of the six possible orderings of S, I and O, that is, of their ‘permutation probabilities’, if we know the ‘pairwise precedence probabilities’ of these constituent types in the queue: \( p(S<<I) \), \( p(I<<O) \) and \( p(S<<O) \). Assume, furthermore, that acceptability judgments about overt S, I and O permutations mirror the precedence probabilities during incremental sentence generation. More specifically, the acceptability rating of a sequence is supposed to be a multiplicative function of the precedence probabilities of the constituent pairs that make up the sequence. In order to estimate the latter probabilities we need to convert acceptability ratings of S+I+O permutations to (quasi-)probabilities and to devise a model that maps hypothetical pairwise prece-

\textsuperscript{2}For an overview of Optimality Theory, see Dekkers, Van der Leeuw & Van Weijer (2000).

\[(3)\] (A) \(+\text{NOM}\) \(<<\) \(-\text{NOM}\) (Subject NP precedes other NPs)
(B) \(+\text{PRO}\) \(<<\) \(-\text{PRO}\) (pronominal NPs precede full NPs)
(C) \(+\text{DAT}\) \(<<\) \(+\text{ACC}\) (Indirect Object precedes Direct Object)
dence probabilities onto (quasi-) probabilities of $S+I+O$ permutations. We decided to apply this method to the acceptability ratings published recently by Keller (2000a) for this purpose.

In his Experiments 6 and 10, Keller applied the psychophysical method of magnitude estimation to elicit particularly fine-graded grammaticality judgments (see Bard, Robertson & Sorace (1996) for details). The sentence material Keller prepared for his Experiment 6 is illustrated in (4). All NPs refer to human protagonists and are introduced by case-marked articles. The sentences were presented without context.

(4) Ich glaube, dass der Produzent dem Direktor den Schauspieler vorschlägt
I believe that the-NOM producer the-DAT director the-ACC actor proposes
'I believe that the producer proposes the director the actor'

The material for Experiment 6 also included sentences with two full and one pronominal NP in nominative, dative or accusative case ($er$, $ihn$, $ihm$). So, every sentence like (4) was presented in 24 different shapes: 6 (permutations of $S$, $I$ and $O$) times 4 (NP versions: 1 full, 3 pronominal). For all 24 data points, Table 2 presents the acceptability ratings transformed into quasi-probabilities.

---

1 Following Keller (o.c.) and Bard et al. (o.c.), we use the terms ‘acceptability’ and ‘grammaticality’ interchangeably, considering both as referring to a graded rather than a discrete (binary) notion.

Keller’s acceptability data show that, overall, subordinate clauses with a full Subject NP in leading position (although after the subordinating conjunction) receive considerably higher acceptability ratings than those where the first full NP is Direct or Indirect Object. The latter two are roughly equally acceptable. This seems to be in contrast with recently published on-line sentence processing data indicating that, after the conjunction, leading Indirect Objects are not more unexpected than leading Subject NPs (ERP measurements reported by Bornkessel, Schlesewsky & Friederici (2002)). These authors write: ‘[D]ative-marked objects in the initial position of an embedded sentence do not elicit a neurophysiologically distinct response from subjects, whereas accusative-marked objects do. These differences are predictable on the basis of grammatical distinctions (i.e. underlying linguistic properties), but not on the basis of frequency information (i.e. a superficial linguistic property)” (p. B21). According to standard linguistic methodology, grammaticality/acceptability judgments constitute the empirical basis of grammatical distinctions. So, here we are confronted with a remarkable discrepancy between a grammatical distinction and its empirical basis. Bornkessel et al.’s frequency counts of leading dative- vs. accusative-marked objects immediately following the conjunction dass ‘that’ yield comparable frequencies, both being considerably lower than the frequency of Subject NPs in the same position. Their frequencies are based on the W-Pub corpus of the Institut für Deutsche Sprache in Mannheim. We conducted a frequency count of $S+I+O$ permutations in the NEGRA treebank (version 2, 2001; cf. Skut, Krenn, Brants & Uszkoreit (1997)) and found, among other things, that there are many more clauses with leading Subject NPs than with leading Indirect or Direct Objects. However, in contrast with Bornkessel et al.’s results, permutations starting with Direct Object tend to be more frequent than those with initial Indirect Object.

The acceptability values reported by Keller range from about −.36 to about .21. In order to transform them to (quasi-) probabilities, we added .5 to each rating, thus projecting them into the interval [0:1] in a linear manner. Then we added the values of the six possible $S+I+O$ permutations of a given sentence type. We repeated this for each column in Table 2. Finally, we divided each value by the total of its column. It follows that the six values in each column add up to 1.
Table 2. (Quasi-)probabilities of S+I+O permutations based on Keller’s (2000a) acceptability ratings. See footnote 4 for the applied linear data transformation method. Columns refer to NP types (full or pronominal), rows represent permutations of the constituents. For a graphical rendering of these data, see the dark bars in Figure 7.

<table>
<thead>
<tr>
<th></th>
<th>Sf,If,Of</th>
<th>Sp,If,Of</th>
<th>Sl,Ip,Of</th>
<th>Sl,Ip,Op</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIO</td>
<td>0.28</td>
<td>0.28</td>
<td>0.26</td>
<td>0.14</td>
</tr>
<tr>
<td>SOI</td>
<td>0.24</td>
<td>0.28</td>
<td>0.18</td>
<td>0.30</td>
</tr>
<tr>
<td>ISO</td>
<td>0.17</td>
<td>0.15</td>
<td>0.25</td>
<td>0.10</td>
</tr>
<tr>
<td>IOS</td>
<td>0.09</td>
<td>0.09</td>
<td>0.10</td>
<td>0.06</td>
</tr>
<tr>
<td>OSI</td>
<td>0.12</td>
<td>0.13</td>
<td>0.09</td>
<td>0.27</td>
</tr>
<tr>
<td>OIS</td>
<td>0.09</td>
<td>0.07</td>
<td>0.12</td>
<td>0.12</td>
</tr>
</tbody>
</table>

Keller’s Experiment 10 yielded acceptability ratings for sentences with only two constituents — S and O, full or pronominal. The Subject NPs were animate, the Direct Objects inanimate (see (5) for an example). Table 3 presents the average acceptability ratings after application of the same linear transformation as on the data of Experiment 6 (cf. footnote 4). That is, first the possibly negative values (ranging from −0.11 to +0.42) were projected into the interval [0:1] by adding 0.5. Second, we added the values for the two possible S+O permutations and divided both by the sum.

(5) *Maria glaubt, dass der Vater den Wagen kauft*

'Maria believes that the father buys the car'

Table 3. (Quasi-)probability estimates of S+O permutations based on Keller’s (2000a) Experiment 10. See footnote 4 for the applied linear data transformation method.

<table>
<thead>
<tr>
<th></th>
<th>Sf,O f</th>
<th>Sf,Op</th>
<th>Sp,Of</th>
<th>Sp,Op</th>
</tr>
</thead>
<tbody>
<tr>
<td>SO</td>
<td>0.59</td>
<td>0.50</td>
<td>0.69</td>
<td>0.67</td>
</tr>
<tr>
<td>OS</td>
<td>0.41</td>
<td>0.50</td>
<td>0.31</td>
<td>0.33</td>
</tr>
</tbody>
</table>

6. A probabilistic model of the acceptability judgments

Under the assumption that Keller’s acceptability judgments are indeed interpretable
as reflections of the relative frequency of the various ordering patterns, we now construct a probabilistic model capable of filling the slots M2–M4 with frequencies that approximate the data in Tables 2 and 3. In line with the idea of incremental grammatical encoding, we assume that the constituents land in slots M2 through M4 of the topology in their order of arrival. Furthermore, we hypothesize that the permutation probabilities are predictable on the basis of the precedence probability of pairs of constituents. For every pair of constituents A and B belonging to the same clause, we need to estimate the probability that A arrives earlier than (i.e. becomes available for being ordered before) B. Given that a clause contains at most one S, I or O constituent, and that each of these may be full or pronominal, we have to estimate precedence probabilities for 12 pairs of constituents. Actually, because the sentence materials used in Keller’s experiment contained at most one pronominal NP, we have no target values for precedence pairs listing two pronominal NPs: \( p(S_p << I_p) \), \( p(S_p << O_p) \) and \( p(I_p << O_p) \). This reduces the number of free parameters to nine.

The probability of a three-constituent string ABC, \( p(\text{ABC}) \), can be predicted from the precedence probabilities of the pairs A<<B, B<<C, and A<<C, via the equation

\[
p(\text{ABC}) \propto p(A << B) \cdot p(B << C) \cdot p(A << C)
\]

For a derivation of this permutation probability formula we refer to the Appendix.

Which combination of values of the nine pairwise precedence probabilities yields the closest fit with the 24 permutation probabilities in Table 2? And is this fit close enough to justify the claim that the model provides a satisfactory account of the observations? To answer these questions we implemented a computer simulation program that searched virtually the entire parameter space defined by the nine precedence probabilities. Each of these variables was assigned values between .05 and .95 (incrementing by .05). For every setting, the program computed the probability of each of the 24 strings referred to in Table 2. Using a Least Squares method, we determined the parameter setting giving the best prediction of the string probabilities in Table 2. The best solution is presented in Table 4.

---

5 In order to obtain the equation for another permutation one only replaces the numerator; e.g., for CAB: \( p(C << A) \cdot p(C << B) \cdot p(A << B) \). The denominator remains the same.
Table 4: Optimal values of the pairwise precedence probabilities estimated by the simulation model.

<table>
<thead>
<tr>
<th>Order</th>
<th>Optimal value</th>
<th>Order</th>
<th>Optimal value</th>
<th>Order</th>
<th>Optimal value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_f &lt;&lt; I_f$</td>
<td>.60</td>
<td>$S_p &lt;&lt; I_f$</td>
<td>.65</td>
<td>$S_f &lt;&lt; I_p$</td>
<td>.50</td>
</tr>
<tr>
<td>$S_f &lt;&lt; O_f$</td>
<td>.70</td>
<td>$S_p &lt;&lt; O_f$</td>
<td>.70</td>
<td>$S_f &lt;&lt; O_p$</td>
<td>.55</td>
</tr>
<tr>
<td>$I_f &lt;&lt; O_f$</td>
<td>.50</td>
<td>$I_p &lt;&lt; O_f$</td>
<td>.55</td>
<td>$I_f &lt;&lt; O_p$</td>
<td>.30</td>
</tr>
</tbody>
</table>

Figure 7 displays the ‘predicted’ string probabilities determined by inserting the parameter settings in Table 4 into the permutation probability formula. The differences between the target values (Table 2) and the ‘predicted’ values are very small. The Spearman rank correlation between obtained and predicted string probabilities is .98. We conclude that the incrementality-driven approach we took in this chapter is viable. Inspection of the optimal parameter settings in Table 4 reveals that they are in line with constraints proposed earlier, such as those in (3). The Subject tends to precede the Indirect and Direct Object with an average precedence probability of .62 (first and strongest constraint in (3)). Pronominal NPs tend to arrive earlier than full NPs (average probability .59; second constraint in (3)). There is no clear preference for the Indirect Object to precede the Direct Object (average precedence probability .48; third and weakest constraint in (3)). Pronominal Direct Objects even strongly prefer to precede full Indirect Objects: $p(O_p << I_f) = .70$. This is in agreement with well-know observations: *Ich habe Maria es gegeben* (literally: I have Mary it given, that is, ‘I gave it to Mary’) is considerably worse than *Ich habe es Maria gegeben*.)
Figure 7. Comparison of observed and predicted acceptability ratings (permutation probabilities) for three-constituent sentences. First panel: $S$, $I$ and $O$ full NPs; second: $S$ pronominal; third: $I$ pronominal; fourth: $O$ pronominal.

The probability estimates for the $S+O$ precedence pairs can be compared directly...
with the observed data in Table 3. Table 5 shows that the predictions (or rather 'postdictions'), although less accurate, are satisfactory (rank correlation .93). Remember that this comparison involves two experiments with different sentence materials (cf. examples (4) and (5)).

Table 5. Comparison of rated and predicted values for sentences with Subject and a Direct Object NPs (Keller’s (2000a) Experiment 10). The $S_pO_p$ values cannot be estimated from the simulations of Experiment 6, where the sentences contained at most one pronominal NP.

<table>
<thead>
<tr>
<th></th>
<th>$S_fO_t$</th>
<th>$S_pO_t$</th>
<th>$S_fO_p$</th>
<th>$S_pO_p$</th>
</tr>
</thead>
<tbody>
<tr>
<td>SO</td>
<td>ratings</td>
<td>.59</td>
<td>.69</td>
<td>.50</td>
</tr>
<tr>
<td></td>
<td>predictions</td>
<td>.70</td>
<td>.70</td>
<td>.55</td>
</tr>
<tr>
<td>OS</td>
<td>ratings</td>
<td>.41</td>
<td>.31</td>
<td>.50</td>
</tr>
<tr>
<td></td>
<td>predictions</td>
<td>.30</td>
<td>.30</td>
<td>.45</td>
</tr>
</tbody>
</table>

7. Summary and discussion

We presented a linearization formalism capable of capturing a broad range of clausal constituent order phenomena in semi-free word order languages such as German. It is part of the psycholinguistically motivated formalism of Performance Grammar (PG), which has separate components for assembling the hierarchical and the linear structure of sentences. The basic data structure used by the linearization component is called *topology*: a one-dimensional array of positions ('slots') serving as landing sites for syntactic constituents. For every constituent there is a set of one or more legal landing sites, which depends on its grammatical function (Subject, Object, Head, etc.) and its syntactic shape. Focusing on the phenomenon of 'scrambling' of $S$, $I$ and $O$ in the Midfield of German clauses, we assume a clausal topology with three special slots, each accommodating at most one of these constituents in any order. Furthermore, we hypothesize that, due to various processing factors at the level of the conceptualizer and/or the semantic-syntactic interface, these constituents tend to 'arrive' at different points in time. That is, they get ready to land in a slot sequentially rather than in parallel (incremental grammatical encoding). If the

---

6 Notice that by predicting the $S+O$ data from the $S+I+O$ data we, in a sense, build in a safeguard against 'overfitting' since we treat the former data as test set for a model trained on the latter.
overt order of S, I and O indeed mirrors their order of arrival, then it should be possible to predict the probability of the various permutations on the basis of the precedence probabilities of pairs of constituents, e.g. on how likely it is for S to arrive before I, etc. Computer simulation of a model based on these assumptions produced a very good fit with experimental acceptability data that we took to reflect probabilities of occurrence of S, I and O permutations.

The success of this approach suggests, first of all, that satisfactory accounts of the scrambling phenomena under scrutiny may well be found within the conceptualizer or the semantic-syntactic interface rather than in the domain of explicit syntactic ordering rules. If so, the syntax of German needs no provisions at all for dealing with these phenomena. Secondly, our incrementality-based account acceptability ratings of S+I+O permutations appears to offer a viable alternative to published accounts of in terms of a hierarchy of ordering constraints (e.g., such as are fashionable in Optimality Theory).

At the end of the paper we should point out an important limitation of the model developed above. Based on the idea of incremental grammatical encoding we have presupposed throughout that the three central constituents become available for linearization sequentially, that is, at clearly distinct points in time. What happens when the encoder works on two or three constituents in parallel and delivers them (nearly) simultaneously? One possible solution is for the simultaneous constituents to assume a default/standard order (e.g. S << I << O). The present model does not take non-incremental grammatical encoding into account. Whether the addition of provisions for this encoding mode yields a more satisfactory model, is a question to be answered in future research.

Appendix. Estimating three-item permutation probabilities from pairwise precedence probabilities

The constituent permutations considered in this paper (SIO, ISO, etc.) can be represented by strings of at most three symbols — call them A, B, and C. Their probability of occurrence, we assume, is a function of the pairwise precedence probabilities \( p(A<<B), p(B<<C) \) and \( p(A<<C) \). Imagine three (biased or unbiased) coins with a precedence relation inscribed on one side and its inverse on the other side. Tossing these coins will reveal the probabilities of the precedence pairs, e.g. \( p(A<<B) \) and
When three coins are tossed, there are eight possible outcomes; see the three leftmost columns in Table A1. Six of these outcomes define unique strings — enumerated in the rightmost column. Two outcomes (the second and the seventh) are 'illegal' because they violate transitivity of the precedence relation (if $A\ll B$ and $B\ll C$ then $A\ll C$).

Table A1.

<table>
<thead>
<tr>
<th>Precedence alternatives</th>
<th>Permutation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A\ll B$</td>
<td>$B\ll C$</td>
</tr>
<tr>
<td>$A\ll B$</td>
<td>$B\ll C$</td>
</tr>
<tr>
<td>$A\ll B$</td>
<td>$C\ll B$</td>
</tr>
<tr>
<td>$A\ll B$</td>
<td>$C\ll B$</td>
</tr>
<tr>
<td>$B\ll A$</td>
<td>$B\ll C$</td>
</tr>
<tr>
<td>$B\ll A$</td>
<td>$B\ll C$</td>
</tr>
<tr>
<td>$B\ll A$</td>
<td>$C\ll B$</td>
</tr>
<tr>
<td>$B\ll A$</td>
<td>$C\ll B$</td>
</tr>
</tbody>
</table>

If we now let $x = p(A\ll B)$, $y = p(B\ll C)$ and $z = p(A\ll C)$, then the probability of getting the first tossing result, which yields the permutation "ABC", is expressed by the product $xyz$. The sum of the probabilities of the six possible legal (i.e. transitive) outcomes therefore can now be written as

$$xyz + x(1-y)z + x(1-y)(1-z) + (1-x)yz + (1-x)y(1-z) + (1-x)(1-y)(1-z)$$

or, equivalently, as

$$1-xy(1-z)-(1-x)(1-y)z = 1-xy+z(x+y-1).$$

It follows that:

For instance,
Substituting the variables $x$, $y$ and $z$ by the original pairwise precedence probabilities gives, for example

There are six such formulae, one for each permutation of A, B and C. In order to obtain estimators for the 24 string probabilities in Table 2, one replaces the symbols A, B and C by the real constituents: $\{S_f, I_r, O_l\}$, $\{S_r, I_r, O_l\}$, $\{S_r, I_f, O_r\}$ and $\{S_r, I_r, O_r\}$.

References


