

Faithfulness violations and bidirectional optimization

Jonas Kuhn

Institut für maschinelle Sprachverarbeitung
Universität Stuttgart

Proceedings of the LFG00 Conference

University of California, Berkeley

Miriam Butt and Tracy Holloway King (Editors)

2000

CSLI Publications

<http://www-csli.stanford.edu/publications/>

Abstract

The systematic assumption of faithfulness violations in Optimality Theory implies an infinite space of candidates. Under the methodological principle of trying to explain as much possible through constraint interaction, control over this infinite space should be exerted by the constraints. Assuming the subsumption-based candidate definition of OT-LFG, the candidate space is indeed sufficiently structured to facilitate computational processing according to this principle. However, the parsing direction in the standard production-based optimization model is not subject to optimization, so for the parsing task, a decidability issue arises. Adopting a bidirectional optimization model is one way of solving this problem, but the required strong concept of bidirectional optimization may not be linguistically desirable. Other possible conclusions are discussed briefly.

1 Introduction

The key insight in the Optimality-theoretic (OT) approach in phonology and syntax¹ has been that variation between languages can be derived in a system assuming a universally invariant set of (conflicting) constraints on well-formed linguistic structures where it is only the relative ranking of these constraints that differs cross-linguistically. A constraint may be violated to satisfy some more highly ranking constraint. The different language-specific constraint rankings will bring out different ways of resolving conflicts between the constraints, leading to a different set of optimal (i.e., by definition grammatical) structures.

Slightly more technically, an OT system is thus set up as the combination of (i) a candidate generation component (*Gen*) that – given some underlying form (the *input*) – produces a set of competing structures which all satisfy some inviolable principles, and (ii) an evaluation component (*Eval*) that checks the candidate structures for constraint violations and determines the optimal (most harmonic) candidates relative to the constraint ranking of the language in question. (The customary tableau notation focuses on component (ii), assuming the candidate set as given and illustrating the constraint violations of the individual candidates and the harmony evaluation across the candidates.)

This general set-up leaves quite some space for variation as to the implementation of a particular OT system for use in a linguistic study or a computational system. One may choose to assume a relatively restrictive set of inviolable principles (as part of *Gen*), leaving a fairly small set of alternatives for the optimization step, or one may assume very weak inviolable principles and leave most of the work to the interaction of violable constraints.

Of course, keeping the candidate space small has the practical advantage of making the optimization task more perspicuous to the theorist, and indeed most OT studies in the literature focus on just some small set of candidates considered relevant for the studied phenomenon. However, this practical move doesn't justify the conclusion that the overall system that OT theorists see themselves as contributing to has a *Gen* component doing that much work. To the contrary, a widely assumed methodological principle is:

(1) *Methodological principle of OT*

Try to explain as much as possible as an effect of constraint interaction.

¹(Prince and Smolensky 1993); see (Kager 1999) for an introduction.

This implies an overall OT model with a very weak *Gen* component.

As an end in itself, principle (1) would not be of much scientific value. What is behind it is the above mentioned observation that for certain linguistic phenomena, OT constraint interaction has been shown to successfully predict the space of cross-linguistic variation (through *factorial typology*, cf. e.g., Kager 1999, sec. 1.7), including the systematic exclusion of certain logically possible languages. So the reason for following (1) is to investigate to what extent OT constraint interaction may serve as the key explanatory device in modelling linguistic knowledge in general. Evaluation of success in this investigation should be based on criteria like the following: Is the empirically observable typological space predicted based on a set of well-motivated constraints?² The strong hypothesis of the Optimality-theoretic approach is thus that all (and only) the observed cross-linguistic variation can be explained as an effect of constraint reranking.

A closer investigation of the formal and computational implications of the strong OT hypothesis is one way of checking to which degree it is tenable. The present paper is an attempt to follow this path, focusing on the division of labour between *Gen* and *Eval*. A key question will be under what circumstances the processing tasks (parsing/recognition and generation) based on an OT model are decidable (Johnson (1998) observes a decidability problem for the general, unrestricted OT model).

The reasoning in this paper is as follows: we can observe certain variations across the languages of the world (whether or not (i) expletive elements are used and (ii) pronominals may be dropped): sec. 2. If we want to model these variations as a mere effect of constraint interaction, *Gen* has to have a certain property (generating particular faithfulness violations); such a system is definable in the OT-LFG framework: sec. 3. Now, if we want all the processing tasks (in particular parsing) to be decidable with this type of *Gen*, we are forced to assume a bidirectional optimization regime. (A processing scheme for OT-LFG is reviewed in sec. 4; sec. 5 addresses the decidability issue and bidirection.) Bidirectional optimization has been variously argued for on empirical grounds, but it also has certain problems and is certainly not the standard model assumed for OT. Thus, it is somewhat surprising that under the given assumptions and hypotheses, bidirectional optimization is enforced on computational grounds. There are different possible conclusions to be drawn from this result (sec. 6).

The approach taken in this paper differs from the approach of (Kuhn 2000a), where the methodological principle (1) wasn't given highest priority, but the goal was to directly exploit results from formal work on classical LFG in order to set up a model for OT-LFG with decidable processing tasks.

2 Variation across languages

The types of cross-linguistic variation that motivate the assumption of considerable differences between the competing candidates' surface strings are very basic ones and were already discussed in the earliest work on OT syntax (cf. Grimshaw 1997): for syntactic reasons, some

²In the motivation of constraints for OT syntax there is a certain danger for circularity, since often an obvious functional motivation (like phonetic restrictions in OT phonology) cannot be given, so the best motivation is through effects of the constraint in interaction, essentially based on factorial typology.

languages require the use of expletive elements where other languages do not (cf. the expletive *do* in English (2a), vs. the German example (2b)).

(2) *Expletive elements*

- a. *Who did John see*
- b. *Wen sah John*
whom saw John

According to the methodological principle of OT (1), this contrast should be explained as an effect of constraint interaction; i.e., the structures of both sentences have to be competitors in the same candidate set. The candidate winning in English is a case where the surface string contains some additional element not present in the underlying *input*. So, quite similarly as in OT phonology, faithfulness to the input has to be a violable constraint in OT syntax. In English, it is outranked by some structural Markedness constraint, thus giving rise to an unfaithful winner. The constraint at stake here is DEP-IO:

(3) DEP-IO: Output segments must have input correspondents.—‘No epenthesis’

(4) MAX-IO: Input segments must have output correspondents.—‘No deletion’

For the MAX-IO constraint, we also find syntactic examples: *Pro-drop* languages like Italian (5b) have no overt correspondence for the subject pronoun referring to a topical entity (5a) (cf. Grimshaw and Samek-Lodovici 1998). This demonstrates that it is possible to leave some *input* material unrealized to satisfy some high-ranking Markedness constraint.

(5) *Dropped pronominal*

- a. *She has sung*
- b. *_ ha cantato*

Consequences for *Gen* As already stated in the introduction, identifying something as an effect of constraint interaction implies that the other component of an OT system, *Gen*, has to preserve the status quo. Assuming faithfulness as a violable constraint means that candidate generation has to be insensitive to the preservation of the *input* information in the surface string.

Assuming a predicate-argument structure with additional tense information as in (6) as *input* (cf. Grimshaw 1997), we are thus faced with all items in (7) as possible candidates, most of them violating DEP-IO or MAX-IO or both (violating MAX-IO twice will for instance lead to a phonologically empty candidate: (7f)).

(6) *Input*

laugh(Ann) & TENSE: PAST

- (7)
- a. Ann laughed
 - b. Ann did laugh
 - c. it laughed Ann
 - d. laughed
 - e. Ann
 - f.
 - g. she laughed
 - h. she did
 - i. Ann yawned
 - j. Ann saw him, etc.

With an appropriate number of MAX-IO violations (precluding the underlying input form to appear overtly) and DEP-IO violations (introducing material that normally denotes something else) we can arrive at every conceivable word string, no matter what the input is. At first sight, such an OT system clearly seems computationally intractable due to an uncontrollable candidate space. As will be shown in the next section, the LFG-based conception of OT syntax provides a natural framework for modelling the intuitions about faithfulness violations addressed in this section in a way that allows one to structure the candidate space adequately for computational processing.

3 Optimality-theoretic LFG: the subsumption-based conception

The starting point for the LFG-based framework for OT syntax, due to (Bresnan 1996; Bresnan 1998), is the observation that the following two intuitions underlying candidate generation can be captured in a formally precise way using LFG:

- All candidates satisfy certain inviolable principles;
- competing candidates are alternative realizations of the input.

The inviolable principles can be encoded in a formal grammar, and using a formalism with a structural representation abstracting away from language-specific realization issues (f-structure), we also have a way of making the role of the input in candidate generation explicit. Inputs are simply formalized as (not yet fully specified) f-structures.

Then the candidates for a given input can be defined as

- analyses of an LFG grammar encoding the inviolable principles (call it G_{inviol})
- containing (i.e., being subsumed by) the input in their f-structure.

More formally, we have the following definition of *Gen* (cf. Kuhn 2000a; Kuhn 2000b – the language generated by an LFG grammar is defined as a set of c-structure/f-structure pairs $\langle T, \Phi \rangle$):

(8) Definition of *Gen*

Φ_{in} : input representation,

$Gen(\Phi_{in}) = \{ \langle T, \Phi' \rangle \in L(G_{inviol}) \mid \Phi_{in} \sqsubseteq \Phi', \text{ where } \Phi' \text{ contains no more semantic information than } \Phi_{in} \}$

This means that candidates may monotonically add (non-semantic) information to the input f-structure, plus they each specify a particular c-structure.

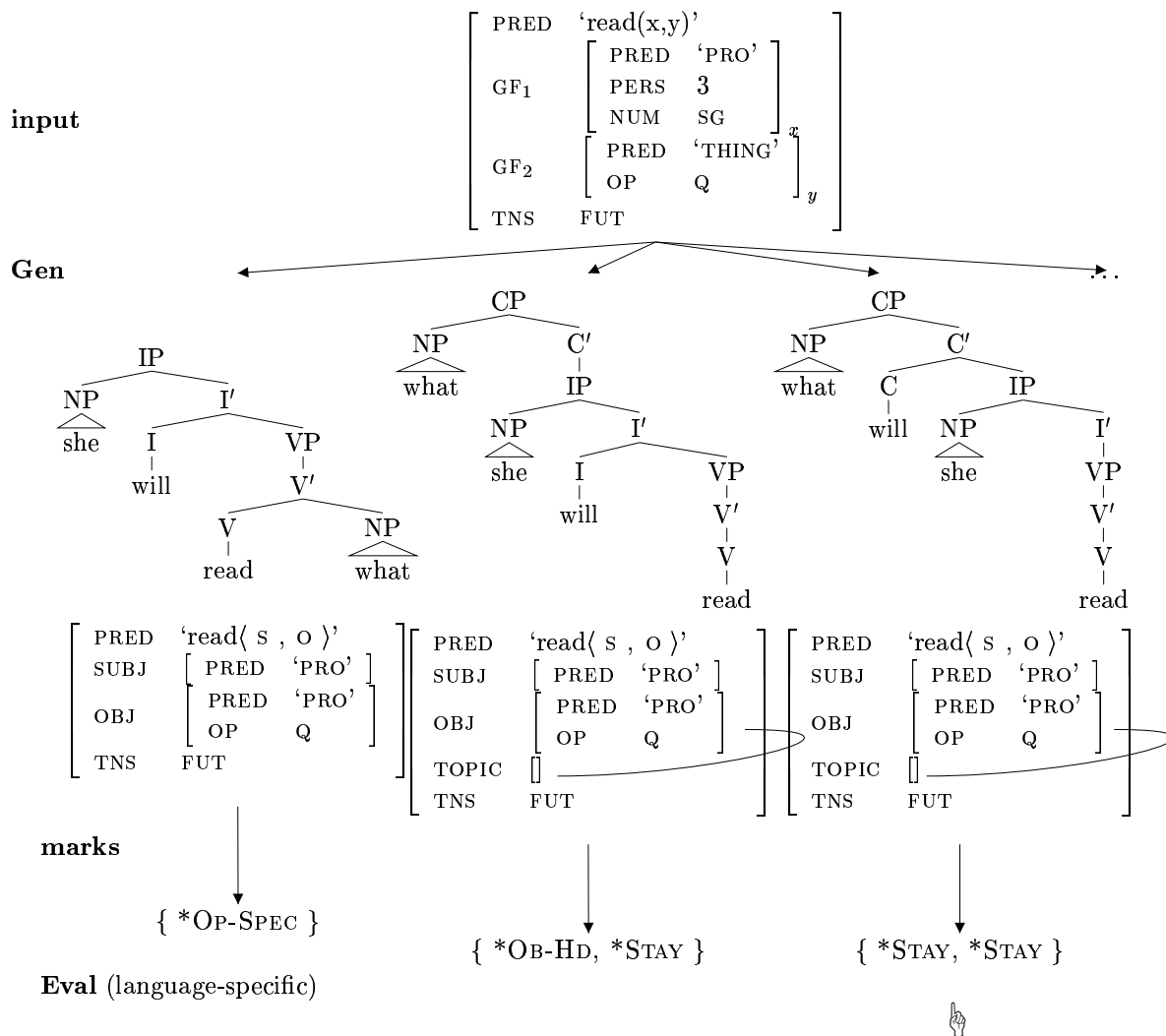


Figure 1: Illustration of the OT-LFG model

With this conception of *Gen*, the Markedness constraints can be formulated straightforwardly as descriptions of structural configurations in the candidate c-structure/f-structure pairs (for more discussion, see Kuhn 2000a, sec. 3.2). So, the standard OT definition of *Eval* (which has been much more explicit in the literature than the definition of *Gen*) can be applied. The entire OT-LFG system is illustrated graphically in fig. 1. The inviolable principles underlying candidate generation are essentially an extended X-bar theory (Bresnan 2000,

ch. 7), the three violable constraints at work (listed in (9)) are taken from Bresnan’s (1998) LFG-based reconstruction of (Grimshaw 1997). (Application of the violable constraints is called *marks*.)

- (9)
- | | |
|---------|--|
| OP-SPEC | An operator must be the value of a DF in the f-structure. |
| OB-HD | Every projected category has a lexically filled [extended, JK] head. |
| STAY | Categories dominate their extended heads. |

The language generated by an OT system is thus defined as follows (note the existential quantification over input representations, which will be of importance when looking at the parsing/recognition task in sec. 4.2):

- (10) *Language generated by an OT system*

A string w is contained in the language defined by an OT system

iff there exists an underlying input representation Φ_{in} s.th. w is the terminal string of the optimal candidate in $Gen(\Phi_{in})$.

3.1 Faithfulness violations in OT-LFG

Let us come back to the faithfulness violations addressed in sec. 2. Following the methodological principle (1), we do not want to *exclude* overly unfaithful candidates from the candidate set, building some limit into the definition of Gen .³ The fact that overly unfaithful candidates play no role when it comes to finding the most harmonic candidate should follow from constraint interaction alone. So, Gen should provide arbitrarily serious faithfulness violations.

Definition (8) looks very restrictive, with the subsumption condition disallowing the deletion of input information (as seems to be required for modelling MAX-IO violations), and an additional clause excluding the addition of semantic information (cf. DEP-IO violations/epenthesis). However, it is a crucial point of the approach taken here that this restrictive definition of Gen is kept up – it will be the basis of keeping control over the candidate in processing. The intended faithfulness violations *can* indeed be captured within the limits of this definition, by regarding unfaithfulness as a tension between f-structure and the categorial/lexical realization:

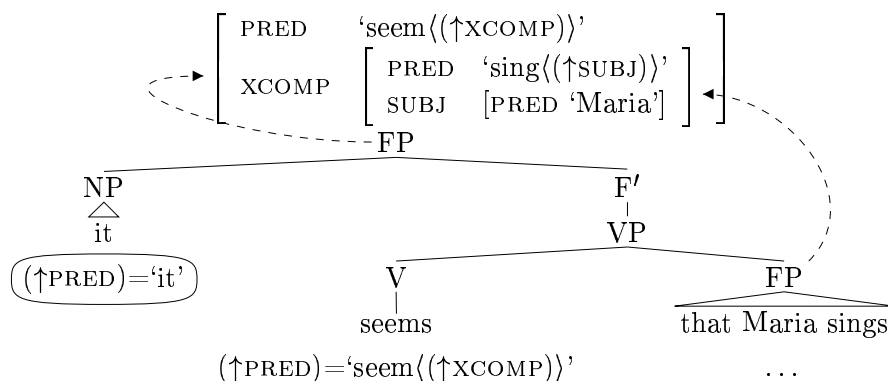
At f-structure, semantic information may neither be added nor removed. C-structure on the other hand may contain material without an f-structure reflex (epenthesis), or leave f-structure information categorially unrealized (deletion). This is illustrated in the following examples ((11)–(14)). Below the lexical entries, the ‘morpholexical constraints’ introduced by the lexical item are shown. Standardly, these functional annotations are treated exactly the same way as annotations in grammar rules, i.e., after instantiation of the meta-variables (\uparrow) they include, they contribute to the overall set of f-descriptions the minimal model of which is the f-structure.

³This would of course be possible, and is a sensible thing to do from a practical point of view. But for our understanding of the theoretical underpinnings of OT syntax we want to be sure that the system will also work otherwise.

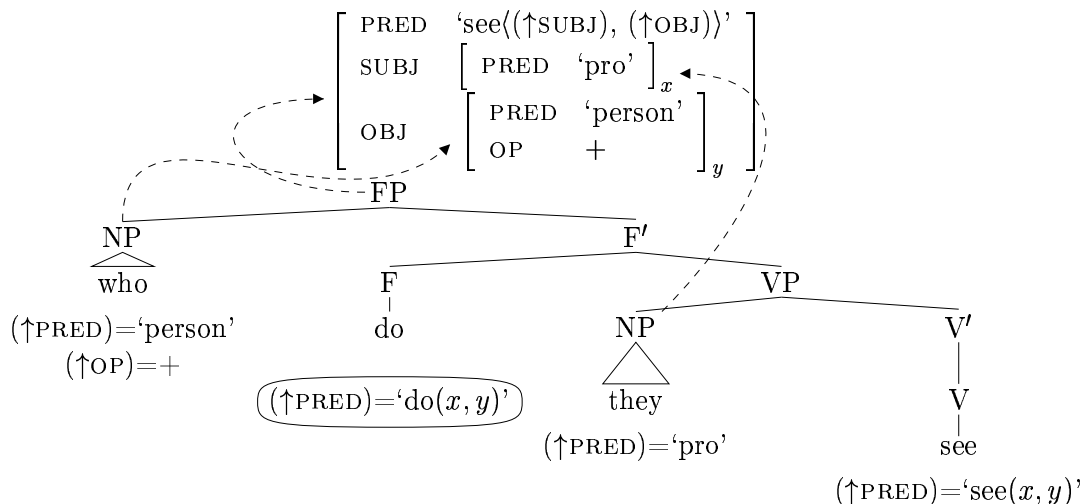
As Bresnan (1998) discusses for the expletive *do*, the DEP-IO-violating use of a lexical item can be modelled by assuming that (part of) its morpholexical contribution is not actually used in the construction of the f-structure. This is illustrated by encircling the respective morpholexical constraint.⁴

(11) is an example of an expletive use of the pronoun *it* in English, as assumed, e.g., in (Grimshaw and Samek-Lodovici 1998, sec. 4). (12) is the well-known example of the expletive *do*. Note that in contrast to classical LFG, in both these cases the ordinary lexicon entry is used (i.e., referential *it*, and full verb *do*). It is merely used in an unfaithful way.⁵

(11) Violation of DEP-IO (3) (or FILL; Grimshaw: FULL-INT)



(12)



MAX-IO violations are the opposite situation. Some part of the f-structure (reflecting the input) isn't contributed by any of the lexical items' morpholexical constraints. In the examples, this is highlighted by encircling the respective part of the f-structure. (13) is a

⁴There are various ways how this can be formalized more rigorously. In (Kuhn 2000a, sec. 3.3.2), I introduce a special λ -projection from c-structure to l-structure. L-structure comprises *all* morpholexical constraints, some of which may not be reflected in f-structure – thus faithfulness constraints can be formulated as structural conditions on c-structure/l-structure/f-structure triples. In a set-up that doesn't assume a strict modular split between *Gen* and *Eval* this extra projection is superfluous. Faithfulness violations can be registered along with lexical access.

⁵In this paper, I do not address the question why it is not some other lexical item that is chosen and used unfaithfully. See (Kuhn 2000a, sec. 3.3.1) for some discussion.

pro-drop example from Italian. Note that – again as opposed to classical LFG – the PRED value of the subject is not introduced by the inflection on the verb; it simply arises “from nothing” as a faithfulness violation.

(13) Violation of MAX-IO (4) (or PARSE)

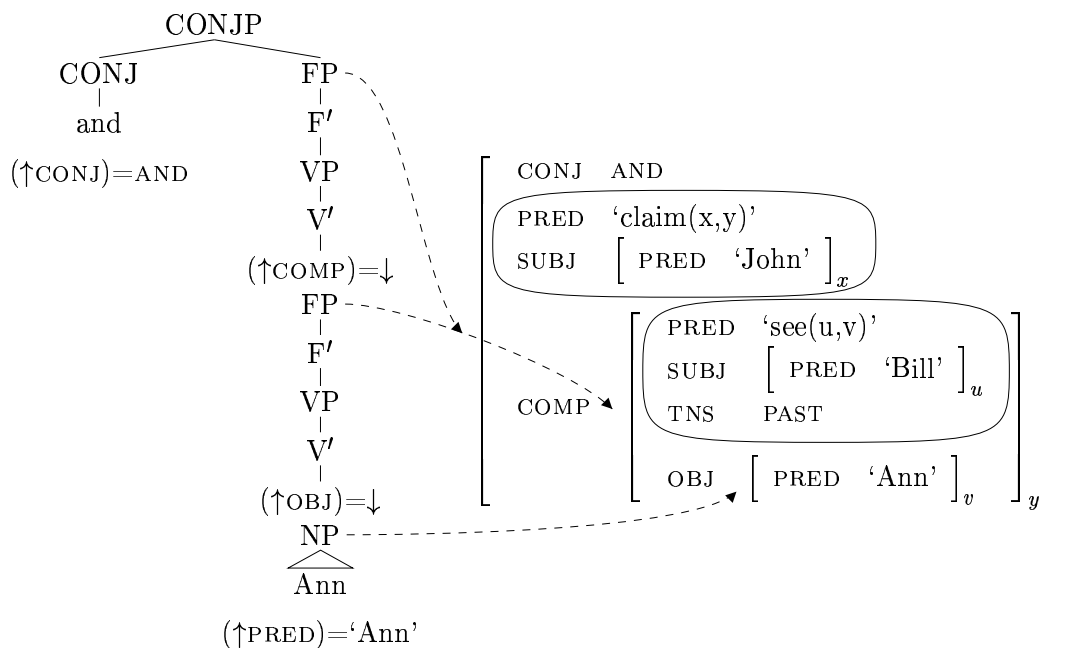
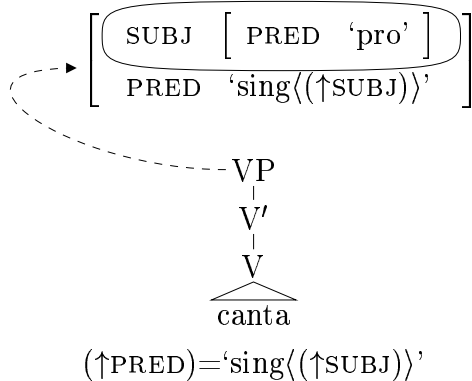


Figure 2: MAX-IO-unfaithful candidate in an ellipsis account

With such MAX-IO violations being part of the candidate space, it becomes conceivable to set up an OT account of ellipsis that explains the (im)possibility of ellipsis in context as an effect of constraint interaction. Let us look at the candidate in fig. 2 as one such MAX-IO-unfaithful candidate. It is the c-structure/f-structure analysis assumed for B’s reply in dialogue (14).⁶ This example is interesting since it illustrates the need for arbitrarily large portions of dropped material (the recursive embedding in A’s utterance could be arbitrarily

⁶The representation builds on L. Levin’s (1982) analysis of sluicing, assuming that at f-structure, the antecedent structure is fully reflected.

deep, which would have to be reflected in the f-structure for B’s reply, according to the account assumed here).⁷

(14) A: John claimed that Bill saw Sue.

B: And Ann.

A short digression on the constraint set required for such an ellipsis analysis: It is quite clear how we can make the candidate in fig. 2 win over less elliptical competitors like *and that Bill saw Ann*, or even *and John claimed that Bill saw Ann*: the assumption of an Economy-of-expression constraint like *STRUCT outranking MAX-IO will do the job – the elliptical utterance is as expressive, using less c-structural material. However, this immediately raises the question how to make sure that Economy of expression does not fire all the time, wiping out most if not all of the linguistic material. Intuitively it is quite clear that only contextually *recoverable* material may be ellided, but this idea has to be implemented more formally. A rather straightforward way is to assume a constraint REC that is violated when some material is left unrealized without there being an antecedent in the local context (cf. Pesetsky 1998). Note that the architecture of the OT system has to be extended in order to make the extra-sentential context visible for the OT constraints (a similar modification would be required in other approaches to capture recoverability too).⁸

To sum up this section, the intuitive way of looking at the relation between the input and the candidates in OT-LFG should be as follows: What is characteristic of an individual candidate is its lexical material and c-structure; a candidate’s f-structure is mostly a reflex of the input.⁹ Input-output faithfulness amounts to comparing a candidate’s f-structure with its morpholexical constraints. Thus one may call this the “lexicalist view of faithfulness” (cf. Kuhn 2000a).

3.2 Varying the input to optimization

Before addressing processing issues in view of the definition of candidate generation (*Gen*) discussed above, it should be noted that a parameter in this definition can be modified. This will give us a formal system with very similar properties, which may however be used to model a different empirical concept.

So far, we have followed the standard application of OT as a definition of the grammatical structures of a language. What is kept constant across candidates is (more or less) the part

⁷The non-branching dominance chain dominating *Ann* in c-structure (which would be excluded by offline parsability in classical LFG) reflects the assumption that the path to [PRED ‘Ann’] in the f-structure cannot arise as an effect of further MAX-IO violations. Instead, the standard X-bar annotation principles are at work, introducing the COMP and OBJ embedding (and along the way, an Economy-of-expression constraint like *XP or *STRUCT is violated). It is however conceivable to devise an account that assumes more compact c-structures on the one hand, but further MAX-IO on the other.

⁸The condition that the REC constraint checks for is rather complicated, so one may hope to replace it by simpler constraint, interacting. This becomes possible in a bidirectional optimization framework as discussed below.

⁹In particular, faithfulness violations cannot lead to the situation that a candidate has a different meaning than the meaning encoded in the input. This excludes a derivation of language-particular ineffability in the style of (Legendre, Smolensky, and Wilson 1998), which works with LF-unfaithful winners. Ineffability is however derivable through bidirectional optimization, without assuming LF-unfaithfulness.

of the structure that defines the meaning. The competing candidates are thus synonymous (potential) realization alternatives. In the procedurally flavoured standard terminology, this is called *production-based optimization*, since *Gen* is in fact defined as an abstract production function.

Alternatively, we may let the f-structure vary freely across candidates and rather leave the terminal string of the c-structure constant.¹⁰ The competing candidates are thus alternative parses of the same string, and we have *comprehension-based optimization*. Overloading the function name *Gen* to also cover the analogous string-based candidate generation function, we get the definition (15). Hendriks and de Hoop (1999) use a comprehension-based optimization model in what they call OT semantics; the winning structure models what native speakers conceive as the preferred reading of a sentence.¹¹

(15) Definition of *Gen* for comprehension-based optimization

w: string,

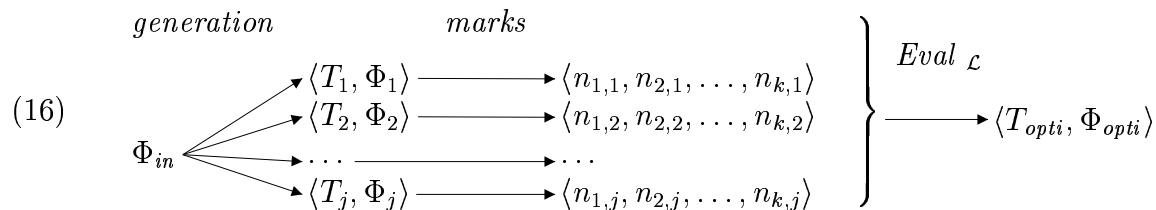
$$Gen(w) = \{ \langle T, \Phi \rangle \in L(G_{inviol}) \mid w \text{ is the terminal string/yield of } T \}$$

A stronger criterion for grammaticality can be formulated if the two concepts are combined conjunctively: the successful candidate must be optimal both among all structures with the same underlying form and among the structures with the same string. This is the underlying idea of *bidirectional optimization*, which will be addressed more extensively in sec. 5.

4 Processing OT-LFG

Let us now turn to the question whether computational procedures can be devised for tasks based on the formal system defined in sec. 3. We will start with the generation task for a production-based optimization system: given an underlying form, what is the optimal candidate according to an OT system? I will call this task (A1).

The initial idea how to approach this task is quite obvious: we can follow the definition of the OT-LFG system illustrated in fig. 1, using standard LFG processing techniques (cf. Kuhn 2000a for a detailed discussion): (i) generate from the input f-structure, using the LFG grammar for inviolable principles; (ii) apply constraints to the candidates (this gives us a sequence of constraint violation counts for each candidate); (iii) pick the most harmonic candidate:



¹⁰Formally, all kinds of other criteria for specifying the candidate set are conceivable. In Minimalism-influenced work in OT syntax, the candidate is often assumed to be defined by a common numeration, i.e., an unstructured bag of lexical items. However, using the underlying (logical) form on the one hand and the surface string on the other has a much clearer conceptual motivation in the broader cognitive context.

¹¹Comprehension-based optimization also plays a role in learning. Tesar and Smolensky (1998) assume it as *robust interpretive parsing* (cf. also Smolensky 1996). It is also being applied as a preference mechanism in the large-scale LFG grammars developed in the Pargram project (Frank, King, Kuhn, and Maxwell 1998; Frank, King, Kuhn, and Maxwell 2000).

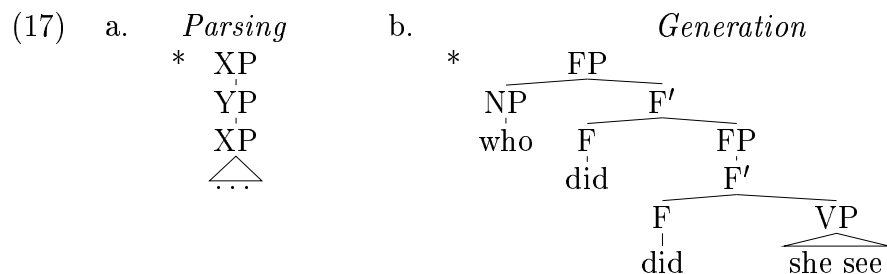
For the parsing task with a comprehension-based optimization system (call it (B1)), the same set-up suggests itself, only starting with a string and applying standard LFG parsing rather than generation as the initial step. But are these obvious approaches possible, given the faithfulness violations allowed by *Gen*?

4.1 Infinite candidate sets in processing

For each of the two directions of optimization, one of the faithfulness constraints is a processing issue (when violated): In *production-based optimization*, DEP-IO violations (epentheses) create an infinite number of possibilities for generation. MAX-IO violations (deletions) are no problem, since there is only a finite amount of information in the input to delete. Vice versa in *comprehension-based optimization*, MAX-IO violations create an infinite number of possibilities for parsing, whereas DEP-IO violations are unproblematic, as there is only a finite number of string elements that could have been inserted.

Why doesn't the problem arise in classical LFG parsing and generation? The bare formalism – a combination of a c-structure grammar as an unrestricted context-free grammar and f-structure projected from c-structure – does actually allow for an infinite number of different structures over a given terminal string: if the context-free grammar contains a rule recursion that can be used in a non-branching way, there are arbitrary many c-structures, including zero to n recursions. To ensure decidability of the parsing task, such recursions are excluded by definition: the offline parsability condition (Kaplan and Bresnan 1982, 266) basically says that if there is a potential recursion in a nonbranching chain, the structure passing this recursion zero times is the only valid LFG structure (see (17a)).

For the classical generation task, there is a parallel issue to take care of: in a unification-based framework, the same feature information can arise from arbitrary many c-structural places, all being unified together. To guarantee decidability of the generation task, an *offline generability* condition has to be assumed, again excluding vacuous application of rule recursion, here with reference to resourced feature structures (cf. the example in (17b), assuming that *did* doesn't introduced a resourced PRED value).¹²



Now, coming back to OT processing, in a set-up that is supposed to reflect the methodological principle (1) we cannot base the candidate generation step on the classical parsing/generation tasks with the offline parsability/generability conditions. The procedure for *Gen* should create an infinite candidate space, and only constraint application acts as a restrictive device. For this to work, we obviously need a procedure where the OT constraints

¹²See (Wedekind 1999; Kaplan and Wedekind 2000) for discussion; the use of an offline generability condition is currently explored by the XLE group at Xerox PARC.

are checked *online*, along with candidate generation. The sequential set-up (16) cannot work, since the first step wouldn't terminate.

Such a procedure is proposed in (Kuhn 2000b), using a chart for generation and parsing (see below). Since the constraints do the work of limiting the search space, we have to be sure that the system contains adequate constraints that will differentiate the candidates arising through rule recursion. We may even call this the *relaxed offline parsability/generability* condition:

(18) *Relaxed offline parsability/generability*

A rule recursion may only be applied if at least one constraint violation is incurred by the recursive structure.

Note that it is quite easy to guarantee for an entire OT system that all candidates satisfy (18): a sufficient condition is that the set of constraints include the Economy-of-expression constraint *STRUCT.

Markedness constraints A decidability problem may still arise if the constraint checking cannot be performed at intermediate points in time during candidate generation. Therefore, the constraints, in particular the Markedness constraints may not be of arbitrary complexity.

Recall that the conditions which markedness constraints check for are expressed as structural descriptions of (parts of) candidate representations. We saw some examples in (9). In order to ensure decidability, we assume the following restriction:

(19) *Restriction on constraints:*

The structure (c-structure/f-structure) denoted by the constraint condition must be bounded.

Note that this restriction is fully compatible with the methodological principle (1) – although here a different facet of the principle is relevant than before: trying to explain as much as possible as an effect of constraint *interaction* means that we're not interested in very expressive *individual* constraints; the explanatory power should really arise out of the *interaction* of several simple constraints (cf. also Grimshaw 1998, making the same point).

Chart-based optimization Here, I will not go into the details of the chart-based optimization approach. A brief summary should be enough. Tesar (1995) proposes a chart-based OT algorithm for generation with regular grammars and context-free “position grammars”. This basic idea is extended to OT-LFG in (Kuhn 2000b), using Earley deduction parsing and generation (following Neumann 1994; Neumann 1998).

The strategy is to store the constraint profile of (partial) constituents in the chart edges. Whenever a constraint may or may not apply, both options are entered into the chart. The assumption of relaxed offline parsability/generability (18) and the boundedness of constraint conditions (19) ensure that recursions not helping to avoid some local constraint violation lead back to an already existing edge.

When an identical edge exists in the chart, the new edge is considered as blocked as usual in chart parsing/generation – however only if the new edge is equally or less harmonic as the existing one. If it is more harmonic, the new constraint profile is propagated through

the chart, which will potentially lead to further options.¹³ Due to this online processing of constraints, the algorithm can deal with an infinite candidate set.

4.2 Directionality in processing

So far, we have looked at procedures for

- (A1) the generation task for production-based optimization (typically modelling grammaticality); and – symmetrically –
- (B1) the parsing task for comprehension-based optimization (typically modelling preference).

This leaves open the respective *recognition tasks*. In a production-based optimization model, we want a procedure telling us whether a given string is contained in the language defined by the OT model. We may also want to know what the correct structure for this string is, which is what Johnson (1998) calls the *universal parsing task*. Recognition and parsing (for production-based optimization) are closely related. Let's refer to both tasks as (A2). According to the definition of the language generated by an OT system (10), the (A2) tasks amount to checking whether there is *some* underlying input representation for which the string is the optimal candidate.¹⁴ From the previous discussion it should be clear that the (A2) task is different from (B1) where we merely choose between alternative parsing analyses of a particular string (for more discussion see (Johnson 1998) and (Kuhn 2000a)). During parsing, (A2) requires a (“backwards”) generation step, since in the production-based optimization model grammaticality is defined that way. With this forward and backward processing involved, we may call the procedure a bidirectional processing procedure. (20) summarizes the steps required; fig. 3 (taken from Kuhn 2000a) illustrates the process graphically for an abstract example (parsing the string ‘a b c’). Note that a given string may have no or many grammatical readings.

(20) *Bidirectional processing*

Task: determine whether a given string is grammatical according to production-based optimization

- parse string to determine possible underlying forms
- “backward generation” from underlying forms
- optimal candidate in a generation-based competition determines grammaticality
- string in optimal candidate has to match the initial string; else, initial string is not grammatical for this particular underlying form

In the context of the present paper we have to ask: Is task (A2) also decidable?

¹³Clearly, the computational complexity of this algorithm is considerable, but the point is just to show that a decidable procedure can be specified. For a more efficient system, various optimizations could be attempted.

¹⁴For comprehension-based optimization (B), the parallel task – (B2) – may be intuitively less interesting (given a logical form, is it the preferred reading for some string in the language under consideration?). I will thus focus on the (A2) task.

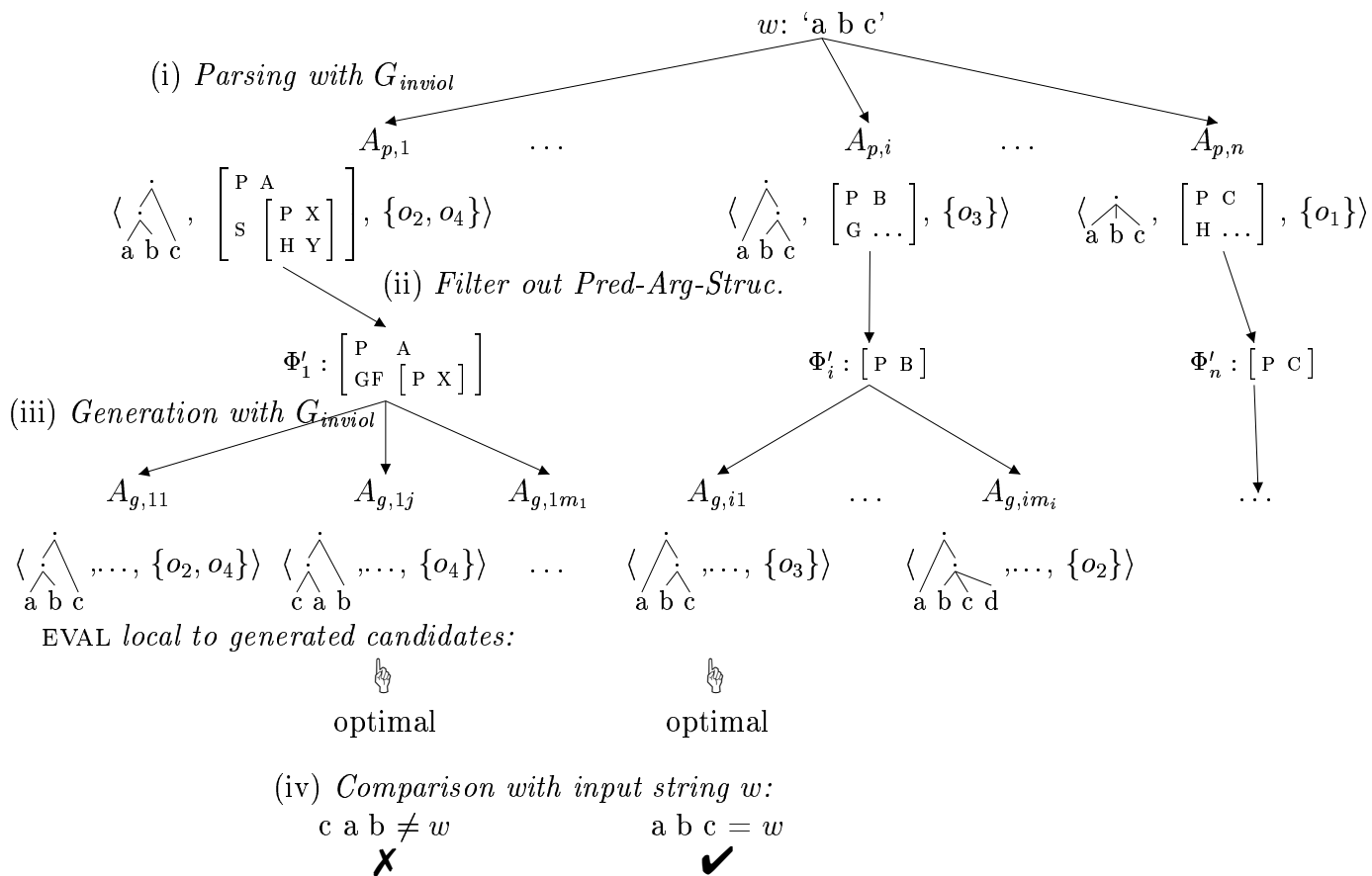


Figure 3: Parsing and “backward generation”

4.3 More infinity issues

For the simpler (A1) task – basically a one-way generation task – we noted above that MAX-IO violations (deletions) don’t pose a decidability problem because there is only a finite amount of information that can be dropped. With the initial parsing step preceding “backward generation” in the bidirectional (A2) task, MAX-IO violations *do* become an issue: an infinite space of potential underlying forms has to be considered.

There are two options for dealing with the situation:

- assume that the space of underlying forms is restricted by some *recoverability principle*¹⁵ (ensuring that only a finite number of *contextually recoverable* options has to be considered), or
- try to derive the effect of such a recoverability principle as a consequence of constraint interaction.

¹⁵Note that the assumption of a violable constraint REC (Pesetsky 1998; discussed in sec. 3.1) does not help to avoid the decidability problem in the parsing direction.

The latter option seems much more in the spirit of the methodological principle of OT (1), so it is this one we will adopt here (I will briefly come back to the other option in the conclusion, sec. 6). It forces us to assume that not only the generation direction of (A2), but also the initial parsing direction is controlled by an OT-style optimization (since it is in this step that we want to demand recoverability). So we get a system that involves not only bidirectional processing, but *bidirectional optimization*.

The intended effect of the constraint interaction modelling recoverability is that during the initial parsing step in (A2) just those MAX-IO violations are postulated (i.e., possible with the optimal candidate) which are justified through an antecedent in the local context (which is finite). This effect can be reached quite simply by assuming that we have a kind of alignment constraints comparing the underlying f-structure for the input string with the given context; these constraints disprefer any kind of divergence. Now, for overt material expressing new information (i.e., diverging from context) there is no alternative to violating the context alignment constraints, but for everything that is not overtly expressed (the MAX-IO violations), the most harmonic option will always be postulating that what's being dropped *does* align with the local context.

5 Bidirectional optimization

Bidirectional optimization has been argued for variously in the theoretical OT literature, on empirical and conceptual grounds (see, e.g., Wilson 1998; Boersma 1998; Smolensky 1998; Lee 2000; Morimoto 2000; Kuhn 2000a; Blutner 2000). It is thus interesting that independent of all this, there are computational arguments for such a model – of course depending on the assumptions about faithfulness violations made in this paper, and taking the methodological principle (1) rather seriously.

With bidirectional optimization, will the (A2) task actually be decidable? If both (A1) and (B1) are decidable, resulting in a finite set of winning candidates,¹⁶ then the bidirectional optimization task is decidable too, since it can be solved by applying (A1) after (B1). Given a string, (B1) is used to determine the optimal candidate(s)¹⁷ according to comprehension-based optimization, then (A1) determines the optimal candidate(s) for the (B1)-winner(s), according to production-based optimization. Ultimately, the terminal string of the (A1)-winner(s) is compared with the initial string (as in (20)), and if they match we have found the bidirectionally optimal candidate. (In the Earley-style chart implementation of (Kuhn 2000b) – following Neumann's (1998) interleaved generation/parsing approach – the strictly sequential set-up need not be kept up, so intermediate backward processing steps can be performed rather early on given chart edges, allowing to prune off erroneous search paths early.)

Note that this scheme implements a strong concept of bidirectional optimality. The successful candidate has to be independently optimal according to both production-based and comprehension-based optimization. So we have changed the definition of the language generated by the OT system:

¹⁶The formal proofs need yet to be written down, so there may be some tacit assumptions underlying the approach discussed above and in (Kuhn 2000b), but this shouldn't affect the overall result.

¹⁷There may be more than one equally harmonic candidate, but due to the relaxed offline parsability/generability condition (18), there will only be finitely many.

(21) *Language generated by a strongly bidirectional OT system*

A string w is contained in the language defined by an OT system

iff it is the optimal candidate in $Gen(w)$

and there exists an underlying input representation Φ_{in} s.th. w is the terminal string of the optimal candidate in $Gen(\Phi_{in})$.

This is not the only conceivable option of combining the two directions of optimization: one may want define the candidate set of one (or both) of the two individual directions as depending on the result of the other optimization. For example, if the idea is kept up that (A1) models grammaticality, while (B1) models preference, the following dependence would make intuitive sense: grammaticality is defined based on (A1) only (i.e., independently of preference), while preference is determined between grammatical candidates. So, if we start out with a string, we have to find all potential underlying forms and check for each of these whether the given string is actually the optimal candidate (since this is the way grammaticality is defined). If there are several options, we compare them to determine the preferred reading. This weaker and asymmetrical scheme was assumed in (Kuhn 2000a, sec. 4.2). The original definition of the language generated by the OT system (10) is left unchanged, assuming that preference is a concept subordinate to grammaticality.

Blutner (2000) proposes a symmetrical concept of *weak bidirection* (contrasting it with the type of *strong bidirection* I have discussed above), assuming mutual dependence of the candidate sets. Impressionistically, one may envisage optimization according to this concept as an inductive process, alternately running (A1) and (B1) optimization. If a candidate wins both directions, it is an acceptable option for the language modelled and is removed from the candidate sets for further induction steps; thus, candidates that couldn't win under strong bidirection can become winners after their competitor has been retracted. Blutner argues that this concept is useful for deriving partial blocking effects in lexical pragmatics (using abstract examples at this stage, rather than detailed empirical examples).

A straightforward way of guaranteeing decidability (with the infinite candidate sets we are confronted with, assuming faithfulness as a violable constraint) exists only for the strong bidirectional optimization model.¹⁸ The decidability problem for the asymmetrical model may become intuitively clear when we go through the (A2)-type parsing of an ungrammatical string involving the potential for arbitrarily many MAX-IO violations: after initially parsing the string, we pick the optimal parsing analysis, applying backward generation (i.e., production-based optimization, modelling grammaticality) to its underlying form. Since the string is ungrammatical, we don't get back to the initial string. In the strong bidirectional case, we would already be finished, but since in the weaker account the comprehension-based

¹⁸It is conceivable that a way of controlling the infinite candidate space in the parsing direction can be found that doesn't throw out candidates that are non-optimal according to comprehension-based optimization (this means that the strong bidirectional interpretation wouldn't be enforced). One would have to find a systematic way of constructing a more harmonic backward competitor for arbitrary parsing analyses constructed through recursions beyond a certain point. Having such a recipe would show that all the interpretations underlying the recursive structures are out of the question (in terms of production-based optimization!) for the string being processed.

Such a procedure could be applied both to unidirectional optimization models and at least to the asymmetrical weaker bidirectional optimization model.

direction models just preference (grammaticality being a stronger requirement), we have to consider the next best parsing candidate, etc. Even after trying the best n underlying forms in the backward optimization, we won't have found a grammatical candidate, but there are infinitely many possibilities, so there is no point where we can be sure that the string is ungrammatical.

Potential problems with strong bidirectional optimization It is not so clear whether the conjunction of the two directions of optimization in the definition of grammaticality is fully desirable from a linguistic point of view. Since the candidate set in comprehension-based optimization is defined on the basis of a common terminal string and the candidates thus differ in meaning, there will be interference of extra-linguistic factors such as world knowledge about what is more plausible etc. For example, word order freezing effects as discussed in (Lee 2000) and (Kuhn 2000a, sec. 4.2) as an application of bidirectional optimization can be overruled by world knowledge.

Of course, it is not necessarily a bad idea to try and model the overall cognitive process of language understanding as some optimization task starting from a perceived speech signal (cf. the broader cognitive scope of Harmony Theory, Smolensky 1986), but as discussed in the introduction, it is an hypothesis of the OT approach that the restricted structure of the OT model is particularly well suited to explain the language faculty. If extra-linguistic cognitive processes are modelled by an optimization process, one wouldn't expect the possibility of systematic re-ranking of constraints with a factorial typology predicted.

Since according to *strong* bidirectional OT, comprehension-based optimization with its extra-linguistic aspects is involved in the definition of the language generated by the OT system, there is no clear way of identifying the scope of a *linguistic* theory as part of the overall cognitive system. Maybe some way of separating out the linguistic part of the comprehension-based optimization can be found. However until this has been clarified, strong bidirectional optimization is presumably inadequate for an explicit formal account of larger sets of data.

The problem can be illustrated with the derivation of the recoverability principle discussed at the end of sec. 4.3: the *strong* bidirectional model forces us to adjust the constraints in a way that makes the contextually adequate candidate optimal in both directions. Now, most non-trivial sentences have more than reading. For the strong model to work we have to assume an intricate conspiracy of constraints that gives us exactly the right reading as the optimal one in parsing. Finding such constraints is clearly not just a linguistic issue. For the (asymmetrical) weaker model in contrast, it would be enough to exclude those candidates from the parsing possibilities that are unfaithful beyond recoverability – however, this model will not guarantee decidability of the parsing task.

6 Possible conclusions

Let us briefly review the reasoning in this paper: based on the methodological principle of OT to try and explain as much as possible as an effect of constraint interaction (1), it was argued that a limit on the unfaithfulness of candidates should not be built into the definition of *Gen*, but should follow from constraint interaction. In a chart-based optimization algorithm, it is actually possible to keep the candidate space under control in this way, provided the critical processing direction undergoes optimization. The standard OT definition assumes

optimization just for the production or generation direction, i.e., parsing with arbitrarily unfaithful candidates poses a decidability problem.

There are (at least) three possibilities of reacting to this problem: first (the option that was mainly explored in this paper), changing the definition of the language generated by an OT system to also include comprehension-based optimization. This way, decidability can be guaranteed and principle (1) is kept up. However it is not so clear whether this extended application of constraint interaction really follows the same restricted scheme of optimization (which includes that one expects predictions on the basis of factorial typology). So, it is not so clear whether we should really regard the second option as refuted: deciding not to follow principle (1) for controlling the space of MAX-IO violations. We could assume a more restrictive definition of *Gen* with a built-in recoverability condition, so comprehension-based optimization would not be required for guaranteeing decidability (this does not exclude the application of comprehension-based optimization to model a different concept like preference).

There is a third possibility: acknowledge undecidability of the parsing task in the general case. This would mean saying that there can be strings for which the parsing (and backward generation) procedure runs forever. Recall the situation of parsing an ungrammatical string. The first n underlying forms have been considered without success, but there are infinitely many possibilities, so there is no point where we can be sure that the string is ungrammatical. In practice one would of course adopt a heuristics enforcing a decision after some finite number of steps – at the risk of wrongly excluding a string that is actually grammatical.

If we look at what is actually being modelled by the theoretical concept of grammaticality – namely acceptability judgements of native speakers –, this implication of the third possibility seems rather plausible. Recall under what circumstances candidates that are heavily unfaithful to MAX-IO (like the one in fig. 2) can turn out to be winners: it is when the context allows ellipsis of large chunks of the underlying (input) form.

Now, looking at the human sentence processor in such a situation is quite revealing: as is well-known when presented with elliptical utterances out of context, our processor breaks down surprisingly fast – in a certain sense. Sentences are judged unacceptable that would be considered perfect if the context was known. For example,

(22) Bill for the doctor's

is likely to be judged ungrammatical if no appropriate context (like *Has anyone left early today?*) is provided (cf. e.g., Klein 1993 for discussion and further examples).

So, the human sentence processing system displays a behaviour suggesting that something like heuristics we just discussed are at work. So undecidability of the parsing task may not be something we have to avoid at any cost.

References

- Blutner, R. (2000). Some aspects of optimality in natural language interpretation. Ms.
- Boersma, P. (1998). *Functional Phonology. Formalizing the interactions between articulatory and perceptual drives*. Ph. D. thesis, University of Amsterdam.
- Bresnan, J. (1996). LFG in an OT setting: Modelling competition and economy. In M. Butt and T. H. King (Eds.), *Proceedings of the First LFG Conference*, CSLI Proceedings Online.
- Bresnan, J. (1998). Optimal syntax. In J. Dekkers, F. van der Leeuw, and J. van de Weijer (Eds.), *Optimality Theory: Phonology, Syntax, and Acquisition*. Oxford University Press. To appear.
- Bresnan, J. (2000). *Lexical-Functional Syntax*. Blackwell. to appear.
- Frank, A., T. H. King, J. Kuhn, and J. Maxwell (1998). Optimality theory style constraint ranking in large-scale LFG grammars. In M. Butt and T. H. King (Eds.), *Proceedings of the Third LFG Conference*, CSLI Proceedings Online.
- Frank, A., T. H. King, J. Kuhn, and J. Maxwell (2000). Optimality theory style constraint ranking in large-scale LFG grammars. Ms., to appear in Peter Sells (ed.), *Formal and Empirical Issues in Optimality-theoretic Syntax*, Stanford: CSLI Publications.
- Grimshaw, J. (1997). Projection, heads, and optimality. *Linguistic Inquiry* 28(3), 373–422.
- Grimshaw, J. (1998). Constraints on constraints in optimality theoretic syntax. Manuscript, Rutgers University.
- Grimshaw, J. and V. Samek-Lodovici (1998). Optimal subjects and subject universals. In P. Barbosa, D. Fox, P. Hagstrom, M. McGinnis, and D. Pesetsky (Eds.), *Is the Best Good Enough?*, pp. 193–219. MIT Press and MITWPL.
- Hendriks, P. and H. de Hoop (1999). Optimality theoretic semantics. *Linguistics and Philosophy*. to appear.
- Johnson, M. (1998). Optimality-theoretic Lexical Functional Grammar. In *Proceedings of the 11th Annual CUNY Conference on Human Sentence Processing, Rutgers University*. To appear.
- Kager, R. (1999). *Optimality Theory*. Cambridge: Cambridge University Press.
- Kaplan, R. M. and J. W. Bresnan (1982). Lexical-functional grammar: a formal system for grammatical representation. In J. W. Bresnan (Ed.), *The Mental Representation of Grammatical Relations*, Chapter 4, pp. 173–281. Cambridge, MA: MIT Press.
- Kaplan, R. M. and J. Wedekind (2000). Lfg generation produces context-free languages. In *Proceedings of COLING-2000*, Saarbrücken, pp. 297–302.
- Klein, W. (1993). Ellipse. In J. Jacobs, A. von Stechow, W. Sternefeld, and T. Vennemann (Eds.), *Syntax: An International Handbook of Contemporary Research*, pp. 763–799. Berlin: de Gruyter.
- Kuhn, J. (2000a). Generation and parsing in optimality theoretic syntax – issues in the formalization of OT-LFG. Ms., IMS, University of Stuttgart, to appear in Peter Sells (ed.), *Formal and Empirical Issues in Optimality-theoretic Syntax*, Stanford: CSLI Publications.
- Kuhn, J. (2000b). Processing optimality-theoretic syntax by interleaved chart parsing and generation. ACL-2000.
- Lee, H. (2000). Markedness and word order freezing. Ms., Stanford University, to appear in Peter Sells (ed.), *Formal and Empirical Issues in Optimality-theoretic Syntax*, Stanford: CSLI Publications.
- Legendre, G., P. Smolensky, and C. Wilson (1998). When is less more? faithfulness and minimal links in *wh*-chains. In P. Barbosa, D. Fox, P. Hagstrom, M. McGinnis, and D. Pesetsky (Eds.), *Is the Best Good Enough? Optimality and Competition in Syntax*. Cambridge, Massachusetts: MIT press.
- Levin, L. S. (1982). Sluicing: A lexical interpretation procedure. In J. W. Bresnan (Ed.), *The Mental Representation of Grammatical Relations*, Chapter 9, pp. 590–654. Cambridge, MA: MIT Press.
- Morimoto, Y. (2000). “Crash vs. Yield”: On the conflict asymmetry in syntax and phonology. Ms. Stanford University, to appear in Peter Sells (ed.), *Formal and Empirical Issues in Optimality-theoretic Syntax*, Stanford: CSLI Publications.

- Neumann, G. (1994). *A uniform computational model for natural language parsing and generation*. Ph. D. thesis, Universität des Saarlandes, Germany.
- Neumann, G. (1998). Interleaving natural language parsing and generation through uniform processing. *Artificial Intelligence 99*, 121–163.
- Pesetsky, D. (1998). Some optimality principles of sentence pronunciation. In P. Barbosa, D. Fox, P. Hagstrom, M. McGinnis, and D. Pesetsky (Eds.), *Is the Best Good Enough? Optimality and Competition in Syntax*, pp. 337–383. Cambridge, Massachusetts: MIT press.
- Prince, A. and P. Smolensky (1993). Optimality theory: Constraint interaction in generative grammar. Technical Report Technical Report 2, Rutgers University Center for Cognitive Science.
- Smolensky, P. (1986). Information processing in dynamical systems: Foundations of harmony theory. In D. E. Rumelhart, J. L. MacClelland, and the PDP Research Group (Eds.), *Parallel distributed processing : explorations in the microstructure of cognition, volume 1: Foundations*, pp. 194–281.
- Smolensky, P. (1996). On the comprehension/production dilemma in child language. *Linguistic Inquiry 17*, 720–731.
- Smolensky, P. (1998). Why syntax is different (but not really): Ineffability, violability and recoverability in syntax and phonology. Handout of a talk given at the workshop *Is Syntax Different?* at Stanford University, December 1998.
- Tesar, B. (1995). *Computational Optimality Theory*. Ph. D. thesis, University of Colorado.
- Tesar, B. B. and P. Smolensky (1998). Learnability in optimality theory. *Linguistic Inquiry 29*(2), 229–268.
- Wedekind, J. (1999). Semantic-driven generation with LFG- and PATR-style grammars. *Computational Linguistics 25*(2), 277–281.
- Wilson, C. (1998). Bidirectional optimization and the theory of anaphora. Ms. Johns Hopkins University. To appear in Jane Grimshaw, Géraldine Legendre, and Sten Vikner (eds.), *Optimality Theoretic Syntax*, MIT Press, Cambridge, MA.

Jonas Kuhn
jonas@ims.uni-stuttgart.de

Institut für maschinelle Sprachverarbeitung
Universität Stuttgart
Azenbergstr. 12
D-70174 Stuttgart
GERMANY