

Processing Optimality-theoretic Syntax by Interleaved Chart Parsing and Generation

Jonas Kuhn

Institut für maschinelle Sprachverarbeitung, Universität Stuttgart
Azenbergstraße 12, D-70174 Stuttgart, Germany
jonas@ims.uni-stuttgart.de

Abstract

The Earley deduction algorithm is extended for the processing of OT syntax based on feature grammars. Due to faithfulness violations, infinitely many candidates must be compared. With the (reasonable) assumptions (i) that OT constraints are descriptions denoting bounded structures and (ii) that every rule recursion in the base grammar incurs some constraint violation, a chart algorithm can be devised. Interleaving parsing and generation permits the application of generation-based optimization even in the parsing task, i.e., for a string input.

1 Introduction

In Optimality Theory (OT), both the (violable) constraints and the set of competing candidate structures are universally invariant; just the relative ranking of constraints is subject to cross-linguistic variation. This makes OT an attractive model for the explanation of typological variation and acquisition. In contrast to OT phonology there has been little computational work on OT syntax. Here, I propose a parsing/generation strategy for OT syntax based on Earley deduction, combining Tesar’s (1995) chart-based computation of OT competition with insights from parsing/generation with constraint-based grammars (in particular Neumann’s (1998) interleaving). A prototype has been implemented in Prolog.

Sec. 2 presents background on OT syntax and a formalization based on Lexical-

Functional Grammar (LFG); sec. 3 identifies two computational challenges; sec. 4 presents the Optimality-theoretic Earley deduction approach, interleaving parsing and generation.

2 OT syntax background

Various OT syntax models have been assumed. I will follow the OT-LFG approach first proposed by Bresnan (1998), since its representations—constituent (c-)structure trees and corresponding feature (f-)structures—are well-understood, and there is both work on formalization¹ and a growing typological literature.² However, the algorithm itself is largely independent from specifics of OT-LFG.

The set of competing candidates is defined by a common underlying representation (“input”), in OT-LFG an underspecified f-structure (1), containing all the information relevant for interpretation (i.e., all candidates have the same meaning).³

$$(1) \left[\begin{array}{l} \text{PRED} \quad \text{'see(x,y)'} \\ \text{GF}_1 \quad \left[\begin{array}{l} \text{PRED} \quad \text{'they' }]_x \\ \text{PRED} \quad \text{'person' } \\ \text{OP} \quad + \end{array} \right]_y \\ \text{GF}_2 \end{array} \right]$$

Candidates are c-structure/f-structure pairs defined by a grammar G_{inviol} encoding the inviolable principles (extended X-bar theory). While each candidate’s f-structure has to be subsumed by the input, the c-structure and lexical filling can deviate

¹(Johnson, 1998; Kuhn, 2000a; Kuhn, 2000b)

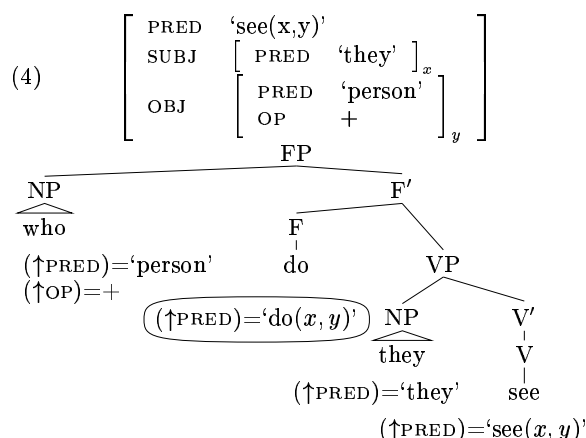
²See e.g., the contributions in (Sells, 2000).

³A richer semantic representation could be assumed, making quantifier scope explicit, etc. In the OT-LFG literature, the argument structure reflected in f-structure has been largely sufficient.

considerably from what is considered “canonical”. Expletive elements on the one hand and dropped pronominals on the other motivate the theoretical assumption that Faithfulness to the input is a *violable constraint*. In English the Markedness constraint OP-SPEC (2) is ranked higher than the Faithfulness constraint DEP-IO (3).

- (2) OP-SPEC
Every operator is in the topmost specifier projection of an extended projection.
- (3) DEP-IO
‘No epenthesis’—Output segments must have input correspondents (i.e., all lexical f-specifications are used in the f-structure).

Thus candidate (4) can arise as grammatical despite a violation of (3)—the lexical contribution of *do* (encircled) is not used in the f-structure.



The tableau (5) shows part of the competition that produces (4) as the winner in English, being the most harmonic.⁴

(5)

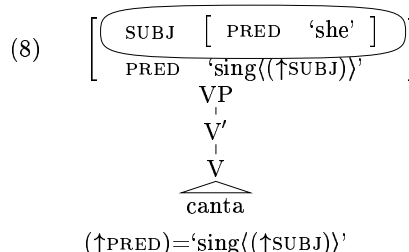
| Candidates | | | OP-SPEC | DEP-IO |
|------------|----|--------------------------|---------|--------|
| | VP | they see who | *! | |
| | FP | they do VP see who | *! | * |
| ☞ | FP | who do VP they see | | * |
| | FP | who do FP they do VP see | | **! |

- (6) Candidate A is *more harmonic* than candidate B iff A’s constraint marking contains less violations for the highest-ranking constraint in which the marking of A and B differs.

⁴For expository reasons, the option of realizing the specifier of FP without realizing the F head (as in *who they see*) is excluded here. This effect is standardly reached with an additional constraint OB-HD favouring projections with an overt head.

In (4), the lexical material is richer than the input. The opposite situation occurs when MAX-IO (7) is violated. Such a candidate is optimal in *pro-drop* languages like Italian (8) (cf. Grimshaw and Samek-Lodovici (1998)).⁵

- (7) MAX-IO
‘No deletion’—Input segments must have output correspondents (i.e., input material to be realized by lexical material).



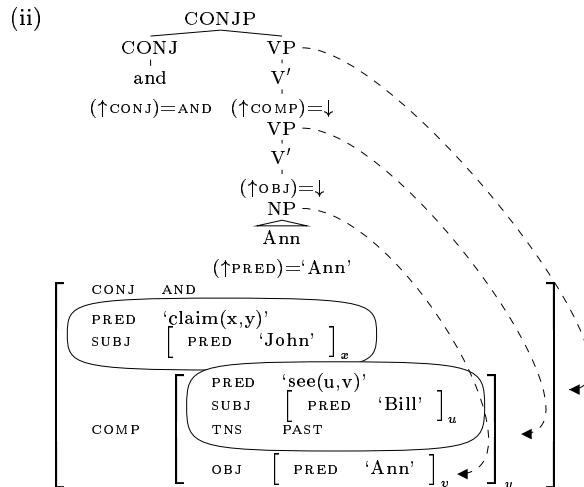
2.1 Formalizing OT-LFG

Following the subsumption-based conception, the candidate set for a competition can be defined as follows:

- (9) Φ_{in} : non-empty input representation; G_{inviol} : (set of c-/f-structure analyses generated by) LFG base grammar; **define**
 $Gen(\Phi_{in}) = \{ \langle T, \Phi' \rangle \in G_{inviol} : \Phi_{in} \sqsubseteq \Phi', \text{ where } \Phi' \text{ contains no more semantic information than } \Phi_{in} \}$

⁵More drastic MAX-IO violations occur when ellipsis is modelled with this mechanism. B’s response in the dialogue (i) could be modelled as in (ii). There is no formal limit to the amount of recursive structure dropped (of course, context-related constraints have to ensure recoverability).

- (i) A: John claimed that Bill saw Sue.
 B: And Ann.



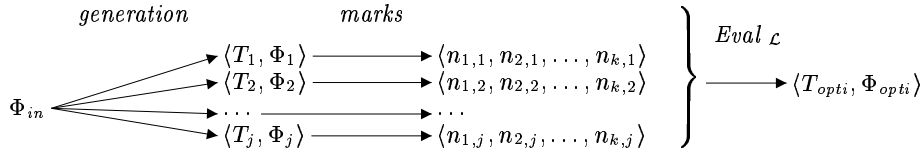


Figure 1: Sketch of generation-based OT processing

The violable constraints can be expressed as (implicative) descriptions of the candidate’s c-structure and f-structure (Kuhn, 2000b). The constraint marking function *marks* is then defined as follows:

- (10) $\langle T, \Phi \rangle$: LFG analysis; $\mathcal{C} = \langle c_1, c_2 \dots c_k \rangle$: constraints; **define**
 $marks(\langle T, \Phi \rangle) = \langle n_1, n_2 \dots n_k \rangle$,
 where n_i is the number of violations that $\langle T, \Phi \rangle$ incurs for constraint c_i .

Eval picks the most harmonic (cf. (6)) from a set of analyses, given a language-specific ranking $\gg_{\mathcal{L}}$ of the constraints.

- (11) Γ : set of LFG analyses $\langle T, \Phi \rangle$; $\langle \mathcal{C}, \gg_{\mathcal{L}} \rangle$: ranked set of constraints; **define**
 $Eval_{\mathcal{L}}(\Gamma) = \{ \langle T_j, \Phi_j \rangle \in \Gamma : marks(\langle T_j, \Phi_j \rangle) \text{ is maximally harmonic for all analyses in } \Gamma, \text{ under ranking } \gg_{\mathcal{L}} \}$

We can now define the language generated by an OT-LFG system:

- (12) G_{inviol} ; $\langle \mathcal{C}, \gg_{\mathcal{L}} \rangle$; *Lex*: lexicon; **define**
 $L(G_{inviol}, \langle \mathcal{C}, \gg_{\mathcal{L}} \rangle) = \{ w \in Lex^+ : \exists \text{ input representation } \Phi_{in}, w \text{ is the terminal string of some } \langle T, \Phi \rangle \in Eval_{\mathcal{L}}(Gen(\Phi_{in})) \}$

3 Two computational challenges

It seems that OT processing can be modelled straightforwardly as sketched in fig. 1, based on “classical” generation from Φ_{in} with G_{inviol} . However, due to unfaithfulness, we have to deal with infinitely many competing candidates: Under a DEP-IO violation, a recursion in the base grammar may be traversed without “consuming” any part of the input—generating for example arbitrarily many instances of expletive *do*. But the following property of OT systems can resolve the problem: with assumptions (13) and (14), there will be a point in any traversal of a base grammar recursion from which on all larger candidates are less harmonic than one given candidate.⁶ To my knowledge, these assumptions

⁶This is a consequence of the pumping lemma for context-free languages. A formal proof is in preparation.

are compatible with the intuitions behind the OT accounts in the literature.⁷

- (13) The constraints are such that every application of recursive rules from G_{inviol} incurs some constraint violation.⁸
 (14) OT constraints are structural descriptions denoting bounded structures.⁹

Tesar (1995) exploits this property in his dynamic programming (or chart-based) algorithm for regular languages and context-free “position grammars”. In sec. 4.2, this is extended to feature grammars.

The second computational challenge arises for the recognition and parsing task for an OT language defined as in (12). As Johnson (1998) notes, a technique symmetrical to the one sketched in fig. 1—i.e., parsing a string and comparing the constraint marking of the different analyses—does not realize the parsing task for the language defined by the original system: strings that are not optimal for any underlying representation Φ_{in} according to (12) will be wrongly predicted to be grammatical—since trivially, one of these will be optimal for this parsing-based optimization.

⁷A straightforward way to satisfy (13) is to assume a constraint AVOIDSTRUC, which is violated once by every c-structure node.

⁸(13) can be seen as a relaxed version of the offline parsability restriction, and a corresponding offline generability restriction: In classical parsing/generation, these restrictions exclude rule recursion to be applied vacuously (wrt. the input, i.e., a string/a resourced feature structure), because this would lead to an infinite number of c-structures. In the OT-LFG setting, *Gen* specifies an infinite set of candidates, and it is the constraints that control the traversal of the candidate space.

A reviewer points out that (13) may not be necessary, since for an infinite set of equally harmonic candidates arising through pumping (cf. fn. 6), the “unpumped” version could be used as a representative. Kaplan and Wedekind (2000) make a similar point for classical generation.

⁹Note that (14) does not exclude that a constraint can have an *effect* on some unboundedly remote part of the structure (through constraint interaction).

Modelling the recognition and parsing task computationally will involve a bidirectional processing effort (Kuhn, 2000b): the string is parsed to detect possible underlying representations, and from these a “backward generation” step is performed, for which OT evaluation is applied. Only if the parsed string is matched by the optimal candidate for its underlying representation, is the string grammatical.¹⁰

Faithfulness violations of the MAX-IO type (7) allow for situations where underlying material is not reflected by any of the lexical material. Unless one introduces a bound for such violations by definition (restricting the generality of the computational model), it turns out that the parsing task in the bidirectional processing scheme cannot be subjected to the standard restriction of offline parsability (cf. fn. 13):¹¹ like in generation, the set of parsing alternatives that have to be considered is infinite, so a systematic traversal of the search space is required (sec. 4.3).

4 Optimality-theoretic Earley deduction

4.1 Earley deduction for parsing and generation

Since the chart-based algorithm to be developed is to work bidirectionally, using Earley deduction (Pereira and Warren, 1983) is a natural choice.¹² Parsing and generation amounts to deduction of a given goal using two basic inference rules: *prediction* (or instantiation) and *completion* (or reduction). In parsing, the goal’s string is specified (and used as its *index*); in generation, its underlying representation is. An agenda is used as a control structure; items on the agenda and in the chart have the form of definite clauses. We distinguish (i) active items, “looking for” material of a certain category and with certain

¹⁰As just laid out, the model involves bidirectional processing, but optimization applies only in one (the generation) direction. It is however straightforward to extend this model to a *bidirectional optimization* model (Kuhn, 2000a).

¹¹This becomes clear in structure (ii) in fn. 5 above.

¹²Cf. (Shieber, 1988); I adopt Neumann’s (1998) uniform agenda-driven tabular algorithm.

features (their selected element), and (ii) passive items, stating that certain material has already been found. When matching, a passive and an active item can be combined by *completion*.¹³ Active items come into existence through *prediction*, based on a grammar rule. Before creating a new chart item and triggering inference rules based on it, it is checked whether such an item exists already (i.e., it is *blocked*), avoiding unnecessary recomputations. The algorithm is initialized by putting an active item looking for the grammar’s root symbol on the agenda, indexed with the complete input. It terminates when the agenda is empty.¹⁴

For indexing the underlying information in generation, I will essentially use the PRED values in the input f-structure, treating them as resources. To identify them uniquely, the semantic index for a given item will be a list of PRED values with their complete paths.

4.2 Chart-based optimization

The extension of Earley deduction to an OT system meeting assumptions (13) and (14) is strikingly simple: (i) a record of the constraint profile of the structure constructed so far is kept for (heads of) items; (ii) as *completion* is applied, the OT constraints are applied, recording the sum of new violations and the constraint profile of the passive item consumed (whenever a constraint may or may not apply, both options are computed¹⁵); (iii) when a passive item is checked for blocking, the constraint profile is taken into account: if the new item is more harmonic, it is not considered as blocked, but is applied. Blocking in (iii) is subjected to a restriction operation, ensuring there is only a finite number of possible types of items.¹⁶

¹³*Scanning* is a special case of *completion*, where a lexicon entry is used to match an active item’s selection.

¹⁴If successful, the chart contains one or more passive items covering the entire input, with the grammar’s root symbol as the category.

¹⁵A book-keeping scheme ensures that constraints on f-structure are applied only once, so unified f-structures are not regarded twice.

¹⁶This operation is justified, since assumption (14) ensures that for comparing the constraint profile of

A sample derivation. (15) specifies G_{inviol} , a small extended X-bar fragment. Categories are assumed to have an internal structure (Bresnan, 2000; Kuhn, 1999) encoding lexical class, bar level, the lexical/functional distinction, and whether they are topmost within an extended projection (i.e., a lexical projection with all its functional projections). For clarity, I will however use abbreviations like VP[+top] for $\langle verb, max, lex, +top \rangle$.¹⁷

- (15) a. ROOT \rightarrow XP[+top]
 $\uparrow=\downarrow$
 b. FP[α] \rightarrow (NP[+top]) F'[α]
 $\{ (\uparrow_{SUBJ})=\downarrow \quad \uparrow=\downarrow$
 $\mid (\uparrow_{TOPIC})=\downarrow$
 $(\uparrow_{OBJ})=\downarrow \}$
 c. F'[α] \rightarrow F[α] XP[-top]
 $\uparrow=\downarrow \quad \uparrow=\downarrow$
 d. VP[α] \rightarrow (NP[+top]) V'[α]
 $(\uparrow_{SUBJ})=\downarrow \quad \uparrow=\downarrow$
 e. V'[α] \rightarrow V[α] (NP[+top])
 $\uparrow=\downarrow \quad (\uparrow_{OBJ})=\downarrow$

The functional annotations follow the annotation principles of (Bresnan, 2000), saying for instance (15e) that the f-structure of the verb's complement is introduced under OBJ. Functional projections are unified with lexical projections at the level of f-structure (thus the $\uparrow=\downarrow$ for both F[α] and XP[-top] in (15c)). Note that XP[-top] in (15c) can be instantiated as FP[-top], leading to a recursion of FP's, all mapped to the same f-structure.

(16) lists the lexicon. Under a faithfulness violation it is possible to skip the lexical contribution, e.g., the PRED introduction by *do*. Thus, the candidates in the tableau (5) (and infinitely many others) are generated by the grammar.

- (16) they NP[+top] (\uparrow_{PRED})='they'
 $(\uparrow_{OP})=-$
 who NP[+top] (\uparrow_{PRED})='who'
 $(\uparrow_{OP})=+$
 see V[] (\uparrow_{PRED})='see(x, y)'
 do F[] (\uparrow_{PRED})='do(x, y)'

two items with the same index, a bounded subset of their feature structures has to be regarded. The restricted structure must be at least the size of the structure maximally denoted by a constraint; using abstraction over all constraints as restriction will guarantee this.

¹⁷FP[+/-top] is used for functional projections (all of which will be verbal in this example); XP[+/-top] is underspecified for the lexical/functional distinction.

Items are notated as follows:

- (17) \langle HEAD \leftarrow REMAINING_BODY; [n, m]; *index*; book-keeping \rangle , where
- REMAINING_BODY (empty for passive items): selected (here generally leftmost) element marked by underlining;
 - n, m : number of violations of OP-SPEC (2) and DEP-IO (3);
 - semantic *index* is written as a list of paths.

The book-keeping keeps track of the structure constructed so far; for illustrative purposes, I use just the string for this.¹⁸ Details of the control strategy are ignored for clarity of presentation.

Let us look at the task of generating from (1), represented here as [PRED-SEE, SUBJ-PRED-THEY, OBJ-PRED-WHO]. The agenda is initialized with item

- (18) \langle ANSWER \leftarrow ROOT; [0,0]; [PRED-SEE, SUBJ-PRED-THEY, OBJ-PRED-WHO]; $\langle \rangle$ \rangle

Prediction will apply, based on rule (15a), giving rise to the following new item:

- (19) \langle ROOT \leftarrow XP[+top]; [0,0]; [PRED-SEE, SUBJ-PRED-THEY, OBJ-PRED-WHO]; $\langle \rangle$ \rangle ,

which will trigger a chain of predictions (note that the index in these active items is the selected element's index, i.e., unless an $\uparrow=\downarrow$ node is selected, the index changes):

- (20) \langle FP[+top] \leftarrow NP[+top] F'[+top]; [0,0]; [SUBJ-PRED-THEY]; $\langle \rangle$ \rangle

- (21) \langle FP[+top] \leftarrow NP[+top] F'[+top]; [0,0]; [OBJ-PRED-WHO]; $\langle \rangle$ \rangle

- (22) \langle FP[+top] \leftarrow F'[+top]; [0,0]; [PRED-SEE, SUBJ-PRED-THEY, OBJ-PRED-WHO]; $\langle \rangle$ \rangle

- (23) \langle F'[+top] \leftarrow F[+top] XP[-top]; [0,0]; [PRED-SEE, SUBJ-PRED-THEY, OBJ-PRED-WHO]; $\langle \rangle$ \rangle

- (24) \langle VP[+top] \leftarrow NP[+top] V'[+top]; [0,0]; [SUBJ-PRED-THEY]; $\langle \rangle$ \rangle

- (25) \langle VP[+top] \leftarrow V'[+top]; [0,0]; [PRED-SEE, SUBJ-PRED-THEY, OBJ-PRED-WHO]; $\langle \rangle$ \rangle

- (26) \langle V'[+top] \leftarrow V[+top] (NP[+top]); [0,0]; [PRED-SEE, SUBJ-PRED-THEY, OBJ-PRED-WHO]; $\langle \rangle$ \rangle

¹⁸The string does *not* represent the string covered by the selected element, but the entire derivation history!

Each of (20) and (24) can undergo scanning with the entry for *they*, leading to reduced items (the index contains the remaining material to be generated by the new selection):

$$(27) \langle \text{FP}[+\text{top}] \leftarrow \underline{\text{F}'[+\text{top}]}; [0,0]; [\text{PRED-SEE}, \text{OBJ-PRED-WHO}]; \langle \text{they} \rangle \rangle$$

$$(28) \langle \text{VP}[+\text{top}] \leftarrow \underline{\text{V}'[+\text{top}]}; [0,0]; [\text{PRED-SEE}, \text{OBJ-PRED-WHO}]; \langle \text{they} \rangle \rangle$$

Let us look at (28). The following prediction

$$(29) \langle \text{V}'[+\text{top}] \leftarrow \underline{\text{V}[+\text{top}]} (\text{NP}[+\text{top}]); [0,0]; [\text{PRED-SEE}, \text{OBJ-PRED-WHO}]; \langle \text{they} \rangle \rangle$$

can be used to scan *see*, resulting in two options, depending whether or not the optional NP is assumed: another active item

$$(30) \langle \text{V}'[+\text{top}] \leftarrow \underline{\text{NP}[+\text{top}]}; [0,0]; [\text{OBJ-PRED-WHO}]; \langle \text{they see} \rangle \rangle$$

or a passive item:

$$(31) \langle \underline{\text{V}'[+\text{top}]} \leftarrow \epsilon; [0,0]; [\text{PRED-SEE}]; \langle \text{they see} \rangle \rangle$$

When creating a passive item, the structural OT constraints are checked; however, in this case, no violations occur.

(30) can be used to scan *who*, leading to another passive item (with a different index). Here, we do have a constraint violation: *who* is marked as an operator, but it is not introduced in the topmost specifier position.

$$(32) \langle \underline{\text{V}'[+\text{top}]} \leftarrow \epsilon; [1,0]; [\text{PRED-SEE}, \text{OBJ-PRED-WHO}]; \langle \text{they see who} \rangle \rangle$$

(32) triggers completion of the VP[+top] item (not causing any further constraint violations), which again triggers completion of the initial ANSWER item. So, we have a first candidate to cover the entire input.

$$(33) \langle \underline{\text{VP}[+\text{top}]} \leftarrow \epsilon; [1,0]; [\text{PRED-SEE}, \text{SUBJ-PRED-THEY}, \text{OBJ-PRED-WHO}]; \langle \text{they see who} \rangle \rangle$$

$$(34) \langle \underline{\text{ANSWER}} \leftarrow \epsilon; [1,0]; [\text{PRED-SEE}, \text{SUBJ-PRED-THEY}, \text{OBJ-PRED-WHO}]; \langle \text{they see who} \rangle \rangle$$

Going back to (21), this active item can be used scanning *who*

$$(35) \langle \text{FP}[+\text{top}] \leftarrow \underline{\text{F}'[+\text{top}]}; [0,0]; [\text{PRED-SEE}, \text{SUBJ-PRED-THEY}]; \langle \text{who} \rangle \rangle$$

predicting

$$(36) \langle \text{F}'[+\text{top}] \leftarrow \underline{\text{F}[+\text{top}]} \text{XP}[-\text{top}]; [0,0]; [\text{PRED-SEE}, \text{SUBJ-PRED-THEY}]; \langle \text{who} \rangle \rangle$$

Here, an interesting case of scanning can occur: in the lexicon there is an entry of the selected category F[_]: *do*. Its f-annotation (introducing [PRED-DO]) does not match the index. But there is the option of using a lexicon entry unfaithfully, introducing a violation of DEP-IO.¹⁹ Note the unchanged index.

$$(37) \langle \text{F}'[+\text{top}] \leftarrow \underline{\text{XP}[-\text{top}]}; [0,1]; [\text{PRED-SEE}, \text{SUBJ-PRED-THEY}]; \langle \text{who do} \rangle \rangle$$

Recall that XP[-top] is either VP[-top] or FP[-top]. The VP option will give us a partial derivation quite similar to the VP[+top] case above ((28)-(31)). Only the object cannot be realized in the VP, since its resource path has already been used. So, we get only the following passive item:

$$(38) \langle \underline{\text{VP}[-\text{top}]} \leftarrow \epsilon; [0,0]; [\text{PRED-SEE}, \text{SUBJ-PRED-THEY}]; \langle \text{who do they see} \rangle \rangle$$

This passive item can be used in completion combined with (37), resulting in the following passive items:

$$(39) \langle \underline{\text{F}'[+\text{top}]} \leftarrow \epsilon; [0,1]; [\text{PRED-SEE}, \text{SUBJ-PRED-THEY}]; \langle \text{who do they see} \rangle \rangle$$

$$(40) \langle \underline{\text{FP}[+\text{top}]} \leftarrow \epsilon; [0,1]; [\text{PRED-SEE}, \text{SUBJ-PRED-THEY}, \text{OBJ-PRED-WHO}]; \langle \text{who do they see} \rangle \rangle$$

This will finally trigger completion of (18). Note that there is already a passive ANSWER item with the same index: (34). So, classically, we would have a case of blocking. But here, the constraint profile is compared, and it turns out that the new ANSWER item is more harmonic. So we replace the first item by this new one:

$$(41) \langle \underline{\text{ANSWER}} \leftarrow \epsilon; [0,1]; [\text{PRED-SEE}, \text{SUBJ-PRED-THEY}, \text{OBJ-PRED-WHO}]; \langle \text{who do they see} \rangle \rangle$$

But processing is not yet finished. The other option for (37) was that the selected XP[-top] is an FP[-top]; then, we get the following new active item (among others):

$$(42) \langle \text{FP}[-\text{top}] \leftarrow \underline{\text{F}'[-\text{top}]}; [0,1]; [\text{PRED-SEE}, \text{SUBJ-PRED-THEY}]; \langle \text{who do} \rangle \rangle$$

This will predict

$$(43) \langle \text{F}'[-\text{top}] \leftarrow \underline{\text{F}[-\text{top}]} \text{XP}[-\text{top}]; [0,1]; [\text{PRED-SEE}, \text{SUBJ-PRED-THEY}]; \langle \text{who do} \rangle \rangle$$

¹⁹Although DEP-IO could be modelled to be checked not until passive items are created, it is natural to keep track of violations along with lexical access.

which can combine with another unfaithful use of *do*:

$$(44) \langle \underline{F'[-top]} \leftarrow \underline{XP[-top]}; [0,2]; [\text{PRED-SEE}, \text{SUBJ-PRED-THEY}]; \langle \textit{who do do} \rangle \rangle$$

Again, we have a choice for $XP[-top]$. We can pick $VP[-top]$ again, being able to reuse (38). Completion gives us

$$(45) \langle \underline{F'[-top]} \leftarrow \epsilon; [0,2]; [\text{PRED-SEE}, \text{SUBJ-PRED-THEY}]; \langle \textit{who do do they see} \rangle \rangle$$

which completes (42) and ultimately (37):

$$(46) \langle \underline{F'[+top]} \leftarrow \epsilon; [0,2]; [\text{PRED-SEE}, \text{SUBJ-PRED-THEY}]; \langle \textit{who do do they see} \rangle \rangle$$

But note that such an item exists already: (39). Comparing the constraint profiles, the new option is less harmonic (the existing item used only one *do*: [0,1]). Further predictions with $XP[-top]$ as $FP[-top]$ are blocked (as in classical chart parsing/generation).

The sample derivation shows how an infinite set of candidates (basically *who do do⁺ they see*) is discarded as an equivalence class. This makes OT processing with an infinite candidate set possible. The example was simple, but the technique carries over to all constraints satisfying assumption (14). It may be required to “pass” a cycle several times before strict harmony decrease (guaranteed by (13)) takes effect. The passes of this cyclic structure have the effect of avoiding the violation of some high-ranked constraint.²⁰ With the constraint size bounded, this construction is guaranteed to terminate. Additional cycle passes will again cause deterioration.

4.3 Interleaving parsing and generation

As discussed in sec. 3, the recognition and parsing task for an OT system involves parsing and backward generation. Following ideas of (Neumann, 1998), I realize this in an interleaved way, assuming a double index for passive items, so they can be used in both directions. For parsing with optimizing backward

²⁰Note that an edge avoiding the high-ranked constraint will have been constructed already before hitting the recursion, anticipating the larger structure required. (This is because for any constraint that may or may not be violated, both options are entered to the chart.) Thus, the violations incurred by the cycle will not cut this branch.

generation, the following procedure is effective:

The agenda is initialized by an active parsing item, indexed by the entire input string. At the point where normally a passive item i_p is added to the chart, an active generation item is put on the agenda, with the semantic index constructed for i_p in parsing. This will trigger an intermediate generation phase, exploring alternative analyses. Here, optimization applies, leading to an optimal item i_g for the semantic index under consideration. When the generation phase is finished, the i_g is compared to i_p . If it is identical, i_p is actually added to the chart; if not, it is ungrammatical and will be discarded (more precisely, a record is kept that this particular item has been shown to be ungrammatical). Later generation phases can use intermediate results of the earlier ones.

MAX-IO violations introduce to parsing similar situations as DEP-IO did to generation: in order to consider all possible underlying representations, a cyclic chart structure has to be dealt with. If *bidirectional optimization* is modeled (i.e., the optimal generation candidates undergo another competition, leaving only a candidate that is best in both directions), the computational solution for generation will carry over directly to parsing: the items constructed in parsing are also checked for constraint violations and filtered accordingly.

But MAX-IO violations are also assumed in models without bidirectional optimization (which *do* require bidirectional *processing*, cf. sec. 3). So, the recognition task based on these models seems to pose a decidability problem. Since there are infinitely many possible underlying forms for a given string, there is no straightforward procedure of applying “backward generation” to each of them. To guarantee decidability, either (strong²¹) bidirectional optimization has to be assumed, or the degree of MAX-IO violations produced by *Gen* has to be limited.²²

²¹For a discussion of weaker models, see (Kuhn, 2000a).

²²It is however conceivable that in the interleaved

Implementation. Vargas (1997) provides an experimental Sicstus Prolog implementation of (Neumann, 1998). On this basis, the algorithm illustrated in sec. 4.2, and the interleaving discussed in sec. 4.3 have been implemented. The parser/generator has been tested with small grammar fragments from the theoretical OT literature.

5 Discussion

I proposed a chart-based OT account for syntax, making crucial use of interleaving of parsing and generation. There are several obvious sources for exponential behaviour:²³ (i) constraint application leads to highly disjunctive structures; (ii) for generation, the number of item sets may grow exponentially in the size of the input f-structure.²⁴

For (i), sophisticated techniques from feature grammar parsing (Maxwell and Kaplan, 1998) may help, exploiting *de facto* independence of structures in most cases. Locality restrictions discussed in (Kuhn, 2000b) may help to limit problem (ii). Also, exploiting precomputed implications of the constraint ranking should have a considerable effect. Having couched computational OT syntax in the well-studied paradigm of Earley deduction will hopefully facilitate such extensions and improvements.

Acknowledgements

This research was supported by the SFB 340 (project B12) of the Deutsche Forschungsgemeinschaft. Thanks to Joan Bresnan, Anette Frank, Ron Kaplan, Martin Kay, John Maxwell, Hadar Shemtov, and Jürgen Wedekind for valuable discussion of various

approach, an indirect control of the infinite candidate space becomes possible (at least for all linguistically interesting cases). The idea would be that for all analyses arising through a recursive loop in parsing, it can be shown systematically that they are production-based losers to some more harmonic competitor with the same input (cf. also (Kuhn, 2000a)). This has to be deferred to future research however.

²³But note that parsing is already worst-case exponential for the base grammars.

²⁴Problem (ii) is a theoretical option even for non-OT generation (Kay, 1996), but the wide-spread unfaithfulness brings it out in the OT case.

issues related to the formalization and processing of OT syntax.

References

- Joan Bresnan. 1998. Optimal syntax. In J. Dekkers, F. van der Leeuw, and J. van de Weijer, editors, *Optimality Theory: Phonology, Syntax, and Acquisition*. Oxford University Press. To appear.
- Joan Bresnan. 2000. *Lexical-Functional Syntax*. Blackwell. To appear.
- Jane Grimshaw and Vieri Samek-Lodovici. 1998. Optimal subjects and subject universals. In P. Barbosa, D. Fox, P. Hagstrom, M. McGinnis, and D. Pesetsky, editors, *Is the Best Good Enough?*, pages 193–219. MIT Press and MITWPL.
- Mark Johnson. 1998. Optimality-theoretic Lexical Functional Grammar. In *Proceedings of the 11th Annual CUNY Conference on Human Sentence Processing, Rutgers University*. To appear.
- Ronald M. Kaplan and Jürgen Wedekind. 2000. LFG generation produces context-free languages. In *Proceedings of COLING-2000*, pages 297–302, Saarbrücken.
- Martin Kay. 1996. Chart generation. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics, Santa Cruz, CA*.
- Jonas Kuhn. 1999. Towards a simple architecture for the structure-function mapping. In M. Butt and T. H. King, editors, *Proceedings of the LFG99 Conference, Manchester, UK*, CSLI Proceedings Online.
- Jonas Kuhn. 2000a. Faithfulness violations and bidirectional optimization. In M. Butt and T. H. King, editors, *Proceedings of the LFG 2000 Conference, Berkeley, CA*, CSLI Proceedings Online. To appear.
- Jonas Kuhn. 2000b. Generation and parsing in Optimality Theoretic syntax – issues in the formalization of OT-LFG. In (Sells, 2000). To appear.
- John Maxwell and Ronald Kaplan. 1998. Unification-based parsers that automatically take advantage of context freeness. Ms., Xerox PARC, February 1998.
- Günter Neumann. 1998. Interleaving natural language parsing and generation through uniform processing. *Artificial Intelligence*, 99:121–163.
- Fernando Pereira and David Warren. 1983. Parsing as deduction. In *Proceedings of the 21st Annual Meeting of the Association for Computational Linguistics, Cambridge, MA*.
- Peter Sells, editor. 2000. *Formal and Empirical Issues in Optimality-theoretic Syntax*. CSLI Publications, Stanford. To appear.
- Stuart Shieber. 1988. A uniform architecture for parsing and generation. In *Proceedings of the 12th International Conference on Computational Linguistics (COLING), Budapest*.
- Bruce Tesar. 1995. *Computational Optimality Theory*. Ph.D. thesis, University of Colorado.
- Sebastian Vargas. 1997. Parsing und Generierung in uniformen Architekturen. Master's thesis, Heinrich-Heine-Universität Düsseldorf.