Optimality-Theoretic Syntax— a Declarative Approach

Jonas Kuhn

[DRAFT] February 2003

**CENTER FOR THE STUDY
OF LANGUAGE
AND INFORMATION**

Contents

Preface

This book is a revised version of my University of Stuttgart dissertation *Formal and Computational Aspects of Optimality-theoretic Syntax* (Kuhn, 2001b). The overall structure of the original dissertation has been left unchanged; the main focus for the revision was on clarity of presentation and on accessibilty. The most significant modifications were made to chapter 5. A section from chapter 6 (which appears in (Kuhn, 2001a)) has been removed.

The research reported in this book grew out of work in two related projects: the B12 project (*Methods for extending, maintaining and optimizing a large grammar of German*) within the Collaborative Research Centre *Linguistic Foundations for Computational Linguistics* (SFB 340), supported by the Deutsche Forschungsgemeinschaft (DFG), and the Parallel Grammar Development Project *ParGram*, a collaborative effort of the Institut für maschinelle Sprachverarbeitung (IMS) Stuttgart, the Palo Alto Research Center (PARC), and the Xerox Research Center Europe (XRCE) Grenoble (and more recently also the University Bergen and Fuji Xerox). Parts of this book are based on articles and conference papers, as indicated in the text and bibliography.

The revision of my dissertation for this book was in part made possible through a postdoctoral research grant by the DAAD (*Deutscher Akademischer Auslandsdienst*, German academic exchange service) for a stay at Stanford University 2001/2002. I am very grateful to Joan Bresnan and Chris Manning for making this postdoctoral stay possible by acting as my sponsors, and for all their help.

I would like to thank all the members of the IMS Stuttgart for providing an excellent working environment over many years. In particular I

wish to thank my supervisors Christian Rohrer and Jürgen Wedekind—not only for input and discussion of material related to this book, but for being a source of advice and motivation over many years. Christian Rohrer has made possible several short-term stays at PARC in the context of the *ParGram* project, which provided me with invaluable input and opportunities for discussion of empirical, formal and computational issues. For all this interaction, I owe thanks to the *ParGram* people, especially Miriam Butt, Dick Crouch, Mary Dalrymple, Anette Frank, Ron Kaplan, Martin Kay, Tracy King, John Maxwell, and Annie Zaenen. Special thanks due to Anette, Tracy and John—my co-authors for (Frank et al. 1998, 2001)—for encouraging me to dig deeper into the formalization of OT-LFG, which turned out to be the starting point for this entire book. Many thanks again to Jürgen for all his help with formal issues.

Extraordinary thanks go to Joan Bresnan for providing crucial input and for raising very important issues in many discussions and email exchanges. I would also like to thank the audience at the Stanford Syntax Workshop (March 2000) for their important feedback, as well as the audiences at the seminars in Edinburgh (July 1999) and Saarbrücken (January 2000), at the GGS conference (May 1999) and the 4th OT syntax workshop in Stuttgart (February 2001), NELS at Rutgers (October 1999), ACL 2000 in Hongkong, the LFG 1999 conference in Manchester, LFG/HPSG 2000 in Berkeley, LFG 2002 in Athens, and ACL 2002 in Philadelphia.

Many thanks to the faculty and students at the Stanford Linguistics Department for a great atmosphere and numerous discussions during my postdoctoral stay, and to the members of Chris Manning's Natural Language Processing group at the Stanford Computer Science Department.

Thanks to Sten Vikner for initiating the lively interaction in the Stuttgart OT syntax project meetings and to the participants, especially Fabian Heck, Gereon Müller, Tanja Schmid and Ralf Vogel. I'm also indebted to Joan Bresnan, Stefanie Dipper, Arne Fitschen, Anette Frank, Ron Kaplan, Peter Sells, and in particular to anonymous reviewers for CSLI Publications for comments on the manuscript for this book and for comments on various related papers. Furthermore, I've benefited from talking to many other people, including Reinhard Blutner, Hans-Martin Gärtner, Mark Johnson, Frank Keller, Chris Manning, and Stefan Riezler.

I owe much to the academic teachers who initiated my interest in

syntax, semantics and computational linguistics, in particular Elisabet Engdahl, Werner Frey, Hans Kamp, Uwe Reyle, Christian Rohrer, and Jürgen Wedekind.

Finally, I wish to thank my parents for their support and encouragement, and my wife Birgit for always finding ways of motivating me, even when I thought I was stuck.

February 4, 2003

1

Introduction

This book addresses the formal and computational properties of Optimality-theoretic (OT) models of grammar, in particular as they are applied in syntactic analysis. The OT approach was first introduced in the 1990s in the area of phonology (Prince and Smolensky, 1993), where it was quickly adopted by a large number of researchers. Soon the framework was also applied in theoretical syntax (Legendre et al., 1993). In this area, (Grimshaw, 1997) was a very influential early contribution (manuscript versions of this paper go back to 1993). The OT approach is compatible with different underlying grammatical frameworks and representations. The setting for this book is the version of OT which is based on the formal framework of Lexical-Functional Grammar (LFG, Kaplan and Bresnan (1982))[1]—the OT-LFG approach, initiated by the work of Bresnan (1996, 2001a, 2000, among others). The choice of a particular formalism is indispensable when the aim is to make the details of the model precise; however I will try to avoid extensive discussion of framework-internal issues. Much of the considerations expressed with the formal notions of LFG can be transferred to other formal accounts of syntax. Setting this account in the framework of LFG has the advantage that the rich literature on formal and computational aspects of this and related unification-based grammar formalisms can be exploited in the investigation of the properties of the OT model.

[1]There are several recent introductory books for the LFG theory of syntax: Bresnan (2001b), Dalrymple (2001), Falk (2001). A collection of contributions on formal aspects of the LFG formalism is (Dalrymple et al., 1995).

OT is based on the following idea: variation across the languages of the world is explained by interaction of a set of universal constraints. Such constraints say for example roughly "realize scope-bearing elements (such as *wh*-words) in a way that makes their scope visible", or "avoid realizing arguments other than in their base position". Constraints are violable and are often in mutual conflict. Depending on priorities, such conflicts can be resolved in different ways (compare *who Ann saw* vs. *Ann saw who*). The priorities, expressed as a dominance relation or ranking between the constraints, constitute the difference between the individual languages (say, English and Chinese). From a set of competing candidate analyses, the one with the least serious constraint violations—that is, the most harmonic one—is defined as grammatical. Learning a language amounts to determining the correct dominance relation over the universal set of constraints.[2]

OT combines the formally exact approach of Generative Grammar and its detailed representations with an empirical learning approach. In a stochastic OT variant like Boersma's 1998, the learning algorithm permits in particular the derivation of frequency effects. This empirical orientation brings with it the potential for resolving a longstanding and crucial problem of syntactic theory, and especially of the grammar development approach in computational linguistics which attempts to benefit from the results of linguistic theory: within the classical grammatical frameworks, the extension of phenomenon-specific linguistic analyses with the goal of broader coverage on real corpus data typically hits a critical point. Beyond this point, the original generalizations have to be restricted or divided into cases in order to match the extended data basis. Formulating the relevant restrictions without causing some other analysis to be accidentally suppressed turns out to be a difficult engineering task,[3] which the classical approach in linguistic theory has little to say about. Under the OT approach with its violable constraints and an empirical learning scheme, "training" on corpus data may lead to a finegrained and well-adjusted system of constraint interaction, which has a similar coverage as the ideal, manually tailored list of special cases. However, linguistic work stays at the level of intact generalizations, since the data tuning is performed by the learning

[2]For an introduction to OT see (Kager, 1999), which has OT phonology as its main focus, but addresses the application to syntax, too.

[3]This type of observation is made in most broad-coverage grammar development projects based on linguistic theory (compare for instance Erbach and Uszkoreit (1990)). As a reaction, work in grammar engineering has put emphasis on testing and profiling methodology (Oepen et al., 1997, Oepen and Flickinger, 1998, Kuhn, 1998).

technique, informed by frequencies of phenomena and lexical elements in the observed data.

Before such an approach can be put to use and particular strategies of corpus-based learning can be evaluated, many issues about the representation formalism and the mechanisms in the processing model have to be settled. The present book is an attempt to contribute to this by proposing a non-derivational formal model of OT syntax and discussing important empirical, formal and computational properties. The focus is on issues of the architecture, i.e., how are the competing candidates determined, what candidates have to be considered, what properties of the candidates are checked? The question of what is the most adequate way of modelling priorization of the different constraints (weighting, ranking, or some stochastic variant of ranking) is to a great extent independent of the architectural issues.

Besides this perspective of applying a linguistically motivated competition-based model of syntax in corpus-based learning, a more immediate need of a spelled-out formalization of OT syntax and a computational account exists for ongoing linguistic work in the OT framework. To date, most work focuses on fairly restricted sets of empirical data and few selected constraints. This is in part due to the fact that OT is still a young field of study, but it seems also that there is a limit to the size of an OT analysis that can be mastered on a piece of paper. A thorough assessment of the benefits of OT presupposes computational devices that allow one to manipulate larger sets of constraints and larger candidate sets with more complex candidate analyses.

Overview

This book consists of six chapters besides this introduction. Chapters 2 and 3 provide some background on OT and identify the conceptual and empirical OT-syntactic issues underlying this book. Chapter 4 presents a formalization of OT syntax in the framework of LFG; chapter 5 discusses an extension of this formalization to bidirectional optimization. Chapter 6 is devoted to processing issues arising from the formalization; chapter 7 presents some conclusions and an outlook.

Chapter 2 starts with a brief introduction of the main concepts of OT, illustrated with an example from syntax. The basic explanatory mechanisms of OT are explained on the basis of simple phonological examples. This establishes the crucial role of constraint interaction in the explanation of variation across the languages of the world.

Chapter 3 moves on to conceptual and empirical issues specific to the application of OT to syntax. The two main themes of this chapter are faithfulness violations and questions of learnability. A treatment of these issues within an OT system presupposes that the formal framework in which the OT system is set have certain properties. I point out the advantages of a non-derivational setup both for candidate analyses and the OT competition system.

At the beginning of chapter 4, I give a short overview of the LFG formalism. The formalization of OT syntax builds mainly on LFG concepts (thus the framework is called OT-LFG): possible candidates are defined by an LFG-style grammar; candidate sets are defined as possible candidates whose f-structure is subsumed by the input f-structures; constraints are defined as structural description schemata using the primitives of LFG. I point out that the OT-LFG framework allows one to address the conceptual and empirical points raised in chapter 3 in an adequate way.

Chapter 5 discusses a variation of the standard production-based (or expressive) optimization model, which is straightforward on the basis of the OT-LFG formalization: comprehension-based (or interpretive) optimization, in which the terminal string is fixed across the members of the candidate set. Formally, this is only a minor modification of the definition of the candidate set, but there are interesting conceptual and empirical issues concerning parallelism between the two "directions" of optimization, and in particular the combination of both in a bidirectional model. I present a bidirectional account of pro-drop in Italian, which derives a recoverability condition as an effect of the interaction of the two optimizations.

Building on computational results for LFG generation, chapter 6 discusses the processing tasks associated with the two types of unidirectional optimization models and with the various combinations in a bidirectional system (chapter 6). The two main issues in processing are the control of the infinite candidate set and directionality of processing. I show that generally, the conceptually and empirically well-motivated formalization of chapters 4 and 5 provides a sufficiently restricted basis for a computational account.

Chapter 7 concludes book with a summary of the main results and a discussion of open points and directions for future research.

2

The foundations of Optimality Theory

This chapter introduces the general ideas of the Optimality-theoretic approach in linguistics. Optimality Theory (OT) was first developed for phonology by Prince and Smolensky (1993), and has quickly attracted many researchers not only in phonology, but also morphology, syntax, semantics and pragmatics. This book is mainly focused on Optimality-theoretic syntax, so the first illustration of constraint interaction in sec. 2.1 is drawn from this area. Nevertheless, it is helpful to introduce the basic explanatory devices with phonological examples, as will be done in sec. 2.2.

2.1 Conflicting violable constraints

This section presents an illustrative example of an analysis in OT syntax. It is taken from Grimshaw's 1997 account of inversion in English, which is set in a syntactic framework working with a representational simulation of movement derivations in the style of *Government-and-Binding* theory (GB).[4] Grimshaw's paper was probably the most influential early work applying OT methods to syntactic phenomena. It presents a very elegant analysis for the notorious auxiliary/inversion facts of English. This paper has prompted many researchers' interest in OT syntax. (Bresnan, 2000, sec. 2) shows that Grimshaw's constraint system can be reconstructed in the LFG framework (see chapter 4),

[4]Cf. Chomsky (1981); for in introduction, see e.g. Haegeman (1994).

and many of the examples I use to illustrate the formalizations in this book will also be based on this fragment. Here, I will just go through a simplified analysis to informally introduce the basic concepts of OT syntax.

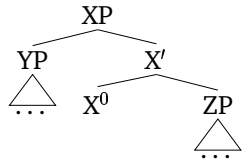
The miniature fragment I will use for illustration is based on the syntactic OT constraints constraints in (1) (Grimshaw, 1997, 374).

(1)

OP-SPEC	Syntactic operators must be in specifier position.
OB-HD	(Obligatory Head) A projection has a head.
STAY	Trace is not allowed.

Note that the constraints are formulated as conditions on syntactic representations, in this case tree configurations following the GB tradition (in particular assuming *traces* marking the base position of elements that have been *moved* to a different position at the level of surface structure). The concepts of head, projection and specifier are standard terms from *X-bar theory*, referring to particular positions in the *X-bar scheme* (2), which underlies all well-formed syntactic trees. The relevant *syntactic operators* (as referred to in the OP-SPEC constraint) for our present purposes are *wh*-phrases like *who* or *what*.

(2)



X^0: head
X', XP: projections of X^0
ZP: complement of X^0
YP: specifier to XP

In general, it is not possible to satisfy all OT constraints at the same time since some of them are in mutual conflict. For example according to the underlying assumptions about representations, some syntactic operators cannot be base-generated in specifier position. So, in order to satisfy OP-SPEC such operators have to be moved to this position, which inevitably gives rise to a trace violating STAY. It is a key assumption of OT that it is legitimate for a well-formed analysis to violate certain constraints (in order to satisfy some other constraints). Thus, contrary to the principles and constraints of pre-OT frameworks of generative linguistics, OT constraints are *violable*.

By hypothesis the set of constraints is universal, but what differs from language to language is the importance of the individual constraints. In some languages it is more important to satisfy OP-SPEC than STAY, in other languages vice versa. This is captured by assuming a language-specific hierarchical dominance ranking of the constraints. The dominance relation is written as $\gg$ ('is more highly ranked than'). For English, the dominance ranking for our three constraints is as follows: OP-SPEC $\gg$ OB-HD $\gg$ STAY.

The type of relation between constraints assumed in standard OT approaches is one of *strict dominance*, meaning that higher-ranking constraints take absolute priority over lower-ranking constraints: if an analysis satisfies a higher-ranking constraint better than any alternative analysis, it does not matter how many violations of lower-ranking constraints the analysis incurs.

The assumption of conflicting violable constraints comes with an inherent need to make comparisons in order to determine what is the well-formed, or grammatical, analysis. This is captured by the abstract process of harmony evaluation —called *Eval*—in which a set of candidate analyses or competitors is evaluated according to the language-particular constraint ranking. In a pair-wise comparison, the most *harmonic* candidate (the "winner") is determined. Harmony is defined as follows.

(3) A_1 is more harmonic than A_2 ($A_1 \succ A_2$) if it contains fewer violations for the highest-ranking constraint in which the marking of A_1 and A_2 differs.

We can subdivide the evaluation process in

- a first step identifying the constraint violations (this is sometimes modelled by a function *marks* from candidates to multisets or bags of constraint violation marks), and
- a second step actually determining the harmony of the candidates.

Only the latter step is language-dependent. When there is no risk of confusion, I will sometimes also call this narrower second step harmony evaluation or *Eval*.

Before looking at an example of harmony evaluation at work, we have to know what candidates enter the competition for the most harmonic candidate. This is a very crucial question, since obviously the result of the comparison may vary significantly depending on how "hard"

the competition is. Intuitively, only genuine *alternatives* should be compared. That means that all candidates should be equivalent in terms of their communicative force. Here, OT accounts are not always very explicit, but it is a widespread assumption that all competing candidates share the same semantic content.[5] If we assume some representation of the semantic content, candidate sets can then be defined by a function that maps a content representation to the set of all analyses expressing this content. This function is called *Gen* (for candidate generation), and since it is tempting to think of this abstract function as some derivational process, the content representation that *Gen* takes as an argument is typically called the *input*. I will follow this standard terminology without implying that *Gen* is indeed a derivational process.[6]

Strictly, in the GB-based model the input should be the LF (logical form), possibly paired with d-structure, but for the purposes of Grimshaw's fragment it suffices to assume an input consisting of "a lexical head plus its argument structure and an assignment of lexical heads to its arguments, plus a specification of the associated tense and aspect" (Grimshaw, 1997, 376). Given a particular such input I, *Gen*(I) is the set of all candidate analyses with I as the underlying argument structure. Note that the definition of *Gen* incorporates certain inviolable principles of what counts as a valid candidate. In the present fragment this comprises the principles of X-bar theory and some principles on chain formation (for representing the movement operations).

So, we can now specify the input for an example (Grimshaw, 1997, 378): (4); the first column of table in (5) shows some sample candidates that are contained in the set that *Gen* assigns to this input.[7] In (6), the full tree structures for the three candidates are given, with the movement transformations illustrated by arrows.

(4) read(x, y)
x = Mary
y = what
tense = future

[5]The issue will be discussed in sec. 3.3.3.

[6]Compare the discussion in sec. 3.3.5.

[7]In this example, I follow Grimshaw's GB-style notation: 't' marks the trace of the moved *wh*-word *what*; '**e**' marks an empty head (of the CP projection); '**will**$_i$—**e**$_i$' is the chain of *will*'s head movement from I to C.

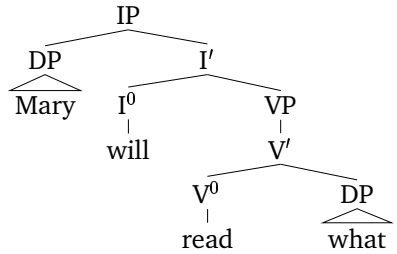
Note that $\mathbf{e}_i$ is a trace, while **e** is not. (5c) does not violate OB-HD, because the I head was filled at one stage of the derivation.

(5)

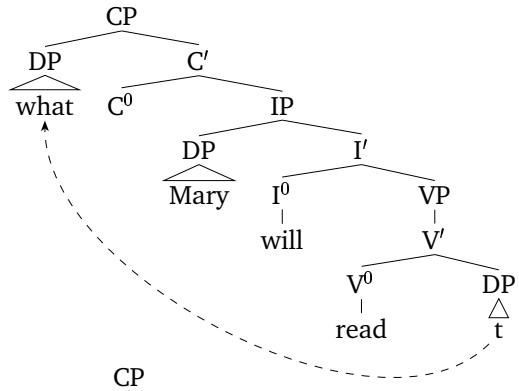
Candidates	Constraint violations
a. [$_{\text{IP}}$ Mary will [$_{\text{VP}}$ read what]]	*OP-SPEC
b. [$_{\text{CP}}$ what **e** [$_{\text{IP}}$ Mary will [$_{\text{VP}}$ read t]]]	*OB-HD, *STAY
c. [$_{\text{CP}}$ what **will**$_i$ [$_{\text{IP}}$ Mary **e**$_i$ [$_{\text{VP}}$ read t]]]	*STAY, *STAY

(6)

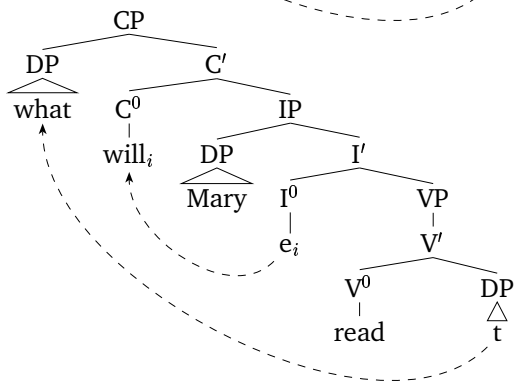
a.



b.



c.



For each candidate derivation, the function *marks* checks which of the constraints it satisfies; a violation is marked in the second column of (5), with an '*' preceding the constraint name. With the given set of constraints, we can effectively check the violations on a single level of representation containing reflexes of transformational derivations,

namely co-indexed traces, or chains (this points towards the possibility of formulating an entirely non-derivational, declarative account, in which not even candidate specification involves a transformational process).

Looking at the three candidates in (5)/(6), candidate a. has the *wh*-operator *what* in the complement position of the verb, thus failing to satisfy the constraint OP-SPEC in (1). In candidate b., the CP doesn't contain a head, leading to a OB-HD violation; furthermore, *what* has been moved to the specifier of CP (so it avoids a OP-SPEC violation), leaving behind a trace and thus violating STAY once. Candidate c. avoids the empty C^0 by moving the auxiliary *will* from I^0 into this position—at the cost of incurring an additional STAY violation. So we see that the constraint conflict triggered by the presence of a *wh*-operator in the input is resolved in different ways in each of the candidates.

Based on this marking of constraint violations for all analyses in the candidate set, and the language-specific constraint hierarchy, now the function *Eval* determines the most harmonic, or optimal, candidate: by definition, this is the only grammatical analysis for the underlying input representation. (There may also be a set of equally harmonic candidates.) The standard notation for the result of the evaluation is a *tableau* (7), with the columns for the constraint reflecting the hierarchy of the language under consideration. It is customary to list the input in the top left-hand table cell.

(7)

Input: read(x, y), x = Mary, y = what, tense = future	OP-SPEC	OB-HD	STAY
a. [$_{IP}$ Mary will [$_{VP}$ read what]]	*!		
b. [$_{CP}$ what **e** [$_{IP}$ Mary will [$_{VP}$ read t]]]		*!	*
c. ☞ [$_{CP}$ what **will**$_i$ [$_{IP}$ Mary **e**$_i$ [$_{VP}$ read t]]]			**

If a candidate loses in a pairwise comparison, the "fatal" mark is highlighted with an '!' (e.g., candidate a. is less harmonic than b., since they differ in the highest-ranked constraint OP-SPEC). Note that the score that the losing candidate has for lower-ranked constraints is completely irrelevant. Ultimately, the candidate that remains without a fatal constraint violation is marked with the symbol ☞ as the winner of the entire competition. In the example, candidate c. is optimal, although it violates the constraint STAY twice. The other analyses are predicted to

be ungrammatical. Different languages are characterized by different relative rankings of the constraints. For instance, a language with *wh* in situ may rank OP-SPEC lower than STAY, which will cause candidate a. in (7), to be the winner. Note that there will always be at least one winning analysis for a given (nonempty) candidate set, since optimality is defined relative to the competitors.[8]

After this informal example, we can identify the notions that a formalization of OT must pinpoint: the input representation, the function *Gen*, the formulation of constraints, and harmony evaluation (*Eval*), consisting of the function *marks* checking for constraint violations, and the determination of harmony based on the language-specific constraint ranking. For some of these concepts, the assumptions made in different incarnations of OT vary significantly.

In chapter 4, I will address the LFG-based approach to formalization of OT Syntax (following work by Bresnan—1996, 2000), which assumes strictly non-derivational candidate analyses. All relevant aspects of the candidates can be encoded representationally, so there is good reason to opt for the conceptually simpler choice of comparing static objects, in particular since the mathematical and computational properties of LFG-like formalisms have been studied extensively.[9]

In the remainder of this chapter and in the following chapter 3, I will discuss some of the assumptions, techniques and methodological principles adopted in the Optimality-theoretic approach in linguistics.

2.2 Factorial typology and the grounding of constraints

One of the main arguments for the OT approach is its ability to make typological predictions. Since this book is mainly about formal and computational aspects of syntactic OT systems, this empirical aspect is not a central topic in the later chapters. However, when one wants to decide between several different, but related architectures it is important to be aware of the aspects that motivate the linguistic application of optimization in the first place. Therefore I will review the structure of a factorial typology argument (based on the example from (Kager, 1999, sec. 1.7)).

Using phonological data for this purpose has not only the advantage of keeping the analyses we have to look at simple, but is also a good

[8]This means that the phenomenon of language-particular ineffability cannot be modelled in the standard OT model. This will be discussed in sec. 3.3.3.

[9]Compare also the discussion in sec. 3.3.5.

starting point for a discussion of the *grounding* of constraints—an issue which is interweaved with the explanatory role of factorial typology.

2.2.1 Factorial typology

Under different dominance rankings over a set of constraints, different analyses in a candidate set come out as the winner. Since by definition, only the winners are in the language described by the OT system, changing the constraint ranking gives us different languages. But with reranking over a fixed set of constraints we do not get *arbitrary* collections of winners (i.e., languages)—the choice of constraints (plus the OT assumptions) enforces certain patterns. This is what factorial typology is about.

Using the illustrative example of (Kager, 1999, sec. 1.7), let us look at nasality of vowels in closed syllables. The phonological analysis is supposed to predict under what conditions vowels are pronounced nasally or not nasally. So in the candidate set both the possibility of keeping the underlying nasality and a change of the underlying nasality have to be provided. Such potential changes of underlying information are controlled by *faithfulness constraints*, which favour candidates without such changes. Typically, faithfulness constraints are in conflict with *markedness constraints*, which favour certain properties of the output form independent of what the underlying form is like. (Faithfulness constraints are discussed in more detail in sec. 3.2.)

For the present illustrative purpose, we are only interested in whether or not a nasal vowel in the input will also be realized as a nasal in the winning candidate, potentially depending on the following consonant. For this consonant, again, it only matters whether it is a (tautosyllabic) nasal, like [n], or not. So, it suffices to look at the following four underlying word forms as representatives for the phenomenon we are interested in: /pan/, /pãn/, /pal/, /pãl/. The input /pan/ can be either realized faithfully as [pan], or, violating faithfulness, as [pãn] (we are not considering faithfulness violations in the realization of the consonant). /pãn/ could come out as [pãn] or [pan]. Likewise for the other words. So, combinatorially, there are 16 different ways natural languages could behave, as shown in (8) Alternative (8a) is faithful in all cases, alternative (8p) is "unfaithful" in all cases. In between, we have any possible combination of faithfulness in some of the cases, with unfaithfulness in the other cases.

(8)
a. {/pan/ → [pan], /pãn/ → [pãn], /pal/ → [pal], /pãl/ → [pãl]}
b. {/pan/ → [pãn], /pãn/ → [pãn], /pal/ → [pal], /pãl/ → [pãl]}
c. {/pan/ → [pan], /pãn/ → [pan], /pal/ → [pal], /pãl/ → [pãl]}
d. {/pan/ → [pan], /pãn/ → [pãn], /pal/ → [pãl], /pãl/ → [pãl]}
e. {/pan/ → [pan], /pãn/ → [pãn], /pal/ → [pal], /pãl/ → [pal]}
f. {/pan/ → [pãn], /pãn/ → [pan], /pal/ → [pal], /pãl/ → [pãl]}
g. {/pan/ → [pãn], /pãn/ → [pãn], /pal/ → [pãl], /pãl/ → [pãl]}
h. {/pan/ → [pãn], /pãn/ → [pãn], /pal/ → [pal], /pãl/ → [pal]}
i. {/pan/ → [pan], /pãn/ → [pan], /pal/ → [pãl], /pãl/ → [pãl]}
j. {/pan/ → [pan], /pãn/ → [pan], /pal/ → [pal], /pãl/ → [pal]}
k. {/pan/ → [pan], /pãn/ → [pãn], /pal/ → [pãl], /pãl/ → [pal]}
l. {/pan/ → [pãn], /pãn/ → [pan], /pal/ → [pãl], /pãl/ → [pãl]}
m. {/pan/ → [pãn], /pãn/ → [pan], /pal/ → [pal], /pãl/ → [pal]}
n. {/pan/ → [pãn], /pãn/ → [pãn], /pal/ → [pãl], /pãl/ → [pal]}
o. {/pan/ → [pan], /pãn/ → [pan], /pal/ → [pãl], /pãl/ → [pal]}
p. {/pan/ → [pãn], /pãn/ → [pan], /pal/ → [pãl], /pãl/ → [pal]}

Let us now look at a linguistically sensible OT analysis of the data (the motivation for picking a particular set of constraints will be discussed further down). We have already talked about faithfulness, so a faithfulness constraint is an important participant in the constraint interaction. The relevant constraint is:[10]

(9) IDENT-IO(nasal)
The specification of the feature [nasal] of an input segment must be preserved in its output correspondent.

Furthermore, we have the following markedness constraints, intuitively saying that certain structural configurations should be avoided—if possible:

(10) $*\mathrm{V}_{\mathrm{NASAL}}$
Vowels must not be nasal.

(11) $*\mathrm{V}_{\mathrm{ORAL}}\mathrm{N}$
Before a tautosyllabic nasal, vowels must not be oral.

(10) is a context-free markedness constraint. If it is not dominated by any other constraints, the language will not contain any nasal vowels at all. (11) is a context-sensitive markedness constraint. Given these three constraints, there are 6 possible rankings (in general for a set of n constraints, there are $n!$ possible rankings, thus the term *factorial* typology):

[10]The input-output identity constraint IDENT-IO is part of the Correspondence Theory approach of (McCarthy and Prince, 1995).

(12) a. IDENT-IO(nasal) ≫ *V_{NASAL} ≫ *V_{ORAL}N
b. IDENT-IO(nasal) ≫ *V_{ORAL}N ≫ *V_{NASAL}
c. *V_{ORAL}N ≫ IDENT-IO(nasal) ≫ *V_{NASAL}
d. *V_{ORAL}N ≫ *V_{NASAL} ≫ IDENT-IO(nasal)
e. *V_{NASAL} ≫ *V_{ORAL}N ≫ IDENT-IO(nasal)
f. *V_{NASAL} ≫ IDENT-IO(nasal) ≫ *V_{ORAL}N

With ranking (12a), we get the competitions in (13). Since Faithfulness (IDENT-IO(nasal)) is undominated, the unfaithful candidate loses in all four cases (note the exclamation marks in the first column of each small tableau). So we get a language that displays full contrast in the output forms. Typologically, this is a widespread pattern. Note that the relative ranking of the two markedness constraints plays no role (given the set of data), so the outcome with ranking (12b) is identical. The ranking *V_{NASAL} ≫ *V_{ORAL}N is said not to be *crucial*—in this situation it is common in the literature that the dominance relation is left unspecified for the two constraints: The hierarchy IDENT-IO(nasal) ≫ { *V_{NASAL}, *V_{ORAL}N } specifies all crucial rankings.[11]

(13) *Full contrast*
IDENT-IO(nasal) ≫ *V_{NASAL} ≫ *V_{ORAL}N

(i) Input: /pan/	ID-IO(nasal)	*V_{NASAL}	*V_{ORAL}N
a. [pãn]	*!	*	
b. ☞ [pan]			*

(ii) Input: /pãn/	ID-IO(nasal)	*V_{NASAL}	*V_{ORAL}N
a. ☞ [pãn]		*	
b. [pan]	*!		*

(iii) Input: /pal/	ID-IO(nasal)	*V_{NASAL}	*V_{ORAL}N
a. [pãl]	*!	*	
b. ☞ [pal]			*

(iv) Input: /pãl/	ID-IO(nasal)	*V_{NASAL}	*V_{ORAL}N
a. ☞ [pãl]		*	
b. [pal]	*!		*

If we look at ranking (12c), we get the competitions in (14):

[11] Often, the dominance ranking is formally defined as a total relation, so effectively for the non-crucial rankings one or the other alternative will actually hold. But for the cases under discussion, this does not have an effect on the outcome.

(14) *Positional neutralization*
$*V_{ORAL}N \gg$ IDENT-IO(nasal) $\gg *V_{NASAL}$

(i) Input: /pan/	$*V_{ORAL}N$	ID-IO(nasal)	$*V_{NASAL}$
a. ☞ [pãn]		*	*
b. [pan]	*!		

(ii) Input: /pãn/	$*V_{ORAL}N$	ID-IO(nasal)	$*V_{NASAL}$
a. ☞ [pãn]			*
b. [pan]	*!	*	

(iii) Input: /pal/	$*V_{ORAL}N$	ID-IO(nasal)	$*V_{NASAL}$
a. [pãl]		*!	*
b. ☞ [pal]			

(iv) Input: /pãl/	$*V_{ORAL}N$	ID-IO(nasal)	$*V_{NASAL}$
a. ☞ [pãl]			*
b. [pal]		*!	

Here, the context-sensitive markedness constraint $*V_{ORAL}N$ outranks faithfulness (IDENT-IO(nasal)). This means for the context in which the condition of $*V_{ORAL}N$ fires (i.e., (i) and (ii) where we have a nasal consonant), it is more important to satisfy this markedness constraint than to be faithful to the input. Thus in the (i) case, we get an unfaithful output, realizing an underlying oral vowel as a nasal. The (iii) and (iv) cases are unaffected, since faithfulness jumps in to level out the effect that $*V_{NASAL}$ could have. This phenomenon is called *positional neutralization*. For certain contexts (i.e., in certain positions), differences in the underlying forms are neutralized: before /n/, both an underlying /a/ and an /ã/ come out the same.

Based on the hierarchy (12d) harmony evaluation yields (15) as a result. With the context-sensitive markedness constraint highest-ranking, followed by the context-free markedness constraints, and faithfulness ranked lowest, we get the same effect as before for the nasal context ((i) and (ii)), plus we get the effect of $*V_{NASAL}$ for the non-nasal context (case (iii) and (iv)). So we observe *allophonic variation*. In both contexts, neutralization of underlying differences occurs. Two of the four cases ((i) and (iv)) display unfaithfulness of the output. This behaviour is found in many dialects of English where vowels before nasal consonants (like in *sand, meant*) are nasalized.

(15) *Allophonic variation*
*$V_{ORAL}N$ ≫ *V_{NASAL} ≫ IDENT-IO(nasal)

(i) Input: /pan/	*$V_{ORAL}N$	*V_{NASAL}	ID-IO(nasal)
a. ☞ [pãn]		*	*
b. [pan]	*!		

(ii) Input: /pãn/	*$V_{ORAL}N$	*V_{NASAL}	ID-IO(nasal)
a. ☞ [pãn]		*	
b. [pan]	*!		*

(iii) Input: /pal/	*$V_{ORAL}N$	*V_{NASAL}	ID-IO(nasal)
a. [pãl]		*!	*
b. ☞ [pal]			

(iv) Input: /pãl/	*$V_{ORAL}N$	*V_{NASAL}	ID-IO(nasal)
a. [pãl]		*!	
b. ☞ [pal]			*

The neutralization caused by the low ranking of faithfulness in allophonic variation languages leads to a somewhat peculiar situation: if one looks at the output (the winning candidates) only, the (i) and the (ii) case are indistinguishable, and so are (iii) and (iv). So there is no context where one could see what the nasality of the actually underlying vowel is. Hence, a learner of this language could never find a clue to distinguish the alternative underlying forms, so it is sufficient to store only one of them in the inventory of underlying lexical forms. The question which one is picked in learning is addressed in OT by Lexicon Optimization (compare (Kager, 1999, sec. 1.6; 7.5.3)): the various underlying inputs which could have given rise to the observed output are compared, using the same constraint ranking as in standard optimization. The most harmonic underlying input is then stored in the lexicon.[12]

As the final ranking options, let us look at (12e) as representative for both (12e) and (12f), since the highest-ranking constraint alone is decisive. The tableaux in (16) result. We again get unfaithfulness in two out of four cases ((ii) and (iv)), where in this case the global markedness of nasal vowels suppresses all other factors—be it the context-sensitive

[12]Lexicon Optimization is an instance of bidirectional optimization, which is addressed in chapter 5 of this book.

markedness of the alternative oral vowel or faithfulness to the underlying input form. So, in this language vowels are never nasal, there is a *lack of variation* for the nasality feature on vowels. Typologically, this kind of behaviour is attested for different features in many languages. Our concrete case here—lack of nasality for vowels—actually holds for the majority of languages of the world. Note that as in the previous case, from the output it is impossible to tell whether the underlying input form had a nasal vowel or not. So again, Lexicon Optimization will apply in learning.

(16) *Lack of variation*
$*\text{V}_{\text{NASAL}} \gg *\text{V}_{\text{ORAL}}\text{N} \gg$ IDENT-IO(nasal)

(i) Input: /pan/	$*\text{V}_{\text{NASAL}}$	$*\text{V}_{\text{ORAL}}\text{N}$	ID-IO(nasal)
a. [pãn]	*!		
b. ☞ [pan]		*	

(ii) Input: /pãn/	$*\text{V}_{\text{NASAL}}$	$*\text{V}_{\text{ORAL}}\text{N}$	ID-IO(nasal)
a. [pãn]	*!		
b. ☞ [pan]		*	*

(iii) Input: /pal/	$*\text{V}_{\text{NASAL}}$	$*\text{V}_{\text{ORAL}}\text{N}$	ID-IO(nasal)
a. [pãl]	*!		
b. ☞ [pal]			

(iv) Input: /pãl/	$*\text{V}_{\text{NASAL}}$	$*\text{V}_{\text{ORAL}}\text{N}$	ID-IO(nasal)
a. [pãl]	*!		
b. ☞ [pal]			*

2.2.2 The predictive power of factorial typology

We have seen all six possible rankings for the three constraints under consideration—with four empirically distinguishable outcomes. Recall that in (8), 16 logically possible language behaviours were observed, of which now only four are predicted: the alternatives a., b., h., and j, repeated below.

(8) *Possible languages*

a. {/pan/ → [pan], /pãn/ → [pãn], /pal/ → [pal], /pãl/ → [pãl]}
b. {/pan/ → [pãn], /pãn/ → [pãn], /pal/ → [pal], /pãl/ → [pãl]}
h. {/pan/ → [pãn], /pãn/ → [pãn], /pal/ → [pal], /pãl/ → [pal]}
j. {/pan/ → [pan], /pãn/ → [pan], /pal/ → [pal], /pãl/ → [pal]}

With the three constraints (9), (10) and (11), no other language behaviour can be derived. If we look at the other logical possibilities, this turns out to be a desirable result: in the languages of the world, only the predicted patterns can be found. Nasal vowels are universally more marked than oral vowels. There are many languages without nasal vowels, but none without oral vowels (as in (8g)). For nasal vowels the position before non-nasal consonants is again universally more marked than the position before nasal consonants. This excludes a language like (8c), in which the nasality of the vowel is neutralized to an oral vowel before /n/ but not before /l/.

(8) *Impossible languages*

c. {/pan/ → [pan], /pãn/ → [pan], /pal/ → [pal], /pãl/ → [pãl]}
d. {/pan/ → [pan], /pãn/ → [pãn], /pal/ → [pãl], /pãl/ → [pãl]}
e. {/pan/ → [pan], /pãn/ → [pãn], /pal/ → [pal], /pãl/ → [pal]}
f. {/pan/ → [pãn], /pãn/ → [pan], /pal/ → [pal], /pãl/ → [pãl]}
g. {/pan/ → [pãn], /pãn/ → [pãn], /pal/ → [pãl], /pãl/ → [pãl]}
i. {/pan/ → [pan], /pãn/ → [pan], /pal/ → [pãl], /pãl/ → [pãl]}
k. {/pan/ → [pan], /pãn/ → [pãn], /pal/ → [pãl], /pãl/ → [pal]}
l. {/pan/ → [pãn], /pãn/ → [pan], /pal/ → [pãl], /pãl/ → [pãl]}
m. {/pan/ → [pãn], /pãn/ → [pan], /pal/ → [pal], /pãl/ → [pal]}
n. {/pan/ → [pãn], /pãn/ → [pãn], /pal/ → [pãl], /pãl/ → [pal]}
o. {/pan/ → [pan], /pãn/ → [pan], /pal/ → [pãl], /pãl/ → [pal]}
p. {/pan/ → [pãn], /pãn/ → [pan], /pal/ → [pãl], /pãl/ → [pal]}

As should be clear by now, the three constraints for the linguistic OT analysis were intentionally chosen to reflect the typological pattern in the languages of the world. This set-up demonstrates the workings of the explanatory machinery of the Optimality-theoretic approach: With a small set of constraints, the space of logically possible formal languages is reduced to a smaller spectrum, which serves as OT's model for the possible natural languages.

Thus, apart from Occam's razor, which will favour a system with a small number of constraints, the following two criteria for the adequacy of an OT system can be identified:

(17) *Criteria for the adequacy of an OT system*

a. The typologically attested spectrum of languages should be correctly predicted by the factorial typology of the constraints assumed.
b. The constraints used should have an independent motivation.

2.2 Factorial typology and the grounding of constraints

I will first illustrate a situation where an attempt of an OT analysis fails to satisfy criterion (17a). Assume a set of three hypothetical constraints:

(18) *Hypothetical constraints*

a. IDENT-IO(NASAL=+)
If an input segment is specified as [NASAL=+], this must be preserved in its output correspondent.

b. $V_{NASAL=\alpha}C_{NASAL=\alpha}$
The nasality features in a vowel and a following consonant are identical.

c. $*V_{NASAL}N$
Before a tautosyllabic nasal, vowels must not be nasal.

(18a) is a more focused variant of the faithfulness constraint (9) IDENT-IO(nasal), punishing fewer instances. It only demands that underlying nasals are rendered faithfully. If we have /pan/ → [pãn], this constraint is not violated. (18b) is a stronger variant of (11) $*V_{ORAL}N$, punishing also the situation of a nasal vowel preceding a non-nasal consonant (as in [pãl]). (18c) is a new contextual markedness constraint.

(19) *Positional neutralization* (*with hypothetical constraint set*)
IDENT-IO(NASAL=+) ≫ $V_{NASAL=\alpha}C_{NASAL=\alpha}$ ≫ $*V_{NASAL}N$

(i) Input: /pan/	ID-IO(NAS=+)	$V_{NAS=\alpha}C_{NAS=\alpha}$	$*V_{NASAL}N$
a. ☞ [pãn]			*
b. [pan]		*!	

(ii) Input: /pãn/	ID-IO(NAS=+)	$V_{NAS=\alpha}C_{NAS=\alpha}$	$*V_{NASAL}N$
a. ☞ [pãn]			*
b. [pan]	*!	*	

(iii) Input: /pal/	ID-IO(NAS=+)	$V_{NAS=\alpha}C_{NAS=\alpha}$	$*V_{NASAL}N$
a. [pãl]		*!	*
b. ☞ [pal]			

(iv) Input: /pãl/	ID-IO(NAS=+)	$V_{NAS=\alpha}C_{NAS=\alpha}$	$*V_{NASAL}N$
a. ☞ [pãl]		*	
b. [pal]	*!		

We may think that the hypthetical constraints in (18) are the correct constraints for modelling nasality of vowels in the languages of the world. As the tableaux in (19) on page 19 show, the ranking IDENT-IO(NASAL=+) $\gg$ $V_{NASAL=\alpha}C_{NASAL=\alpha}$ $\gg$ *$V_{NASAL}N$ gives us the positional neutralization language (8b) (NASAL is abbreviated as NAS).

With the ranking IDENT-IO(NASAL=+) $\gg$ *$V_{NASAL}N$ $\gg$ $V_{NASAL=\alpha}C_{NASAL=\alpha}$ (reversing the two lower constraints), all cases apart from (19-i) are the same. So we would get the free-variation language (8a). With $V_{NASAL=\alpha}C_{NASAL=\alpha}$ highest-ranking, it is easy to see that independent of the relative ranking of the other two constraints, we will always get the pattern {/pan/ → [pãn], /pãn/ → [pãn], /pal/ → [pal], /pãl/ → [pal]}, i.e., the allophonic-variation language (8h). Lastly, we might argue that the ranking *$V_{NASAL}N$ $\gg$ $V_{NASAL=\alpha}C_{NASAL=\alpha}$ $\gg$ IDENT-IO(NASAL=+) gives us the fourth language with lack of variation: (20).

(20) *Lack of variation* (*with hypothetical constraint set*)
*$V_{NASAL}N$ $\gg$ $V_{NASAL=\alpha}C_{NASAL=\alpha}$ $\gg$ IDENT-IO(NASAL=+)

(i) Input: /pan/	*$V_{NASAL}N$	$V_{NAS=\alpha}C_{NAS=\alpha}$	ID-IO(NAS=+)
a. [pãn]	*!		
b. ☞ [pan]		*	

(ii) Input: /pãn/	*$V_{NASAL}N$	$V_{NAS=\alpha}C_{NAS=\alpha}$	ID-IO(NAS=+)
a. [pãn]	*!	*	
b. ☞ [pan]			*

(iii) Input: /pal/	*$V_{NASAL}N$	$V_{NAS=\alpha}C_{NAS=\alpha}$	ID-IO(NAS=+)
a. [pãl]		*!	
b. ☞ [pal]			

(iv) Input: /pãl/	*$V_{NASAL}N$	$V_{NAS=\alpha}C_{NAS=\alpha}$	ID-IO(NAS=+)
a. [pãl]		*!	
b. ☞ [pal]			*

The problem is however that we have not yet looked at the complete factorial typology! There is still the ranking which is like the previous one, but with the lower two constraints swapped: *$V_{NASAL}N$ $\gg$ IDENT-IO(NASAL=+) $\gg$ $V_{NASAL=\alpha}C_{NASAL=\alpha}$. For (iv), this rank-

ing *does* make a difference, so unlike the original constraint set, the hypothetical constraint set wrongly predicts a fifth possible language:

(8) c. {/pan/ → [pan], /pãn/ → [pan], /pal/ → [pal], /pãl/ → [pãl]}

This demonstrates the workings of criterion (17a), which demands that the factorial typology formally predicted has to be checked at the empirically observed typology.

2.2.3 The grounding of constraints

Besides the typological criterion (17a) for the adequacy of an OT constraint set, we have criterion (17b), demanding independent motivation for the constraints. In phonological OT, phonetic circumstances can often give a very clear indication of what segments are marked or what combination of segments is marked. Such evidence can originate from properties of either articulation or perception, and it provides *phonetic grounding* of a constraint. Let us go through the original three constraints (9) IDENT-IO(nasal), (10) $*V_{NASAL}$ and (11) $*V_{ORAL}N$, starting with the last one. Without going into much articulatory detail, it is plausible that the sequence of an oral vowel and a (tautosyllabic) nasal consonant requires more effort (lowering the velum to allow airflow through the nose) than that of a nasal vowel and a nasal consonant. This provides phonetic grounding for constraint (11) $*V_{ORAL}N$. Note that the constraint $*V_{NASAL}N$ from the hypothetical constraint set (18) lacks such a motivation.

For faithfulness constraints like (9) IDENT-IO(nasal), the grounding issue does not pose itself in the same way, since we can never *observe* the underlying forms directly; their shape is a theoretical construct that we can define as adequate. What is important is that we use them in a consistent way. Formally, it is conceivable to assume the mirror-image of the IDENT-IO constraint (something like *IDENT-IO or DIFF-IO, if we have binary valued features), which would mean that a free-variation language would be modelled as (8p) rather than (8a).

(8) a. {/pan/ → [pan], /pãn/ → [pãn], /pal/ → [pal], /pãl/ → [pãl]}
p. {/pan/ → [pãn], /pãn/ → [pan], /pal/ → [pãl], /pãl/ → [pal]}

We would no longer have four faithful instances, but rather what looks like four unfaithful instances (defined as the unmarked case in this

thought experiment).[13] However, it would not be very helpful to name the underlying forms in this confusing way.

Let us look at the phonetic grounding of the other markedness constraint, (10) $*V_{\text{NASAL}}$. Here, the situation may already be a little less obvious than it was for $*V_{\text{ORAL}}N$: Why should nasal vowels be more marked a priori (taking into account that ability to close off the nasal cavity has evolved only quite recently in terms of evolution of *homo sapiens*)? An indirect functional motivation may recur on perceptual factors: in order to exploit the space of different vowel qualities ([a] vs. [u] vs. [i], etc.) in a maximally effective way, the perceived vowel qualities should be maximally distinct—and indeed the difference between oral vowels can be perceived more clearly than the difference between nasal vowels. Thus, using oral vowels is more efficient from a functional perspective, which motivates a constraint $*V_{\text{NASAL}}$.

As the example shows, criteria (17a) and (17b) will typically interact in finding a suitable set of candidates—the typological observation that many languages lack nasal vowels, while all languages have oral vowels is so striking that it may give sufficient indication for postulating a constraint $*V_{\text{NASAL}}$. Based on this example, one might think that the typological predictions alone may suffice to characterize an appropriate constraint system. However the following example will show that the constraints one postulates should indeed also be evaluated according to criterion (17b)—independent motivation. Assume another hypothetical set of constraints:

(21) *Hypothetical constraints*
- a. $*V_{\text{NASAL}}$
- b. $V_{\text{NASAL}=\alpha}C_{\text{NASAL}=\alpha}$
- c. EXISTFAITHFUL-V_{NASAL}
  The output must contain a faithful nasal vowel.

Constraint (21a) is the known constraint disfavouring nasal vowels (10). (21b) is the same constraint as (18b) in the previous hypothetical constraint set. It demands that the value of the nasality features on a vowel-consonant combination be identical and is thus violated by the forms [pan] and [pãl]. Note that this constraint is not at all implausible. Now, if we also assume constraint (21c) (for which we do not have any independent motivation), the striking result is that we can derive

[13] As one can check, the typological data would be easily derivable with this upside-down system; (8n) would replace (8b); the neutralization paradigms (8h) and (8j) would stay the same.

exactly the same set of languages as with the constraints (9), (10) and (11) from above: Constraint (21c) is violated by both /pan/ → [pan] and /pan/ → [pãn] (neither of them contains a *faithful* nasal vowel, since the underlying form does not contain one); furthermore it is violated by the unfaithful candidate /pãn/ → [pan]. With the /pal/ and /pãn/ competitions, the situation is parallel.

With $V_{NAS=\alpha}C_{NAS=\alpha}$ (21b) ranked the highest, we get language (8h) (with [pãn] and [pal] as the only overt forms), no matter how the other two constraints are ranked. With $*V_{NASAL}$ (21a) ranked the highest, we always get (8j) (like in (16)). With the ranking EXISTFAITHFUL-V_{NASAL} ≫ $*V_{NASAL}$ ≫ $V_{NAS=\alpha}C_{NAS=\alpha}$, we get (8a), with EXISTFAITHFUL-V_{NASAL} ≫ $V_{NAS=\alpha}C_{NAS=\alpha}$ ≫ $*V_{NASAL}$, the resulting language is (8b). The last result is illustrated in (22).

(22) *Positional neutralization* (*with hypothetical constraint set*)
EXISTFAITHFUL-V_{NAS} ≫ $V_{NAS=\alpha}C_{NAS=\alpha}$ ≫ $*V_{NAS}$

(i) Input: /pan/	EXFAITH-V_{NAS}	$V_{NAS=\alpha}C_{NAS=\alpha}$	$*V_{NAS}$
a. ☞ [pãn]	*		*
b. [pan]	*	*!	

(ii) Input: /pãn/	EXFAITH-V_{NAS}	$V_{NAS=\alpha}C_{NAS=\alpha}$	$*V_{NAS}$
a. ☞ [pãn]			*
b. [pan]	*!	*	

(iii) Input: /pal/	EXFAITH-V_{NAS}	$V_{NAS=\alpha}C_{NAS=\alpha}$	$*V_{NAS}$
a. [pãl]	*	*!	*
b. ☞ [pal]	*		

(iv) Input: /pãl/	EXFAITH-V_{NAS}	$V_{NAS=\alpha}C_{NAS=\alpha}$	$*V_{NAS}$
a. ☞ [pãl]		*	*
b. [pal]	*!		

This hypothetical example shows that just having a minimal set of constraints which predicts the typological data correctly does not guarantee that the most adequate constraint set has already been found. (21c) EXISTFAITHFUL-V_{NASAL} is hard to motivate as a primitive constraint. So, if we take into account criterion (17b), the original set of constraints turns out superior.

2.2.4 Simplicity in the logical structure of constraints

Obviously the constraint system (22) just discussed was not entirely wrong. It described the same data as the original system, with the same number of constraints. Two of the constraints were plausibly grounded. So how can a single further constraint get the data right while being rather implausible? Are there any clues to be read off the formal structure of the constraint that might tell us that something is not quite right? Indeed there are: EXISTFAITHFUL-V_{NASAL} is a combination of a faithfulness and a markedness constraint. The relevant effect is reached if we form a logical conjunction of the ID-IO(nasal) and the *V_{ORAL}N constraints:[14] (23).

(23) COMBINATION of IDENT-IO(nasal) and *V_{ORAL}N
The specification of the feature [nasal] of an input segment must be preserved in its output correspondent
and
before a tautosyllabic nasal, vowels must not be oral.

The combined constraint (23) is violated by all candidates violating either of the two constituent constraints. It is also violated just a single time if both are violated (for candidate b. in (24-ii)).

As the comparison in (24) on page 25 shows, the combination behaves exactly like our hypothetical constraints EXISTFAITHFUL-V_{NASAL}, except for candidate b. in (iii). Looking back at (22) however, this difference does not have any effect, since in (iii) both candidates violate EXISTFAITHFUL-V_{NASAL}, and the remaining constraints make b. the winner under any ranking.

The lesson to be learned from this hypothetical example is that one should be skeptical about constraints that cannot be formulated in a straightforward and simple way (cf. also Grimshaw (1998), arguing for logically simple constraints). These are likely to be combinations of simpler constraints, so a constraint system based on the more primitive constraints should be considered.

[14] This way of conjoining two constraint specifications should not be confused with the OT concept of local conjunction (see e.g., (Kager, 1999, 392ff)). The local conjunction of two constraints C^1 and C^2 relative to a domain δ is an additional, rankable constraint that is violated if and only if both C^1 and C^2 are violated within the same instance of δ. Logically, this corresponds to a *disjunction* of the individual constraint specifications (which is false if and only if the two disjuncts are false), rather than a conjunction.

(24) Comparison of EXISTFAITHFUL-V_{NASAL} and the COMBINATION of IDENT-IO(nasal) and *V_{ORAL}N

(i) Input: /pan/	EXFAITH-V_{NASAL}	COMBINATION	ID-IO(nasal)	*V_{ORAL}N
a. [pãn]	*	*	*	
b. [pan]	*	*		*

(ii) Input: /pãn/	EXFAITH-V_{NASAL}	COMBINATION	ID-IO(nasal)	*V_{ORAL}N
a. [pãn]				
b. [pan]	*	*	*	*

(iii) Input: /pal/	EXFAITH-V_{NASAL}	COMBINATION	ID-IO(nasal)	*V_{ORAL}N
a. [pãl]	*	*	*	
b. [pal]	*			

(iv) Input: /pãl/	EXFAITH-V_{NASAL}	COMBINATION	ID-IO(nasal)	*V_{ORAL}N
a. [pãl]				
b. [pal]	*	*	*	

On the other hand, a system with a small number of simple constraints that predicts the typological data correctly suggests that one is on the right track, even if one does not yet have a thorough independent motivation for every single constraint. The logical structure of constraints is discussed in more detail in sec. 4.4.

2.3 Summary

In this chapter, the components of a linguistic OT account were demonstrated with the example of Grimshaw's 1997 analysis of inversion. The paramount role of constraint interaction as an explanatory device—which will be at the centre of discussion throughout this book—was pointed out, and the mechanics of cross-linguistic empirical predictions through factorial typology were demonstrated.

Factorial typology with its precisely specified relation between the assumptions made by a linguist (in the form of constraints) and their *cross-linguistic* empirical consequences is one of the major innovations of the OT framework, going along with the departure from classical

grammar models based on the concept of formal grammars which define sets of strings. In the classical string-language-oriented view, the analyses assigned to strings are subordinate to the main goal of specifying the correct string language. In the OT view, the objects of the theory are crucially pairings of surface forms and underlying content representation, as the discussion of sec. 2.2 showed. With this move, the learning task for a grammar model receives an entirely new character: learning no longer means trying to get the string language correct by making decisions on a subset of the rule specifications in a formal grammar (*parameters*, which appear rather arbitrary from a formalist point of view). Now learning means checking whether the pairing of form and content observed is predicted as optimal by the learner's own system, and if necessary correcting the constraint ranking. (In sec. 3.3, I will address learning in some more detail.)

3

Some observations about Optimality-Theoretic Syntax

In this chapter, some important observations about the basic components of a *syntactic* OT system are made—postponing detailed formal considerations to chapter 4. Sec. 3.1 contrasts the options for motivating OT constraints in phonology with the options available for a syntactic account. Sec. 3.2 addresses faithfulness violations in OT syntax and consequences for candidate generation; sec. 3.3 introduces the OT learning theory and discusses implications for the character of the input.

3.1 On the motivation of syntactic OT constraints

The formal rigour of typological predictions illustrated in sec. 2.2 is clearly one of the strengths of the Optimality-theoretic approach. At the same time, the examples discussed in that section point to a potential methodological problem—in particular when we are moving from phonology to syntax: above, we were dealing with toy examples of three constraints which have a clearly observable effect. But what if there are more constraints and their effect on the observed data is more indirect? (Phonological representations are more closely linked to observable effects than most aspects of syntactic representations.) More constraints give us a larger factorial typology to check, and what we have to check depends in part on theoretical assumptions. So it is not so easy to get an overview of the actual typological spectrum in the languages of the world. The question of whether or not a specific language

type (like (8c) in sec. 2.2) exists may be hard to answer. Not finding any evidence for some language type in the typological literature does not imply that it is not a possible language type.

This does not mean that the typological criterion is less important for evaluating syntactic OT systems, but it suggests that occasionally one has to make do without rigorous evidence according to this criterion.

Unfortunately, the issue of independent, functional evidence for syntactic constraints is even more problematic. Syntactic constraints are based on abstract representations, which are only indirectly related to observable evidence. Thus, the form that the constraints take always depends in part on the kind of representations one assumes. This makes it hard to find theory-independent grounding for constraints as can arguably be provided in the phonological domain (through phonetic grounding).

To a certain degree this reflects a problem that any theoretical account of syntax has—the value of the representations assumed can only be judged when looking at the interplay of all aspects of the theory. A special twist to the problem arises for OT syntax since the representations assumed are often inherited from another theoretical framework (GB, Minimalism, LFG). Obviously, some of the explanatory burden carried by some aspect of the original framework is now replaced by OT's mechanism of constraint interaction. This in turn may influence the "grounding" of the representations—or in other words, the same representation may be suitably motivated in the original framework, but may be foreign to the Optimality-theoretic variant of the framework.

There are a number of consequences one may draw from these circumstances. The ones picked up in this book are the following:

- Syntactic OT systems should be formulated in a precise way; this should help one isolate the effect of particular constraints. In particular the principles governing candidate generation have often been left implicit so far.
- Ultimately, a computational simulation of a complex syntactic OT system should facilitate the checking of empirical, typological consequences of the assumed constraints.[15]

[15]While this book does not provide a fully implemented system for general OT models, it discusses many of the relevant computational issues. For investigations that are compatible with certain restrictions discussed further down, a classical pro-

- A system containing only simple violable constraints (according to some precise measure of complexity) has fewer degrees of freedom than a system in which already the individual constraints are very powerful. A simple constraint system is thus preferable as an initial hypothesis; working out the consequences of such a system should lead to insights about the character/usefulness of OT-style constraint interaction.
- For the formal and computational approach, it is important to know what types of cross-linguistic variation occur (word order variation, presence/absence of certain overt elements and other aspects); this allows for conclusions about the character of the candidate generation function *Gen* and the expressiveness of the violable constraints.

Generally, the focus in this book is on formal and computational properties of syntactic OT systems and the role that these properties play in predicting certain empirical facts. Individual phenomena and particular sets of constraints for modelling them typologically are only used for occasional illustrations. (Quite obviously, the investigation of entire factorial typologies is impractical under this focus.)

3.2 Faithfulness violations in phonology and syntax

In this section, a central methodological principle of OT is identified (sec. 3.2.1). If this principle is taken seriously, the cross-linguistic observations discussed in sec. 3.2.2 for phonology and in sec. 3.2.3 for syntax enforce a particular liberty in the candidate generation *Gen*—impressionistically speaking, the deletion of material from the input and the addition of material to the output. In this section, the OT analyses are presented in a relatively informal way. In sec. 4.5, the issue will be reconsidered against the formalization of OT syntax in the LFG framework provided in chapter 4.

3.2.1 Constraint interaction as the main explanatory device

As has become clear from chapter 2, one of the key insights in the OT approach in phonology and syntax has been that variation between languages can be derived in a system assuming a universally invariant set of constraints on well-formed linguistic structures, where it is only the relative ranking of these constraints that differs cross-linguistically. Technically, an OT system is thus set up as the combination of

cessing system like the LFG parser/generator XLE (Xerox Linguistic Environment; `http://www.parc.xerox.com/istl/groups/nltt/xle/`) can be readily applied.

(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 does not 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:

(25) *Methodological principle of OT*
Try to explain as much as possible as an effect of constraint interaction.

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

As an end in itself, principle (25) would not be of much scientific value. What is behind it is the observation discussed in sec. 2.2: For certain linguistic phenomena, OT constraint interaction with its inherent *factorial typology* has been shown to successfully predict the space of cross-linguistic variation, including the systematic exclusion of certain logically possible languages. So the reason for following (25) 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? 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 re-ranking. A closer investigation of the formal and computational implications of this strong OT hypothesis is one way of checking to which degree the hypothesis is tenable. The present book can be seen as an attempt to follow this path, focusing on the division of labour between *Gen* and *Eval*. An important 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 critical constraints for the decidability issues are faithfulness constraints that lead to a significant discrepancy between the input and the output when violated. This way, they may lead to an infinite number of candidates. Do we need such constraints?

The reasoning on this issue in this book is as follows: In sec. 3.2.3, it is observed—rather informally—that a certain type of variation across the languages of the world exists (whether (i) expletive elements are used and (ii) pronominals may be dropped). If we want to model this type of variation as an exclusive effect of constraint interaction (following principle (25)), *Gen* has to have a certain property (generating particular faithfulness violations). The formal character of this property is discussed again in chapter 4. In chapter 6 the consequences for processing are addressed.

3.2.2 Epenthesis and deletion in phonology

Before going into the syntactic data, here is some background on deletions and epentheses as they are known from the phonological theory of syllable structure (cf. e.g., (Kager, 1999, ch. 3)). An important markedness constraint of syllabic well-formedness is ONSET:

(26) ONSET
*[$_\sigma$V—'Syllables must have onsets.'

Thus, when possible the underlying phoneme sequence is grouped into syllables in such a way that consonants will act as the onset of a new syllable: In Axininca Campa (an Arawakan language spoken in Peru), the input /no-N-čʰik-i/ is syllabified as [noɲ.čʰi.ki].

However, if two vowels are adjacent in the underlying form (as in /no-N-koma-i/),[16] this is not possible. Since (26) ONSET is high-ranking in Axininca Campa, the language makes use of an epenthetical consonant in such cases: (27a).

(27) Consonant epenthesis in Axininca Campa (Kager, 1999, 93)
a. /no-N-koma-i/ [noŋkomat**i**] 'he will paddle'
b. /no-N-čʰik-i/ [noɲčʰiki] (*[noɲčʰikt**i**]) 'he will cut'

This shows that faithfulness to the underlying input segments has to be a violable constraint. According to the Correspondence Theory account of McCarthy and Prince (1995), this is formulated as an instance of the DEPENDENCE constraint: input-output correspondence (28).[17]

(28) DEP-IO
Output segments must have input correspondents.—'No epenthesis'

The opposite option, dropping one of the two adjacent vowels, exists also. In languages like Modern Greek, the conflict between (26) ONSET and faithfulness is resolved in this way: /kanona-es/ ('rules') is realized as [kanones].[18] Hence, we have another violable faithfulness constraint—an instance of MAXIMALITY:

(29) MAX-IO
Input segments must have output correspondents.—'No deletion'

With all three relevant constraints in place, we can have a look at the tableau for (27) in Axininca Campa:

(30)

(i) Input: /no-N-koma-i/	ONSET	MAX-IO	DEP-IO
a. ☞ [noŋ.ko.ma.**ti**]			*
b. [noŋ.ko.ma]		*!	
c. [noŋ.ko.ma.i]	*!		

[16]Note that the assumption of this underlying form for (27a) is based on the observation that both examples in (27) end in the same underlying suffix.

[17]Correspondence is a relation between elements in two strings s_1 and s_2, where DEPENDENCE demands that every element of s_2 has a correspondent in s_1. Its dual is MAXIMALITY demanding that every element of s_1 has a correspondent in s_2. (Cf. also (Kager, 1999, 205).)

[18]Thanks to Effi Georgala for help with the Greek data.

The factorial typology of the three constraints (26), (28) and (29) leaves open a third option:[19] ranking both faithfulness constraints above the markedness constraints. The result are onset-less (but faithful) output structures as they exist in many languages, including English.

This typology tells us a lot about the candidate generation function *Gen*. Since it is universal (by assumption), it must produce candidates with epentheses and deletions in all languages. In the outcome, the freedom of generating candidates undergoes the filter of constraint interaction, which limits the number of grammatical sentences for a particular given language. One should bear in mind that this *Gen/Eval* generation/filtering system is an *abstract* model allowing us to structure the space of typological possibilities—rather than a processing model for human language production (or comprehension). Hence the conceptually most simple instantiation of this system will assume unrestricted epenthesis and deletion in *Gen*: since the constraints will filter out overly unfaithful candidates anyway it would be redundant to exclude them in candidate generation. Consequently, the abstract *Gen* function will generally "produce" an *infinite set of candidates*.

3.2.3 Syntactic variation across languages

Expletive elements

Now turning to syntax, the types of cross-linguistic variation one can observe motivate the assumption of a similarly liberal *Gen* function on the level of words as on the level of phonological segments. The cross-linguistic differences between the surface strings of winning candidates are very basic ones and were already discussed in the earliest work on OT syntax (cf. Grimshaw (1997)): for syntactic reasons, some languages require the use of expletive elements where other languages do not (cf. the expletive *do* in English (31a), vs. the German example (31b)).

(31) *Expletive do in English*

a. Who did John see

b. Wen sah John
whom saw J.

[19] There are six possible rankings of the three constraints, but the relative ranking of the two highest-ranking constraints never plays a crucial role.

According to the methodological principle of OT (25) discussed in sec. 3.2.1, 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 to 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 faithfulness constraint at stake here is input-output dependence: DEP-IO—we can assume exactly the same constraint as in phonology (28), interpreting words as the relevant output segments for syntax and semantic predicates as their input correspondents.

Note that one might also try to argue for the expletive status of *did* based only on a single language like English: one could point out the difference between minimal pairs of inverted and non-inverted clausal structures (*John saw her* vs. *Who did John see*), which suggest that the latter case contains an instance of overt material not present in the underlying form. However, this evidence would not exclude the possibility that the underlying form of questions contains some additional element that is being rendered faithfully in the English question. And if all languages behaved like English, *Gen* should not generate any choice for the questions. However, if one finds a cross-linguistic difference (as in (31) above), one can be sure that according to the OT assumptions, *Gen* has to provide both options.

Strictly speaking, the hypothesis of an underlying question-marking element cannot be excluded when other languages like German are taken into account that do not display this element overtly, since these other languages could rely on deletion, violating MAX-IO (29). The underlying form is not directly observable, but it rather models some semantic concept of content. However, for this concept of content we do not have any theory-independent intuition about what might count as a segment. So, in a sense it is an arbitrary decision whether English is unfaithful—violating DEP-IO—or German is unfaithful—violating MAX-IO. (Recall that a similar choice was discussed for the "polarity" of the IDENT-IO constraint in sec. 2.2.3.) The latter choice would however result in a rather unorthodox theory, so we can safely keep to the former option.

(32) is a tableau showing that (31a) does actually arise as the winner of optimization in English, according to the analysis of Grimshaw (1997).[20]

(32) *Optimization with unfaithful winner*

Input: $\langle$ read(x, y), x = Mary, y = what $\rangle$	OP-SPEC	NO-LEX-MVT	OB-HD	DEP-IO (FULL-INT)	STAY
a. [$_{VP}$ Mary reads what]	*!				
b. [$_{CP}$ what **e** [$_{VP}$ Mary reads t]]			*!		*
c. [$_{CP}$ what **reads**$_i$ [$_{VP}$ Mary **e**$_i$ t]]		*!			**
d. ☞ [$_{CP}$ what **does**$_i$ [$_{IP}$ Mary **e**$_i$ [$_{VP}$ read t]]]				*	**
e. [$_{CP}$ what **e** [$_{IP}$ Mary **does** [$_{VP}$ read t]]]			*!	*	*

The constraints OP-SPEC, OB-HD and STAY have already been introduced in sec. 2.1. The definitions (1) are repeated below for convenience. NO-LEX-MVT (33) is an additional markedness constraint Grimshaw assumes:

(1) OP-SPEC Syntactic operators must be in specifier position.
OB-HD (Obligatory Head) A projection has a head.
STAY Trace is not allowed.

(33) NO LEXICAL HEAD MOVEMENT (NO-LEX-MVT)
A lexical head cannot move.

In Grimshaw's terminology the DEP-IO faithfulness constraint is called FULL INTERPRETATION (FULL-INT). As (32) shows, the fact that NO-LEX-MVT outranks DEP-IO causes candidate d. to win over c. Under a ranking in which both OB-HD and DEP-IO dominate NO-LEX-MVT, candidate c. would be the most harmonic, i.e., we would have a language with inversion of the main verb.

Another well-known example for an expletive element is the English *it* filling the structural subject position in (34a) (cf. (Grimshaw and Samek-Lodovici, 1998, sec. 4)). Semantically, the verb *seem* takes just a proposition (realized as the *that*-clause) as an argument. Comparison with Italian (34b) shows that again, this type of expletive does not occur universally, so in English we have a DEP-IO violation.

[20]Grimshaw assumes that auxiliaries are always inserted in I^0, thus there is no candidate *what does Mary read* without an intermediate IP projection.

(34) *Expletive pronoun in English*

a. It seems that John has left

b. Sembra che Gianni è andato
seems that G. is gone

(36) and (37) show the tableaux deriving this effect in English and Italian, according to the analysis by Grimshaw and Samek-Lodovici (1998) and Samek-Lodovici (1996). SUBJECT (35) is the relevant markedness constraint.[21]

(35) SUBJECT
The highest A-specifier in an extended projection must be filled.

(36)

Input: ⟨ seem(x), x = ⟨ leave(y), y = John ⟩⟩	SUBJECT	DEP-IO (FULL-INT)
a. seems [that . . .]	*!	
b. ☞ it seems [that . . .]		*

(37)

Input: ⟨ sembra(x), x = ⟨ andare(y), y = Gianni ⟩⟩	DEP-IO (FULL-INT)	SUBJECT
a. ☞ sembra [che . . .]		*
b. *expl* sembra [che . . .]	*!	

Null elements

If we look at Italian examples with verbs that clearly take a semantic argument like *cantare* 'sing', we have an example of the opposite type of faithfulness violation, a MAX-IO violation: In *pro-drop* languages, there is no overt correspondent for the subject pronoun referring to a topical entity (cf. (Grimshaw and Samek-Lodovici, 1998, sec. 3)).[22] English

[21] The "highest A-specifier" refers to the specifier-to-IP position.

[22] (Classical) LFG construes the inflection on the verb as the subject in *pro-drop* sentences; in an OT version of LFG, one may assume that there is a constraint that prefers realization of arguments as full syntactic phrases, which is violated in the *pro-drop* case (compare the Iconicity and Avoid-Allotaxy constraints of Bresnan (2001a)). The *pro-drop* property of a language is then triggered by the ranking of this constraint.

(38a) contains an overt subject pronoun, Italian (38b) does not. This demonstrates that it is possible to leave some *input* material unrealized to satisfy some high-ranking Markedness constraint.

(38) *Dropped pronominal in Italian*

a. He has sung

b. _ ha cantato
has sung

Grimshaw and Samek-Lodovici's analysis deriving these results relies on the additional constraint (39) DROPTOPIC, which favours the *pro-drop* option, provided the element in the previous coreferent with the dropped subject pronoun is already the topic.

(39) DROPTOPIC (Grimshaw and Samek-Lodovici, 1998, 194)
Leave arguments coreferent with the topic structurally unrealized. Failed by overt constituents which are coreferential with the topic.

English (40) and Italian (41) differ in the relative ranking of DROPTOPIC and MAX-IO (called PARSE in Grimshaw and Samek-Lodovici's terminology); the other two constraints play no role in this particular comparison. In sec. 5.3.3, I will come back to this analysis discussing the status of the DROPTOPIC constraint.

(40)

Input: $\langle$ sing(x), x = topic, x = he $\rangle$	MAX-IO (PARSE)	DROPTOPIC	SUBJECT	DEP-IO (FULL-INT)
a. has sung	*!		*	
b. ☞ he has sung		*		

(41)

Input: $\langle$ cantare(x), x = topic, x = lui $\rangle$	DROPTOPIC	MAX-IO (PARSE)	DEP-IO (FULL-INT)	SUBJECT
a. ☞ ha cantato		*		*
b. lui ha cantato	*!			

3.2.4 Consequences for *Gen*

As already stated in sec. 3.2.1, identifying something as an effect of constraint interaction implies that the other component of an OT system, *Gen*, has to leave open all options for this effect. 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 and aspect information as in (42) as *input* (cf. (Grimshaw, 1997, 375)), we are thus faced with all items in (43) 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: (43f)).

(42) Input: laugh(x), x = Ann, TENSE = PAST

(43)
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. John yawned
k. 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. But as I argue in the following chapters, the LFG-based conception of OT syntax (chapter 4) provides a natural framework for modelling the intuitions about faithfulness violations addressed in this and the previous section in a way that allows one to structure the candidate space adequately for computational processing (chapter 6).

3.3 Learning and the character of the input

3.3.1 Learning in OT

As mentioned in chapter 1, it is one of the outstanding features of the Optimality-theoretic approach that it is intimately coupled with a formal theory of language learning. This means that not only can predictions for a single language be tested empirically—based on a formal specification of *Gen*, the constraints and the language-specific ranking, but moreover, the validity of a set of constraints can be determined empirically by using a learning algorithm on corpora of language data. For typologically different languages, the algorithm should arrive at different rankings over the constraint set provided as input.

Two approaches to the learning problem should be mentioned: the original *Constraint demotion algorithm* of Tesar (1995), Tesar and Smolensky (1998, 2000), and the *Gradual learning algorithm* of Boersma (1998), Boersma and Hayes (2001).

Constraint demotion

The *Constraint demotion algorithm* (Tesar, 1995, Tesar and Smolensky, 1998, 2000)[23] should be seen as an idealized and abstract algorithm whose main purpose is to show that a formal OT system is indeed learnable. Besides the set of constraints and the *Gen* function, the algorithm requires language data which contain an explicit specification of the underlying (input) form.

The algorithm assumes that several constraints can have the same rank—they can form a single *stratum*. Initially, *all* constraints are assumed to be in the same stratum, i.e., no constraint outranks any other constraint. The algorithm uses language data as evidence for removing a constraint from an existing stratum, because it should be dominated by some other constraint in that stratum. At each stage the current ranking of the learner is used as the hypothesis with which data are analyzed. A piece of data can be exploited as evidence when the learner's prediction does not meet the observed facts, i.e., the system makes an *error*, predicting an incorrect candidate to be the winner. What is important to note is that only *positive* evidence is needed for getting the ranking correct. This is possible since it follows from the OT assumptions that knowing the winning candidate in a competition implies that all other possible realizations of the input are ungrammatical. So the

[23] See (Kager, 1999, ch. 7) for a more thorough introduction.

ranking has to exclude all candidates but the winner. Since the constraints and *Gen* are known, the constraint marking (the number of stars in the tableau cells) of all candidates in a given competition are known. Now, the target ranking has to be such that for each losing candidate, the winner is more harmonic.

Let us assume for example a learner that has to deal with five constraints; and let us look at a stage where it has already built up a ranking with three strata: { CONSTR. 1, CONSTR. 2 } ≫ CONSTR. 3 ≫ { CONSTR. 4, CONSTR. 5 }. With this ranking, candidate A in (44) would be predicted to be the winner. However, B is the observed form for the underlying input. Hence, the assumed ranking must have been incorrect: CONSTR. 3 should outrank CONSTR. 1.

(44) *Detecting an error in the learner's system*

Candidate set:	CONSTR. 1	CONSTR. 2	CONSTR. 3	CONSTR. 4	CONSTR. 5
candidate A		*	*		
observed: candidate B	*!	*			*

More generally, we have to look at the constraint violations of two competitors in the following way: constraints violated by both the predicted winner (A) and the observed output (B) and constraints violated by neither of the two are ignored in a learning step (this excludes CONSTR. 2 and CONSTR. 4 from further consideration). Of the remaining constraints, the ones violated by observed output are called the *winner marks* (CONSTR. 1, CONSTR. 5), the ones violated by the incorrectly predicted output are called the *loser marks* (CONSTR. 3). For a correct prediction, the highest-ranking winner mark should be ranked below some loser mark. For example, CONSTR. 1 should be below CONSTR. 3.

At an early point in the process of building up the ranking for a given language, there is typically a choice of alternative constraint rerankings all of which lead to the correct prediction for a particular piece of data. However, some of these possibilities will lead to problems later on in learning, as more data are considered. The problematic cases would require a correction of decisions made earlier. The constraint demotion algorithm of (Tesar, 1995, Tesar and Smolensky, 1998, 2000) avoids the need for corrections by proceeding in a conservative way: at each step the minimal commitment is made that is required for getting

the current piece of data correct. This conservativity results if each of the winner-mark constraints is *demoted minimally*, i.e., just below the highest-ranking loser-mark constraint. (So, CONSTR. 1 is removed from the highest stratum and is placed in the stratum just below CONSTR. 3, and we get CONSTR. 2 ≫ CONSTR. 3 ≫ { CONSTR. 1, CONSTR. 4, CONSTR. 5 }, as in (46).)

(45) *Minimal constraint demotion*

Candidate set:	CONSTR. 1	CONSTR. 2	CONSTR. 3	CONSTR. 4	CONSTR. 5
candidate A		*	*		
observed: candidate B	*	*			*

(46) *Constraint ranking after learning step*

Candidate set:	CONSTR. 2	CONSTR. 3	CONSTR. 1	CONSTR. 4	CONSTR. 5
candidate A	*	*!			
observed: candidate B	*		*		*

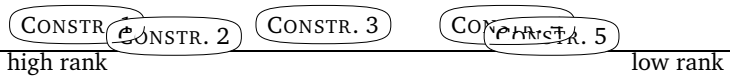
For a given candidate set, this *constraint demotion* step is performed repeatedly on all winner-loser pairs, until no further demotions occur. For a consistent set of learning data from a particular language (which can be described with the constraint set), the algorithm converges in a stable ranking, independent of the order in which the winner-loser pairs are considered.

Gradual learning

One idealization of the constraint demotion algorithm that is clearly unrealistic for modelling the real learning task is the following: the algorithm is designed to work on a perfect sample of language data. A single piece of erroneous data (e.g., an utterance from a different language or a slip of the tongue) may cause a constraint demotion that is inadequate for the target ranking, but from which the learner may not recover (an infinite loop of demotions may follow).

This is the main motivation for Boersma's 1998 proposal of a gradual learning algorithm (see also Boersma and Hayes (2001)), which is *robust* in the sense that a small amount of noise in the training data has no negative effect on the outcome of the learning process. The key idea is that on encountering a piece of data that is incompatible with the current constraint ranking, no radical reranking is performed but just a gradual adjustment. The constraints are ranked on a *continuous* scale, and whenever a harmony evaluation is performed, a small amount of noise is added in determining the actual rank (hence the framework is often called the Stochastic OT framework).[24] Diagram (47) is a schematic illustration of such a noisy constraint ranking. The actual rank of the constraints is more likely to be in the center than towards the edges of the ellipses; however for constraints with a small difference in the medium rank, like CONSTR. 4 and CONSTR. 5, both actual relative rankings have a fair chance of occurring.

(47)



When an error arises in learning (i.e., again, a piece of data is not predicted to be the winner under the learner's current ranking), the constraint ranking is not altered according to the demotion-only regime discussed above, but rather according to a global small-increment-promotion/demotion regime. This means that *all* winner-marks are demoted on the continuous scale by a small increment and *all* loser-marks are promoted slightly.

(48) *Promotion/demotion in the Generalized learning algorithm*

Candidate set:	CONSTR. 1	CONSTR. 2	CONSTR. 3	CONSTR. 4	CONSTR. 5
candidate A		*	*		
observed: candidate B	*	*			*
	→		←		→

[24]Random noise with a normal distribution is added when the effective ranking—the *disharmony*—of the constraints is determined. So, each evaluation is based on a slightly different effective constraint ranking. As learning proceeds, this ensures that the constraints maintain a safety margin in their ranking difference and the system can stabilize.

If the observed ranking error originated from noise in the data, its effect is soon neutralized by other evidence. If on the other hand, the piece of data revealed a real problem in the ranking, additional evidence to the same effect re-enforces the tendency. Data types occurring with sufficient frequency will cause a repeated demotion/promotion, so a quasi-categorical separation of the constraint strengths can result. If the data contain some degree of variability, opposing tendencies of constraint demotion/promotion will ultimately balance out in a way corresponding to the frequencies in the data (assuming a large enough sample is presented to the learner). So the probability distribution of the data generated by the OT system reflects the probability distribution in the observed data. This makes the stochastic OT approach very attractive for modelling optionality[25] (Asudeh, 1999) and frequency effects (Bresnan and Deo, 2001, Koontz-Garboden, 2001, Dingare, 2001). Moreover, with the assumption of a universal set of constraints, the stochastic OT architecture predicts that strict categorical effects in one language may surface as stochastic tendencies in another language; such evidence is adduced by Bresnan et al. (2001). Corpus-based learning experiments using the Generalized learning algorithm for syntax are reported in (Kuhn, 2002).

3.3.2 Criteria for learnability—implications for the input

Both learning algorithms sketched in the previous subsection are error-driven, i.e., it is crucial that the learner recognize the need for an adjustment in her constraint ranking. This need arises when for a given input, the winner predicted according to the learner's current constraint ranking differs from the actually observed winner. Almost trivially, this situation can only be recognized when the input for the observed winner is known.

In the idealization addressed above, one assumes that the input is given in the training data, along with the output form. However, realistically, the learner has to infer the underlying input form from the observed output and contextual and world knowledge. [26] If based only on the linguistic output form, this inference is non-trivial, since the input-output mapping may be highly ambiguous (most notably due to

[25] For discussion of optionality in standard OT, see (Müller, 1999, Schmid, 2001).

[26] Note that assuming the input to be explicitly represented in the candidate structures, even if deletions occurred (as ensured by the original *containment* condition of Prince and Smolensky (1993)), does not change the situation, since the learner will only have access to the surface form of the candidate representations.

the possibility of faithfulness violations).[27] Still, human learners do not seem to have problems with this inference. Presumably this is due to the predominant role played by contextual and world knowledge: from the immense space of possible underlying inputs for a given output, learners can exclude most options since it would be highly implausible that the speaker wanted to express this input in the actual context of utterance. To provide a simple example, assume a child hears the utterance *put the cup on the table* (and let us assume that she knows the denotation of *the cup* and *the table*). In an OT system modelling the learning of syntax, the possible underlying inputs must at least include options where the object denoted by *the table* is moved with respect to "the cup", or where "the cup" is moved under "the table", and others. However, knowledge about the acceptable positions of cups and the present physical position of the objects will make it fairly clear which possible underlying input the speaker had in mind.

It is a conjecture adopted in this book that this extra-linguistic reinforcement is not just a convenient guide in language learning, but that learning would be impossible without it. So, language learning is only possible once general cognitive abilities put the child in a position to *understand* the content of the utterance from which it learns—or at least relevant aspects of the content.

Note that it may be sufficient if the extra-linguistic information is available in a (possibly small) number of learning situations. As soon as the relevant aspects of grammar have been learned, they can themselves be exploited to narrow down the space of choices for underlying inputs. So the claim is not that generally in verbal communication, extra-linguistic information is crucial.

As these considerations about the prerequisites for learning suggest, the idealization of providing both input and output in the training data is not so far off.[28] The child's understanding of the learning situation

[27] A deep investigation of the task would have to take the bidirectional optimization architecture to be discussed in sec. 5.3 into account. This would make it even clearer that error-driven learning can work only if there is an independent way of judging whether the underlying interpretation predicted by the current OT system is correct (i.e., it corresponds to the message that the speaker intended to convey). Tesar and Smolensky (1998) assume the device of *robust interpretive parsing* to determine the underlying form (which we might view as a comprehension-based—or interpretive—optimization based on the strong bidirectionality account to be discussed in sec. 5.3). However, the assumption of such a device does not make the need for restriction (49) below redundant. It only provides a characterization of the inference process.

[28] See also (Kager, 1999, sec. 7.5.3) for a discussion of learning without given inputs.

provides enough independent information to make the task of input reconstruction from the output feasible. However, if this reasoning is right, certain restrictions follow for the character of the input we may assume for an OT system.[29]

(49) *Restriction on the character of the input*
All aspects of the OT input must be such that they can be in principle inferred from world knowledge and the general context of utterance.

A simple way of satisfying this restriction for OT syntax is to follow standard assumptions and work with an input that consists of the truth-conditional semantics of the utterance: all truth-conditional aspects of an utterance can in principle be inferred from the context of usage, since contextual and world knowledge (if available) will allow the hearer to verify whether the input proposition is true. But (49) is also compatible with a more finely structured input including further "pragmatic" clues, such as information structural status.

Note that it is not stated that for an arbitrary utterance, it must be possible to effectively infer all aspects of the input. The restriction is much weaker, demanding that the input information has to be of a character that can at least be inferred from the context under favourable circumstances. For instance, it sanctions the inclusion of speakers' intentions in the input, which are not normally clear in a conversation, but which may be deducible in a particular situation. A constraint relying on such input information may be difficult to learn, but it is nevertheless possible if a sufficient amount of "training material"—utterances in certain conversational contexts—are available. (And if we think of irony or sarcasm, which one might want to treat as very intricate forms of faithfulness violations, it is hard to imagine how one can learn to assess them without ever having seen a rather obvious case of them.)

Since the restriction (49) is formulated so weakly, it may appear almost ineffective; however it *does* exclude the assumption of a highly

[29] I would postulate this restriction not only for OT syntax, but also for the inputs in OT phonology. The assumption of input-level distinctions that get neutralized in the output is only justified when the learner can exploit independent evidence for this—for instance from a morphologically related form not displaying the neutralization effect. (For example, the [t] in the Axininca Campa example (27) [noŋ.ko.ma.**t**i] can only be identified as epenthetical if the learner realizes that (i) [noŋ.ko.ma.**t**i] is future tense, (ii) in the general case the future morpheme /-i/ appears without the t, and (iii) the t is not part of the verb stem.)

abstract concept of input with no connection to the utterance situation at all. Such a concept of input has been proposed to deal with the phenomenon of language-particular ineffability (Legendre et al., 1998), discussed briefly in the following section.

3.3.3 Language-Particular Ineffability

One of the puzzles for the Optimality-theoretic approach is how to account for situations where in a particular language, there is no grammatical way of stating a certain state of affairs, while this is perfectly possible in another language. This phenomenon is called language-particular ineffability. The standard example is multiple *wh*-questions in Italian (although they are often just discussed in the abstract). The English question (50), transferred literally into Italian yields an ungrammatical sentence: (51); and there is no other grammatical way of expressing this thought.[30]

(50) Who ate what?

(51) *Chi ha mangiato che cosa? (Zeevat, 2000, (2))
who has eaten what

The phenomenon constitutes a well-known problem for the Optimality-theoretic approach: optimization is defined in such a way that there is at least one winning candidate for every competition. Assuming furthermore that all competitions are universal (apart from the aspect of which candidate wins), the grammaticality difference between English and Italian is unexpected. For the competition that (50) is the winner of in English (with the input in (52)), there should be some winner for Italian too.

(52) $\langle$ eat(x, y), $x =$ who, $y =$ what $\rangle$

Legendre et al. (1998) propose a solution to this puzzle: they assume a highly abstract input, which they call *Index* in order to circumvent potential derivational connotations. The only purpose of this Index is to define candidate sets for optimization; it is unrelated to anything playing a role in a particular application of the grammar in language production. In what Legendre et al. (1998) call the *inventory view* of an OT grammar, optimization has its place only in the abstract definition of the inventory of forms available in a particular language. The most important consequence of the abstract view of the input/Index is that

[30] At least in a single clause.

the part of the candidate structures that fixes their interpretation (the logical form/LF) can be affected by faithfulness violations. So the candidates in a competition need no longer have the same interpretation.

This set-up opens up the possibility of a neutralization account for the ineffability data: both in Italian and in English, we have two competitions with the following two inputs/Indices:

(53) a. $\langle$ Q_i Q_j eat(x_i, y_j) $\rangle$
b. $\langle$ Q_i eat(x_i, y), y = something (INDEFINITE) $\rangle$

(53a) is the interpretation for the multiple *wh*-question as in (52) (with explicitly marked scope, using Q-operators which bind variables in the argument positions). (53b) is the interpretation of a different question, with just a single *wh*-operator asking for the subject, while the object is indefinite.

In English, the two competitions have different winners: (50) for (53a) and (54) for (53b).

(50) Who ate what?
(54) Who ate anything?

In Italian, neutralization occurs, and we get the same candidate—presumably something like (55)—as the winner in both competitions.[31]

(55) Chi ha mangiato qualche cosa?
who has eaten anything

This is because in Italian, a markedness constraint that is violated by multiple *wh*-questions (*ABSORB, according to Legendre et al. 1998) is ranked higher than faithfulness to the input/Index LF. As the relevant faithfulness constraint Legendre et al. (1998) assume a constraint PARSE(wh), which would be IDENT-IO(wh) in the terminology adopted in the book. So the conflict that English resolves at the cost of violating *ABSORB is resolved in Italian at the cost of a winner whose LF is unfaithful to the input/Index LF. Another option of resolving the conflict is

[31]For the detailed discussion of their approach Legendre et al. (1998) use Chinese examples—I am simplifying their discussion here, filling in examples for the discussion of Italian. Roberta d'Alessandro (p.c.) points out that a more natural way of expressing the existentially quantified object with *eat* in Italian would be with an understood object:
(i) Chi ha mangiato?
The argumentation would go through with this form as the neutralization target too, but I wanted to avoid adding MAX-IO violations as additional faithfulness effects to the present discussion.

by fronting both *wh*-words, as Bulgarian does, but this violates another markedness constraint *ADJOIN, which also dominates the faithfulness constraint in Italian. Note that the conflict between faithfulness to *wh*-marking and the *ABSORB/*ADJOIN markedness constraints does not arise for the simple *wh*-question input (53b): with a single question operator no adjunction or absorption is required.

The competition for (53a) is sketched in tableau (56), with the English ranking. Candidate a. uses absorption to make sure that the two question operators are interpreted correctly; candidate b. uses adjunction for the same purpose; candidate c. unfaithfully drops the *wh*-character of the object argument to avoid the problem. In English, candidate a. wins. With IDENT-IO(wh) ranked lowest, candidate c., which has a meaning that is different from the input, becomes the winner (as in Italian). Note that c. has a different meaning than the input.

(56)

Input: $\langle$ Q_i Q_j eat(x_i, y_j) $\rangle$	*ADJOIN	IDENT-IO(wh)	*ABSORB
a. ☞ who$_{i[j]}$ t$_i$ ate what$_{[j]}$			*
b. who$_i$ what$_j$ t$_i$ ate t$_j$	*!		
c. who$_i$ t$_i$ ate anything		*!	

For comparison, here is the corresponding part of a tableau for the alternative input (53b), in which the constraints focused on here take no effect:

(57)

Input: $\langle$ Q_i eat(x_i, y), y = something (INDEFINITE) $\rangle$	*ADJOIN	IDENT-IO(wh)	*ABSORB
a. ☞ who$_i$ t$_i$ ate anything			

One should bear in mind that these results have to be regarded under the inventory view of optimization. This means it is inappropriate to think of ineffability in the following naive way: when Italian speakers have in mind a message like (53a), they will try to produce an utterance; but due to the way the constraints are ranked Italian speakers

are bound to fail conveying the message, so they will say something else instead. The input/Index has no status in actual language use, it merely defines the inventory of linguistic structures available in a language. Multiple *wh*-questions are not available in Italian, so speakers will not try to use them.

3.3.4 The problem for learnability

The abstract view of the input/Index assumed by Legendre et al. (1998) works fine for an adult OT system with the constraint ranking fixed—and it works also for the idealized learning task, in which the training data include an explicit indication of the input/Index. As discussed in sec. 3.3.2, this idealization is arguably even justified as a model for real language learning as long as the input keeps the semantic content fixed across all candidates. Assuming the semantic input representation to be known in learning allows one to factorize out all kinds of general cognitive abilities involved in language learning, without making any unjustified idealizations: at least in favourable contexts the semantic input will be effectively deducible, which suffices to guarantee learnability over time.

However, for the abstract input/Index approach, it is a totally unnatural assumption that information about the input/Index is context-deducible for the learner: By assumption there is no connection between the utterance situation and the input/Index LF (as opposed to the candidate's LF, which is potentially unfaithful to the input/Index LF). In other words, when the neutralized Italian *who ate anything* (55) is uttered by an adult, the learner cannot tell whether it is the winner of a competition (56) or (57)—competitions are abstract and determine only the inventory.

But then there is no way of learning a language with a low-ranking LF-sensitive faithfulness constraint.[32] At the beginning, IDENT-IO(wh) has the same ranking level as the markedness constraints *ADJOIN and *ABSORB (they are all in the same constraint *stratum*). Then the learner will always hear neutralized data (since no others are in the adults' inventory). What should cause the learner to assume the more complicated competition (56) rather than (57)? Comparing the inputs according to a Lexicon Opimization scheme (as mentioned briefly in

[32]If one could assume explicit negative evidence in learning, the situation would be different of course. But it is one of the basic tenets of the OT learning theory to assume only positive evidence.

sec. 2.2.1 on page 16), would lead to (57), since it has the more harmonic constraint profile. But note that only the former would trigger the demotion of IDENT-IO(wh) required for adult Italian.

Even if there were some way of enforcing this move, it is not clear whether the option of assuming low-ranking LF-faithfulness is sufficiently constrained. Is there anything that would keep learners from re-interpreting a piece of evidence for which their current ranking predicts wrong results under the assumption of an LF-faithful winner? At least, the complexity added by the possibility of assuming an LF-unfaithful winner is enormous. Essentially, any semantically relevant feature of the training data evidence can be reconstructed as rendered unfaithfully, leading to a great numbers of possible abstract competitions. So it is highly likely that some of the learner's errors that should trigger learning steps involving markedness constraints (according to the error-driven scheme), will be incorrectly attributed to LF-unfaithfulness.

Learning is only a side issue in this book, but I take the concerns discussed here as a further basis for enforcing the restriction on the character of the input (49) introduced in sec. 3.3.2. The interpretation of the candidates has to be fixed by the input/Index. This excludes the derivation of language-particular ineffability proposed by Legendre et al. (1998), but as will be discussed in sec. 5.3.2 and sec. 5.3.5, the assumption of bidirectional optimization provides the basis for a different explanation that avoids the learnability problems discussed here.[33]

3.3.5 The non-derivational view of the input-output connection

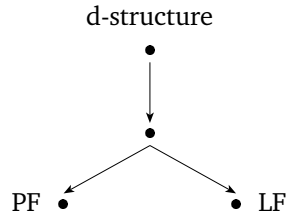
At this point, I should note that while for learnability reasons I do not follow Legendre et al. (1998) in the assumption of LF-unfaithful candidates, the insistence on an inventory view of optimization is very helpful—especially for derivational syntactic theories embedded in OT. (For inherently non-derivational approaches like Optimality-theoretic Lexical-Functional Grammar, OT-LFG, the danger of getting caught in a misleading thinking is much smaller for this matter.)

In a (Chomskyan) derivational account of syntax, each candidate analysis is a derivation, starting out from a d-structure representation and resulting in a phonetic form (PF) and a logical form (LF) —known as the Y-model (58).[34]

[33] Compare also (Smolensky, 1998, Lee, 2001a).

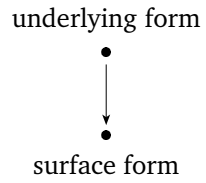
[34] See for instance (Chomsky, 1995, ch. 1), Haegeman (1994).

(58) *The Chomskyan Y-model for syntax*

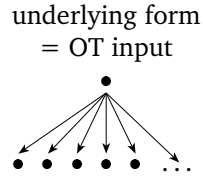


On a technical level, this seems to suggest a suitable parallel with OT phonology: generative phonology is also based on a derivational process from an underlying, "deep" structure to a surface form that is realized phonetically (59). This permits a straightforward visualization of candidate generation in OT as a parallel derivation following all possible paths simultaneously (60), and then picking the most harmonic candidate as the actual output. (The derivation of *all* candidates can of course only be meant metaphorically, since there may be infinitely many of them. Hence, even here an inventory perspective is more realistic.)

(59) *The generative model of phonology*

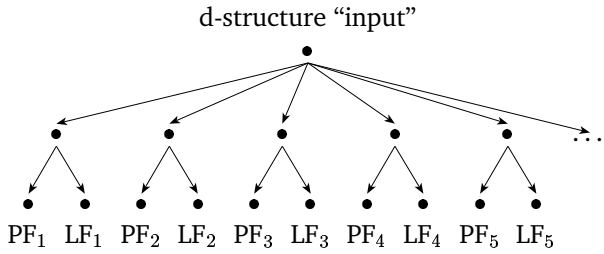


(60) *Candidate generation in OT phonology*



Treating the syntactic Y-model in parallel with the phonological model, gives us the following conception of syntactic OT:

(61) *Simple model of candidate generation with the syntactic Y-model*



However, this parallel misses an important conceptual difference: in phonology, the underlying form is indeed the interface to the conceptual level—by assumption, the underlying forms are what is stored in the mental lexicon along with the semantic and morphosyntactic information for the entry. Thus, we can envisage language production as a process involving the selection of some lexicon entry for which the stored phoneme string is retrieved and a phonological optimization is performed. Whether or not one adopts an inventory view or a particular input view does not have any important implications for the candidate set and for the learning situation. The learner will detect errors based on independently inferred underlying forms (e.g., assuming uniformity of stem forms across a morphological paradigm[35]).

The status of d-structure is *technically* closely related—it is the only "input" to the derivational process of syntax. However, contrary to what holds in the generative process of phonology, d-structure is not the sole interface to the conceptual parts of the cognitive system. The logical form/LF plays as important a role, such as fixing the scope of quantifiers. So modelling the OT input (the Index defining candidate sets) as only comprising the "input" of syntactic derivations (i.e. d-structure, or a numeration, that is an unstructured bag of lexical items) leads to unnatural results—at least when OT is viewed as more than just a technical device without any intuitive connection to the learning problem. For instance, LF_1 in (61) may have an interpretation different from LF_2 (see e.g., Sternefeld (1997)).

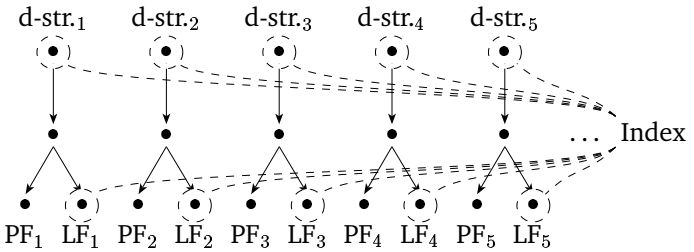
The obvious solution is to extend the OT input to also comprise the LF information. For technical reasons, this is not compatible with a literally derivational OT model: the LF information for a candidate is not available until its derivation is finished.[36] Thus, it is impossible to

[35]Compare footnote 16 on page 32 and footnote 29 on page 45.

[36]This fact leads to a terminological conflict, which other researchers resolve in a dif-

keep up the derivational character of *both* the OT competition with its candidate-set-defining input *and* the individual candidate analyses with the input d-structure. Legendre et al. (1998) give up the derivational view of OT candidate set generation from an input. Their abstract Index with its inventory-defining character applies at two points during the candidate derivation, as suggested by the sketch in (62): The specification of the Index consists of a d-structure with target positions for the operators (cf. (53) above); thus both the candidates' d-structure and their LF are fixed (subject to faithfulness violations). For the individual candidate analyses, Legendre et al. (1998) *do* still assume a derivational character; one may think of the Index as a filter on candidate derivations.

(62) *Candidates in OT syntax according to the inventory view of Legendre et al. (1998)*



An alternative solution to the problem is to assume static, non-

ferent way. Heck et al. (2000) make a similar observation as the one just reported and come to the following conclusion: candidate set specification in syntax cannot be performed by the input (in the sense of *starting point of derivational syntactic analyses*) exclusively—some independent criterion is needed. This independent criterion (which they do not investigate any further) is what I continue to call input/Index in this book (with the sole function of specifying the candidate set).

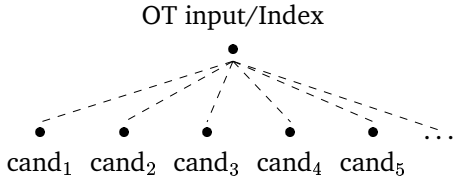
An independent issue discussed in (Heck et al., 2000) is the question to what extent the input has to be referred to when checking faithfulness constraints. With the rich representations assumed for syntactic candidates, they conclude that separate bookkeeping of the input is redundant in syntax. (This fact follows quite naturally if candidates are made precise in a formal framework, like in the OT-LFG account of (Bresnan, 1996, 2000, Kuhn, 2001c), which I will introduce in chapter 4; the redundance of separate input-bookkeeping is addressed specifically in sec. 4.5.2, on page 121.)

The general conclusion drawn by Heck et al. (2000)—that in OT syntax, no input is required at all—should be seen against their specific terminological background. Certainly, candidate set specification on the basis of some partial linguistic information is needed throughout all OT approaches in phonology or syntax. However, the straightforward derivational view of candidate set specification is unproblematic for phonology, but incompatible with syntactic candidate analyses based on a standard derivational approach.

derivational candidate representations. In principle, this could be realized by a redefinition of the Chomskyan framework: the chain representations standardly assumed for movements just have to be governed by strictly declarative principles. Then all structural levels will come into existence simultaneously, satisfying the well-formedness principles. This intuition seems to underlie in much OT syntax work, although the derivational metaphor of movement tends to thwart intuitions occasionally. This is one of the motivations for assuming a strictly non-derivational basis for OT syntax as first proposed by Bresnan (1996, 2000) for the formalism of Lexical-Functional Grammar. This framework is also adopted in this book (cf. the formalization in chapter 4).

The advantage of assuming non-derivational candidate representations is that the relation between the candidate-set-defining Index (or OT input) and the individual candidate analyses is conceptually simpler than in the model sketched in (62). The input/Index can be viewed as a formal object of the same type as the analyses (with the possibility that the input/Index is specified only partially). Faithfulness constraints and potential restrictions on the candidate generation function *Gen* can then be defined as an abstract relation between two formal objects of this type: what arises can be sketched as in (63), assuming that the dots represent formal objects comprising all structural levels relevant for a particular range of phenomena.

(63) *Strictly non-derivational view of candidate generation (for OT phonology or syntax)*



Although this conception is radically different from a simple derivational OT model (as sketched for phonology in (60)), it is isomorphic in its general structure. So the inventory view of OT competition, based on non-derivational candidate analyses emerges as a simple, but general framework adequate for all application areas of OT. A further advantage of the declarative model—the straightforward "reversability" of the direction of candidate generation—will be discussed in sec. 5.1.

For the non-derivational architecture, the term "input" might still be misleading if understood literally. However, firstly the terminolog-

ical conflict with the candidate-level derivations of (62), which have their own "inputs", has been eliminated; secondly, a metaphorical terminology suggesting abstract processes is rather customary for formal systems with a strictly declarative formulation (compare for instance automata theory). Therefore, I will continue to use the term *input*, even though the view of OT competition I adopt comes closer to the inventory view that Legendre et al. (1998) describe.

3.4 Summary

In this chapter, I identified some of the empirically relevant aspects of the OT system that a formalization must pinpoint and which furthermore have to be addressed in a computational implementation of an OT system. Sec. 3.1 provided a brief discussion of the problems of finding independent motivation for syntactic OT constraints—a circumstance which can be viewed as motivation for doing foundational work on the formal and computational properties of syntactic OT systems. In sec. 3.2, differences across the languages of the world were investigated under one particular aspect: is there a difference in the practice of inserting semantically empty, i.e. epenthetical, material on the one hand, and in the overt realization of underlying material? As is well-known the answer is positive. This has consequences for the formal setup of OT syntax, in particular for the candidate space to be generated by the function *Gen*—assuming one takes the methodological principle seriously that one should try to explain all cross-linguistic differences through constraint interaction. In sec. 3.3, the basic assumptions of the OT learning theory were introduced, motivating a restriction on the character of the input. To ensure learnability it must be possible in principle to infer all aspects of the input from the utterance context. This excludes the assumption of LF-unfaithful candidates based on an abstract input. Furthermore, it was observed that a non-derivational formalization of an OT system has conceptual advantages and will therefore adopted in the remainder of this book.

February 4, 2003

4

The formalization of OT Syntax in the LFG framework

In sec. 3.3.5, the conceptual advantages of a non-derivational framework for OT syntax (and OT systems in general) were discussed. A strictly non-derivational framework for OT syntax on a formally rigid basis was first proposed by Bresnan 1996, 2001a, 2000. In (Bresnan, 2000, sec. 2), Bresnan presents a relatively close reconstruction of Grimshaw's 1997 OT system with the formal tools of LFG, providing further arguments for the non-derivational approach.

In this chapter, I introduce an LFG-based formalization along Bresnan's lines, discussing choices in the exact specification against the background of the general empirical and learning issues of chapter 3.

There may be various reasons for adopting the LFG formalism as the basis for an OT account of syntax, including for instance the fact that a lot of typological research has been conducted in the LFG framework. For the present purposes, the main advantage of picking LFG as the base formalism is however that its formal and computational properties have undergone thorough research and that there are highly developed systems for processing the formalism. In fact, one might say that one goal in developing the OT-LFG model is to arrive at a sufficiently restricted formalism for OT syntax in general to allow computational processing—in much the same way as the design of the LFG formalism was guided by linguistic and computational objectives: finding a framework that is expressive enough for an explanatory linguistic theory, but which at the same time guarantees that the processing tasks for grammars in that formalism are computationally tractable.

It should be noted that there is a fairly rich literature on the formalization and computational properties of OT phonology: Ellison (1994), Frank and Satta (1998), Karttunen (1998), Hammond (1997), Eisner (1997), Gerdemann and van Noord (2000), Jäger (2002c,b), and others. However, the approach adopted by these researchers is based on regular languages and rational relations, which are not expressive enough for modelling the syntactic domain.[37]

4.1 Background on Lexical-Functional Grammar

In this section, I present a brief review of the most important formal concepts of Lexical-Functional Grammar (LFG). For more details the reader is referred to the papers in (Dalrymple et al., 1995) (for aspects of the formalism) and to the recent textbooks (Bresnan, 2001b, Dalrymple, 2001, Falk, 2001).

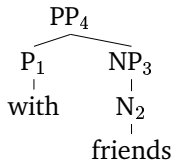
LFG is a non-derivational, monostratal paradigm for the representation of grammatical knowledge, first defined in Kaplan and Bresnan (1982). The key idea is to assume a number of different mathematical objects for the formal representation of different dimensions of linguistic utterances—most centrally the categorial dimension, represented through the phrase structure trees of c-structure, and the functional dimension, represented through the directed graphs of f-structure. [38] As a simple example, the c-structure and f-structure for the prepositional phrase *with friends* is given in (64)[39] and (65).

[37] An interesting way of generalizing the results from formal/computational OT phonology to syntax is the generalization from regular languages to regular tree languages, as in Wartena (2000) (compare also Jäger (2002c)). However, following the thread of computational-linguistic work on LFG and related formalism (as I do in this book) has the advantage that much theoretical and implementational work is already in place and can be applied with little need for adjustments.

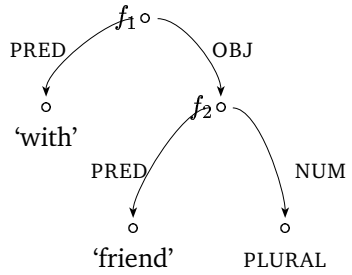
[38] (Kaplan, 1995, 11) characterizes f-structures as (hierarchical) finite functions; the graph model may however be more intuitive to readers familiar with work on unification. In the graph, identity of the values of two features (possibly under differnt paths) means that there is a single node, to which two feature arcs are pointing.

[39] The categories are indexed in this example to suggest that we are dealing with particular instances of the category types PP, P, NP, and N.

(64) *C-structure*



(65) *F-structure*



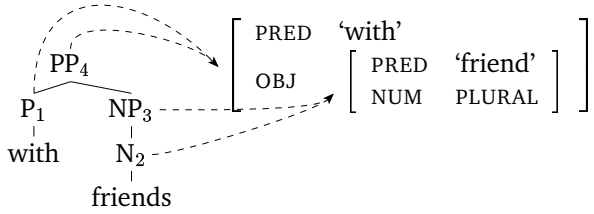
The arcs in the f-structure graph are labelled with features (PRED, OBJ and NUM); the atomic (i.e., leaf) f-structures are constants ('with', 'friend' and PLURAL). The common notation for f-structure graphs is as an attribute value matrix, like in (66).

(66) $$\begin{bmatrix} \text{PRED} & \text{'with'} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{'friend'} \\ \text{NUM} & \text{PLURAL} \end{bmatrix} \end{bmatrix}$$

The elements of the different structures stand in a correspondence relation, in the case of c- and f-structure a function ϕ mapping c-structure category nodes to f-structures. This function is often called a *projection*. In the example, ϕ maps both P_1 and PP_4 to f_1; f_2 is the ϕ image for both N_2 and NP_3. Note that the function ϕ need not be one-to-one, nor onto.[40] Sometimes, the function ϕ is shown explicitly in the representations, using arrows:

[40]Examples for f-structures that are not the image of any c-structure node are those representing arbitrary subjects like the subject of 'read' in the LFG analysis of sentences like *reading her book is fun*. In LFG, one does not assume a phonologically empty c-structure node for such a sentence.

(67)



As the system is non-derivational it is important to note that one should think of all structures as coming to existence simultaneously. Projection from c-structure to f-structure should not be seen as a process; neither c-structure nor f-structure is prior to the other.

Knowledge about legal linguistic representations is formulated through propositions in a description language. For the different formal objects (trees and directed graphs), different description languages exist: the trees of c-structure are described by a context-free grammar, [41] the f-structure graphs by formulae of a feature logic—by so-called *f-descriptions*. A valid f-structure is defined as the minimal model satisfying all f-descriptions (plus the additional well-formedness conditions of Completeness and Coherence, which I will discuss briefly below).

(68) *Context-free grammar as a description of c-structure*
PP → P NP
NP → N
Lexicon entries:
with P
friends N

(69) *Feature logic formulae as a description of f-structure*
$(f_1 \text{ PRED}) = \text{'with'}$
$(f_1 \text{ OBJ}) = f_2$
$(f_2 \text{ PRED}) = \text{'friend'}$
$(f_2 \text{ NUM}) = \text{PLURAL}$

Obviously, a list of equations is interpreted conjunctively. But the feature description language contains also other Boolean connectives, so

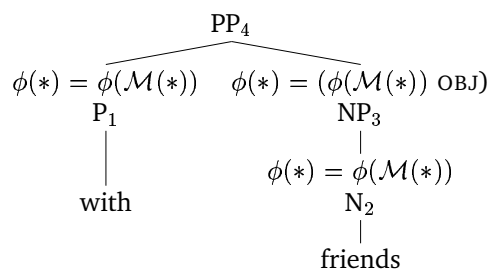
[41] To be precise, a generalization over context-free grammars is used: the right-hand side of the production rules is not defined as a string of non-terminals/terminals, but as a regular expression over non-terminals/terminals. For each language generated there exists a weakly equivalent context-free language. The advantage of allowing regular expressions in the productions is that optionality etc. can be expressed without loss of generalization, as the following rule illustrates: VP → V (NP) PP*

a formula can be negated, two formulae can be connected disjunctively etc.[42]

The correspondence relation between the structural dimensions is also expressed in the description language. Since the relation between c- and f-structure is a function, functional terms can be used to denote the f-structure corresponding to a given c-structure node. Thus, $\phi(\mathrm{PP}_4)$ refers to the f-structure projected from PP_4, i.e., f_1; $\phi(\mathrm{NP}_3)$ refers to f_2, etc. To allow for generalized statements about the c-/f-structure correspondence we want to be able to formulate *schemata* for f-descriptions, relative to a particular c-structure node. Such schemata can be attached to their category nodes (this is sometimes shown by writing the descriptions above the node in the tree representation). In the schemata, a metavariable $*$ can be used, which is instantiated to the particular c-structure node that the description is attached to.

$\phi(*)$ denotes the f-structure projected from the node that the f-description is attached to. For expressing the relation between the f-structures projected from different nodes, we also need to be able to refer to the mother of the current node; this is done by $\mathcal{M}(*)$. Again, $\phi(\mathcal{M}(*))$ denotes the f-structures projected from the mother node. So we can stipulate that the f-structures projected from the current node and its mother node are identical by stating $\phi(*) = \phi(\mathcal{M}(*))$ (used for both P_1 and for N_2). The f-description attached to NP_3 stipulates that the f-structure projected from this NP is the value of the feature OBJ in the mother node's f-structure.

(70)

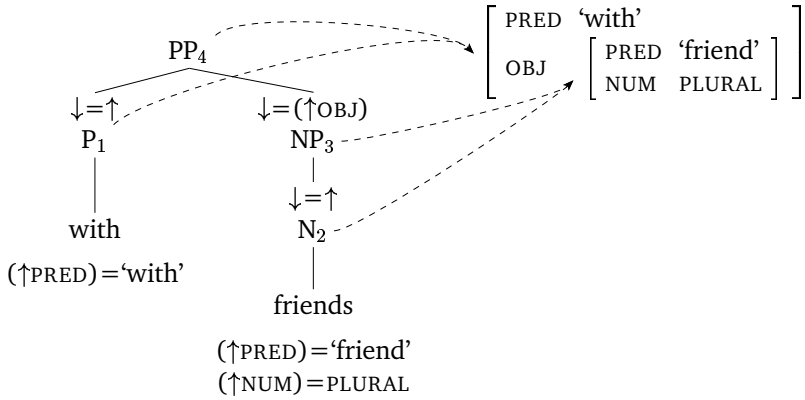


Since $\phi(*)$ and $\phi(\mathcal{M}(*))$ are used extensively, there are abbreviations for these expressions, which are also called metavariables (for

[42]Apart from (defining) equations, there exist other primitive types of formulae: existential constraints about feature paths, which are true when the path exists (through some other defining equation); constraining equations, which are true when a feature bears a particular value (again, defined elsewhere); and set membership constraints.

f-structures): ↓ and ↑. As becomes clear when the f-descriptions are written above the nodes, the ↓ symbol is mnemonic for the (f-structure projected from the) current node—the arrow points to it. Likewise ↑ is mnemonic for the (f-structure projected from the) mother node:

(71)



The f-descriptions originating from the lexicon (or more generally, representing morpholexical information) are typically written below the phonological/orthographic form shown as the leafs of the c-structure tree. Nevertheless, ↑ is used to denote the preterminal nodes in the tree representation (P_1 and N_2 in the example). The phonological/orthographic forms—"with", "friends" – should not be seen as full syntactic objects, but rather as indications of the phonological properties of the syntactic words, represented as P_1 and N_2.

With these means of expression, all grammatical knowledge can be encoded in a context-free grammar with feature annotations (or short: f-annotations). The annotations are schemata for f-descriptions to be attached to the category nodes in the c-structure tree. In the rule specification, f-annotations are typically written below the category that they are attached to. (72) shows the LFG grammar and lexicon required to analyze our example PP.

(72) *LFG grammar: Context-free grammar with f-annotations*

PP	→	P	NP
		↑=↓	(↑OBJ)=↓
NP	→	N	
		↑=↓	

Lexicon entries:

with	P ∗	(↑PRED)=‘with’
friends	N ∗	(↑PRED)=‘friend’
		(↑NUM)=PLURAL

As mentioned briefly above, there are two additional conditions on the well-formedness of f-structures: Completeness and Coherence. These ensure that the subcategorization frame introduced by verbs, prepositions and other lexical categories is actually filled in syntactically. For example *John devoured* fails to satisfy Completeness, while *John yawned a car* is ruled out by Coherence. Technically this is achieved by specifying the selected governable grammatical functions (SUBJ, OBJ etc.) in the PRED value of verbs, prepositions etc., so we would have (↑PRED)=‘with⟨(↑OBJ)⟩’ and (↑PRED)=‘devour⟨(↑SUBJ) (↑OBJ)⟩’.[43] The composite PRED values may be seen as abbreviatory for feature structures as in (73).

(73) *Composite* PRED *values*

$$\begin{bmatrix} \text{PRED} & \text{‘with}\langle(\uparrow\text{OBJ})\rangle\text{’} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{‘friend’} \\ \text{NUM} & \text{PLURAL} \end{bmatrix} \end{bmatrix}$$

expands to

$$\begin{bmatrix} \text{PRED} & \begin{bmatrix} \text{FUNCTOR} & \text{with} \\ \text{ARGUMENT1} & \end{bmatrix} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \begin{bmatrix} \text{FUNCTOR} & \text{friend} \end{bmatrix} \\ \text{NUM} & \text{PLURAL} \end{bmatrix} \end{bmatrix}$$

The expanded feature structure encodes the functor/argument structure of the predicate[44] and forms the interface to a conceptual representation. The arc indicates that the value of the feature path PRED ARGUMENT1 and OBJ are the same f-structure object. The two values are then said to be *re-entrant* or *structure shared* (compare footnote 38).

Based on these composite PRED values, Completeness and Coherence can be formulated as follows:

[43]Under a more generalized account, only the semantic arguments are specified and the choice of particular grammatical functions is derived through mapping principles (Bresnan, 2001b, ch. 14).

[44]Implementations of LFG parsers like the Xerox Grammar Writer's Workbench Kaplan and Maxwell (1996) and the Xerox Linguistic Environment (XLE; `http://www.parc.xerox.com/istl/groups/nltt/xle/`) implement composite PRED values along these lines.

(74) *Completeness*
All arguments specified in a predicate's subcategorization frame are also realized in this predicate's f-structure.

(75) *Coherence*
Only those governable grammatical functions are realized in a predicate's f-structure that are specified as arguments in the predicate's subcategorization frame.

One further peculiarity about PRED values should be noted. Their values are by definition interpreted as *instantiated symbols*. This means that even if the same predicate, say 'friend', is introduced twice in a sentence, the two instances will be distinct f-structure objects 'friend'$_1$ and 'friend'$_2$ and thus cannot be unified. This reflects the resource sensitivity of language and excludes that arguments are doubled, inserting them simultaneously in several c-structural position where they may occur alternatively. An example would be the following ungrammatical German sentence with the subject both in the preverbal *Vorfeld* position, and in the *Mittelfeld*:

(76) *Mein Freund ist heute mein Freund angekommen
my friend is today my friend arrived

The language generated by an LFG grammar can be defined as the set of c-structure/f-structure pairs, such that the c-structure is generated by the context-free grammar and the f-structure is the corresponding minimal model for the f-descriptions, satisfying Completeness and Coherence. There is however one proviso in the definition (Kaplan and Bresnan, 1982, 266): Only those c-structures are considered which do not contain recursive non-branching dominance chains, as illustrated in (77).

(77) *Non-branching dominance constraint/Offline Parsability*

This restriction, commonly known as *offline parsability*, ensures that for a given string there are only finitely many c-structure analyses and thus the parsing task for LFG grammars is decidable (i.e., a procedure can be devised for the task which terminates after a finite number of steps). Quite obviously, since without this restriction a context-

free grammar may predict an infinite number of analyses for a single string, a procedure constructing f-structure models from c-structures could never stop. I will not come back to processing details until chapter 6, but it is worthwhile noting at this point that the standard LFG formalism has such a built-in restriction motivated by processing considerations.[45]

(78) *Language generated by an LFG grammar*
The language $L(G)$ generated by an LFG grammar G is the set of c-structure/f-structure pairs $\langle T, \Phi \rangle$, such that
T is a tree generated by the context-free grammar in G—subject to offline parsability—and
Φ is the minimal model satisfying all f-descriptions that arise from instantiation of the f-annotation schemata in G, and satisfying the Completeness and Coherence condition.

The *string language* $\mathcal{L}(G)$ generated by an LFG grammar can easily be defined as the set of terminal strings derived from the c-structures in the language $L(G)$.[46]

It is straightforward to extend the formal model of c- and f-structure outlined in this section to include further levels of representation, such as a semantic structure projected from f-structure. If a graph structure similar to f-structure is assumed for this semantic structure, we would have further feature-logic descriptions annotated to the rules and lexicon entries.[47] A formally equivalent way of including such a semantic representation in the LFG structures would be to assume a special fea-

[45] Strictly speaking, an LFG analysis is not uniquely specified by a pair of a c-structure tree and an f-structure; several ϕ mappings may be possible. But rather than mentioning a particular ϕ as a third component every time we refer to an LFG analysis, let us assume that the labeling function for the tree categories A of T in a pair $\langle T, \Phi \rangle$ incorporates a reference to $\phi(A)$.

[46] Throughout this book I deviate from the standard terminology of formal language theory, in which my *string language* is simply the *language*. This is to avoid clumsiness when referring to the set of analyses generated by a formal grammar—which is done much more frequently than reference to the string language. From the context, the usage should generally be obvious. Furthermore I use a typographical distinction.

[47] The situation changes if a special resource-sensitive logic is assumed to underlie semantic construction as in the glue-language approach of Dalrymple and colleagues (see Dalrymple et al., Dalrymple et al. 1993, 1997, the contributions in Dalrymple (1999), and Dalrymple (2001) as a textbook). Then the formal treatment of semantic construction is no longer a special case of f-structure construction. In the present book I cannot go into this framework, since too little is known about generation from semantic structures—a crucial building block for an OT system.

ture SEMANTICS in the f-structures.[48] Given this close formal relationship between the syntactic level of f-structure and the level of semantic structure, it suffices for most formal and computational purposes relevant to the present book to just consider c-structure/f-structure pairs.

4.2 Optimality-Theoretic LFG—the overall architecture

In chapters 2 and 3, some general empirical and learning issues were discussed at a rather informal level. Defining an OT system in a formal framework such as LFG will allow us to state the issues and their consequences for the formal system in a more precise way. In particular, this will permit an investigation of computational procedures to model language processing within an OT system.

In this section, an abstract specification of the components of an OT system is given, based on the formal devices of LFG introduced in sec. 4.1. This will provide the context for a more detailed discussion of the character of the input, *Gen* and the violable constraints in the remainder of this chapter.

4.2.1 Abstract formal specification

Candidates The candidate analyses that OT-LFG deals with are tuples of structures as known from LFG, i.e., pairs of c-structure and f-structure (and as just mentioned possibly more levels of analysis) that are in a correspondence relation. All analyses satisfy certain basic inviolable principles, which we can assume to be encoded in an LFG grammar G_{inviol}; thus the set of all possible candidate analyses is defined as the structures generated by this grammar G_{inviol}. Sec. 4.3 contains a discussion of what principles are encoded in this grammar G_{inviol}.

(79) *Definition of possible candidates*
The set of possible candidates is defined as the language $L(G_{inviol})$ generated by a formal LFG-style grammar G_{inviol}.

There are some issues as to how closely the LFG-style grammars used in OT-LFG systems resemble standard formal LFG grammars (see the definition of the language generated by an LFG grammar (78)). In

[48] There are some technical issues I cannot go into here: in order to be able to express the sharing of information correctly and in a general way, either the overall feature geometry has to be changed, or a special restriction operator has to be assumed Wedekind and Kaplan (1993).

sec. 4.3, I discuss whether the Completeness and Coherence conditions apply. In sec. 4.5 and chapter 6 the issue is addressed whether the offline parsability condition should apply for the LFG-style grammars used in OT-LFG systems.

Index/Input With the candidate analyses being fully specified LFG analyses, an appropriate representation for the input in the OT sense is a partially specified representation of LFG analyses. This gives us the strictly non-derivational system I argued for in sec. 3.3.5.

For recasting Grimshaw's 1997 analysis within LFG (Bresnan, 2000, sec. 1.1) assumes as the input "a (possibly underspecified) feature structure representing some given morphosyntactic content independent of its form of expression". An example (that in English would have *I saw her* as its optimal realization) is given in (80).[49]

(80)
$$\begin{bmatrix} \text{PRED} & \text{`see}(x,y)\text{'} \\ \text{GF}_1 & \begin{bmatrix} \text{PRED} & \text{`PRO'} \\ \text{PERS} & 1 \\ \text{NUM} & \text{SG} \end{bmatrix}_x \\ \text{GF}_2 & \begin{bmatrix} \text{PRED} & \text{`PRO'} \\ \text{PERS} & 3 \\ \text{NUM} & \text{SG} \\ \text{GEND} & \text{FEM} \end{bmatrix}_y \\ \text{TNS} & \text{PAST} \end{bmatrix}$$

More generally, we may want to assume an input comprising a feature representation of the semantic content of an utterance (and potentially some further "pragmatic" clues, such as information structural status etc., cf. sec. 3.3.2). So, the input is defined as follows:

(81) *Definition of the* ***input***
The *input* is a member of the set of well-formed (partial) f-structures $\mathcal{F}$.

Note that contrary to the situation with derivational candidates assumed in some OT syntactic work (cf. the discussion in sec. 3.3.5), there is no issue at what point of a candidate derivation the input information is available. Both the candidates and the input are formal objects that we should think of as static (with information about all

[49]The status of the arguments x and y in the semantic form 'see(x, y)' will be discussed in sec. 4.3.2.

levels being available simultaneously). The relations between them are just mathematic relations between formal objects. Two kinds of input-candidate relations are relevant for a formal OT system: (i) *Gen* – involving a relation between the input and a set of candidates –, and (ii) the faithfulness constraints—involving a relation between an individual candidate and the input.

Candidate generation We can now give the following general definition of the function *Gen*, depending on the grammar G_{inviol}:

(82) *Definition of* ***Gen***
$Gen_{G_{inviol}}$ is a function from the set of f-structures to the power set of the analyses (c-structure/f-structure pairs $\langle T, \Phi \rangle$) in $L(G_{inviol})$:

$$Gen_{G_{inviol}}: \mathcal{F} \mapsto \wp(L(G_{inviol}))$$

In other words, $Gen_{G_{inviol}}$ takes each input f-structure Φ_{in} to a set of candidate analyses, which are contained in G_{inviol}:
$Gen_{G_{inviol}}(\Phi_{in}) \subseteq \{\langle T, \Phi' \rangle | \langle T, \Phi' \rangle \in L(G_{inviol})\}$. Further restrictions will be discussed below.

Constraint marking The OT constraints come into play when the alternative candidate analyses in the $Gen_{G_{inviol}}$ image of a given input f-structure are evaluated. The function *marks* assigns counts of constraint violations to each member of the candidate set. There are different ways in which the counts of constraint violations can be captured formally: in sec. 2.1, a multiset of constraint violation marks was assumed—for instance $\{*C^2, *C^2, *C^4\}$ for a candidate violating constraint C^2 twice and constraint C^4 once (compare the example (5) in sec. 2.1). An alternative, but equivalent way is to represent the number of violations that each constraint incurs as a natural number. I will adopt this representation in this definition.

The constraints $\mathcal{C}$ used in an OT system are given as a sequence $\langle C^1, C^2, \ldots, C^k \rangle$. So we can represent the violation counts for a particular candidate as a sequence of natural numbers: $\langle n^1, n^2, n^3 \ldots n^k \rangle : n^i \in \mathbb{N}_0$. So, assuming a constraint sequence $\langle C^1, C^2, C^3, C^4, C^5 \rangle$ for the above example, the violation counts would be $\langle 0, 2, 0, 1, 0 \rangle$.

An individual constraint $C^i \in \mathcal{C}$ is then defined as a function taking a pair of an input f-structure and an LFG analysis to a natural number. The function *marks* depends not only on the candidates, but on

the input structure too, for the following reason: we not only have markedness constraints (which are defined on the candidate structures alone) but also faithfulness constraints (which are defined relative to the input).[50])

(83) *Definition of* ***OT constraints***
$\mathcal{C}$: a sequence of constraints $\langle C^1, C^2, \ldots, C^k \rangle$;
for each constraint $C^i, i = 1..k$:

$$C^i(\Phi_{in}, \langle T, \Phi' \rangle) \in \mathbb{N}_0$$

The specification of markedness constraints based on LFG analyses is discussed in sec. 4.4. Faithfulness constraints are discussed in sec. 4.5.

For the function *marks*, which takes into account all constraints, we get:

(84) *Definition of* ***marks***
$marks_{\mathcal{C}}$ is a function from (input) f-structures and LFG analyses to a sequence of k natural numbers
(where $k = |\mathcal{C}|$, the size of the constraint set)

$$marks_{\mathcal{C}} : \mathcal{F} \times L(G_{inviol}) \mapsto \mathbb{N}_0^k, \text{ such that}$$

$$marks_{\mathcal{C}}(\Phi_{in}, \langle T, \Phi' \rangle) = \langle C^1(\Phi_{in}, \langle T, \Phi' \rangle), C^2(\Phi_{in}, \langle T, \Phi' \rangle), \ldots, C^k(\Phi_{in}, \langle T, \Phi' \rangle) \rangle$$

Harmony evaluation The key concept of optimization—harmony evaluation in the narrow sense—depends on the constraint violation counts for each input/candidate pair from a given candidate set $Gen_{G_{inviol}}(\Phi_{in})$, and on the language specific constraint ranking over the constraint set $\gg_{\mathcal{L}}$. The function *Eval* formalizing this concept determines the most harmonic candidate according to definition (3), repeated here.

(3) Candidate A_i is more harmonic than A_j ($A_i \succ A_j$) if it contains fewer violations for the highest-ranked constraint in which the marking of A^i and A^j differs.

[50] In sec. 4.5 the input-dependence of *marks* will be eliminated, since under the definition of $Gen_{G_{inviol}}$ adopted in sec. 4.3.1 the relevant faithfulness constraints can be defined on the candidate structures alone.

(85) *Definition of* ***Eval***
Given a set of LFG analyses Γ,
$Eval_{\langle \mathcal{C}, \gg_{\mathcal{L}} \rangle}(\Gamma) =$
$\{ \langle T_j, \Phi_j \rangle \in \Gamma \mid \langle T_j, \Phi_j \rangle$ is maximally harmonic for all analyses in Γ, under ranking $\gg_{\mathcal{L}}$ $\}$

Note that this definition is compatible both with a classical, strict constraint ranking and with a stochastic constraint ranking as assumed by Boersma (1998). As discussed briefly in sec. 3.3.1, the latter model is superior from the perspective of learnability, and it also constitutes an interesting mechanism for deriving optionality and frequency effects.

Language generated by an OT-LFG system

An OT-LFG system is specified by three components.

(86) *Definition of an OT-LFG system*
An OT-LFG system $\mathcal{O}$ is defined as $\langle G_{inviol}, \langle \mathcal{C}, \gg_{\mathcal{L}} \rangle \rangle$, where
G_{inviol} is a formal LFG-style grammar defining the possible candidate analyses,
$\mathcal{C}$ is a sequence of OT constraints, and
$\gg_{\mathcal{L}}$ is a ranking over $\mathcal{C}$.

Finally, the language generated by an OT-LFG system can be defined as the set of analyses $\langle T_j, \Phi_j \rangle$ for which there exists an input f-structure Φ_{in} such that $\langle T_j, \Phi_j \rangle$ is among the most harmonic candidates for that input (i.e., $Eval_{\langle \mathcal{C}, \gg_{\mathcal{L}} \rangle}(Gen_{G_{inviol}}(\Phi_{in}))$).

(87) *Definition of the language generated by an OT-LFG system* $\mathcal{O} = \langle G_{inviol}, \langle \mathcal{C}, \gg_{\mathcal{L}} \rangle \rangle$
$L(\mathcal{O}) = \{ \langle T_j, \Phi_j \rangle \in L(G_{inviol}) \mid \exists \Phi_{in} : \langle T_j, \Phi_j \rangle \in Eval_{\langle \mathcal{C}, \gg_{\mathcal{L}} \rangle}(Gen_{G_{inviol}}(\Phi_{in})) \}$

The string language $\mathcal{L}(\mathcal{O})$ generated by an OT-LFG system can be defined accordingly as the set of terminal strings for such analyses $\langle T_j, \Phi_j \rangle$.

An example of a linguistic OT-LFG system satisfying this definition will be developed in the subsequent sections (specifically in sec. 4.3 and 4.4) which will focus on the concrete specification of the components.

4.2.2 Degrees of freedom in this OT-LFG architecture

Based on the definitions just presented, there are three essential components specifying a particular OT-LFG system: the "base grammar"

G_{inviol}, the OT constraints $\mathcal{C}$, and the constraint ranking $\gg_{\mathcal{L}}$. The formalization leaves these components open to be filled out by empirical and conceptual linguistic work. Recall that by assumption of the linguistic OT approach, the former two are universal, and only the latter is language-specific. So, it is a goal for linguistic research to determine the specification of the former two (G_{inviol} and $\mathcal{C}$) in such a way that the language-specific ranking $\gg_{\mathcal{L}}$ is learnable based on language data.

Note however that in addition there are some further degrees of freedom in this OT-LFG architecture, on a more technical level:

(88) *Degrees of freedom in the definitions* (79)–(87)

a. The definition of possible candidates is based on the language generated by a "formal LFG-style grammar G_{inviol}"; this leaves open whether G_{inviol} is interpreted as an LFG grammar in the strict sense or whether the Completeness and Coherence condition is modified and/or the offline parsability condition is loosened up.

b. The exact specification of the function $Gen_{G_{inviol}}$ is left open.

c. It has not been fixed how the constraints are formally specified, given the input and a candidate structure.

For clarification of the last two points note that the definitions (82) and (83) given above specify only what kind of function $Gen_{G_{inviol}}$ is (mapping an f-structure to a set of candidates) and what kind of function a constraint is (mapping a candidate analysis to a natural number). This leaves completely open how the functions are specified. For example, the function $Gen_{G_{inviol}}$ may be a constant function that assigns the same candidate set to all possible input f-structures; or it may assign different candidate sets to different f-structures, based on some structural relationship. Likewise for the constraints, there is a wide spectrum of possibilities: the numbers they assign to candidate structures may be based on the counting of simple structural patterns, or they may involve complicated conditions (for instance, it is conceivable to formulate a constraint as a context-free grammar, assigning 0 when the candidate c-structure is included in the language and 1 otherwise).

All three aspects listed in (88) have to be pinned down to complete the formal specification of an OT-LFG system. The choices do not seem to have as clear an empirical impact as variations of the main components of the OT-LFG system have. But (i) they *do* have significant impact on processing complexity of the formal system (as I show in

sec. 4.2.3), and (ii) the basic assumptions of the OT approach discussed in chapter 2 and the empirical and conceptual observations of chapter 3 may be reflected more or less adequately in the formalization, depending on these choices.

As an example for point (ii) note that a system relying on LF-unfaithful candidates as discussed in sec. 3.3.3 is compatible with the definitions. There need not be any structural connection between the input/Index that $Gen_{G_{inviol}}$ takes as its arguments and the set of candidates it assigns to this input. In sec. 3.3.4, I argued that this circumstance is undesirable from the point of view of learnability.

4.2.3 Undecidability arguments for unrestricted OT systems

To see the impact of the choices in (88) on the processing complexity, let us look at two non-linguistic examples of OT systems both of which allow the construction of an undecidability argument (they can be seen as a variant of the sketch of an undecidability argument for unrestricted OT systems that Johnson (1998) presents).

The first construction creates a scheme of OT systems $\mathcal{O}_1$ for which an effective optimization procedure could only be devised in case another problem—the emptiness of the intersection of two context-free languages (89)—could be solved. However, since problem (89) is known to be undecidable, the optimization problem for $\mathcal{O}_1$ must be undecidable too.

(89) *The emptiness problem of the intersection of two context-free languages*
Given two arbitrary context-free grammars G_1 and G_2, is $L(G_1) \cap L(G_2) = \emptyset$?

Undecidability due to powerful constraints

(90) *A constructed OT-LFG system (schema)* $\mathcal{O}_1$
Let G_1 and G_2 be context-free grammars.
G_1 has the starting symbol S_1.
Specify the c-structure part of G_{inviol} by adding the following productions to G_1:
$S \rightarrow S_1$
$S \rightarrow$ *yes*,
where S is a new symbol, used as the start symbol of G_{inviol} and *yes* is a new terminal symbol. (The f-annotations in G_{inviol} are irrelevant. We may assume arbitrary f-structures.)

Define $Gen_{G_{inviol}}$ as follows:
$Gen_{G_{inviol}}(\Phi) = L(G_{inviol})$, for all Φ
Assume two constraints:

$$C^1(\Phi, \langle T, \Phi' \rangle) = \begin{array}{l} 0 \text{ if the terminal string of } T \text{ is in } \mathcal{L}(G_2) \cup \{yes\} \\ 1 \text{ otherwise} \end{array}$$

$$C^2(\Phi, \langle T, \Phi' \rangle) = \begin{array}{l} 1 \text{ if the terminal string of } T \text{ is } yes \\ 0 \text{ otherwise} \end{array}$$

Assume the ranking $C^1 \gg_{\mathcal{L}} C^2$.

Note that $Gen_{G_{inviol}}$ is a constant function, assigning the full set of possible candidates to any input; C^1 is a constraint based on the context-free language G_2.

The system works fine if we assume simple grammars for G_1 and G_2. For example, we may assume $\mathcal{L}(G_1) = \{a, aa\}$ and $\mathcal{L}(G_2) = \{b, c\}$. The possible candidate strings generated by G_{inviol} are then $\{a, aa, yes\}$. Now, let us check whether the string *yes* is in the string language generated by the OT-LFG system. According to the definition there has to be some input such that *yes* is the terminal string of the optimal analysis from the candidate set assigned to that input by $Gen_{G_{inviol}}$. Since all candidate sets are the same (and the input plays no role) this is easy to check, we need only look at one tableau:

(91)

Input: (arbitrary)	C^1 "$\in \mathcal{L}(G_2) \cup \{yes\}$?"	C^2 "*yes"
a. a	*!	
b. aa	*!	
c. ☞ *yes*		*

So, *yes* is indeed in the string language. In fact, it is the only string in that language. If we change G_2 from the previous example to make $\mathcal{L}(G_2) = \{a, b, c\}$, we get the following tableau:

(92)

Input: (arbitrary)	C^1 "$\in \mathcal{L}(G_2) \cup \{yes\}$?"	C^2 "*yes"
a. ☞ a		
b. aa	*!	
c. *yes*		*!

The string *yes* is no longer in the language generated (instead, all strings that are in the intersection of the two languages are, since they

all have the same constraint profile). The only way *yes* can ever win is that all other candidates fail to satisfy C^1. So what the OT-LFG system schema $\mathcal{O}_1$ in (90) effectively does is check whether the intersection of two context-free languages is empty. This problem is known to be undecidable in the general case; so the recognition problem for unrestricted OT-LFG systems (checking whether a certain string is in the string language) is also undecidable in the general case.

Since the formal system used for this argument contained an unnecessarily powerful constraint, it is worthwhile to investigate more restricted definitions.

Undecidability due to powerful candidate generation

This second type of undecidability argument was suggested to me by Jürgen Wedekind (p.c.). It exploits the fact that the emptiness problem for an LFG language is known to be undecidable[51] and uses a powerful G_{inviol} in candidate generation. This system can be constructed as follows:

(93) *Another constructed OT-LFG system (schema) $\mathcal{O}_2$*
Assume an LFG grammar G_1 with start symbol S_1, for which the problem $\mathcal{L}(G_1) = \emptyset$ is undecidable.
Construct from G_1 the grammar G_{inviol} with the new start symbol S, a new nonterminal symbol Y and the new terminal symbol *yes*. The following productions are added to the productions of G_1:

$$\begin{array}{ll} S \rightarrow & Y \\ & (\uparrow \text{PRED}) = \text{'yes'} \\ S \rightarrow & S_1 \\ & (\uparrow \text{CHECK}) = + \\ & (\uparrow \text{PRED}) = \text{'yes'} \\ Y \rightarrow & \textit{yes} \end{array}$$

Assume a single constraint:

$$C^1(\Phi, \langle T, \Phi' \rangle) = \begin{array}{l} 0 \text{ if } \Phi' \text{ has the feature } [\text{CHECK } +] \\ 1 \text{ otherwise} \end{array}$$

The candidate *yes* violates the constraint C^1 (since its f-structure does not contain the feature [CHECK +]). Analyses making use of the rule $S \rightarrow S_1$ satisfy C^1. But we can nevertheless get *yes* as the optimal

[51] According to the introduction to part V in (Dalrymple et al., 1995, 333), this was first shown in 1983 by Kelly Roach.

candidate for Φ_{in} =[PRED 'yes']: there may be no other candidates in the candidate set. This is the case iff no candidates can be derived with the S_1 symbol, i.e., when $\mathcal{L}(G_1) = \emptyset$. Since by assumption this problem is undecidable it is also undecidable whether *yes* $\in \mathcal{L}(\mathcal{O}_2)$.

What is unintuitive about this OT system is that there is no structural relationship between the f-structure constructed in the original LFG grammar G_1 (i.e., $\phi(S_1)$) and the f-structure projected from the new start symbol, i.e., $\phi(S)$. S is always mapped to the f-structure

$$\begin{bmatrix} \text{PRED} & \text{'yes'} \\ \text{CHECK} & + \end{bmatrix}$$

—no matter what $\phi(S_1)$ is like. But in a linguistic grammar, this ignored part of the structure reflects the candidate's interpretation. The intuition that all candidates should have the same underlying interpretation, modelled by the semantic part of f-structure, is not observed.

The two undecidability arguments motivate that both the expressive power of individual constraints and the expressive power of candidate generation have to be suitably restricted.[52]

4.2.4 Fixing the choices in the definitions

An objective of this book is to arrive at a formalization of OT syntax that is sufficiently restricted to permit realistic processing, while at the same time meeting the assumptions and intuitions underlying the theoretical OT approach. Further details of processing complexity will not be discussed until chapter 6; the strategy I adopt is to first fix the degrees of freedom mentioned in (88) in a way compatible with the empirical and conceptual criteria noted in the previous chapters and show afterwards that the resulting system is also adequate for processing.

Point (88a)—addressing the freedom of candidate generation based on G_{inviol}–is relevant for the treatment of unfaithful candidates. I will come back to it briefly in sec. 4.3 and sec. 4.5 and particularly in chapter 6. For the moment, the decision is not of crucial relevance.

The points (88b) and (88c)—the exact specification of $Gen_{G_{inviol}}$ and the individual constraints—appear to be antagonistic. If $Gen_{G_{inviol}}$ goes

[52] I will discuss a third undecidability argument, related to the second one presented here, in sec. 6.3.1: ensuring that the f-structure is considered for all candidates does not suffice to guarantee decidability of the recognition problem for OT-LFG systems (without offline parsability). F-structure must furthermore be related to the surface string, or possibly a context representation.

a long way in restricting the possible candidate structures to the observable language data, then the constraints need not be highly expressive, since they only have to control comparatively small choices. However, this goes against the methodological principle of the OT approach (25), discussed in sec. 3.2.1 and repeated here for convenience.

(25) *Methodological principle of OT*
Try to explain as much as possible as an effect of constraint interaction.

Prima facie this principle would suggest a maximally general and unrestricted $Gen_{G_{inviol}}$. But as we have seen in sec. 4.2.2 (and sec. 3.3.2), when applied in syntax, such a concept of $Gen_{G_{inviol}}$ leads to problems both with learnability considerations and with decidability.

For the standard conception of OT phonology the learnability and decidability problem with a weak $Gen_{G_{inviol}}$ do not pose themselves. Learnability is not negatively affected if all candidate sets are identical and just the input differs: the candidate analyses do not *contain* a "deep", interpreted level of representation that may differ from the input (like LF or the f-structure in syntax); rather the interface to the rest of the linguistic and cognitive system is concentrated in the input. So inferences from the utterance context permit direct conclusions about the input.[53]

The decidability problem does not arise either—if rational relations are assumed for modelling $Gen_{G_{inviol}}$ and regular expressions for the constraints, as is assumed in computational OT phonology. Following insights of Frank and Satta (1998), the entire OT system can be implemented as a single finite-state transducer implementing $Gen_{G_{inviol}}$ and the constraints composed by "lenient" composition (in the terminology of Karttunen (1998)).[54]

[53] As pointed out in footnote 29 on page 45, such inferences from utterance context may indeed be required for phonological OT systems, since surface forms *can* be ambiguous (cf. Hale and Reiss (1998) for the ambiguity problem posed by a simple strong bidirectional OT system not making use of information from the utterance context in the determination of the underlying form). But if the inferences work properly, they suffice to determine the underlying input. This is *not* guaranteed for unrestricted OT syntax systems, in which LF and the input may differ. (Note that contextual inferences can provide only information about the candidate's *actual* LF, and thus not necessarily about the input which may not be faithfully rendered at LF.)

[54] However for multiple violations of a single constraint a fixed upper bound has to be assumed.

However for OT syntax we need candidate structures which contain their own level of interpretation (related to the input in some way or another, let us assume here it is identical to the input). The set of meaning representations, e.g., predicate argument structures, which underlie syntax cannot be described with regular expressions, we need at least a context-free grammar. Of course, well-formedness of the input need not be checked by the OT system itself, but it has to be able to maintain the structure (for the fully faithful candidates). As a consequence, the overall OT system will be capable of generating a context-free language when fed with appropriate predicate-argument structures: if we rank faithfulness to the input structure highest, the strings generated will be based on a phrase structure isomorphic to the underlying meaning structure. In this sense the candidate generation component $Gen_{G_{inviol}}$ for syntax needs to be stronger than in phonology (in terms of automata theory, a string transducer is not sufficient, but we need a tree transducer; cf. e.g., Gécseg and Steinby (1997), and Wartena (2000) for a discussion of OT syntax). The individual constraints need not be very powerful since they can make reference to the "structural skeleton" provided by the input and maintained by $Gen_{G_{inviol}}$.

Within these limits, principle (25) tells us to choose the weakest possible $Gen_{G_{inviol}}$, so the main explanatory burden is on constraint *interaction*. (A criterion for deciding what should definitely follow as an effect of constraint interaction is crosslinguistic comparison, as was done for the syntactic MAX-IO and DEP-IO violations in sec. 3.2.3.)

The setup I propose uses (formally) very simple individual constraints (discussed in sec. 4.4) and a conception of $Gen_{G_{inviol}}$ that disallows divergences at the level of the interpreted part of the candidate structures, i.e., the part of f-structure that corresponds to the input (sec. 4.3). (Translated to the Chomskyan terminology, this covers both the predicate-argument structure at d-structure and the interpretation-relevant aspects of LF.)

Although this may sound like a massive restriction, this $Gen_{G_{inviol}}$ conception is compatible with a view that maintains maximal generality at the surface level of syntax, as will be discussed in sec. 4.5.[55]

[55] So the stronger $Gen_{G_{inviol}}$ restrictions in a comparison with the OT phonology model may be viewed as no more than a compensation for the different type of representations used in syntax. A totally different formalization of OT syntax might attempt to use only the surface form as candidates and use more expressive constraints, circumventing the decidability problem addressed in sec. 4.2.2 in a different way.

4.3 Candidate generation and the inviolable principles

4.3.1 The restricted definition of *Gen*$_{G_{inviol}}$

Based on the considerations of sec. 4.2.4, the candidate generation function *Gen*$_{G_{inviol}}$ is defined as follows:

(94) *Restricted definition of* ***Gen***

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

So only those of the possible candidate analyses in $L(G_{inviol})$ are picked whose f-structure is *subsumed* by the input f-structure, and which do not add any semantically relevant information. There is some leeway for the exact definition of *semantic information*. But note that in every concrete formulation of G_{inviol}, this can be specified by declaring particular features or feature combinations as contributors to semantic information.[56]

This formalization meets the intuition of Bresnan's 2000 original account of candidate generation in OT-LFG: at an abstract level, *Gen*$_{G_{inviol}}$ is indeed generation from the input with an LFG grammar. The candidate analyses can be viewed as being *generated* from the input structure by monotonically adding (non-semantic) information. Conveniently, the task of computing *Gen*$_{G_{inviol}}(\Phi_{in})$ is then exactly the classical task of generation from an underspecified f-structure, given an LFG grammar (G_{inviol}). Processing is discussed further in chapter 6.

4.3.2 Completeness and Coherence in OT syntax

There is an alternative way of stating the restricting condition in candidate generation, drawing a parallel to the standard LFG concepts of Completeness (74) and Coherence (75).[57] What we want to implement is that candidates may contain neither more nor less semantic information than specified in the input. We could assume a special place in the

[56] The qualification "where Φ' contains no more semantic information than Φ_{in}" could be expressed more formally as follows:

$$\Phi' \mid_{F_{sem}} \sqsubseteq \Phi_{in}$$

where $\Phi' \mid_{F_{sem}}$ is defined as the largest f-structure $\Phi'' \sqsubseteq \Phi'$, such that all features contained in its path are members of F_{sem} (the set of semantic features).

[57] Compare also Wedekind's 1995 terminology in his discussion of generation.

f-structure for encoding the input information, e.g., within the composite PRED values (compare (73)). The input (80) would then look roughly as follows (note that all input information has to be coded into the structures under PRED—here I use *ad hoc* feature names like REF-NUM for the semantic referent's number specification etc.):

(95) *Potential way of re-coding the OT input*

$$\begin{bmatrix} \text{PRED} & \begin{bmatrix} \text{FUNCTOR} & \text{see} \\ \text{TNS} & \text{PAST} \\ \text{ARGUMENT1} & \begin{bmatrix} \text{PRED} & \begin{bmatrix} \text{FUNCTOR} & \text{PRO} \\ \text{REF-PERS} & 1 \\ \text{REF-NUM} & \text{SG} \end{bmatrix} \end{bmatrix} \\ \text{ARGUMENT2} & \begin{bmatrix} \text{PRED} & \begin{bmatrix} \text{FUNCTOR} & \text{PRO} \\ \text{REF-PERS} & 3 \\ \text{REF-NUM} & \text{SG} \\ \text{REF-GEND} & \text{FEM} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

(If the example contained an adjunct, this would have been included under PRED too.) Now we can express restricted candidate generation through Completeness and Coherence conditions:

(96) *OT-Completeness*
All information specified as a predicate's input information (= under PRED) is also realized in this predicate's f-structure.

(97) *OT-Coherence*
Only the semantic information that is specified as a predicate's input information (= under PRED) is also realized in this predicate's f-structure.

This would permit us to move the subsumption restriction on $Gen_{G_{inviol}}$ into the formal specification of the underlying LFG-style grammar, i.e., into the definition of possible candidate structures in general (79). Candidates with an over- or under-informative f-structure would be incoherent or incomplete and thus no longer exist in G_{inviol}, so they would not have to be excluded in the definition of $Gen_{G_{inviol}}$.[58] However, I will keep to definition (94), assuming it to be more perspicuous. We may think of the OT-Completeness and OT-Coherence conditions to be (redundantly) at work within G_{inviol} nevertheless.

[58] We would get

(i) $Gen_{G_{inviol}}(\Phi_{in}) = \{ \langle T, \Phi' \rangle \in L(G_{inviol}) \mid \Phi_{in}$ is the input information of $\langle T, \Phi' \rangle \}$

It is interesting to note a change in the source of the reference information for checking Completeness/Coherence. In classical LFG, this information is contributed by the *lexical entries* of the predicates (i.e., the verbs, prepositions etc.). The PRED value with the subcategorization frame is conveniently written in one line, using angle brackets as in (↑PRED)='devour⟨(↑SUBJ) (↑OBJ)⟩' (compare (73)). In OT-LFG, the relevant reference information is contributed by the *input*. This means that if we nevertheless assume subcategorization frames in the lexicon entries then they are pretty much redundant. Only those matching the input will be usable (since they have to be unified with the input). The only interesting case arises when we look at pathological inputs like a strictly transitive verb such as *devour* used with just a single argument. Here, a lexicon lacking (↑PRED)='devour⟨(↑SUBJ)⟩' would lead to an empty candidate set, predicting that the input cannot be expressed in any language. It is questionable however whether such an approach would be in the spirit of OT. For instance, OT phonology will predict a surface realization for any underlying nonsense string of phonemes. The learner will just never come across such a string and thus it will not be entered into the inventory of underlying forms. Of course, the situation is slightly different in syntax, since we are not dealing with finite lists of underlying forms. However, we have to assume that the inputs are processed further in the conceptual cognitive system, and this further processing detects and excludes nonsense inputs. The syntactic system itself will be more perspicuous and general if it does not attempt to anticipate such conceptual restrictions.

In other words, it seems most adequate for the OT syntax setup to assume lexicon entries without a specification of the subcategorization frame, compatible with all underlying argument combinations. Lexical preferences may then be derived through constraint interaction. For example, the thought of someone eating very fast, leaving unexpressed what he eats, would turn out in English as *he was eating hastily* or something similar (note that the input would certainly contain no particular lexical items, like English *devour*, but a more abstract conceptual representation). The verb *devour* would not be used because it is suboptimal for expressing this thought. Such a constraint-based account is essential if we want to have a uniform learning theory for syntax and the lexicon. It also bears much more explanatory potential for argument frame variation in the lexicon. Of course, this type of account presupposes the assumption of rather fine-grained constraints which are sensitive to individual lexical items—an issue which I cannot pursue any further in this book.

In the representation of OT analyses, I will not use angle brackets which would suggest a subcategorization frame that is checked within the formal system of LFG. Rather I will use parantheses following the semantic form in the PRED value of input f-structures. For example, we would get [PRED 'devour(x, y)']. This notation is suggestive for the interpretation of dependents as semantic arguments, without the standard LFG mechanism of Completeness and Coherence being at work.

4.3.3 The base grammar G_{inviol}

In this subsection, an impression of the kinds of inviolable principles encoded in G_{inviol} is given. As an example, assume we want to encode Grimshaw's 1997 fragment of inversion data in English in OT-LFG, following the reconstruction by (Bresnan, 2000, sec. 2). The formal LFG-style grammar G_{inviol} will have to formalize a theory of extended projections. This can be done on the basis of LFG's extended head theory that (Bresnan, 2001b, ch. 7) discusses in detail. The principles Bresnan assumes can be fleshed out in a set of LFG rules, i.e., context-free rules[59] with f-annotations. Kuhn (1999b) contains a more detailed discussion of how extended head theory can be captured in concrete rule formulations; here it may suffice to assume that the effect of the principles can be envisaged as a set of classical LFG rules like the rules in (98),[60] generating X-bar-configurations with an extension to functional categories (like the verbal functional categories I and C). A lexical category and the corresponding functional categories on top of it form an *extended projection* Grimshaw (1991), Bresnan (2001b). (I do not actually assume a fixed limitation to a maximum of two functional categories, so G_{inviol} really generates arbitrarily many FP's within an extended projection.)

(98) *Some of the rules in G_{inviol}*

CP	$\longrightarrow$	(XP)	(C′)
		(↑DF)=↓	↑=↓
C′	$\longrightarrow$	(C)	(IP)
		↑=↓	↑=↓
IP	$\longrightarrow$	(XP)	(I′)
		(↑DF)=↓	↑=↓

[59] More precisely, a generalization of context-free rule notation which allows regular expressions on the right-hand side, cf. footnote 41 on page 60.

[60] DF is a generalization over discourse functions (TOPIC, FOCUS, Q-FOCUS and SUBJECT); CF generalizes over complement functions (OBJ, OBL$_\theta$, COMP etc.). The non-endocentric category S that Bresnan 2000, 2001b assumes is ignored here.

$$\begin{array}{lcll} \mathrm{I}' & \longrightarrow & \mathrm{(I)} & \mathrm{(VP)} \\ & & \uparrow{=}\downarrow & \uparrow{=}\downarrow \\ \mathrm{VP} & \longrightarrow & \mathrm{V}' & \\ & & \uparrow{=}\downarrow & \\ \mathrm{V}' & \longrightarrow & \mathrm{(V)} & \mathrm{(XP)} \\ & & \uparrow{=}\downarrow & (\uparrow\mathrm{CF}){=}\downarrow \end{array}$$

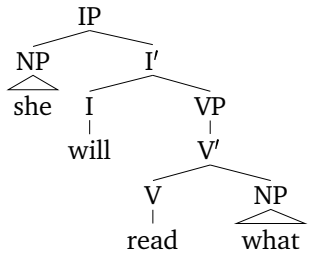
There are two crucial points to note about (98): first, the ↑=↓ annotation of both the C and the IP category in the C′ rule, and the I and the VP in the I′ rule. The functional head and its complement (in c-structure) act as "co-heads" on f-structure, i.e., their f-structures are identified. (All categories of an extended projection are thus mapped to the same f-structure.) Second, all categories are optional. These points together ensure that a given input f-structure has a wide range of realization alternatives—as required in an OT account with a weak $Gen_{G_{inviol}}$ component and strong effects of constraint interaction (sec. 4.2.4): since the f-structures of all heads within the extended projection are identified, each of them is a potential site for a category realizing information from the input f-structure. The structures generated for the underspecified input f-structure in (99) include the LFG analyses in (100) on page 83, for example.

(99) $$\begin{bmatrix} \mathrm{PRED} & \text{`read}(x,y)\text{'} \\ \mathrm{GF}_1 & \begin{bmatrix} \mathrm{PRED} & \text{`PRO'} \end{bmatrix}_x \\ \mathrm{GF}_2 & \begin{bmatrix} \mathrm{PRED} & \text{`PRO'} \\ \mathrm{OP} & \mathrm{Q} \end{bmatrix}_y \\ \mathrm{TNS} & \mathrm{FUT} \end{bmatrix}$$

The grammar specification given in (98) is not totally sufficient for deriving analyses (100b) and (100c). The f-annotation of the specifier XP in the CP rule will introduce the f-structure for *what* only under a discourse function, i.e. Q-FOCUS. The fact that the value of the Q-FOCUS feature and the grammatical function OBJ are structure-shared (or reentrant) follows from an additional f-annotation not shown in (98). There are different ways of achieving the effect. Bresnan (2001b) assumes an inside-out functional uncertainty introduced by a phonologically empty object NP (or DP) under V′. I assume here an outside-in functional-uncertainty equation as an additional f-annotation in the CP rule (Kaplan and Zaenen, 1989/95, 154), shown in (101) on page 83.[61]

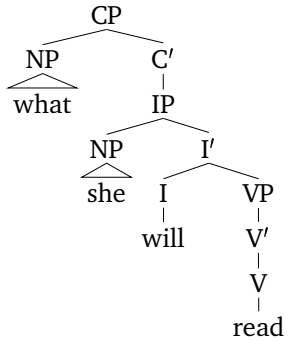
[61]The main reason for adopting the outside-in functional-uncertainty approach is presentational: the freedom of $Gen_{G_{inviol}}$, providing all kinds of candidate analyses is

(100) a.



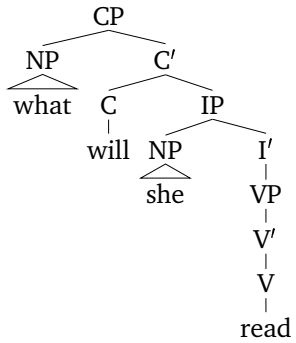
$$\begin{bmatrix} \text{PRED} & \text{`read}(x,y)\text{'} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{`PRO'} \end{bmatrix}_x \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{`PRO'} \\ \text{OP} & \text{Q} \end{bmatrix}_y \\ \text{TNS} & \text{FUT} \end{bmatrix}$$

b.



$$\begin{bmatrix} \text{PRED} & \text{`read}(x,y)\text{'} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{`PRO'} \end{bmatrix}_x \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{`PRO'} \\ \text{OP} & \text{Q} \end{bmatrix}_y \\ \text{Q-FOCUS} & [] \\ \text{TNS} & \text{FUT} \end{bmatrix}$$

c.



$$\begin{bmatrix} \text{PRED} & \text{`read}(x,y)\text{'} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{`PRO'} \end{bmatrix}_x \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{`PRO'} \\ \text{OP} & \text{Q} \end{bmatrix}_y \\ \text{Q-FOCUS} & [] \\ \text{TNS} & \text{FUT} \end{bmatrix}$$

brought out clearer with this variant (without having to go into great technical detail). In (Kuhn, 2001c, 335) I present essentially the same fragment, but based on the inside-out functional-uncertainty specification.

(101) CP $\longrightarrow$ (XP) (C′)
$(\uparrow \text{DF}) = \downarrow$ $\uparrow = \downarrow$
$(\uparrow \text{DF}) = (\uparrow \{\text{COMP} \mid \text{XCOMP}\}^* \ (\text{GF}-\text{COMP}))$
where the two instances of DF are the same (i.e., both TOPIC or both Q-FOCUS, etc.) and GF: the grammatical functions.

The regular expression in the feature path expression in (101) (note the operations disjunction "{ | }", Kleene star "*" and complementation "–") allows one to cover arbitrary long-distance dependencies between the topicalized argument and the verb introducing the f-structure predicate.

Many further aspects of the non-derivational framework of LFG play a role in its use as the base formalism of an OT system. I cannot go into the details here (see Bresnan (2001b)). Note however the differences between the representations in (100) and the corresponding ones in (6), sec. 2.1, which are set in the GB-based system used by Grimshaw (1997): In the c-structures in LFG, all elements are "base-generated" in their "final" position. No movements or chains are assumed; the kind of information captured by these concepts in GB-style approaches is available in LFG's f-structures and the correspondence mapping. The OT constraints can be straightforwardly specified by referring to (c- or f-)structural configurations (see sec. 4.4).

To close this section, it should be noted that although the resulting grammar is formally an LFG grammar, it is certainly unusual since it "overgenerates" vastly, producing all universally possible c-structure-f-structure pairings. This is due to the special role that this LFG grammar plays as part of an OT model: given the different definition of grammaticality, the set of analyses generated by the LFG grammar is not the set of *grammatical* analyses of a particular language (as classically assumed). Rather, it is the union over all possible candidate sets—for any input.

4.4 The violable constraints: markedness constraints

According to the formalization in sec. 4.2, a constraint is generally a function mapping an input/candidate pair to a natural number (83). OT distinguishes two major types of constraints: faithfulness and markedness constraints. Markedness constraints are checked on the candidate analysis alone, independent of the underlying input. Faithfulness constraints are violated when the surface form diverges from the underlying form (empirical examples from phonology and syntax

were discussed in sec. 3.2). For this reason, the formalization apparently has to rely on a comparison of the input and the candidate analysis. It turns out that with the conception of candidate generation introduced in sec. 4.3, it is redundant to check the candidate analysis against the input; so even for faithfulness constraints, a consideration of just the candidate analysis suffices. This point will be discussed in more detail in sec. 4.5. For the time being, I will put faithfulness constraints aside and concentrate on markedness constraints. The objective is to establish a concrete way of specifying a constraint as an instance of the following restricted definition:

(102) *Restricted definition of an OT constraint*
Each constraint $C^i \in \mathcal{C}$ is a function, such that

$$C^i(\langle T, \Phi \rangle) \in \mathbb{N}_0$$

In sec. 4.4.1, the formulations of markedness constraints used in OT-LFG are reviewed, to provide a basis for discussion of the precise formalization. Sec. 4.4.2 addresses the universally quantified character of constraints, coming to the conclusion that this should *not* be reflected in the individual constraints. Sec. 4.4.3 introduces the formalization of OT constraints as constraint schemata. Finally, in sec. 4.4.4 I point out that the standard LFG means of expression are sufficient to specify OT constraints in accordance with the assumptions behind OT.

4.4.1 Markedness constraints in OT-LFG

In most OT work the constraints are formulated in prose. However, it seems to be a central assumption that the constraints can be formalized as structural descriptions of the type of representations output by $Gen_{G_{inviol}}$—i.e., LFG structures in our case. As a set of sample constraints, let us again look at an adaptation of the constraints that (Bresnan, 2000, sec. 2) uses in her illustrative reconstruction of Grimshaw's 1997 fragment (parts of which I sketched in sec. 2.1). Due to their original illustrative purpose, these specific constraints do not necessarily play a role in original OT-LFG accounts (viewing OT-LFG as a theoretical paradigm, rather than just a formalism). But the means of expression used in these constraints can be regarded as representative.

The constraints involved in the derivation of *What will she read*, which was also used for illustration in sec. 2.1, involves the constraints in (103) to be discussed in the following. The tableau for the sample candidates from (100), based on these constraints, is anticipated in (104).

(103) a. OP-SPEC Bresnan (2000)
An operator must be the value of a DF [discourse function] in the f-structure.

b. OB-HD (Bresnan, 2000, (21))
Every projected category has a lexically filled [extended, JK] head.

c. DOM-HD (Bresnan's 2000 STAY (24))
Categories dominate their extended heads.

d. ARG-AS-CF
Arguments are in c-structure positions mapping to complement functions in f-structure.

(104)

Input: $\begin{bmatrix} \text{PRED} & \text{'read}(x,y)\text{'} \\ \text{GF}_1 & \begin{bmatrix} \text{PRED} & \text{'PRO'} \end{bmatrix}_x \\ \text{GF}_2 & \begin{bmatrix} \text{PRED} & \text{'PRO'} \\ \text{OP} & \text{Q} \end{bmatrix}_y \\ \text{TNS} & \text{FUT} \end{bmatrix}$	OP-SPEC	OB-HD	ARG-AS-CF	DOM-HD
a. [$_{\text{IP}}$ she will [$_{\text{VP}}$ read what]]	*!			
b. [$_{\text{CP}}$ what [$_{\text{IP}}$ she will [$_{\text{VP}}$ read]]]		*!	*	
c. ☞ [$_{\text{CP}}$ what will [$_{\text{IP}}$ she [$_{\text{VP}}$ read]]]			*	*

(103a) OP-SPEC and (103b) OB-HD are correspondents to Grimshaw's constraints with the same name (cf. (1)), now based on the LFG representations. Since in G_{inviol} (cf. (98)) the only way of introducing something under a discourse function is as the specifier of CP or IP (and since the specifier of VP is never filled), (103a) amounts to the same as (1a) "Syntactic operators must be in specifier position". (103b) relies on a slightly more complicated reconstruction on the placement of lexical and functional X^0 heads within extended projections. The concept of the *extended head* is defined to accomodate for the possibility of an I^0 or C^0 category to act as the head of categories further down in the same extended projection.[62]

[62] I made condition (ii) explicit to ensure that IP does not qualify as the extended head of C′ in the following configuration (cf. the discussion of (120b) below):

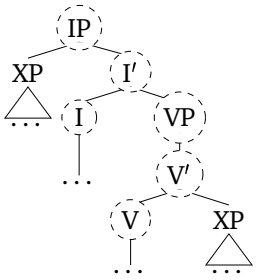
(i)


(105) *Definition of* ***extended head*** (Bresnan, 2001b, 132)
Given a c-structure containing nodes $\mathcal{N}, \mathcal{C}$ and c- to f-structure correspondence mapping ϕ, $\mathcal{N}$ is an **extended head** of $\mathcal{C}$ if
[(i)] $\mathcal{N}$ is the minimal node in $\phi^{-1}(\phi(\mathcal{C}))$ that c-commands $\mathcal{C}$ without dominating $\mathcal{C}$ [and
(ii) the X-bar level of $\mathcal{N}$ is less or equal to the X-bar level of $\mathcal{C}$, JK].

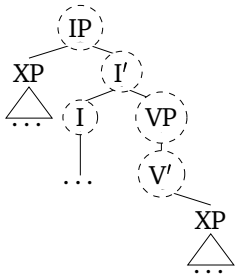
For our purposes, it suffices to note that there are two ways of satisfying this condition: either (i) the extended head is just the ordinary X-bar-categorial head, if present (i.e., X^0 or X′ for X′, and X′ or XP for XP); or (ii) if there is no X-bar-categorial head, an X^0 category in the same extended projection (and thus mapping to the same f-structure) becomes the extended head, if it is the lowest one c-commanding the category in question ($\mathcal{C}$). This is illustrated in the following examples.

In (106a) and (106b), the categories belonging to the extended projection are circled. In (106a), all projected categories have ordinary X-bar-categorial heads: I′ is the head of IP, I the head of I′, etc.[63] In (106b) however, the V category is c-structurally unrealized (recall that in the rules all categories are optional). So, V′ has no ordinary head; but the extended head definition applies, making I the extended head of V′: it is mapped to the same f-structure and c-commands V′, without dominating it (and being the only such node, it is also the minimal one).[64]

(106) a.



b.



Let us move on to constraints (103c) and (103d). For Grimshaw's STAY, Bresnan also introduces a purely representational formulation, given in (103) as (103c) DOM-HD. I use an additional constraint (103d) ARG-AS-CF, to differentiate between head mobility (covered

[63]The X^0 categories themselves do not have extended heads; but note that constraint (103b) OB-HD does not apply to them.

[64]Note that VP has an ordinary extended head again: V′.

by DOM-HD) and argument mobility (covered by ARG-AS-CF). (Note that the formalism would also permit collapsing the two.[65])

A further point to note about (103c) DOM-HD is that there are two ways one may interpret this constraint, depending on how the presuppositional phrase "their extended heads" is resolved. Under the strong interpretation, DOM-HD is violated even by categories that do not have an extended head at all, so only categories that have an extended head *and* dominate it satisfy the constraint. Under the weaker interpretation, categories without an extended head do not violate DOM-HD (they *do* violate OB-HD of course). As can be seen in the tableau (104) on page 86, the interpretation adopted here is the latter one: Candidate (104b) (=(100b) incurs no violation of DOM-HD, although the C′ category has no extended head, so it does not dominate one either. A precise formulation of the interpretation adopted would be the following:

(107) DOM-HD (revised formulation)
If a projected category has an extended head, it dominates the extended head.

Pinning down the formulation of constraints with means of a formal language, as will be done in the following, is a guarantee that ambiguities as just observed are excluded.

4.4.2 Universal quantification of constraints

To sum up the observations of the previous subsection, the violable markedness constraints in OT-LFG are formulated as conditions on the structural representations. Both f-structure and c-structure (and the relation between them) are referred to. For formalizing the primitive relations in the f-structural configurations triggering a constraint violation, a description language is already in place: the standard specification language used in LFG's annotations, as introduced in sec. 4.1. As regards c-structure, the configurations referred to in OT constraints may

[65] Bresnan's 2000 system works with just a single STAY constraint: (103c). The c-structural introduction of arguments in DF positions incurs a violation of this constraint, since the inside-out functional-uncertainty approach is used (cf. the discussion of rule (101) above), so there is a phonologically empty argument XP under V′, establishing identity (or structure sharing) of the f-structure under the DF and the complement function (typically OBJ):

(i) $[_{CP}\ [_{DP}$ what$]$ will $[_{IP}$ she read $[_{DP}\ \epsilon]]]$

go beyond the local tree accessible within a single context-free rule (the mother and the immediate daughters), but the extension is rather straightforward; it is discussed in sec. 4.4.4.

So, the problem with constraint formalization certainly does not lie in the primitive configurational relations. What is more of an issue is the overall logical structure of constraints. The constraints are not formulated with reference to a particular structure, they typically take the shape of universally quantified implications: whenever a structure satisfies the description A, it should also satisfy the description B (or if the constraint is specified negatively, no structure satisfying A should also satisfy B).

A natural reaction is to try and formulate constraints as universally quantified formulae in a feature logic, to range over the complete candidate analysis to be evaluated. This move would make precise the logical structure underlying the natural language formulations of the constraints. I will pursue this idea in the following. Anticipating the result, this approach will turn out to be unnatural for modelling multiple constraint violations.[66]

Universal quantification

To express universal quantification in the constraints, we need a language that permits universal quantification over the structural objects (f-structures and c-structure nodes), and that contains negation (to express implication). With a feature logic including general negation and universal quantification, we can thus express (103a) OP-SPEC as (108a).

Following B. Keller 1993, one could alternatively use a logic without the universal quantifier, but with general negation and unrestricted functional uncertainty: [67] (108b), which is closer to the standard LFG specification language.

(108) a. $\forall f.[\exists g.(f\ \text{OP}) = g \rightarrow \exists h.(h\ \text{DF}) = f]$
b. $\neg[(\uparrow\text{GF}^*) = f \wedge (f\ \text{OP}) \wedge \neg(\text{DF}\ f)]$

[66] This section follows the reasoning in (Kuhn, 2001c, sec. 3.2).

[67] Kaplan and Maxwell (1988) assumed a restricted interpretation of functional uncertainty, excluding cyclic interpretation in proving decidability of the satisfaction problem. However, Ron Kaplan (p.c., August 1999) points out that for functional uncertainty outside the scope of negation, the satisfaction problem is generally decidable (correlate of results of Blackburn and Spaan (1993)).

(108b) is expressed here as an f-annotation with ↑ referring to the root node of the grammar, f is a local metavariable for an f-structure, similar to the metavariables ↑ and ↓.

For the constraints on c-structure and the c-structure/f-structure correspondence, the language has to refer to c-structure nodes and tree-geometric relations as well. The options for doing this are discussed in more detail in sec. 4.4.4. For the current purpose, these general considerations may suffice, since the approach will be rejected based on problems with multiple constraint violations.[68]

Constraint marking

With the general form of a constraint being a function from analyses to the natural numbers, we have yet to specify in which way the feature logic formulae are to be applied. Since they are specified in a highly general form, it is conceivable to evaluate them as if attached to the root category of the grammar (for instance, (108b) could be technically used in this way). Of course, treating the constraints as ordinary feature logic descriptions to be satisfied by the correct analysis fails to capture violability. Or in other words, the "constraints" would be a part of $Gen_{G_{inviol}}$.

But there is a simple way of allowing candidates to violate the constraint formulae *once* per constraint: The OT constraints are disjoined with their negation, and a constraint violation mark is introduced in case their negation is satisfied. Assume constraint C^1 is specified by the feature logic formula ψ^1, then we can model its application as a violable constraint as

(109) $\psi^1 \vee (\neg\psi^1 \wedge \text{`}{*}C^1\text{'} \in (\uparrow\text{MARKS}))$

attached to the root node. From this MARKS set, the constraint marks can be simply read off: 1 if the constraint mark is in the set; 0 otherwise.[69]

[68]Note that decidability is not an issue with such a logic: Although the general satisfiability problem for feature logics with universal quantification and general negation is undecidable (B. Keller 1993, sec.4.4, (Blackburn and Spaan, 1993, sec. 5)), the use of this logic for constraint checking on given candidate analyses is unproblematic, since the expressions are not used constructively. (This was pointed out by Ron Kaplan (p.c.) and Maarten de Rijke (p.c.).) The task performed is not checking satisfiability, but model checking, which is easy: the given candidate structure has a finite number of nodes and f-structures, thus it can be checked for each of them whether it satisfies the constraints by instantiating the variables to the given elements.

[69]Note that the result will be an LFG grammar that models not only the function

Multiple constraint violations

What happens in a scheme based on the mechanism (109) when a candidate analysis contains several instances of the configuration excluded by the constraint? Obviously, the formula ψ^1 cannot be satisfied, so the other disjunct is picked, introducing a single violation mark. Of course, multiple violations of a given constraint up to a fixed upper bound could be simulated by formulating extra constraints that check for the presence of several instances of the violation in the candidate structure.[70] This may be acceptable when the domain of competition is locally confined to non-recursive structures, but it is unnatural for the fully recursive generative system of syntax.

If the possibility of multiple constraint violations is to be granted generally and without an upper bound, the mechanism checking for constraint satisfaction has to be modified. As the candidate structure is traversed, the constraint has to be checked over and over. Whenever the application of a constraint leads to inconsistency, a violation has to be counted, but the rest of the structure has to be checked for further violations of the same constraint.

Since the constraint checking has to traverse the candidate structure anyway, one may ask if there is still the need for formulating the constraint in a universally quantified way. Note that the original idea of this format was to ensure that the constraint ranges over the entire candidate structure (and not just the outermost f-structure, to give a concrete example). Moreover, is it clear at all for arbitrary constraints in such a highly expressive logic what constitutes a multiple violation? For simple implicational constraints with universal quantification over one structural element (an f-structure or a c-structure node) it is intuitively clear what it means to violate this constraint more than once; but it seems that expressions involving more than one universal are more problematic. Assume we wanted to work with the following constraint (110a):

Gen$_{G_{inviol}}$ (when used to generate from an input f-structure), but also the function *marks*. We may thus call the grammar $G_{inviol,marks}$. I will briefly come back to this idea in sec. 4.4.5.

[70] Karttunen (1998) proposes this for his computational model of OT phonology, which does not allow arbitrarily many violations of a constraint either. For certain constraints, multiple violability can however be modelled with finite-state means, as pointed out by Gerdemann and van Noord (2000) (see Jäger (2002b) for a generalization).

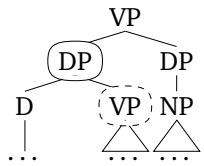
(110) *Hypothetical constraint*

a. For all DP categories, all their daughters are nominal (i.e., either N or D projections).

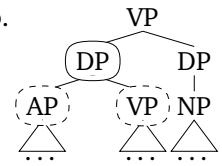
b. $\forall n.[\mathrm{DP}(n) \rightarrow \forall m.[dtr(m,n) \rightarrow nom(m)]]$

Now, the following structures are evaluated:

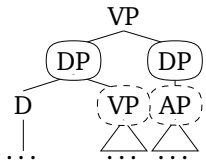
(111) a.



b.



c.



None of the three satisfies (110). But how many violations does each of them incur? In (111a) and (111b), one DP fails to satisfy the condition that all its daughters are nominal, while in (111c), both do. So, under one interpretation, (111a) and (111b) should violate (110) once, and (111c) twice.

On the other hand, (111a) is better than (111b), because only one of the DP's daughters violates the inner implication. Shouldn't one expect then that (111b) incurs two violations? In fact, the modified checking mechanism sketched above, which counts sources of inconsistency, would presumably have this effect (unless the mechanism is explicitly set back to the outermost universal whenever an inconsistency is encountered).

The problem is also clearly brought out if we look at the following two reformulations of (110):

(112) a. Each daughter of a DP category is nominal.

b. $\forall m.[\exists n.[dtr(m,n) \wedge \mathrm{DP}(n)] \rightarrow nom(m)]$

(113) a. For every category, if it has a non-nominal daughter, then it is not a DP.

b. $\forall n.[\exists m.[dtr(m,n) \ \wedge \ \neg nom(m)] \rightarrow \neg \mathrm{DP}(n)]$

Both are equivalent to (110) in terms of classical predicate logic, but read with the intuitions behind violable constraints, they clearly differ

in the number of violations ascribed to (111b): (111b) violates (112) twice, but it violates (113) only once. This indicates that the use of general formulae of this type of feature logic is inappropriate for modelling the intuitions behind OT constraints, for which we need a precise way of stating what it means to incur multiple violations of a given constraint.

4.4.3 Constraint schemata

The recursive applicability of constraints over the entire candidate structures has to be inherent to the general constraint marking mechanism in order to allow arbitrary multiple violations. Hence, the universal range does not have to be stated explicitly in each individual constraint. To the contrary, this makes the constraints counterintuitive, as was discussed in the previous section.

Thus, we should give the individual constraint formulations a simpler logical structure—this is also in line with the methodological principles discussed particularly in sec. 2.2.4 (cf. also Grimshaw (1998)). Now, the universal applicability is implicit to all constraints and will be made effective in the checking routine that the candidate structures have to undergo after they have been constructed. At every structural object (either a c-structure node *or* an f-structure), all constraints are applied. This application of the constraints to multiple objects is the only source for multiple violations—a single structural element can violate each constraint only once. At each application, the individual constraints are again interpreted classically.

In order for this to work, the structural object which is being checked with a given constraint has to be clearly identified. I will assume a metavariable $\star$ for this (reminiscent of the $*$ used in standard LFG f-annotations of categories to refer to the category itself, cf. page 61). (112) will for example take the following shape:

(114) $\exists n.[dtr(\star, n) \wedge \mathrm{DP}(n)] \rightarrow nom(\star)$

When the constraints are checked, the metavariable $\star$ will be instantiated to one structural element after the other. Thus, the constraints are actually specified as constraint schemata, generating classical constraints.

Note that we could now express (110b) in either of the following two ways, reaching the two different effects for the structures in (111) discussed above:

(115) a. $\mathrm{DP}(\star) \rightarrow \forall m.[dtr(m,\star) \rightarrow nom(m)]$
b. $\forall n.[\mathrm{DP}(n) \rightarrow (dtr(\star,n) \rightarrow nom(\star))]$

Expressing (115b) in the equivalent form (114) may actually be more intuitive (note that now, equivalences of classical logic apply again).

So, we can state the following restriction on constraint formulation, which allows for a simple concept of multiple violations that is compatible with a classical concept of satisfiability, and also meets the intuitions behind violable constraints in OT:

(116) *Restriction on the form of constraints*
Violable constraints are formulated with reference to a unique structural element, which is referred to by a metavariable ($\star$).

Scalar constraints

Note that it is compatible with this restriction to assume a "scalar" interpretation of alignment constraints like, e.g., HEAD LEFT (117) from Grimshaw (1997).[71] (Under a scalar interpretation, this constraint is violated twice if there are two intervening elements between a (single) head and the left periphery of its projection.)

(117) HEAD LEFT: (Grimshaw, 1997, 374)
The head is leftmost in its projection.

The metavariable-based formulation allows a clear distinction between the non-scalar and the scalar version of this constraint as shown in (118). (The function $proj(n)$ is assumed to denote the projection of the node n; the relation $dtr(n,m)$ holds if n is a daughter of m, the relation *precede* is obvious.)

(118) HEAD LEFT
non-scalar interpretation
$head(\star) \rightarrow \neg\exists n.[dtr(n, proj(\star)) \wedge precede(n,\star)]$
scalar interpretation
$cat(\star) \rightarrow \neg\exists n.[head(n) \wedge dtr(\star, proj(n)) \wedge precede(\star, n)]$

The first formulation is stated from the point of view of the head; since the instantiated schema is interpreted classically (i.e., incurring maximally one constraint violation for each structural element), a given

[71] In recent work, Sells (e.g., 1999, 2001) has proposed an antisymmetric constraint system for deriving the typologically attested space of c-structure configurations, using alignment constraints of this kind.

head can violate this constraint only once (even if there are several intervening nodes to the left of it). The second formulation is from the point of view of the intervening category; thus if there are several of them, the overall structure will incur several violations of this constraint.

Formalization of the example constraints

With the formal means of constraint schemata, we are in a position to give the example constraints from (103) a precise formulation. English paraphrases for the formal specifications are given below the formulae. It is assumed that all operators introduce a feature OP (question operators have OP-value Q, for example), non-operators do not introduce this feature. *f-str*(f) holds of f-structures, *cat*(n) of c-structure nodes.

(119) a. OP-SPEC Bresnan (2000)
An operator must be the value of a DF in the f-structure.

$$(\textit{f-str}(\star) \wedge \exists v.[(\star\ \text{OP}) = v]) \\ \rightarrow \exists f.[(f\ \text{DF}) = \star]$$

"If $\star$ is an f-structure bearing a feature OP (with some value), then there is some (other) f-structure f such that $\star$ is embedded in f under the feature DF."

b. OB-HD (Bresnan, 2000, (21))
Every projected category has a lexically filled [extended, JK] head.

$$(\textit{cat}(\star) \wedge (\textit{bar-level}(\star, 1) \vee \textit{bar-level}(\star, 2))) \rightarrow \exists n.[\textit{ext-hd}(n, \star)]$$

"If $\star$ is an X-bar or X-max category, then there is some node n which is the extended head of $\star$."

c. DOM-HD (revised formulation)[72]
If a projected category has an extended head, it dominates the extended head.

$$\forall n.[(\textit{cat}(\star) \wedge \textit{ext-hd}(n, \star)) \rightarrow \textit{dom}(\star, n)]$$

"For all nodes n such that category $\star$ is their extended head, n dominates $\star$."

[72]Note that the alternative, stronger interpretation of STAY discussed in connection with (107) can be easily expressed too:

$$(\textit{cat}(\star) \rightarrow (\exists n.[\textit{ext-hd}(n, \star) \wedge \textit{dom}(\star, n)]))$$

d. ARG-AS-CF
Arguments are in c-structure positions mapping to complement functions in f-structure.

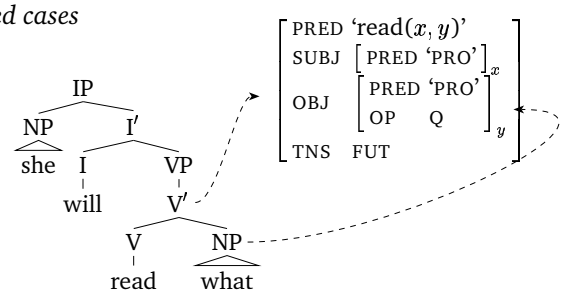
$$\exists f.[(f\ \text{CF}) = \star] \rightarrow \exists n.[cat(n) \wedge \phi(n) = \star \wedge \textit{lex-cat}(\mathcal{M}(n))]$$

"If $\star$ is embedded under a complement function CF (in some f-structure f), then there exists a c-structure node n projecting to $\star$, whose mother is a lexical category, i.e., $\star$ is c-structurally introduced as the complement of a lexical category (the canonical position for CF introduction, according to the mapping principles, (Bresnan, 2001b, sec. 6.2))."

When these constraints are applied to the sample candidate set of (100)/(104), we get the following constraint violations listed under the analyses:

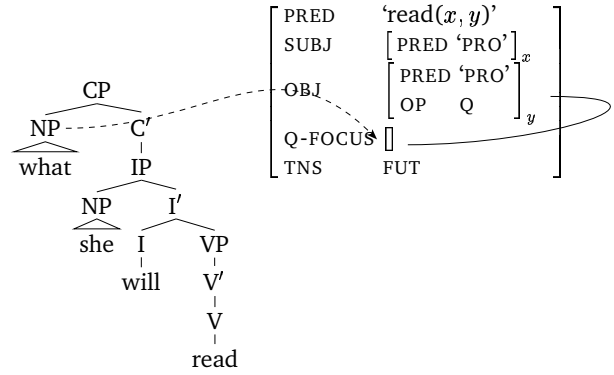
(120) *Candidate analyses for tableau* (104), *with ϕ mapping shown in selected cases*

a.



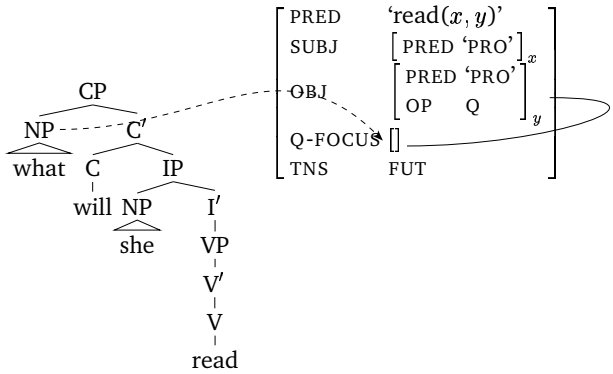
*OP-SPEC

b.



*OB-HD, *ARG-AS-CF

c.



*DOM-HD, *ARG-AS-CF

The constraint violations are derived as follows: In candidate (120a), we have an f-structure that satisfies the antecedent of the implication (119a), the OP-SPEC constraint: the f-structure under OBJ bears the feature OP Q. Let us call this f-structure $\star$ for the moment. To satisfy OP-SPEC, $\star$ has to be embedded in some f-structure under a feature DF. This is not the case in candidate (120a), thus we get a violation *OP-SPEC. Note that both of the two other candidates satisfy OP-SPEC, since there the OBJ value is identical with the value of Q-FOCUS (an instance of DF).

The f-structure $\star$ we were looking at in candidate (120a) *does* however satisfy the remaining three constraints: Constraints (119b) and (119c) are satisfied trivially, since our $\star$ is not a c-structure category so the antecedent is already false (making the implication true). However let us look at constraint (119d) ARG-AS-CF: with f instantiated to the outermost f-structure (the only possibility), the antecedent is satisfied: $\star$ is indeed embedded under a CF, namely OBJ. So does $\star$ also meet the consequent? There is only one category projecting to $\star$: the lower NP node, and fortunately its mother—V′—is a lexical category, as required. So we get no violation of ARG-AS-CF. All other c-structure and f-structure elements of candidate (120a) satisfy all four constraints, so *OP-SPEC is the only constraints violation we get.

Let us now look at candidate (120b). Since there is no C category in the tree and IP does not qualify as the extended head of C′ (cf. definition (105) and footnote 62 on page 86), we get a violation of (119b) OB-HD. Checking (119d) ARG-AS-CF, we can again instantiate f as the outermost f-structure, and $\star$ as the f-structure under OBJ. There is a single category node mapping to $\star$: the NP in the specifier to CP

(note that the f-structure under OBJ and Q-FOCUS is identical). In order to satisfy the constraint, the mother node of this NP—CP—would have to be a lexical category. This is not the case, so we get *ARG-AS-CF. The other constraints are fully satisfied.

Candidate (120c) shares with (120b) the configuration violating ARG-AS-CF. If we look at C′ however, we note that (119b) OB-HD is satisfied here: since C is filled we do find an extended head. Note that I′ too has an extended head, although there is no X-bar-categorial head I; C is in the appropriate c-commanding position and has the right bar-level. However, I′ violates (119c) DOM-HD: since it has an extended head, it would also have to dominate this head for satisfying DOM-HD.

With the constraint ranking OP-SPEC, OB-HD ≫ ARG-AS-CF, DOM-HD for English, we get (120c) as the most harmonic of these three (and in fact all possible) candidates, as shown in tableau (104) already.

Constraint marking

Based on the constraint schemata proposed in this section, the marking of constraint violations incurred by a given candidate structure is conceptually straightforward. For each structural element (c-structure node and f-structure), the set of constraints is applied, with the metavariable instantiated to the respective element. When the application of a constraint fails, the candidate is not rejected, but the count of constraint violations is increased.

So, the constraint marking function from candidate analyses to natural numbers can be given a precise specification, based on the cardinality of the set of structural elements for which the instantiated constraint schema is not satisfied by the candidate analysis:[73]

(121) *Schema-based definition of an OT constraint*

$$C^i(\langle T, \Phi\rangle) = \quad |\{x \mid \quad x \text{ is a structural element of } T, \Phi \text{ and } \langle T, \Phi\rangle \not\models \psi^i[\star/x] \quad \}|,$$

where $\psi^i[\star/x]$ is a constraint schema ψ^i formalizing C^i, with x instantiating the metavariable $\star$.

4.4.4 Constraint schemata as standard LFG descriptions

In this section, I adjust the formulation of OT constraints further, based on the observation that it can be brought closer to the formulation of

[73] The adoption of a set-theoretical view was suggested by Ron Kaplan (p.c.) and Dick Crouch (p.c.).

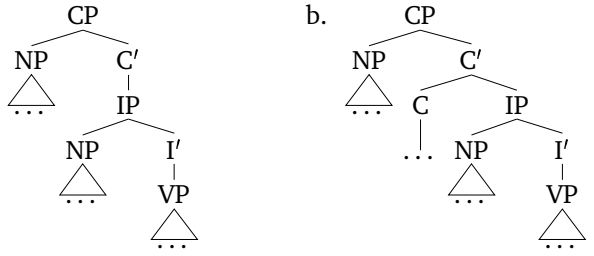
familiar, classical constraints in the LFG framework. The expressiveness is slightly reduced by this move, which should be considered an improvement, since the linguistically relevant constraints for OT syntax can still be expressed.

OT constraints of c-structure

The constraint formalization of the previous section (and Kuhn (2001c)) makes use of intuitively named predicates over categories and tree configurations (e.g., *ext-hd*, *dom*, *lex-cat*, etc.). A precise definition for these predicates was not given since the focus of presentation was on the question how to formalize the idea of multiply violable constraints. Nevertheless we need to find a more explicit account.

A slight difficulty in this task is that the intuitive difference between classical LFG constraints and typical OT constraints seems larger for constraints on c-structure than for constraints on f-structure. In the latter case, propositions about certain graph configurations have classically been expressed as Boolean combinations of various types of equations. This can easily be transferred to violable OT constraints, which add certain implications etc. For constraints on c-structure, the classical means of expression in LFG have been context-free rewrite rules (extended to allow regular operations such as optionality, union/disjunction, and Kleene closure). Here, it is not so obvious how the typical implications of OT constraints should be added. For instance, at what level should the difference between the I′ nodes in the following tree structures (one satisfying, one violating OB-HD) be stated?

(122) a. b.



There are various possible formats for stating such c-structure constraints. Underlying the previous section was a tendency to move away from the standard means of grammar specification in LFG, suggesting the introduction of a totally new tree specification language. So, on the one hand the trees in (122) are constructed according to the inviolable

constraints in $Gen_{G_{inviol}}$, which are formulated in the form of (extended) context-free rewrite rules. But for the violable OT constraints, a new, more general tree description language has been assumed.

For an analysis of further computational properties of the account, this move is problematic, since it would make the transfer of formal results from classical LFG to our situation unnecessarily difficult. It is also possible to state the constraints in a more familiar format.

We have still at least two choices: one could follow the proposal of Andrews and Manning (1999), who modify the classical projection architecture of LFG to encode categorial structure as attribute-value matrices (inspired by HPSG[74]). In such a framework, implicational OT constraints could be expressed as combinations of feature equations describing these attribute-value matrices.

The other alternative, which I will follow here, does not involve importing c-structure information into the feature structure representation. The additional constraints on c-structure are expressed as regular expressions over sequences of categories in local subtree configurations.[75] For instance, a constraint could demand that in a local subtree dominated by I′, an I daughter is present: (123).[76] The general format of constraints is implicational $A \Rightarrow B$, to be read as "if a local configuration satisfies description A, then it also has to satisfy description B in order to satsify the constraint".

(123) *Illustrative constraint:* I′-HAS-I-HEAD

$\star$=I′ $\Rightarrow$ $\star$?* I ?*

With just the ordinary set of c-structure symbols available, this account has a very limited expressiveness. The different status of (122a) and (122b) could not be detected, since at the local subtree level, the two representations are indistinguishable.

However, we can extend the set of category symbols to allow for more fine-grained distinctions to be made. The standard convention

[74]Head-Driven Phrase Structure Grammar, Pollard and Sag (1994)

[75]I use the term local subtree to refer to subgraphs of trees with depth 1.

[76]As in the XLE system's rule notation, the '?' is assumed to denote arbitrary category symbols; the stars ('*') are Kleene stars. So, the regular expression on the lower side of the local tree in (123) is matched by all category strings containing at least one occurrence of I.

is to encode the lexical class (N, V, A, P), the bar level and the functional/lexical distinction in the category symbols. Some of this information would be recoverable from the tree structure; for example, if the lexical class was only encoded in the X^0 categories and not in their projections, one could still detect it by tracing down the X-bar projection line.[77] With the explicit encoding of lexical class in higher projections, this information is made available up to the XP level, where it is required to express constraints on possible combinations of maximal categories with other material. So, the definition of the X-bar categorical head is effectively precomputed: the rewrite rules are specified in a way that ensures that the lexical class of a projection is the same as the one of its projecting X^0 head.

Now, the same idea of precomputing relevant information from the c-structure configuration and providing it as part of a category name can be applied in other situations. For instance we could distinguish between IPccxhy and IPccxhn (for *c-commanding extended head yes/no*), based on rules like the following:

(124)

C′	→	C	IPccxhy
C′	→		IPccxhn
IPccxhy	→	(XP)	I′ccxhy
IPccxhn	→	(XP)	I′ccxhn
I′ccxhn	→	(I)	(VP)
I′ccxhy	→	(I)	(VP)

The symbol IPccxhn is introduced in the second of the two C′ productions, i.e., when a c-commanding extended head (i.e., the C category) is missing: this information is propagated down to the I′ level. (Of course, the same construction would have to be introduced for the VP, but this is omitted here for simplicity.) We can now formulate the exact conditions for the OB-HD constraint at this level:

(125) *Partial specification of constraint:* OB-HD

$\star$=I′ccxhn ⇒ $\star$?* I ?*

Without the c-commanding extended head being present, the I′ has to

[77] This reasoning is based on a more classical set-up, where the X^0 heads cannot be missing; to extend it to the current LFG set-up where all categories including heads are optional, one might assume that projections with missing X^0 heads are underspecified for lexical class.

have a local I head in order to satisfy OB-HD; when a c-commanding extended head is present, there are no restrictions on the local subtree.

With the described technique of threading the relevant information through the c-structure skeleton, all distinctions with a finite set of relevant choices can be "imported" to the scope of a local subtree.

Explicit generalizations over c-structure

A possible objection to the proposed scheme of formulating c-structural constraints is that generalizations are not expressed explicitly in the $Gen_{G_{invviol}}$ rules and in the OT constraints. Instead, a massively disjunctive specification seems to be required for non-trivial systems.

However, there are several ways of ensuring that at the level of grammar specification, the underlying generalizations are indeed explicit. In (Kuhn, 1999b, 4.1), I distinguish two strategies: the representation-based vs. the description-based formulation of c-structural generalizations. In both cases, the idea is that entire classes of categories can be constrained with a single rule or principle. The representation-based strategy modifies the representation of categories, no longer viewing them as atomic symbols but assuming an internal structure (thereby making explicit a conceptual classification that is assumed anyway). The IP category, for instance, can be seen as a combination of category-level "features" for lexical class V, bar-level 2 (or *maximal*), and category type *functional*, which we may write as ⟨V, 2, *func*⟩. It is obvious how further distinctions can be added to this scheme (to give us ⟨V, 2, *func*, ccxhy⟩, for instance, to signal the existence of a c-commanding extended head). As long as all features have a finite range of values and there is a finite number of features, the fundamental LFG set-up remains unchanged.[78] But now, generalizations over rules can be expressed more explicitly by constraining the co-occurrence of category features within rules. In particular, OT constraints can be formulated that express implications based on particular category features, leaving the rest underspecified. A more general specification of (125) would be

(126) *Generalized (partial) specification of constraint:* OB-HD
$\star = \langle \alpha, 1 \vee 2, \beta, \text{ccxhn} \rangle \Rightarrow$ $\star$?* $\langle \alpha, 0, \beta, \_ \rangle$?*

[78] The XLE implementation provides the discussed concept of complex categories, constraints about which are formulated in *parametrized rules*. For the category-level features a positional encoding is used; the format for complex categories is `XP[a,b]`.

Note that as a prerequisite for this technique of constraint specification, principles about the inheritance of all relevant category-level features in the G_{inviol} have to be explicitly stated. It is beyond the scope of the present discussion to go into details of such a specification. I will assume that G_{inviol} is specified in such a way that (i) the c-structure in all candidate analyses is in accordance with the intuitive definition of the properties encoded by the category-level features, and (ii) all combinations of categories conforming with the definitions are effectively generated. The G_{inviol} grammar specification is simply taken as the formal definition of all category-level features involved, so there is no danger of generating any incorrect candidates (in a technical sense).

The other strategy for expressing generalizations about c-structure (the description-based formulation) keeps up atomic category symbols, but provides a richer language for describing categories: the category IP is in the denotation of meta-category names like *V-cat*, *Xmax-cat*, and *Func-cat*, which can be intersected as appropriate. This strategy can be directly integrated into the classical regular-language specification of LFG rules. However, it does not allow one to generalize over entire classes of meta-categories, so I keep to the representation-based strategy assuming complex categories.

Excluding universal quantification from the constraint schemata

Having simplified the formulation of c-structural OT constraints, we should now take another look at universal quantification (which is not included in the means of expression of standard LFG). According to sec. 4.4.3 (and Kuhn (2001c)) universal quantification is still allowed in the constraint schemata, although it is no longer used to account for multiple constraint violations (cf. the discussion of sec. 4.4.2). In the formulation of sample constraints in sec. 4.4.3, we do find a use of the universal quantifier, in the DOM-HD constraint (119c), repeated below.

(119) c. DOM-HD (revised formulation)

If a projected category has an extended head, it dominates the extended head.

$$\forall n.[(\mathit{cat}(\star) \wedge \mathit{ext\text{-}hd}(n,\star)) \rightarrow \mathit{dom}(\star,n)]$$

"For all nodes n such that category $\star$ is their extended head, n dominates $\star$."

Clearly, this universal quantification has been made obsolete by the move to the more canonical formulation of c-structural OT constraints,

according to which the property of having an extended head is now encoded in the c-structure categories. But even without this move, the use in (119c) would not justify the need for full universal quantification: according to the definition of extended head (see (105), adapted from (Bresnan, 2001b, 132), and note the minimality condition), the relation $\textit{ext-hd}(A, B)$ could be rewritten in a functional notation, with $\textit{ext-hd}'(B) = A$. So, (119c) could be reformulated as follows:[79]

(127) $\textit{cat}(\star) \rightarrow \textit{dom}(\star, \textit{ext-hd}'(\star))$

Further uses of universal quantification occur in the two versions of the hypothetical constraint first introduced in (110) and discussed in sec. 4.4.3. Their schema-based specification is repeated below.

(115) a. $\mathrm{DP}(\star) \rightarrow \forall m.[\textit{dtr}(m, \star) \rightarrow \textit{nom}(m)]$
b. $\forall n.[\mathrm{DP}(n) \rightarrow (\textit{dtr}(\star, n) \rightarrow \textit{nom}(\star))]$

Concerning (115b), we can again exploit the functional character of the daughter relation: $\textit{dtr}(A, B)$ iff $\mathcal{M}(A) = B$, which gives us

(128) $\mathrm{DP}(\mathcal{M}(\star)) \rightarrow \textit{nom}(\star)$

In (115a), we do have a "real" example of universal quantification. The intuition is that in order to satisfy this constraint, *all* daughters of category $\star$ have to be nominal. But since the property checked for in the constraint – having nominal daughters exclusively—is locally confined, it would be possible to introduce a complex-category-level distinction on mother categories encoding whether or not they have this property. Then the constraint would only need to check for the relevant category feature. Alternatively, a binary-branching re-coding of the c-structure could be assumed and the property *nom* could be checked with a simple constraint schema at each level, so the universal-quantification effect would be shifted to the universal application of constraint schemata.

I take these examples to indicate that after the move to constraint schemata, which are evaluated at every structural object, universal quantification is no longer required for linguistically interesting constraints. In fact it goes against the general principle of keeping the individual constraints simple and exploiting constraint *interaction* for higher-level effects.

[79] $\textit{ext-hd}'$ is a partial function—some categories have no extended heads. One might define a proposition containing an undefined functional expression to be false, then the consequent of (127) would become false, and the entire constraint would become true/satisfied for a category which has no extended head.

This means in particular that for the f-structural OT constraints we can restrict the format available for the formulation of constraint schemata to the format of standard LFG f-annotations (with the metavariable $\star$ added). Note that this excludes constraints like the following, universally quantifying over all possible feature paths in an f-structure:

(129) $\forall \text{PATH}, f.[(\star\ \text{PATH}) = f \rightarrow (f\,\text{CHECK}) = +]$

This constraint checks an entire f-structure with all substructures for a particular property ([CHECK +]). Note that when this schema is instantiated with different partial f-structures, we get a funny behaviour: a violation originating from a certain embedded f-structure g lacking the specification [CHECK +] is counted over and over again; so, we get multiple violations of the constraint, but the number of violations does not depend on the number of offensive partial f-structures, but rather on the size of the overall f-structure. A more reasonable behaviour results if we use the metavariable for the local f-structure that is demanded to have the specification [CHECK +]. In other words, we do away with universal quantification within the constraint schemata.

So, concluding this section, the constraints in the OT-LFG setting can be restricted to (i) conditions on local subtrees in c-structure (making use of complex category specification for generality), and (ii) functional annotations containing the metavariable $\star$. In addition, we allow for a combination of c-structural and f-structural restrictions, so OT constraints can express conditions on the structure-function mapping. This restricted format will be most relevant for decidability considerations discussed in sec. 6.2. For other discussions in the following, I will keep up the more perspicuous constraint schema format using intuitive predicates and universal quantification.

4.4.5 Digression: Constraint marking as description by analysis vs. codescription

Given the schema-based definition of constraints, the overall constraint violation profile of a given candidate results when the counts for the individual constraints are taken together. The way candidate set generation and constraint marking have been defined suggests a clear conceptual split between the two abstract processes: Candidate generation is prior to constraint marking, and one may want to think of the latter starting only once the former has finished. However, with the restriction to constraint schemata referring to single structural elements, an

alternative way of conceiving of constraint marking is opened up: we may envisage candidate generation and constraint marking as a combined abstract process. This alternative view has certain advantages, for instance in practical grammar writing, where the constraint marking can be coded into the grammar rules specifying the inviolable principles. As long as the effect of the constraint schemata is spelled out for every single element in the grammar, there is no difference in the resulting OT system, however coding the marking into the grammar makes it possible to be selective (which has certain advantages, but bears some risks too). In this section, I will make a few remarks about the two views.

There is a parallel to different approaches of realizing syntax-driven semantic analysis in an LFG grammar: the *description by analysis* approach vs. the *codescription* approach. In the former, semantic construction starts only once the f-structure analysis has been created; in the latter, f-structure and semantic structure are built up simultaneously (see e.g. Kaplan (1995) for discussion). The constraint marking approach working with two separate abstract processes works as description by analysis. The constraint violation profile of a candidate (corresponding to semantic structure) is built by traversing the previously constructed syntactic analysis.

But as long as there is a unique place in the grammar/lexicon from which a certain structural element can be introduced, we can attach the constraints it has to meet in the rule or lexicon entry already. This becomes possible since the constraints are formally restricted to refer to a single structural element. What we have got now is a codescription approach. For c-structure categories, it is true that there is such a unique place: we can attach the constraints in the rules for nonterminals and in the lexicon entries for terminals. With f-structures, we have to be careful, since due to unification, there is not generally a unique source in the grammar/lexicon for an f-structure. For all PRED-bearing f-structures, the instantiated symbol interpretation of semantic forms (cf. page 64) guarantees uniqueness however. Hence, attaching the constraints wherever a PRED-value is introduced captures this subset of f-structures, and a generalization to all f-structures is possible.[80]

[80]The generalization would work as follows: even for f-structures without a PRED-value, a feature taking an instantiated symbol as its value is assumed. Let us call it ID. The Completeness condition is extended to demand that in a well-formed f-structure, all substructures contain this feature ID. The value for ID is introduced in the constraint marking schemata for f-structures, which are *optionally* applied in all rules and all lexicon entries, whenever there is a reference to an f-structure. Since the value is an instantiated

This means that a general conversion from description by analysis to codescription—and vice versa—is possible.

As in the grammar-based constraint marking discussed briefly in sec. 4.4.2 (cf. footnote 69 on page 91), we have to ensure the violability of the attached OT constraints. We can again disjoin the constraints with their negation—this time not just for the root category, but potentially for every category in each rule. As a result we will again have a grammar that performs both the task of candidate generation and constraint marking; thus we might call it $G_{inviol,marks}$.

Constraint violation marks are introduced to the MARKS multiset in the places in the grammar where the structure violating a constraint is created. For example, we may want to formulate the constraint (119c) DOM-HD by making the following annotation in each of the grammar rules (cf. (98)):

$$\text{(130)} \quad \text{C}' \longrightarrow \left\{ \begin{array}{c} \text{C} \\ \uparrow = \downarrow \end{array} \middle| \begin{array}{c} \epsilon \\ \text{*DOM-HD} \in (\uparrow\text{MARKS}) \end{array} \right\} \text{(IP)}$$

In the original rule, the head C was simply optional "(C)", now we have an explicit disjunction of C and the empty string ϵ. At c-structure, this is equivalent (ϵ is not a phonologically empty category, it is indeed the empty string in the formal language sense). However, at f-structure, we can now annotate the option not realizing C with the introduction of the constraint violation mark *DOM-HD.

Since constraint marks can now be introduced into the MARKS multiset in every rule, we have to ensure that all contributions are collected and made available at the root node of the analysis.[81] This can be achieved by identifying the MARKS feature of all daughter constituents with the mother's by the equation (↑MARKS)=(↓MARKS), creating a single multiset for the complete analysis. Note that multiple violations of a single constraint fall out from the use of a multiset.[82]

---

symbol, only one application of the schemata can be performed on each f-structure. When there is unification, i.e., other rules or lexicon entries referring to the same f-structure, the optionality of application will ensure that they are not applied again. But Completeness ensures that they are applied once. Of course there is a non-determinism leaving open which rule/lexicon entry is the one setting the ID value. But this does not affect the result.

[81] As Ron Kaplan (p.c.) pointed out, the collection of marks need not be realized within the grammar of course, since evaluation is a grammar-external process anyway; thus, the identification of the MARKS feature is an unnecessary overhead.

[82] An alternative way using a standard set would be to interpret the constraint marks

If we now use a special projection o instead of the feature MARKS (so the membership constraint from (130) reads as "*DOM-HD $\in o$") we are very close the system of Frank et al. (2001), which is built into the Xerox Linguistic Environment (XLE) LFG parsing/generation system. The projection o is always a multiset of constraint violation marks and we can assume implicit trivial equations ($\uparrow = \downarrow$) in the rules, so the o-structure of all constituents is identified. The XLE system also provides an (extended) implementation of the *Eval*-function, based on the marks introduced to the o-projection, and a dominance hierarchy specified in the configuration section of the grammar.[83]

One advantage of the grammar-based or codescription approach is that often the constraint formulation becomes simpler: the triggering configurations for constraint satisfaction/violation do not have to be restated when they have an obvious location in the grammar specification. A good example is constraint (119d) ARG-AS-CF:

(131) ARG-AS-CF
Arguments are in c-structure positions mapping to complement functions in f-structure.
Description-by-analysis formulation

$$\exists f.[(f\ \text{CF}) = \star] \rightarrow \exists n.[cat(n) \wedge \phi(n) = \star \wedge \textit{lex-cat}(\mathcal{M}(n))]$$

Codescription formulation (just adding the violation mark introduction to rule (101))

$$\begin{array}{lccc} \text{CP} \longrightarrow & (\text{XP}) & & (\text{C}') \\ & (\uparrow\text{DF}) = \downarrow & & \uparrow = \downarrow \\ & (\uparrow\text{DF}) = (\uparrow\{\text{COMP} \mid \text{XCOMP}\}^* \ (\text{GF–COMP})) & & \\ & {}^*\text{ARG-AS-CF} \in o & & \end{array}$$

The description-by-analysis approach with the strict separation of candidate generation and constraint marking makes it necessary to reanalyze the c-structure configuration in which complements of lexical categories are mapped to the embedded CF. In the codescription account, a constraint violation mark can be introduced when an argument is c-structurally introduced in the non-canonical position, making use of functional uncertainty.[84]

---

introduced as instantiated symbols (like the PRED values in standard LFG), i.e., as pairwise distinct.

[83] As Frank et al. (2001) discuss in detail, XLE distinguishes several types of constraint marks—in particular preference marks besides dispreference marks. For the purposes of modelling a standard OT account, the dispreference marks suffice.

[84] The two formulations are not strictly equivalent. When an argument is realized in

A further, practical advantage of the grammar-based constraint marking is that it makes it easy to focus on some specific phenomenon, abstracting away from irrelevant interactions. This allows a linguist to write an experimental grammar fragment rather fast: writing the LFG grammar that models $Gen_{G_{invio\,l}}$ happens simultaneously to thinking about the constraints. So, in particular one can focus attention on a small set of relevant constraints, only generating the candidate distinctions at stake.[85] Different hypothesis can be checked very quickly. Parts of the grammar that are not at the center of attention can be realized with a classical LFG analysis.

Of course, the fact that the grammar writer herself/himself can decide which constraints to check in which rule bears a certain risk, especially when the fragment grows over time: important interactions may be misjudged as irrelevant and thus left out. Later decisions are then set against an incorrectly biased background, and it may get hard to keep the entire constraint system under control. This is a familiar risk in grammar writing, occurring whenever some underlying generalizations are not made explicit in the grammar code.[86] So the selective strategy of constraint checking is presumably best applied to small or medium-sized grammar fragments of theoretical interest. For larger fragments, the learnability of OT systems based on empirical data should be exploited and thus the space of candidate variance should not be restricted too much by manual preselection.

4.5 Faithfulness constraints

In sec. 4.4, OT constraints were discussed, with a limitation to those constraints that can be checked on the candidate analysis alone (i.e., without reference to the input). This seems to exclude faithfulness constraints, which are violated exactly in those cases where the candidate analysis diverges from the input. Furthermore, the restriction of the candidate generation function $Gen_{G_{invio\,l}}$ made in sec. 4.3 (definition (94) is repeated below for convenience) seems to preclude (overly)

the two c-structural positions simultaneously (like split NPs in German, according to the analysis of Kuhn (1999a, 2001d)), the description-by-analysis formulation is satisfied, whereas the codescription formulation is violated. It depends on the linguistic account which variant meets the intentions.

[85] With this strategy, it was relatively easy to implement the relevant aspects of the OT fragment of (Bresnan, 2000, sec. 2) in the XLE system, i.e., leaving aside instances of constraint violation where they were obviously irrelevant.

[86] See Kuhn and Rohrer (1997), Kuhn (1998, 1999b), Butt et al. (1999a,b), King et al. (2000) for relevant discussion.

unfaithful candidates from getting into the candidate set in the first place: by definition, all candidates are f-structurally subsumed by the input f-structure and may not add any semantically relevant information.

(94) *Restricted definition of* ***Gen***

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

Excluding unfaithful candidate in $Gen_{G_{inviol}}$—even if it was just for the massively unfaithful candidates—would go against the methodological principle (25), according to which as much as possible should be explained as a consequence of constraint interaction. 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, $Gen_{G_{inviol}}$ should provide arbitrarily serious faithfulness violations. As discussed in sec. 3.2.4, the candidate set for *Ann laughed* should for example contain the following candidate strings (and infinitely many more):

(43)
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. John yawned
k. Ann saw him, etc.

As I will show in this section, the restriction of $Gen_{G_{inviol}}$ *is* indeed compatible with the intuitive concept of unfaithfulness in syntax, as discussed in sec. 3.2.3. Morevover, this restriction makes redundant a dependence of faithfulness constraints on the input (besides the candidate analysis), so the form of constraints introduced in sec. 4.4 encompasses both markedness constraints and what is intuitively regarded as faithfulness constraints.

The last point involves a terminological issue, which I would like to clear up in advance: it is conceivable to *define* faithfulness constraints

as opposed to markedness constraints by their reference to the input. Under this terminology the point of this section is to show that the representations assumed in syntax work best with a system employing no faithfulness constraints at all. However, I will keep up the intuitive terminology where a faithfulness violation occurs when the surface form diverges from the underlying form, not implying that the constraints do really access the input representation.

4.5.1 Faithfulness and the subsumption-based conception of $Gen_{G_{inviol}}$

Definition (94) looks very restrictive, with the subsumption condition disallowing the deletion of input information (as seems to be required for modelling MAX-IO violations, cf. (38), repeated here), and an additional clause excluding the addition of semantic information (cf. DEP-IO violations/epenthesis, cf. (31)).

(38) *Dropped pronominal in Italian*

a. He has sung

b. _ ha cantato
has sung

(31) *Expletive do in English*

a. Who did John see

b. Wen sah John
whom saw John

However, the restrictive definition of $Gen_{G_{inviol}}$ has at least two motivations: learnability (discussed in sec. 3.3) and computational complexity (which will be discussed further in chapter 6). Hence, it would be problematic to relax the restriction in order to permit the generation of unfaithful candidates. But 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 (thus the interpretation of all candidates is identical, which is important for the learner to rely on). C-structure on the other hand may contain material without an f-structure reflex (epenthesis), or leave f-structure information categorially unrealized (deletion). In the setup of LFG, this possibility can be conveniently located in the (morpho-)lexical annotations.

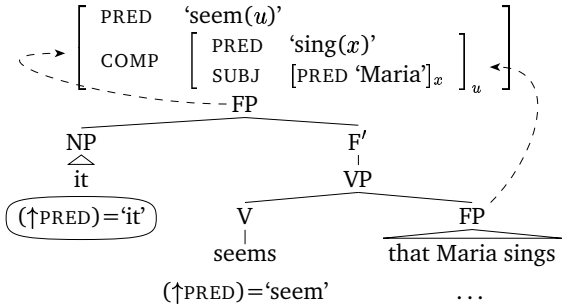
The lexical f-annotations specify semantic and morphosyntactic information. (132) shows the lexicon entry for (the full verb) *did* with two equations in the f-annotations.

(132) *did* V * (↑PRED) = 'do'
(↑TNS) = PAST

In analysis trees, the lexical f-annotations are sometimes shown below the phonological/orthographic forms for the terminal symbols (cf. (71)). Since they convey the morphological and lexical information, these f-annotations are called the 'morpholexical constraints' in Bresnan's 2000 terminology (note that the term *constraints* is not used in the OT sense of violable constraints here). Standardly, these functional annotations are treated exactly the same way as annotations in grammar rules. This means that after instantiation of the metavariables (↑) they include, they contribute to the overall set of f-descriptions the minimal model of which is the f-structure.

As Bresnan (2000) 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. In the examples to follow, I illustrate this by encircling the respective morpholexical constraint. The ways of checking such a violation technically are discussed in sec. 4.5.2.

(133) *Violation of* DEP-IO

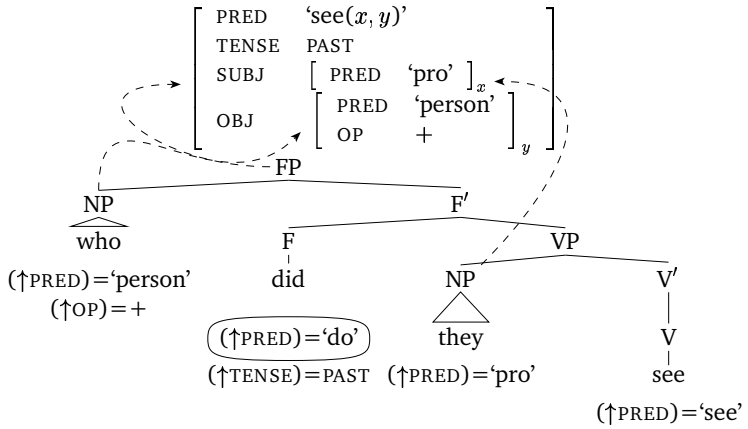


(133) is an example of an expletive use of the pronoun *it* in English, as assumed, e.g., in (Grimshaw and Samek-Lodovici, 1998, sec. 4). (134) is the well-known example of the expletive *do*.[87] Note that in contrast to classical LFG, in both these cases the ordinary lexicon entry

[87] In these examples, I use the category symbol FP for the functional categories, rather than a a concrete instance like IP. This is meant to suggest that in principle, arbitrarily

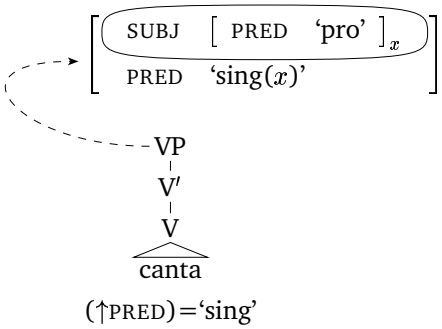
is used, i.e., referential *it*, and full verb *do*. They are just used in an unfaithful way. (In classical LFG, special lexicon entries are assumed that do not introduce their own PRED value.) In Grimshaw's terminology this type of faithfulness violation is also referred to as a case of *unparsing* a lexical item's lexical conceptual structure.

(134)



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

(135) *Violation of* MAX-IO (29)



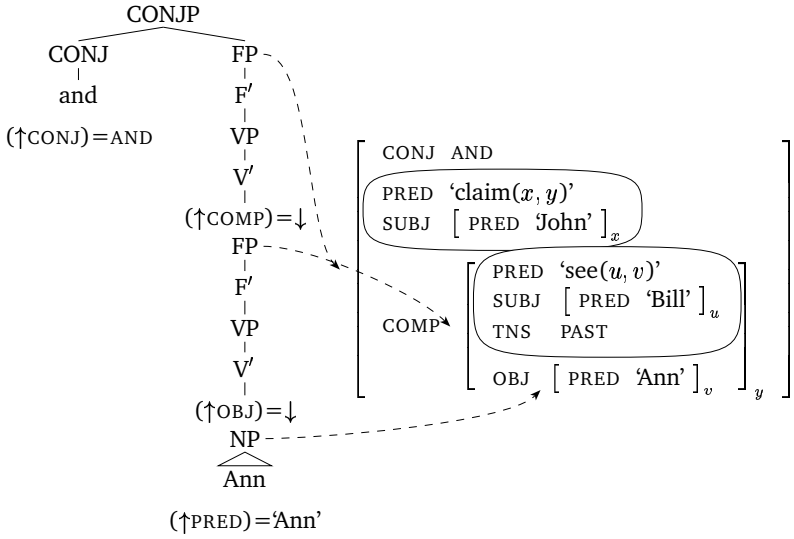
(135) is a *pro-drop* example from Italian. Note that—again as opposed

many functional projections can occur in an extended projection. The structures that we actually observe arise through the interaction of markedness and faithfulness constraints.

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.

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 (136) as one such MAX-IO-unfaithful candidate. It is the c-structure/f-structure analysis assumed for B's reply in dialogue (137).[88]

(136) MAX-IO-*unfaithful candidate in an ellipsis account*



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

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

Note the non-branching dominance chain dominating *Ann* in (136). With the re-occurrence of the categories, this structure would be ex-

[88]The representation builds on L. Levin's 1982 analysis of sluicing, assuming that at f-structure, the antecedent structure is fully reflected.

cluded by the offline parsability condition in classical LFG (77).[89] So, if we want to model the ellipsis data with such representations, construing the amount of context-recovered information as an effect of constraint interaction, the grammar G_{inviol} defining the possible candidate analyses cannot be subjected to the offline parsability condition. This will fix the choice (88a) discussed in sec. 4.2.2 and 4.2.4. We get an LFG-style grammar that is not strictly an LFG grammar, producing a superset of analyses. The additional analyses do not produce any new terminal strings, but they provide strings already covered with infinitely many new f-structural interpretations (as required for the ellipsis analysis).

Giving up the classical offline parsability condition poses questions about the decidability of the processing tasks. This is discussed in chapter 6. In essence, the restricting effect of the OT constraints can be exploited to control the set of candidates that have to be effectively generated to ensure that the optimal one is among them.

Before moving on to a more rigorous formalization of faithfulness constraints, a few remarks are due on the constraint set required for the ellipsis analysis just hinted at: It is quite clear how we can make the candidate in (136) 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).[90] The role that the context-representation plays in an OT analysis with arbitrary MAX-IO violations is discussed further in sec. 6.3.2, under a computational perspective.

[89] This is so independent of a modification of offline parsability that is usually assumed LFG (compare (237) on page 211 below).

[90] The condition that the REC constraint checks for is rather complicated, so one may hope to replace it by simpler, interacting constraints. This becomes possible in a bidirectional optimization framework as discussed in chapter 5, in particular in sec. 5.3.4.

To sum up this subsection, 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.[91] 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 (2001c)).

4.5.2 The formalization of faithfulness constraints

There are various ways how the idea discussed in the previous subsection can be formalized more rigorously. Bresnan's 2000 original proposal is based on a special way of instantiating the metavariable ↑ in morpholexical constraints. Classically, all metavariables ↑ in the set of morpholexical constraints introduced by a given lexical entry have to be instantiated to the same f-structure—the one projected from the lexical item's category. For the OT-LFG model, Bresnan assumes that some of the metavariables may be instantiated by an element that does not occur in the candidate's f-structure—at the cost of a faithfulness violation. In order to facilitate the formulation of constraints according to the description-by-analysis account discussed in sec. 4.4.5, I proposed in (Kuhn, 2001c, sec. 3.3.2) a formalization of morpholexical constraints using a special λ-projection from c-structure to a special l-structure, which I will go through in the following.[92]

The λ-projection-based formulation: DEP-IO

I will assume that all morpholexical constraints for a lexical item will be introduced in a separate feature structure projected from the pre-

[91] In particular, faithfulness violations cannot lead to the situation that a candidate has a different meaning than the meaning encoded in the input. (Compare the discussion of the ineffability account of Legendre et al. 1998 in sec. 3.3.3 and 3.3.4, which works with LF-unfaithful winners. Ineffability is however derivable through bidirectional optimization, without assuming LF-unfaithfulness, cf. sec. 5.3.5.

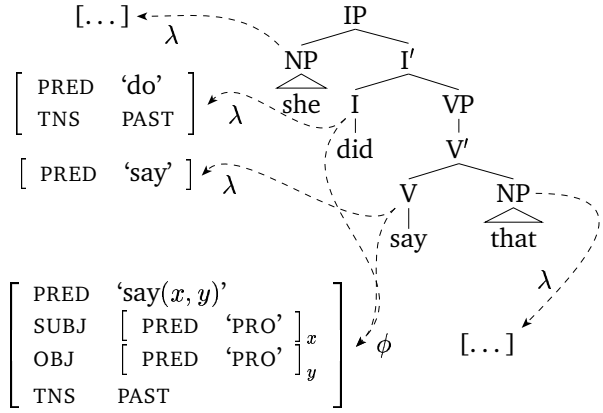
[92] In a codescription account, the formulation for the DEP-IO constraint would be rather straightforward: all morpholexical constraints can be assumed to be wrapped in a two-way disjunction – either make use of the morpholexical constraint or accept a constraint violation mark. The entry for *did* would thus look as follows:

(i) *did* V * { (↑PRED) = 'do'
| *DEP-IO ∈ *o* }
{ (↑TNS) = PAST
| *DEP-IO ∈ *o* }

terminal node.[93] Let us call this new projection the λ-projection (for "lexical"), describing a correspondence between c-structure and "l-structure". When faithfulness is met, all elements in this set will subsume the f-structure projected from the same category; however, for unparsed morpholexical constraints, a mismatch occurs.

In (138), (a modification of (Bresnan, 2000, (44))), the idea is illustrated for an example violating faithfulness: the PRED constraint introduced by *did* does not re-appear in the f-structure.

(138) DEP-IO *violation (with* PRED 'do' *missing in f-structure)*



In order to reach the intended effect, the lexicon entries have to look as follows (recall that $\uparrow$ is short for $\phi(\mathcal{M}*)$ – i.e., the f-structure projected from the current node's mother, in this case the pre-terminal):

(139) *did* I * $(f_1$ PRED$)$ = 'do'
$f_1 = \lambda(\mathcal{M}*)$
$(\ f_1 = \uparrow\ )$
$(f_2$ TNS$)$=PAST
$f_2 = \lambda(\mathcal{M}*)$
$(\ f_2 = \uparrow\ )$

[93] In (Kuhn, 2001c, sec. 3.3.2), a *set* of feature structures was assumed in the l-structure, containing a small feature structure for each morpholexical constraint. The faithfulness constraints were then checked for each of the small feature structures. Here, I assume that the faithfulness constraints are checked for the atomic values, so it is unnecessary to assume this additional complication in the geometry of l-structure.

say V * $(f_1 \text{ PRED}) = \text{'say'}$
$f_1 = \lambda(\mathcal{M}*)$
$(\ f_1 = \uparrow\ )$

For every morpholexical constraint, there are three annotation schemata, making use of a distinct local metavariable referring to a feature structure $(f_1,\ f_2,\ \ldots)$. The three schemata are: (i) the lexical constraint itself,[94] (ii) an f-equation introducing the morpholexical constraint to the l-structure projected from the pre-terminal, and (iii) an *optional* f-equation introducing the constraint at the level of f-structure. The optionality of schema (iii) leads to the presence of unfaithful analyses.

The faithfulness constraint DEP-IO can now be formulated as follows:[95]

(140) DEP-IO
$\forall n, P.[(\textit{atomic-f-str}(\star) \wedge \textit{cat}(n) \wedge (\lambda(n)\ P) = \star) \rightarrow (\phi(n)\ P) = \star]$
"For all categories n and feature paths P, if $\star$ is an atomic value under P in the λ-projection from n, then $\star$ is also the value under P in the ϕ-projection from n."

Since the metavariable $\star$ is generally instantiated to every structural element, it is now in particular instantiated to the feature structures in l-structure. Note that the value 'do' of the PRED feature in (138) fails to satisfy this constraint: instantiating n as the I category and feature path P as PRED, we have $(\lambda(\text{I})\ \text{PRED}) = \text{'do'}$, but not $(\phi(\text{I})\ \text{PRED}) = \text{'do'}$.

An issue not addressed so far is the following: What controls the choice of expletive elements (*do* rather than *shout* etc.)? This question is briefly addressed by (Bresnan, 2000, sec. 2), adopting the basic idea from (Grimshaw, 1997, 386): the assumption is that for a verb like *do*, "[t]he unparsing of its semantically minimal PRED feature is a smaller

[94]Note that the assumption is that all constraints are expressed as *defining* equations, rather than *constraining* equations, and that Completeness and Coherence are checked only on f-structure, not on l-structure.

[95]This formulation contains a universal quantification over category nodes and features paths, as was excluded in sec. 4.4.4. However, the domain of the variables is restricted by the (finite) lexicon, so a reformulation avoiding universal quantification would be possible (using the metavariable $\star$ for the category which is being quantified over in (140)). A parametrization of the constraint to a particular type of information as is usually assumed—DEP-IO(FEATURE$_i$)—is straightforward too:
$\forall n.[(\textit{atomic-f-str}(\star) \wedge \textit{cat}(n) \wedge (\lambda(n)\ \text{FEATURE}_i) = \star) \rightarrow (\phi(n)\ \text{FEATURE}_i) = \star]$

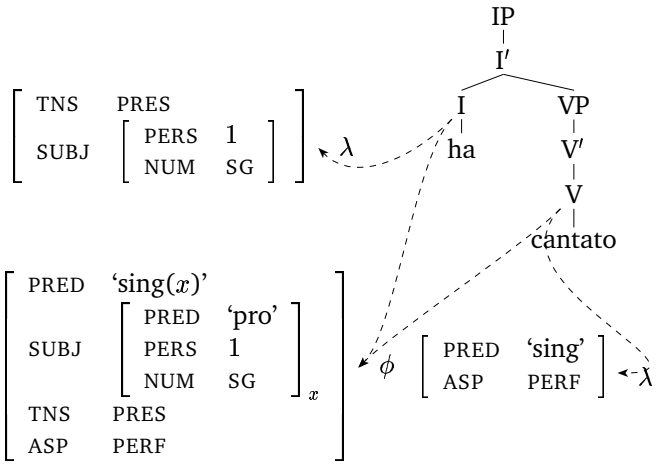
violation of faithfulness than that incurred by unparsing the semantically richer PREDs of *shout*, *obfuscate*, or any other verb in the English lexicon."

For concreteness, let us assume that this intuition is modelled by a conceptual hierarchy of PRED values—or lexical conceptual structures. More specific sub-concepts will inherit all the information from their super-concepts, plus they will add some information. Now, to evaluate faithfulness constraints on "unparsed" PRED values, the conceptual contribution they would have made is considered piece by piece, i.e., concepts embedded more deeply in the concept hierarchy will incur more violations than the more general ones. In effect, everything else being equal, the most general available concept will be picked as an expletive element.[96] I will not pursue this issue further in this book.

The MAX-IO constraint

For the MAX-IO we get a symmetrical picture as with DEP-IO. An example is given in the structure (141) for Italian *ha cantato* (has sung). Note that none of the λ-projected (i.e., morpholexical) feature structures introduces the PRED values under SUBJ, which *does* appear in the f-structure.

(141)



[96] To turn this idea into an account with reasonable empirical coverage, it clearly has to be complemented by some additional device allowing for conventionalization of the use of a particular lexical item out of a choice of semantically very similar items. For example, most Romance languages use the verb derived from Latin *habere* ('have') as a perfect auxiliary, while Portuguese uses the verb derived from *tenere* ('hold').

For technically "assembling" the f-structure, the grammar G_{inviol} has to provide a way of optionally introducing "pseudo-lexical constraints" for each piece of information occurring in the input. There are several ways in which this could be done. One is to have a pseudo-lexical annotation in the grammar rule for the root symbol, using functional uncertainty to reach arbitrary embedded f-structures and optionally provide some missing information, such as the PRED-value 'pro'. If we have a recursive rule for the root symbol, this way of adding information can be used over and over again:

(142) *Gen grammar rule,*
with the potential of providing pseudo-lexical constraints

$$\text{ROOT} \rightarrow \left\{ \begin{array}{c} \text{ROOT} \\ (\uparrow \text{GF}^* \text{ PRED}) = \text{'pro'} \end{array} \middle| \begin{array}{c} \ldots \\ \\ \end{array} \right\}$$

Alternatively, pseudo-lexical constraint introduction without the functional uncertainty could be foreseen for all maximal projections, by adding a recursion (this would change the c-structure representation, but the original structure can be systematically recovered if an appropriate marking is used):

(143) *Pseudo-lexical constraint introduction at the level of maximal categories*

$$\text{XP} \rightarrow \left\{ \begin{array}{c} \text{XP} \\ (\uparrow \text{PRED}) = \text{'}\ldots\text{'} \end{array} \middle| \begin{array}{c} \ldots \\ \\ \end{array} \right\}$$

Such pseudo-lexical constraints can of course be used only at the cost of incurring a MAX-IO violation. In example (141), we made use of this option for the PRED-value under SUBJ. We can formalize MAX-IO as follows:[97]

(144) MAX-IO
atomic-f-str$(\star) \rightarrow \exists n, P.[cat(n) \wedge (\phi(n)\ P) = \star \wedge (\lambda(n)\ P) = \star]$
"If $\star$ is an atomic value then there is some category n, such that $\star$ is embedded under some path P in the ϕ-projection from n and $\star$ is also the value under P in the λ-projection from n."

Again one can check that this constraint is violated in (141): for the value 'pro' embedded in the f-structure under the path SUBJ PRED, we

[97] Again a parametrization MAX-IO(FEATURE$_i$) is possible (compare footnote 95):
atomic-f-str$(\star) \rightarrow \exists n.[cat(n) \wedge (\phi(n)\ \text{FEATURE}_i) = \star \wedge (\lambda(n)\ \text{FEATURE}_i) = \star]$
Note that universal quantification over c-structure nodes would not give us the right result since ϕ may map several nodes to the same f-structure, and it is sufficient if $\star$ was introduced lexically by one of them.

do not find any category such that it is embedded in this category's l-structure under the same path as in f-structure (in particular, we do not find the value 'pro' under SUBJ PRED in λ(I)).

Conclusion

Concluding the section on faithfulness constraints, we can note that the subsumption-based conception of $Gen_{G_{inviol}}$ is indeed compatible with the idea of arbitrarily heavy faithfulness violations: the unfaithfulness arises as a tension within the LFG candidate analyses, since the categorial/lexical structuring need not reflect the f-structure faithfully. As a consequence of this conception, even the faithfulness constraints can be checked on the candidate analyses alone, without explicit reference to the input.[98] Both types of faithfulness violations raise certain issues for the processing which will be discussed in chapter 6.

4.6 Summary

In this chapter, I proposed a formalization of OT syntax in the formal framework of LFG, elaborating the original ideas of Bresnan (1996, 2000). This formalization meets the requirements developed in chapter 3 under empirical and conceptual considerations: the cross-linguistic variance in surface realization of underlying arguments can be derived as an effect of constraint interaction. At the same time, the precondition for learnability is met, since the semantically (and pragmatically) interpreted part of the candidate representations is identical for all members of a candidate set.

This was reached by assuming non-derivational candidate analyses based on LFG's system of correspondence between parallel structures (most notably c- and f-structure). Defining possible candidate analyses as the structures produced by a formal LFG-style grammar (G_{inviol}) comprising the inviolable principles, the entire OT system can be defined in a declarative, non-derivational fashion. The only purpose of the input (or Index) is the definition of candidate sets; since the input fixes the interpretation of the candidates, the formal representation used for the input is that of a partially specified f-structure.

The formal relation between the input f-structure and the candidates in the candidate set defined by this input is subsumption. The input

[98] Compare (Heck et al., 2000) for a similar result, and the discussion in footnote 36 on page 53.

subsumes the candidate's f-structure, and at the same time the candidate may not specify additional semantic information. Thus, the candidates differ only in terms of c-structural information and f-structural information insofar as it is not semantically relevant (i.e., purely morphosyntactic feature information). Despite this limitation as to the degree of formal divergence between input and candidates, all empirically motivated cases of faithfulness violation can be modelled. Faithfulness emerges as a candidate-internal concept and can be checked by comparing a candidate's f-structure and the morpholexical specification of the lexical items used.

Thus, it is sufficient for both markedness constraints and faithfulness constraints to refer to the candidate structure exclusively. Constraints are formulated as schemata in a tree/feature description logic over LFG representations. For constraint checking/marking, the schemata are instantiated to all structural elements (c-/f-structures) in a candidate analysis. Every structural element for which a constraint is not satisfied increases the constraint violation count for this constraint. The final evaluation step based on the constraint counts for all analyses in the candidate set is the canonical harmony-based evaluation step of OT with no specific modification for the OT-LFG scenario.

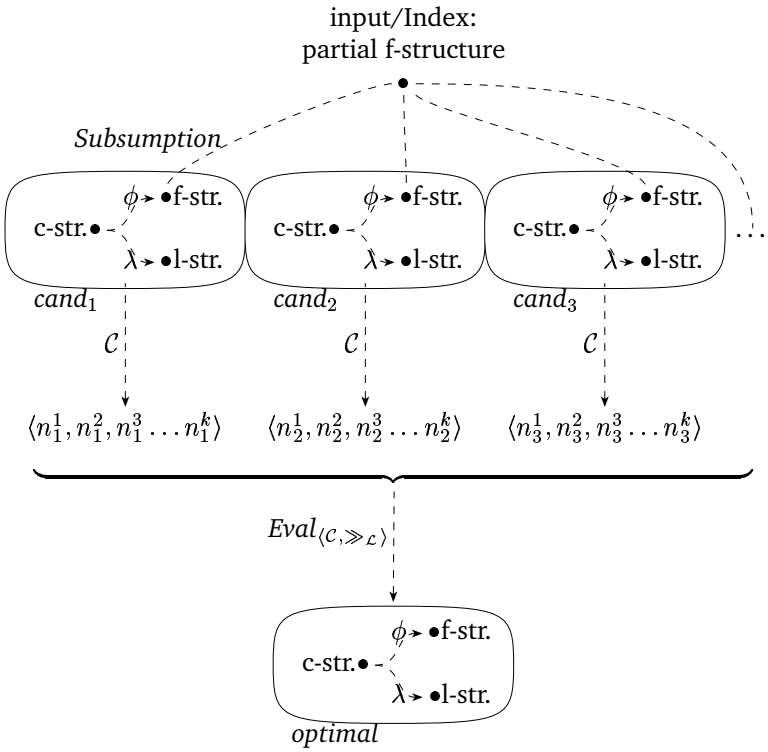
The diagram in (145) illustrates the formal setup graphically. The broken lines suggest that the relations between the formal elements making up the OT-LFG systems should be seen in a declarative way, as opposed to derivational processes.

Ultimately, it is important to note that the language generated by an OT-LFG system is defined with an existential quantification over underlying inputs. The definition is repeated here:

(87) *Definition of the language generated by an OT-LFG system* $\mathcal{O} = \langle G_{inviol}, \langle \mathcal{C}, \gg_{\mathcal{L}} \rangle \rangle$
$L(\mathcal{O}) = \{\langle T_j, \Phi_j \rangle \in L(G_{inviol}) \mid \exists \Phi_{in} : \langle T_j, \Phi_j \rangle \in Eval_{\langle \mathcal{C}, \gg_{\mathcal{L}} \rangle}(Gen_{G_{inviol}}(\Phi_{in}))\}$

So, given an LFG-style grammar G_{inviol} for the inviolable principles and a set of constraints $\mathcal{C}$ with a language-specific ranking $\gg_{\mathcal{L}}$, the set of grammatical analyses is defined as those analyses $\langle T_j, \Phi_j \rangle$ produced by G_{inviol}, for which there exists an underlying input Φ_{in} such that $\langle T_j, \Phi_j \rangle$ is optimal (based on $\mathcal{C}$ and $\gg_{\mathcal{L}}$) in the candidate set defined by Φ_{in}.

(145) *The OT-LFG setup*



February 4, 2003

5

The direction of optimization

In this chapter, some variations of the formal setup defined in chapter 4 are discussed. So far in this book, optimization has been used to find the most harmonic way of expressing some meaning, but we can also use optimization to find the most harmonic interpretation of a given utterance (i.e., a string of words, as we are focusing on the field of syntax). If we call the standard optimization we have looked at so far *production-based* or *expressive* optimization, we now get *comprehension-based* or *interpretive* optimization.

To illustrate the character of the two difference optimizations, let us look at a very simple example. We assume a G_{inviol} grammar which leaves open the serialization of arguments (say, of a transitive verb). For simplicity, the verb position is assumed to be fixed: all candidates produced by G_{inviol} have the verb in final position. So for a transitive verb, G_{inviol} will produce the orders subject-object-verb or object-subject-verb. Both variants can occur in embedded clauses in German:

(146) (weil) Anna den Film kannte
*(because) A. the movie (*ACC*) knew*
'...because Anna had seen the movie'

(147) (weil) ihn der Geruch störte
*(because) him the smell (*NOM*) bothered*
'...because he didn't like the smell'

Now let us assume two constraints (for less *ad-hoc* constraints see the discussion in sec. 5.1.1 below): SUBJPRECOBJ—'The subject precedes the object', and PRONPRECFULLNP—'A pronoun precedes a full NP'. In

standard production-based/expressive optimization, we can use these constraints to determine the most harmonic realization of an underlying meaning like in (148). (In English, the underlying meaning would be realized as *Peter saw her*.) The meaning, encoded as an f-structure, determines the candidate set; formally, it is our input. For both of the two orderings of the argument phrases we get a candidate:

(148)

Input F-Structure: $\begin{bmatrix} \text{PRED} & \text{'sehen}(x,y)\text{'} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{'Peter'} \end{bmatrix} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{'PRO'} \\ \text{GEND} & \text{FEM} \end{bmatrix} \end{bmatrix}$	SUBJPRECOBJ	PRONPRECFULLNP
a. ☞ (weil) Peter sie sah *(because) Peter her saw* $\begin{bmatrix} \text{PRED} & \text{'sehen}(x,y)\text{'} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{'Peter'} \end{bmatrix} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{'PRO'} \\ \text{GEND} & \text{FEM} \end{bmatrix} \end{bmatrix}$		*
b. (weil) sie Peter sah *(because) her Peter saw* $\begin{bmatrix} \text{PRED} & \text{'sehen}(x,y)\text{'} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{'Peter'} \end{bmatrix} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{'PRO'} \\ \text{GEND} & \text{FEM} \end{bmatrix} \end{bmatrix}$	*	

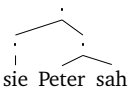
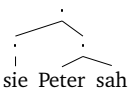
With the given constraint ranking, the subject-object-verb candidate wins.

In comprehension-based/interpretive optimization, which this chapter adds to the discussion (compare Smolensky (1996)), all candidates have the surface word string in common, but not necessarily the meaning (f-structure). Let us look at the string

(149) (weil) sie Peter sah
(because) she/her P. saw
'. . . because she saw Peter'
[*or in principle*: '. . . because Peter saw her']

The two NPs are both ambiguous between nominative and accusative, so each can either be the subject or the object of the sentence. This is reflected in the candidate set, which includes candidates with different f-structures.

(150)

Input String: (weil) sie Peter sah *(because) she/her P. saw*		SubjPrecObj	PronPrecFullNP
a. ☞  (weil) sie Peter sah	PRED ‘sehen(y, x)’ SUBJ [PRED ‘PRO’, GEND FEM] OBJ [PRED ‘Peter’]		
b.  (weil) sie Peter sah	PRED ‘sehen(x, y)’ SUBJ [PRED ‘Peter’] OBJ [PRED ‘PRO’, GEND FEM]	*	

The fact that candidate (150a) is the winner tells us that the most harmonic analysis for the given string involves the meaning which has *sie* (‘she’) as the subject and *Peter* as the object. Note that this is different from the meaning underlying all candidates in tableau (148). Candidate (150a) was not in the candidate set of the production-based/expressive optimization (148), only (148b)/(150b) is a shared candidate.

The interpretation in (150a) is indeed the strongly preferred reading for (149). This indicates that it may be possible to capture generalizations by using the same set of constraints in the two types of optimizations.

This may suffice as an initial illustration. Differences and commonalities between the two optimizations, and possible ways of combining them are discussed in the remainder of this chapter. Sec. 5.1 specifies comprehension-based/interpretive optimization in the declarative formal framework introduced in chapter 4 and reviews the motivation for assuming a close formal similarity of the two optimizations; sec. 5.2 discusses some potential counterarguments against the close similarity. Sec. 5.3 discusses various ways of combining the two optimizations in a *bidirectional optimization* model.

5.1 Varying the input to optimization

The declarative character of the OT formalization proposed in chapter 4 makes it very easy to characterize the two types of optimization as variants of the same abstract mechanism.

In the definition of candidate generation $Gen_{G_{inviol}}$ (94) (repeated below), which was discussed in the previous chapter, the argument that $Gen_{G_{inviol}}$ takes is a partial LFG analysis (a partial f-structure), which is used as the common part of all LFG analyses in the candidate set.

(94) *Definition of* ***Gen***
For Φ_{in} a partial f-structure:
$Gen_{G_{inviol}}(\Phi_{in}) = \{ \langle T, \Phi' \rangle \in L(G_{inviol}) \mid \Phi_{in} \sqsubseteq \Phi'$, where Φ' contains no more semantic information than $\Phi_{in} \}$

A formally very similar function can be defined to create the candidate set required for comprehension-based or interpretive optimization. Rather than using the (partial) input f-structure as the common substructure across candidates, we can use a part of c-structure: the string of terminal symbols (i.e., words). The remainder of c-structure and f-structure will vary freely across candidates.

(151) *Extension of the definition of* ***Gen***
For w a word string:
$Gen_{G_{inviol}}(w) = \{ \langle T, \Phi \rangle \in L(G_{inviol}) \mid w$ is the terminal string/yield of $T \}$

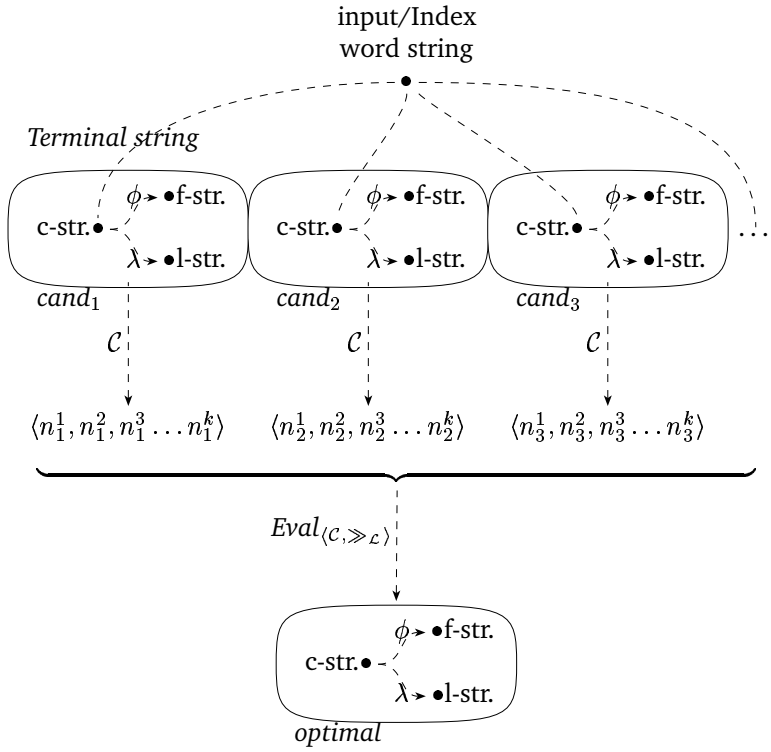
Beside these two criteria of specifying the candidate set ((94) and (151)), all kinds of other criteria for specifying the candidate set are conceivable from a purely formal point of view. For example, one might keep some higher-level part of c-structure trees fixed, while the c-structure terminal symbols and f-structure varies freely. However, in sec. 3.3 the observation was made that in order to ensure learnability, the input has to be deducible from the utterance or utterance context (formulated as restriction (49)).

(49) *Restriction on the character of the input*
All aspects of the OT input must be such that they can be in principle inferred from world knowledge and the general context of utterance.

For production-based/expressive optimization, the consequence was that only semantically and pragmatically relevant information could be included in the input (and that at the level of interpretation, all candidates had to be faithful to this information). Now, when we consider a more surface-oriented input, the learnability consideration forces us to assume a *fully* surface-detectable concept of the input. So we should assume the string of words without any fixed phrase structure representation. Definition (151) is compatible with this restriction.

Inserting the modified definition (151) of the candidate generation function $Gen_{G_{inviol}}$ into the context of the full optimization scheme gives us the diagram in (152)—a modification of (145).

(152) *Comprehension-based/interpretive optimization in OT-LFG*



Note that the overall declarative system of candidate comparison remains unaffected when the input is changed as just argued. This gives us a perspicuous high-level model of the sound-meaning relation, abstracting away from all processing details. Since the constraint set and

the candidate evaluation mechanism is identical for expressive and interpretive optimization, empirical predictions of the two formal devices are also directly linked.

It is worthwhile pointing out that the straightforward reversability of the architecture is a consequence of the strictly representational nature of the candidate analyses, which was argued for in sec. 3.3.5. The candidate status and the constraint profile of a particular candidate analysis is independent of its "use" in a candidate set; this was illustrated by candidate (148b)/(150b) which showed up in a candidate set for expressive optimization (along with other candidates sharing its meaning representation) and also in a candidate set for interpretive optimization (with other candidates sharing its surface string). If the candidate-internal relation between meaning and form were modelled as a derivational sequence of structural transformations, a reversal of the direction of candidate generation would involve a reverse application of the individual transformational steps. This does not *necessarily* affect the two optimizations: the proposed reversal remains feasible if declarative OT constraints are applied on a single (the "final") structure that includes a record of the entire derivational history, in the form of traces or chains. (Note however that in this case the question arises why the definition of candidates is still stated in derivational terms and not declaratively too; compare the discussion of (62) in sec. 3.3.5.) If on the other hand application of the OT constraints *is* sensitive to the derivational order (for instance, through stepwise filtering of the candidate set as in a "local optimization" approach, e.g. Heck and Müller (2000)), a reversal of the derivational order may change the candidate status of a particular analysis: in one direction the candidate may survive until the final optimization step, while in the other direction it is filtered out at the very beginning. This may have a snowball effect on other candidates, for which the candidate would have been a competitor later on. So, a controlled reversability of optimization seems to be guaranteed only if we adopt declarative constraints that apply simultaneously (or "globally") on comprehensive candidate representations.

Comprehension-based optimization has been proposed in various contexts in the literature. For example, Hendriks and de Hoop (2001) use such a comprehension-based optimization model in what they call OT semantics; the winning structure models what native speakers conceive as the correct interpretation in the given context. Comprehension-based optimization is also being applied as a preference mechanism in the large-scale LFG grammars developed in the ParGram project (Kuhn and Rohrer (1997); Frank et al. (1998), 2001).

Comprehension-based optimization also plays a role in learning. Tesar and Smolensky (1998, 2000) assume it as *robust interpretive parsing*. Gibson and Broihier (1998) explore to what degree such an optimization model can be used to derive disambiguation strategies in human sentence processing. [99]

Smolensky (1996) proposes to explain the lag of children's production abilities behind their ability in comprehension without having to assume a special grammar or special processing devices. The lag is predicted by an OT system if one assumes that in comprehension a simple interpretive optimization along the lines of (152) is performed, which permits processing the strings that the child hears with the same constraint ranking that is applied in production. Thus in comprehension, many analyses are accepted that are not grammatical under the child's current constraint ranking (according to the $Gen_{G_{invio\,l}}$-based definition of grammaticality). The simple parsing task is liberal enough not to filter them out. However in production, the common underlying structure *does* determine the candidate set, and the constraints will have a strong filtering effect. The result is a very reduced production ability for the initial constraint ranking.

In the following, I will address the question whether there are any empirical phenomena in adult language indicating that the constraint set used in production-based optimization has also an explanatory impact on comprehension-based optimization. This is done by applying a constraint set motivated independently for a production-based optimization account to the disambiguation task. Indeed, the disambiguation preferences observed for native speakers follow. The presentation follows (Kuhn, 2001c, 4.2). A similar analysis of word-order freezing was proposed independently by Lee (2000, 2001a,b), who goes into much more empirical detail than I do here. Here, the analysis is just used to make a formal point about the OT-LFG architecture.

5.1.1 Word order freezing

With the formalization of OT-LFG, we are in a position to address the question whether constraint systems with production-based competi-

[99] They use special constraints modelling the *Minimal attachment* and *Late closure* strategy in an OT framework and come to the conclusion that a strict ranking of constraints is inadequate for this task. Note however that the *Minimal attachment* and *Late closure* constraints are distinct from the constraints one would assume in production-based/expressive optimization models of grammaticality. Fanselow et al. (1999)) argue that human sentence processing facts *can* be derived in an OT framework, based on an incremental application of the standard grammaticality-defining constraint set.

tion are necessarily distinct or even incompatible with systems designed for a comprehension-based competition. As was indicated in the introduction to this chapter, there are phenomena where using the same constraints in both directions makes the correct empirical predictions for preference among readings.

Here I will continue to interpret optimality in comprehension-based/interpretive competition as preference of the respective reading of the input string.[100] So we hope to find empirical cases where the comprehension-based winner (based on a general constraint set) coincides with the intuitively preferred reading. It is best to look at an empirical domain which involves a fair amount of realization alternatives and ambiguity. Therefore, the relatively free word order in German is a good example. In this section, I will go beyond the somewhat ad-hoc constraints used in (148) and (150); I will show that the production-based/expressive optimization account proposed by Choi (1999) extends straightforwardly to the comprehension-based/interpretive application, making correct preference predictions.

In the German *Mittelfeld* (the region between the finite verb in verb second position and the clause-final verb position), nominal arguments of the verb can appear in any order. However, as has been widely observed (cf., e.g., Lenerz (1977), Höhle (1982), Abraham (1986), Uszkoreit (1987)), a certain "canonical" order is less marked than others (cf. also Kuhn (1995)). Deviations from this canonical order are used to mark a special information structure (or topic-focus structure), i.e., these non-canonical orderings are more restricted through context. Sentence (153) reflects the neutral order as it would be uttered in an out-of-the-blue context. Variant (154a) will be used to mark *dem Spion* as the focus; (154b) furthermore marks *den Brief* as the topic.

(153) dass der Kurier dem Spion den Brief zustecken sollte
*that the courier (*NOM*) the spy (*DAT*) the letter (*ACC*) slip should*

(154) a. dass der Kurier den Brief dem Spion zustecken sollte
b. dass den Brief der Kurier dem Spion zustecken sollte

[100] A different interpretation is the *recoverability* interpretation: while the preference interpretation assumes that all candidates have undergone production-based/expressive optimization, the recoverability interpretation uses interpretive optimization as the initial filter. A later expressive optimization is performed on the output of this filter (compare Jäger (2002a), and sec. 5.3.1).

(Choi, 1999, 150) models these data assuming competing sets of constraints on word order: the canonical constraints, based on a hierarchy of grammatical functions (and, in principle also a hierarchy of thematic roles) (155); and information structuring constraints (distinguishing the contextual dimensions of novelty and prominence, each marked by a binary feature) (156).[101]

(155) CANON (Choi, 1999, 150)

a. CN1:
SUBJ should be structurally more prominent than (e.g. 'c-command') non-SUBJ functions.

b. CN2:
non-SUBJ functions align reversely with the c-structure according to the functional hierarchy.
(SUBJ > D.OBJ > I.OBJ > OBL > ADJUNCT)

(156) Information Structuring Constraints: (Choi, 1999, 150)

a. NEW:
A [−New] element should precede a [+New] Element.

b. PROM:
A [+Prom] element should precede a [−Prom] Element.

Based on an appropriate ranking of these constraints (PROM ≫ CN1 ≫ {NEW, CN2}), Choi can predict the optimal ordering for a given underspecified f-structure (which in this case will also contain a description of the informational status of the verb arguments). When the arguments do not differ in informational status—e.g., everything is *new* but nothing is *prominent*—the canonical constraints will take effect, leading to the order in (153), as illustrated in the tableau (157), for a few sample candidates.

[101] Following more recent work (in particular by Aissen (1999)) on the OT treatment of alignment across hierarchies, the mechanism of Harmonic alignment could be used to formulate these constraints more precisely.

(157)

Input F-Structure:

$$\begin{bmatrix} \text{PRED} & \text{'sollen}(x,y)\text{'} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{'Kurier'} \\ \text{NEW} & + \\ \text{PROM} & - \end{bmatrix} \\ \text{XCOMP} & \begin{bmatrix} \text{PRED} & \text{'zustecken}(x,u,v)\text{'} \\ \text{SUBJ} & \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{'Brief'} \\ \text{NEW} & + \\ \text{PROM} & - \end{bmatrix} \\ \text{OBJ}_\theta & \begin{bmatrix} \text{PRED} & \text{'Spion'} \\ \text{NEW} & + \\ \text{PROM} & - \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

	PROM	CN1	NEW	CN2
a. ☞ der Kurier dem Spion den Brief zustecken sollte				
b. der Kurier den Brief dem Spion zustecken sollte				*
c. den Brief der Kurier dem Spion zustecken sollte		*		*

When there *are* differences in informational status, the unmarked order will however violate information structuring constraints, such that competitors with a different ordering can win out:

(158)

Input F-Structure:

$$\begin{bmatrix} \text{PRED} & \text{'sollen}(x,y)\text{'} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{'Kurier'} \\ \text{NEW} & - \\ \text{PROM} & - \end{bmatrix} \\ \text{XCOMP} & \begin{bmatrix} \text{PRED} & \text{'zustecken}(x,u,v)\text{'} \\ \text{SUBJ} & \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{'Brief'} \\ \text{NEW} & - \\ \text{PROM} & + \end{bmatrix} \\ \text{OBJ}_\theta & \begin{bmatrix} \text{PRED} & \text{'Spion'} \\ \text{NEW} & + \\ \text{PROM} & - \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

	PROM	CN1	NEW	CN2
a. der Kurier dem Spion den Brief zustecken sollte	**		*	
b. der Kurier den Brief dem Spion zustecken sollte	*			*
c. ☞ den Brief der Kurier dem Spion zustecken sollte		*		*

Like the Grimshaw/Bresnan fragment assumed in most examples so far, Choi's assumptions about *Gen* can be formulated as an LFG grammar G_{inviol}. The tableaux above showed expressive optimization. What are the predictions if the same constraint set is used in interpretive optimization?

For sentence (153) and its ordering variants, an application of

comprehension-based/interpretive optimization does not produce any interesting results, since in parsing the NPs can be unambiguously mapped to argument positions. However, if we look at sentences with ambiguous case marking like (159) and (160), the situation changes.

(159) dass Hans Maria den Brief
*that H. (*NOM/DAT/ACC*) M. (*NOM/DAT/ACC*) the letter (*ACC*)*
zustecken sollte
slip should

(160) dass Otto Maria Hans vorschlagen
*that O. (*NOM/DAT/ACC*) M. (*N/D/A*) H. (*N/D/A*) suggest*
sollte
should

Parsing (159) with the appropriate G_{inviol}-grammar will result in two classes of analyses: one with *Hans* as the subject, and *Maria* as the indirect object, and one with the opposite distribution. The latter reading is strongly preferred by speakers of German (i.e., we observe a "freezing effect"). Note that there is no way of avoiding this ambiguity with hard constraints. For (160), even more readings become possible: any of the three NPs can fill any of the three available argument positions. Nevertheless, speakers clearly prefer one reading. If we apply interpretive optimization, Choi's original constraints will predict exactly these observations: In the comprehension-based/interpretive optimization the string is fixed for all competing candidates; in addition to the string, we have to assume some representation of the context which clarifies the informational status of the referents that the new sentence is about.[102] If this informational status is neutral for all referents (e.g., all are known—*not new*—and not *prominent*), the analysis which violates the fewest constraints will be the one which interprets the arguments in such a way that the observed order is in line with the canonical order. For (160), we will get the following competition:

[102]Some discussion of the character of such a context representation will follow in sec. 5.3.4. The representation proposed there does not immediately accommodate Choi's representation of context, but it is not difficult to reformulate the underlying grammar and constraints in a way that has the intended effect. I do not go into these details here, since I only want to make a point about the general architecture of optimization.

(161)

Contextual information about set of referents: $\left\{ \begin{bmatrix} \text{PRED} & \text{'Otto'} \\ \text{NEW} & - \\ \text{PROM} & - \end{bmatrix}, \begin{bmatrix} \text{PRED} & \text{'Maria'} \\ \text{NEW} & - \\ \text{PROM} & - \end{bmatrix}, \begin{bmatrix} \text{PRED} & \text{'Hans'} \\ \text{NEW} & - \\ \text{PROM} & - \end{bmatrix} \right\}$ *Input string:* dass Otto Maria Hans vorschlagen sollte	PROM	CN1	NEW	CN2
a. ☞ $\begin{bmatrix} \text{PRED} & \text{'sollen}(x,y)\text{'} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{'Otto'} \end{bmatrix} \\ \text{XCOMP} & \begin{bmatrix} \text{PRED} & \text{'vorschlagen}(x,u,v)\text{'} \\ \text{SUBJ} & \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{'Hans'} \end{bmatrix} \\ \text{OBJ}_\theta & \begin{bmatrix} \text{PRED} & \text{'Maria'} \end{bmatrix} \end{bmatrix} \end{bmatrix}$				
b. $\begin{bmatrix} \text{PRED} & \text{'sollen}(x,y)\text{'} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{'Otto'} \end{bmatrix} \\ \text{XCOMP} & \begin{bmatrix} \text{PRED} & \text{'vorschlagen}(x,u,v)\text{'} \\ \text{SUBJ} & \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{'Maria'} \end{bmatrix} \\ \text{OBJ}_\theta & \begin{bmatrix} \text{PRED} & \text{'Hans'} \end{bmatrix} \end{bmatrix} \end{bmatrix}$				*
c. $\begin{bmatrix} \text{PRED} & \text{'sollen}(x,y)\text{'} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{'Maria'} \end{bmatrix} \\ \text{XCOMP} & \begin{bmatrix} \text{PRED} & \text{'vorschlagen}(x,u,v)\text{'} \\ \text{SUBJ} & \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{'Otto'} \end{bmatrix} \\ \text{OBJ}_\theta & \begin{bmatrix} \text{PRED} & \text{'Hans'} \end{bmatrix} \end{bmatrix} \end{bmatrix}$		*		*
d. $\begin{bmatrix} \text{PRED} & \text{'sollen}(x,y)\text{'} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{'Hans'} \end{bmatrix} \\ \text{XCOMP} & \begin{bmatrix} \text{PRED} & \text{'vorschlagen}(x,u,v)\text{'} \\ \text{SUBJ} & \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{'Otto'} \end{bmatrix} \\ \text{OBJ}_\theta & \begin{bmatrix} \text{PRED} & \text{'Maria'} \end{bmatrix} \end{bmatrix} \end{bmatrix}$		**		*

Thus, for the constraints that Choi (1999) assumes, the standard OT view of expressive optimization can be generalized to the interpretive scenario, giving rise to predictions of preferred readings that are clearly met by native speakers' intuitions.

The present discussion leaves open how exactly the interpretive optimization and the expressive optimization are combined. The application of both optimizations is called bidirectional optimization. I will discuss the possibilities for spelling out the combination in a rigorous way in sec. 5.3. Before this, I will address some issues about the character of comprehension-based/interpretive optimization, posed by the parallelism suggested by the formal treatment.

5.2 The character of comprehension-based optimization

The formal similarity of the two optimization concepts and the prospect of deriving intuitive concepts like grammaticality and preference are intriguing. However, when the idea of comprehension-based/interpretive optimization for disambiguation is applied to a non-trivial set of data, a serious issue arises, challenging the parallelism between the two optimizations.

Although the word-order freezing effect observed in sec. 5.1.1 does occur in the absence of contextual or world-knowledge clues, it can be easily overridden by such non-syntactic information. For example, take (162), which like (159) and (160) contains ambiguous case marking.

(162) dass diese Oper Mozart komponiert hat
*that this opera (*NOM/ACC*) M. (*NOM/ACC*) composed has*

Here the selectional restrictions of the verb *komponieren* 'compose'—in combination with knowledge about the argument phrases *Mozart* and *diese Oper* 'this opera'—clearly overrule the ordering preferences. The absurd reading of the opera composing Mozart does not occur to a speaker of German (neither does the sentence sound odd), even if the linguistic context of the sentence is neutral. But the straightforward comprehension-based optimization account predicts the sentence to have only the odd reading.[103]

As an immediate reaction, one might try to augment the set of constraints by additional constraints taking these extra-syntactic dependencies into account.[104] Rather than developing such an account in detail, I will address some general issues raised by this move.

[103] A similar influence of context and world knowledge in comprehension-based optimization is observed by Lee (2001a).

[104] Müller (1998) proposes an OT account for deriving markedness judgements regarding certain word order variants in German. Working with the ordinary production-based optimization, he assumes a *subhierarchy* of constraints which is used only for determining markedness, while the matrix hierarchy is used for grammaticality. The constraints in the subhierarchy are in part based on extra-syntactic concepts like definiteness, animacy, and focus; however, no explicit reference to the actual context of an utterance is made. So the empirical predictions can only be checked against intuitions about gradual differences in markedness of isolated sentences. Since to some degree such intuitions depend on the informant's ability to make up plausible contexts, they have to be regarded with caution. The model that I will argue for in sec. 5.2.2 and demonstrate in sec. 5.3.4, assumes an explicit formal representation of context as part of the input and thus circumvents the vagueness problem in the validation of empirical predictions.

The direction of optimization

One general concern which I will only mention in passing is that clearly the boundaries of what is traditionally considered the scope of syntactic theory are passed when such effects of discourse context and world knowledge are incorporated into a single optimization task. *Per se*, this is neither completely new (since "pragmatic effects" on syntax have frequently been observed), nor necessarily a bad thing. Ultimately, the parallelism assumption of OT leads us to expect such a global cognitive optimization ("Parallelism: all constraints pertaining to some type of structure interact in a single hierarchy", (Kager, 1999, 25)). However, so far, reasonable idealizations factorizing out details of the utterance context etc. have been vital for progress in linguistic theory. So one might become skeptical in case there is no way at all of pinning down the linguistic part of the problem by a suitable interface specification (this may indeed be possible as discussed in sec. 5.3.4).

5.2.1 Apparent counterevidence against constraint ranking

Let us assume we want to capture the extra-syntactic factors in disambiguation by additional constraints within comprehension-based optimization. I will concentrate on the subtask of getting the filling of the argument positions correct (this ignores other ambiguities like ambiguous tense/aspect forms, quantifier scope, resolution of definite descriptions etc.). We have to take into account at least the knowledge sources in (163). The inferences, based on linguistic knowledge are informally sketched by implication arrows. Examples are given below. Unless one of the knowledge types leads to an unambiguous result, one typically gets the situation of several knowledge sources interacting to lead the hearer to a particular disambiguation.

(163) *Knowledge sources involved in syntactic disambiguation & type of knowledge derivable*

• Phonological string

a.word order ⇒ (information structural status of argument phrases ⇒) grammatical functions of argument phrases ⇒ argument filling

b.morphological (head or dependent) marking ⇒ grammatical functions ⇒ argument filling

c.intonational marking (or typographical emphasis) ⇒ information structural status of argument phrases ⇒ grammatical functions ⇒ argument filling

- Discourse context

d.information structural status of argument phrases ⇒ grammatical functions ⇒ argument filling

e.anaphoric dependency ⇒ semantic class of (pronominal or polysemous) argument phrases ⇒ argument filling

- Extra-linguistic context

f.reference of deictic expressions ⇒ semantic class of argument phrases ⇒ argument filling

g.partial disambiguation of argument filling through given situation

- Idiosyncratic/encyclopaedic knowledge

h.selectional restrictions of the predicate ⇒ argument filling

i.semantic class of (polysemous) argument phrases ⇒ argument filling

j.exclusion of particular readings (encyclopaedic knowledge to the contrary)

Languages like English have word order (a.) as a fairly reliable direct source for function specification, which again allows inferences about the argument filling. In languages with a freer word order, like German, word order does not permit hard inferences, but may be an important clue, presumably via inferences about the information-structural status. We have seen case marking as an example for morphological dependent marking of grammatical function (b.), subject-verb argeement is a widespread example for morphological head marking. Intonational marking (c.) in spoken language provides clues about information-structural status (e.g., topic-marking pitch accent), which may for instance explain a certain word order, thus giving additional clues about function specification.[105]

Interacting with intonational marking, the discourse context provides clues about the information-structural status of argument phrases (d.). For instance, the second sentence in (164) contains ambiguous case marking of the familiar kind. With the first sentence as context, it is fairly clear that we have two predications about the same topic, *Anna Schmidt*. The noun phrase *die spätere Wahlsiegerin* will typically bear a rising accent, and the intonation contour will rise over the rest of the second sentence up to the predicate *vorgeschlagen* with a falling accent. Furthermore drawing on encyclopaedic knowledge about elections, *die*

[105] Some discussion of such effects can be found in (Kuhn 1996a, 1996b, 1996c).

spätere Wahlsiegerin is inferred as the theme argument of *vorschlagen* 'propose'.

(164) Anna Schmidt stellte sich einer Kampfabstimmung
A. S. stood-for REFL *a competitive election*
Die spätere Wahlsiegerin hatte Otto Müller für die Kandidatur
The later winner had O. M. for the candidacy
vorgeschlagen
proposed

(164) contains an example of discourse-based anaphora resolution illustrating knowledge type (e.) in (163): pronominals and definite noun phrases have to be resolved in order to allow inference about the semantic class of an argument phrase. Likewise, deictic phrases allow the hearer to include information from the extra-linguistic context (f.). Of course, this context may also narrow down the choice of readings (g.) (for example, it may be obvious who is the agent of a certain action).

A fairly reliable source for direct inferences about argument filling (without the reasoning via grammatical functions) is knowledge about selectional restrictions (h.). An example of this was given in (162), where it was clear that *compose* has to take a human being as its agent. This inference may nevertheless be distorted by polysemy of argument phrases (i.). For instance (165), which normally has a preference for the subject-object reading, may also have a object-subject reading in case *Mozart* is used for the works of Mozart and *diese Oper* refers to an opera house. A context for bringing out this reading would be (166).

(165) Mozart hat diese Oper nie aufgeführt
M. has this opera never performed

(166) Unser Haus ist spezialisiert auf die Italienische Oper der
Our house is specialised in the Italian opera of
Romantik
Romanticism

Most of the context-based inferences can also be made without the relevant knowledge being actually introduced in the linguistic context; in this case, the inferences are based purely on encyclopaedic knowledge ((j.) in (163)).

The purpose of presenting this fairly long list of knowledge types involved in disambiguation is not to start the formalizion of a particular

constraint set for comprehension-based optimization. The validation of a non-trivial account of the problem would require a larger-scale computer simulation of learning based on empirical data and is thus clearly beyond the scope of this book. Here, the list of knowledge types should serve as background for considerations about the general character of this comprehension-based optimization.

At first, it may seem as if OT with its soft constraints is ideal for modelling this interaction of uncertain knowledge, which contributes defeasible information. Roughly, we might model each knowledge type as a constraint, introducing violation marks for the marked options. (For example, reverse word order with respect to the functional hierarchy would be marked; likewise type coercions from individuals to their works, as we had it in the *Mozart* polysemy; for the discourse-context-sensitive knowledge, the constraint system of Beaver (2000) could be applied.) From the different interpretations of a string, the most harmonic one would then be predicted as the preferred reading.

But does constraint interaction under the OT assumptions really capture the way in which the knowledge sources are interrelated in disambiguation? What ranking should we assume for the constraints? Some sources are more reliable than others (in part depending on the language), so these should be high in the constraint hierarchy. But an important knowledge source can be overridden by a sufficient amount of lower-priority clues. This points towards a constraint *weighting* regime—contrary to OT assumptions.[106] Within a ranking regime, an additive effect could be reached only if all relevant constraints are unranked with respect to each other (or have a very similar rank in a stochastic OT approach), which again would defeat the idea of modelling reliability of knowledge sources by the rank of the corrsponding constraint.

A possible conclusion at this point might be that the two optimizations are not instances of the same formal setup (compare fn. 99 on page 131, pointing to a similar conclusion made in Gibson and Broihier (1998)): constraint *ranking* has proven a very useful formal restriction on production-based/expressive optimization model of grammaticality, but it may not extend to an optimization model of interpretive preference. Preference might be better captured with a weighting model. If

[106]Compare also arguments from the modelling of gradient grammaticality judgements Keller (2000), Keller and Asudeh (2001). Keller (2000) provides psycholinguistic results showing that under the assumption of agreed-upon constraint sets, gradience judgements cannot be derived with a pure constraint *ranking* system.

this conclusion is correct, the formal similarity of the two optimization types in the declarative OT formalization would seem to be of limited empirical value.

However, the issues raised in this section could be interpreted differently, which is attempted in the following section.

5.2.2 Optimization with a fixed context

Due to the "softness" of OT constraints, i.e., the fact that they are violable by the ultimate output analysis, they seem to lend themselves to the modelling of defeasible information used in disambiguation. This was sketched in the previous section: if there is a sufficient amount of defeasible evidence (say, definiteness of argument phrases, intonational marking, plus encyclopaedic likelihood) pointing towards a particular reading of an utterance, this will overwhelm a residual of defeasible evidence in favour of a different reading (say, word order).

However, I conjecture that this interpretation of softness is different in a principled way from the typology-inducing character of constraint softness as used in the OT constraint ranking model of grammar. The typology-inducing constraint ranking model is crucially based on the (idealizing) assumption that there is no uncertainty in the knowledge sources underlying the constraint profile. The fact that a particular markedness constraint (say, OB-HD) outranks a particular faithfulness constraint (maybe DEP-IO) in language A has no implications to the effect that in language A, the clues suggesting that OB-HD is violated in a given analysis are more reliable than the clues for DEP-IO. Each candidate in a "pure" OT competition is a point in the abstract space of possibilities; the epistemological question of which are the matching abstract candidates for a given utterance lies outside the model. My conjecture is that this matching task is the locus of the uncertainty observed in the previous section. But this uncertainty cannot be modelled with the ranking of constraints too (without leading to an inconsistent overall picture).

In order to accomodate for both aspects of constraint softness, we can assume a two-stage optimization model (where the two stages do not imply any sequentiality, but just different regimes of defeasibility): the ("outer") context-determining optimization and the ("inner") optimization relative to a fixed context.

(167) *The two-stage optimization model*

Space for context-determining optimization			
Selection based on clues from linguistic/real-world context, inference and experience.			
Formal context c_1	Formal context c_2		Formal context c_n
Classical OT, profiles fixed by context c_1.	Classical OT, profiles fixed by context c_2.	...	Classical OT, profiles fixed by context c_n.

The term (formal) *context* is used here in a technical sense. It comprises all the details that have to be fixed in order to unambiguously determine the full constraint profiles required for the inner optimization; a real-world context of utterance may leave some such decisions open, so several technical contexts are possible (maybe with different probability). While by assumption, the inner optimization follows the typology-inducing ranking approach of OT, the additive effect discussed in the previous section suggests that the outer optimization follows a less restrictive regime. (Only the inner optimization focuses exclusively on linguistic knowledge, the outer system includes the interface to the overall cognitive system; so we do not expect typological predictions to arise in this part. It is a research question whether the encapsulated inner system can be specified in a manner that comprises all and only the predictions pertaining to the typologically relevant linguistic knowledge.)

Why did the issue of the two types of constraint defeasibility arise only when we started looking at preferences in the disambiguation task (as modelled by comprehension-based/interpretive optimiztion)? When we are interested in native speakers' (or rather hearers') preferences for a particular reading of a sentence, it is next to impossible to abstract away from the outer optimization task. Too many information sources are effectively underspecified—especially under the standard linguistic methodology where examples are presented with no or very little linguistic context and in written form.

For modelling grammaticality in an expressive optimization framework, it is less unnatural to assume the idealization that a speaker has direct access to the fully specified contexts of the inner optimization. The input is not just the surface part of an utterance made by some other speaker, but it is the meaning that the speaker wants to convey

in the actual context. The only way the speaker can go wrong in making a choice in the outer optimization is by making wrong assumptions about the common ground shared with the hearer.

So if this two-stage optimization model is indeed adequate, we actually expect that superficially, a simplified model using the ranking architecture of the inner optimization for the entire problem makes (almost) correct predictions for expressive optimization, while it fails for interpretive optimization (where a less restrictive model, i.e., a constraint *weighting* approach seems to be required). However, for an approach taking into account both directions, the two-stage architecture captures the relation between the two directions, in particular in the inner, constraint-ranking based optimization. The effects of this ranking in interpretive optimization may be generally masked by the outer optimization, but one may nevertheless be able to prompt them by narrowing down the contextual choices.

In this section, I argued that apparent counterevidence against a formal parallelism of the two directions of optimization *can* be accomodated in a parallel architecture that is not necessarily more complicated than an the asymmetrical model one would be forced to assume otherwise.

5.3 Bidirectional optimization

So far in this book, production-based optimization and comprehension-based optimization have been discussed in isolation (in chapters 2–4 and in sec. 5.1, respectively) or compared (sec. 5.2), but quite obviously the question arises of how the two optimizations relate to each other.

If one of them models grammaticality and the other some concept of preference, we expect that they have to make some reference to each other. A model taking into account both directions of optimization is called a *bidirectional optimization* model. (Recall that the term *direction* should not be taken literally, which would suggest a procedural/derivational definition; the formalization in chapter 4 and the extension in sec. 5.1 showed that a strictly non-derivational/declarative formulation is possible.)

Bidirectional optimization has been argued for variously in the theoretical OT literature, on empirical and conceptual grounds (see, e.g., Wilson (2001), Boersma (1998), Smolensky (1998), Lee (2001a),

Kuhn (2001c), Blutner (2000)). There are surprisingly many alternative formal options of combining the two unidirectional optimizations. Here, I will point out the most important options and discuss the general consequences for the character of the different resulting combined models (for a comparison with a similar goal, compare Beaver (2003)). Since there are various open issues about the individual optimization models (in particular the comprehension-based one, as discussed in sec. 5.2), it is certainly too early to draw any final conclusions.

5.3.1 Sequential bidirectional optimization models

There are two conceptually simple ways of combining the two unidirectional optimizations: one could run one after the other (resulting in a choice of which comes first), or one could run both simultaneously. The first variant is discussed in this section, the other one in sec. 5.3.2. Note that the terms "sequential" or "simultaneous" are used to refer to the *logical* sequence between the optimization relations, they are not meant to model any cognitively relevant order of processing. Procedures for computing these abstract relations will be discussed in chapter 6.

Recall the definition of the language generated by an OT system (based on the original concept of $Gen_{G_{inviol}}$, which was defined for production-based optimization—the essential detail is that the candidate set is defined by a common input f-structure Φ_{in}):

(87) *Definition of the language generated by an OT-LFG system* $\mathcal{O} = \langle G_{inviol}, \langle \mathcal{C}, \gg_{\mathcal{L}} \rangle \rangle$

$$L(\mathcal{O}) = \{ \ \langle T_j, \Phi_j \rangle \in L(G_{inviol}) \mid \exists \Phi_{in} : \langle T_j, \Phi_j \rangle \in Eval_{\langle \mathcal{C}, \gg_{\mathcal{L}} \rangle}(Gen_{G_{inviol}}(\Phi_{in})) \ \}$$

The grammaticality/preference model

We may base the construction of a bidirectional model on the linguistic intuitions we have about the concepts being modelled by the two optimization tasks. The original production-based optimization models grammaticality, while comprehension-based optimization models preference of readings. If we would like to give preference only a secondary status with regard to grammaticality, the production-based grammaticality model has to "feed" the comprehension-based preference model. That means that losers in the first optimization are discarded as ungrammatical, while the losers in the second optimization represent dispreferred readings of a string, but are nevertheless grammatical.

The direction of optimization

To illustrate the intuitions behind the second optimization step as modelling the preferred reading of a string, let us use a very simple concrete example. Let us assume a constraint FIRSTLASTNAME that is satisfied when a sequence of a first name and a last name (like *John Smith*) is interpreted as forming a single full name. It is violated when such a sequence is split into two nominal phrases. (There are various ways of implementing this constraint technically, which is not relevant here.) In comprehension-based optimization, the string is fixed across candidates, while the syntactic structure and interpretation (c-structure and f-structure) may vary. Assume the sentence *Today, Sue called John Smith* as the input string (as in tableau (168)). The FIRSTLASTNAME will be satisfied in the reading which we may paraphrase as *Today, Sue gave Mr. John Smith a call*, but it is violated in the reading *Today, Sue called John by the name "Smith"*. So, interpreting optimality as hearer preference, our tiny OT system will predict the former reading to be the preferred interpretation of the sentence.

(168) *Comprehension-based optimization*

		FIRSTLASTNAME
Input String: Today, Sue called John Smith		
a.	VP: V, NP, NP; … called John Smith	PRED 'call(x, y, z)'; SUBJ [PRED 'SUE']; OBJ [PRED 'John']; OBJ$_\theta$ [PRED 'Smith']	*!
b. ☞	VP: V, NP; … called John Smith	PRED 'call(x, y)'; SUBJ [PRED 'SUE']; OBJ [PRED 'John Smith']	

As pointed out, in this scenario comprehension-based optimization has only a secondary status. Prior to it, we may assume the original production-based optimization model of grammaticality. Both of the two candidate analyses of the string are actually grammatical for English, so we model both of them as winners of two different initial production-based optimizations, based on their respective input f-structures. So, candidate (168a) is grammatical, but not the preferred reading of the given string. This distinguishes (168a) from losers in the initial optimizations, like *Today, Sue John Smith called*.

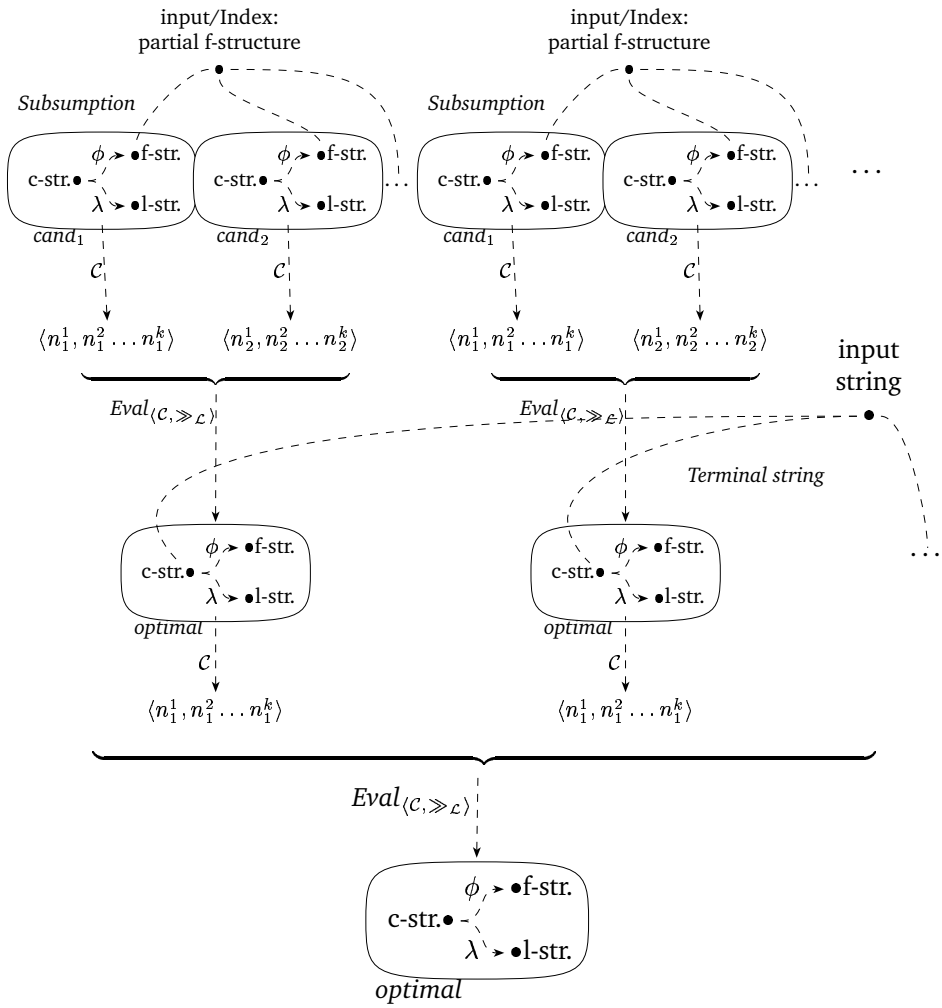
Formally, this means that in the sequential grammaticality/preference model, the definition of the language (=the set of grammatical analyses) generated by an OT system is not changed. But we are adding an additional notion, the notion of the *preferred reading* of a string on top of this.

(169) *Preferred analyses in the grammaticality/preference model*
$\langle T_j, \Phi_j \rangle$ is among the preferred analyses of a string w
iff
$\langle T_j, \Phi_j \rangle \in \mathit{Eval}_{\langle \mathcal{C}, \gg_{\mathcal{L}} \rangle}(\; L(\mathcal{O}) \cap$
$\{ \; \langle T', \Phi' \rangle \in L(G_{inviol}) \mid$
w is the terminal string of $T'\} \;)$

Note that here the set of analyses that is evaluated—the argument of $\mathit{Eval}_{\langle \mathcal{C}, \gg_{\mathcal{L}} \rangle}$—is not defined by $\mathit{Gen}_{G_{inviol}}$, but it is a subset of the language defined by the (production-based) OT-LFG system. The interactions are visualized in the schematic illustration in (170) on page 148.

The upper half reflects the production-based optimization underlying the definition of the language $L(\mathcal{O})$. Only the winners which contain the relevant input string are considered further on; this is indicated by the broken lines from the fixed *input string* on the right-hand side in the middle. The lower half then shows how from this (infinite) set of analyses with a particular terminal string, the preferred analysis is determined by optimization.

(170) *The grammaticality/preference model*



The comprehension-based/production-based sequence

Besides the grammaticality/preference sequence, which leaves the concept of grammaticality untouched, it is conceivable to adopt the opposite order of the two optimizations. An initial step of comprehension-based optimization filters out certain readings of a string, and only the winners are candidates for the final production-based optimization. Such a model is assumed by Jäger (2002a). Intuitively, the initial step filters out readings that would be irrecoverable for the hearer; grammaticality is defined through the final optimization. Note that this opposite order leads to different empirical predictions (if the constraint set is left unchanged). Candidate (168a) would no longer be called grammatical: it would have been filtered out in the first step since it loses against (168b).

The comprehension-based/production-based sequence of optimization can be useful in predicting certain ways of expressing a thought because another alternative would be hard to interpret for the hearer. Take the string *Mary told John Smith left*. If we split *John* from *Smith*, we get a reading that should be perfectly acceptable. But unless we get very clear prosodic clues that disambiguate the sentence in this direction, the string is odd. A speaker would say *Mary told John that Smith left*. This rephrasing is not predicted by the grammaticality/preference model (169), since preference is only modelled over identical strings.

In the comprehension-based/production-based sequence of optimization, the winner of the initial comprehension-based optimization would be an analysis with the bracketing *Mary told* [$_{\text{NP}}$ *John Smith*] *left*, as shown in (171).[107] The verb *left* has a null subject in this analysis. (We assume here that the constraint FIRSTLASTNAME is highly ranked.) However, in the subsequent production-based optimization starting from (171b)'s f-structure, it will turn out that the winner of the first optimization does not survive as a grammatical analysis for English: The analysis with the original string loses against *Mary told John Smith she left* (compare (40) on page 37). Therefore, the initial string *Mary told John Smith left* is predicted to be unacceptable. If we rephrase it as *Mary told John that Smith left*, the constraint FIRSTLASTNAME will no longer filter out the intended reading, so grammaticality follows.

[107]For the SUBJECT constraint, which is just used as an example for further lower-ranking constraints, compare (40) on page 37.

(171) *Comprehension-based optimization*

Input String: Mary told John Smith left		FIRSTLASTNAME	SUBJECT
a.	VP [V … told, NP John, NP Smith, S [VP left]] PRED 'tell(x, y, z)' SUBJ [PRED 'MARY'] OBJ [PRED 'John'] COMP [PRED 'leave(u)' SUBJ [PRED 'SMITH']]	*!	
b. ☞	VP [V … told, NP John Smith, S [VP left]] PRED 'tell(x, y, z)' SUBJ [PRED 'MARY'] OBJ [PRED 'John Smith'] COMP [PRED 'leave(u)' SUBJ [PRED 'PRO']]		*

5.3.2 Strong bidirectional optimization

As an alternative to the sequential bidirectional models, we could assume that both directions are equally important and have to apply conjuntively. In a successful candidate, the string must be optimal among all strings for the underlying meaning *and* the meaning must be optimal among all possible meanings of the string. This is what Blutner (2000) calls the strong bidirectional optimization model (the corresponding *weak* model will be discussed in sec. 5.3.3).

The crucial difference between the strong bidirectional model and the sequential comprehension-based/production-based model is the following: in the strong bidirectional model, the candidate sets for both optimizations are independent, while in the sequential model, the candidate set in production-based optimization (the second optimization) consists only of comprehension-based winners. This may result in different predictions. In strong bidirection it may happen that the two optimizations do not *agree* on a candidate as the winner. This cannot happen in the sequential model, since the second optimization will only have initial winners as candidates.

For concreteness, let us add a very simple Economy-of-expression constraint *STRUCT to out sample OT system, which is violated by any c-structure node. With this constraint, an independent production-based optimization for the input f-structure (172) will produce the

string *Mary told John Smith left* as the winner, since it violates *STRUCT fewer times than *Mary told John that Smith left*.

(172) *Production-based optimization*

Input F-Structure:	*STRUCT
$\begin{bmatrix} \text{PRED} & \text{'tell}(x, y, z)\text{'} \\ \text{GF}_1 & \begin{bmatrix} \text{PRED} & \text{'MARY'} \end{bmatrix} \\ \text{GF}_2 & \begin{bmatrix} \text{PRED} & \text{'John'} \end{bmatrix} \\ \text{GF}_3 & \begin{bmatrix} \text{PRED} & \text{'leave}(u)\text{'} \\ \text{GF}_4 & \begin{bmatrix} \text{PRED} & \text{'SMITH'} \end{bmatrix} \end{bmatrix} \end{bmatrix}$	
a. ☞ Mary told John Smith left	*...*
b. Mary told John that Smith left	*...**!

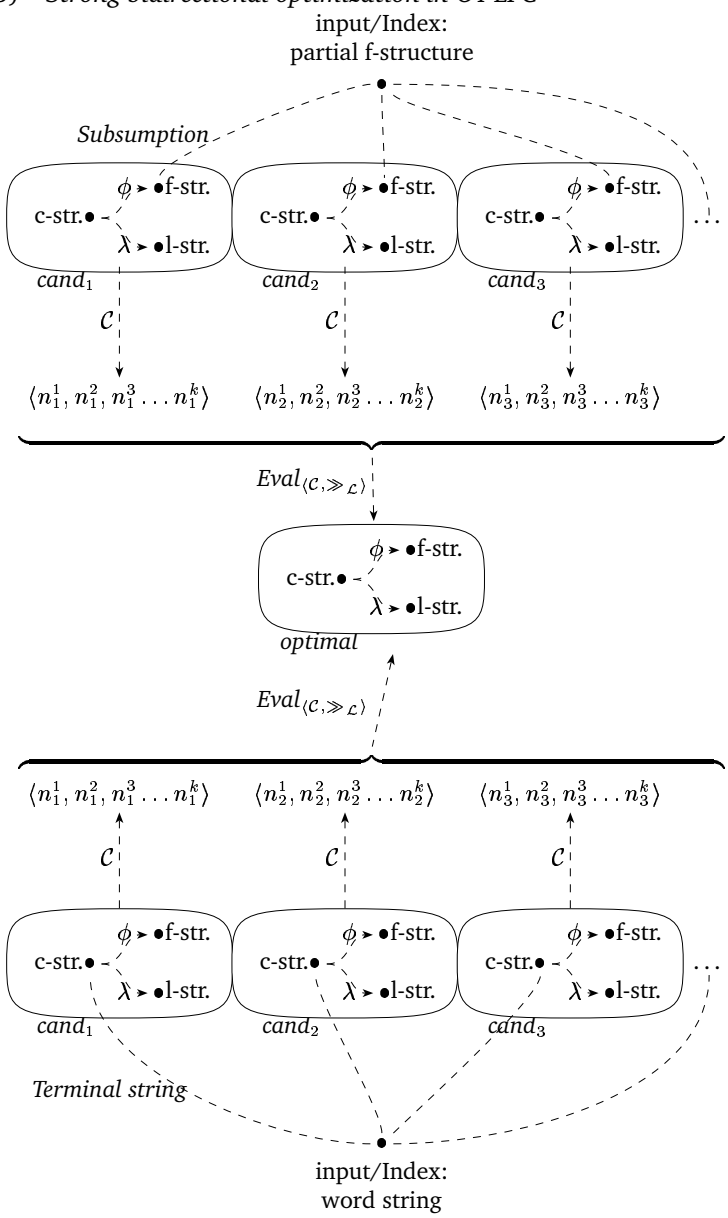
The comprehension-based optimization works exactly as shown in (171). Note that (171a) and (172a) are identical LFG analyses: this candidate shows up in both candidate sets. However, it wins only in (172), while losing in (171). Similarly, (172b) is the winner for its comprehension-based optimization (not shown here), but a loser for its production-based optimization (namely (172)).

The two directions do not agree on a winner. In strong bidirectional optimization this means that none of the mentioned candidate analyses is in the language generated by the system. This includes candidate (172b) which would be a final winner in the sequential system, but which is here *blocked* by the simpler (though ungrammatical) f-structure-realization (172a). This blocking effect can be evoked fairly easily in a strong bidirectional system.

The definition of strong bidirectionality and its consequences

In sec. 5.1, an extended definition of $Gen_{G_{inviol}}$ was considered which generalizes to the application for both optimization directions (see (151)). So we can define a language consisting of those analyses that are optimal for *both* directions: the formal definition is given in (174) on page 153. Illustrating the overall system again graphically, we get the scheme (173) on page 152. Note the independence and symmetry of the two optimizations.

(173) *Strong bidirectional optimization in OT-LFG*



(174) *Language generated by a strong bidirectional OT system*

$$\mathcal{O} = \langle G_{inviol}, \langle \mathcal{C}, \gg_{\mathcal{L}} \rangle \rangle$$

$$L_{2/strong}(\mathcal{O}) =$$
$$\{ \langle T_j, \Phi_j \rangle \in L(G_{inviol}) \mid$$
$$\exists \Phi_{in} : \langle T_j, \Phi_j \rangle \in \mathit{Eval}_{\langle \mathcal{C}, \gg_{\mathcal{L}} \rangle}(\mathit{Gen}_{G_{inviol}}(\Phi_{in}))$$
$$\textit{and}$$
$$\exists w : \langle T_j, \Phi_j \rangle \in \mathit{Eval}_{\langle \mathcal{C}, \gg_{\mathcal{L}} \rangle}(\mathit{Gen}_{G_{inviol}}(w)) \quad \}$$

The independence of the two optimizations in the strong bidirectional model has the advantage that the system is conceptually rather simple (this goes along with certain computational advantages, as we will see in sec. 6.3). With the *blocking* technique, strong bidirectionality also opens up a simple way of deriving Language-Particular Ineffability, a phenomenon discussed in sec. 3.3.3 which poses a problem for standard unidirectional OT (recall that the assumption of LF-unfaithful candidates made by Legendre et al. (1998) is highly problematic from a learnability point of view). In strong bidirectional OT, all those underlying forms are predicted to be ineffable for which the production-based winner is suboptimal in the comprehension-based optimization (like in the above example where the input f-structure of (172) is effectively ineffable). See sec. 5.3.5 for a more detailed discussion and illustration.

The simplicity of an Ineffability analysis in the strong bidirectional model obviously excludes the danger of overgeneration that one may see for a standard production-based OT system. It is not so clear however whether this effect is reached at the price of undergeneration. Here, a problem of the independence of the two optimizations may become relevant: for every well-formed analysis one has to ensure that it comes out optimal in both directions. Without further adjustments (such as context-sensitive constraints), the strong bidirectional system predicts that ambiguity of strings is practically nonexistent. This *ambiguity problem* was first pointed out by Hale and Reiss (1998). But even a system with context-sensitive constraints, which could in principle resolve the ambiguity problem, requires much further research. With the enforced context-dependence of the comprehension-based optimization discussed in sec. 5.2, there is at least a practical problem for strong bidirectional OT: the constraint system taking context-effects into account must be extensive for a non-trivial grammar fragment.

A related potential disadvantage of strong bidirectionality is that due to mutual independence, no interaction between the optimizations can be exploited for explanatory purposes. Thus, the inherent idea of con-

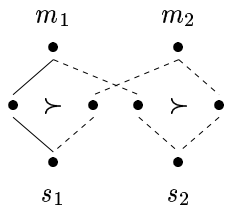
straint interaction as the main explanatory device is defeated at the interface of the two directions. If interaction effects across the two optimization directions *are* allowed, this can simplify the constraint systems considerably, as will be discussed in the following subsection.

5.3.3 Weak bidirectional optimization

Perhaps most problematic about the strong concept of bidirectionality is the strict blocking effect that any analysis exerts which is optimal in one of the directions. Competitors that are less harmonic cannot play any further role at all. In his OT system for lexical pragmatics, Blutner (2000) proposes a concept of *weak bidirection* that remedies this problem. Like strong bidirection, weak bidirection is symmetrical—none of the two directions is prior to the other. However, the candidate sets in the two directions are not independent of each other, but highly interrelated—based on a recursive definition: only the ultimate winners are allowed as candidates. In order to visualize the effect of such a recursive optimality relation, one of course has to work with preliminary candidate sets containing analyses that have to be eliminated later on. When the weakly bidirectional winners are computed this way, analyses that would be a loser in strong bidirection "get a second change" because a blocking competitor gets removed.

This is best seen in a small abstract example. Assume we have two strings s_1 and s_2, both of which could have either meaning m_1 or m_2 according to G_{inviol}. So we have four possible candidates $\langle s_1, m_1\rangle$, $\langle s_2, m_1\rangle$, $\langle s_1, m_2\rangle$, $\langle s_2, m_2\rangle$. Let us assume that $\langle s_1, m_1\rangle$ has the most harmonic constraint profile, and $\langle s_2, m_2\rangle$ the least harmonic profile (the other two candidates are in between).

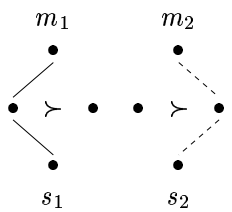
(175)



So if we do a comprehension-based optimization for input s_1, candidate $\langle s_1, m_1\rangle$ will win. For input s_2, candidate $\langle s_2, m_1\rangle$ wins. Similarly for production-based optimization. In strong bidirection, the analysis $\langle s_1, m_1\rangle$ is the only acceptable candidate: the s_2 and m_2 candidates are blocked by this candidate.

In weak bidirection, only the ultimate winners ("superoptimal candidates") are actually defined to be candidates in a competition. So, for instance, we cannot tell ahead of time which are the candidates of the comprehension-based optimization for string s_2. We have to compute the correct candidate sets (=the superoptimal candidates) recursively. In the recursion, we have to start at the base case. The base case are those analyses for which we know that no other (potential) candidate can be more optimal in either direction. These are exactly the "old" strong bidirectional winners—in our case only $\langle s_1, m_1 \rangle$. It is the winner for the comprehension-based optimization starting from s_1 and for the production-based optimization starting from m_1. But since in a single optimization, there can only be one winner (we know there are no ties), we now know that $\langle s_2, m_1 \rangle$ *cannot* be a superoptimal candidate! Likewise for $\langle s_1, m_2 \rangle$. They are both blocked by $\langle s_1, m_1 \rangle$ in their respective competitions. So the following picture arises:

(176)



Since two potential candidates turned out not to be actual candidates (because they fail to be superoptimal), we realize that there are no competitors left in the optimizations based on s_2 and m_2 that would block candidate $\langle s_2, m_2 \rangle$. Hence it is superoptimal itself. So, the language includes two analyses: $\langle s_1, m_1 \rangle$ and $\langle s_2, m_2 \rangle$. (In the end we know the exact candidate sets for the optimizations: they are the singleton sets of the respective winner: $\{\langle s_1, m_1 \rangle\}$ and $\{\langle s_2, m_2 \rangle\}$, respectively.)

Blutner (2000) provides examples indicating that weak bidirection is useful for deriving *partial blocking* effects in lexical pragmatics. For example, the existence of a lexeme *kill* in the lexicon leads to a special interpretation of the expression *cause to die*: it is usually interpreted as involving an indirect chain of causation (e.g.: 'Black Bill caused the sheriff to die: he caused the sheriff's gun to backfire by stuffing it with cotton'). *kill* corresponds to s_1, *cause to die* is the less economical expression s_2. *kill* is interpreted as a direct causation (m_1), so the more complex indirect-causation interpretation (m_2) is paired up with *cause to die*.

Strong bidirection would only predict strict blocking, i.e., one would expect that *cause to die* could not be used as it is blocked by *kill*.

The definition of weak bidirectionality

Transferred to the terminology of this book, we get the following definition of a weak bidirectional OT system.[108]

(177) *The weak bidirectional OT system*

$$O_{prod}(\mathcal{O}) = \{\langle T_j, \Phi_j\rangle \in L(G_{inviol}) \mid \exists \Phi_{in} : \langle T_j, \Phi_j\rangle \in \mathit{Eval}_{\langle \mathcal{C}, \gg_{\mathcal{L}}\rangle}(O_{compr}(\mathcal{O}) \cap \{\langle T', \Phi'\rangle \in L(G_{inviol}) \mid \Phi_{in} \sqsubseteq \Phi'\}\ )\}$$

$$O_{compr}(\mathcal{O}) = \{\langle T_j, \Phi_j\rangle \in L(G_{inviol}) \mid \exists w : \langle T_j, \Phi_j\rangle \in \mathit{Eval}_{\langle \mathcal{C}, \gg_{\mathcal{L}}\rangle}(O_{prod}(\mathcal{O}) \cap \{\langle T', \Phi'\rangle \in L(G_{inviol}) \mid w \text{ is the terminal string of } T'\}\ )\}$$

$$L_{2/weak}(\mathcal{O}) = O_{prod}(\mathcal{O}) \cap O_{compr}(\mathcal{O})$$

O_{prod} and O_{compr} are auxiliary languages specifying the production-based optima and the comprehension-based optima. What is crucial is the interdependence of the candidate sets: optimization in the production-based direction is performed on comprehension-based optima which have the same input f-structure; and vice versa, comprehension-based optimization is performed on production-based optima with the same string. Ultimately, the language generated by a weak bidirectional system is the intersection of the two auxiliary notions.[109]

[108] Without making the system sensitive to context, as discussed in sec. 5.2.2, the application of such a system to the syntactic domain is not very satisfactory. For comparability, I nevertheless keep to the format of the definitions given so far, i.e., ignoring context-dependence of the analyses.

[109] Concerns that the notion of weak bidirectionality may not be well-defined, due to the apparent circularity, are refuted by Jäger (2002c). A recursive definition is possible, since the harmony relation, on which the application of *Eval* is based, is transitive and well-founded (Jäger, 2002c, sec. 2). Intuitively, the recursive computation has to start at the "base cases"—the most harmonic candidates, and OT's harmony relation guarantees that we can identify them (they are the winners of strong bidirection).

Jäger also proves that the results of Frank and Satta (1998), Karttunen (1998)—that in regular-language based OT systems, the entire system is a rational relation—carry over to weak bidirectionality.

So far, there has been little empirically oriented work elaborating the details of such an account. But the concept seems very appealing for the derivation of various aligment scales (cf. also Beaver (2000)). On the other hand, as Beaver (2003) argue, there is an overgeneration problem with weak bidirectionality: at least in the finite case, every form is ultimately paired up with some meaning. So it is unclear how ungrammticality can be modelled in this architecture.

In the following I discuss null subjects as an example from syntax making use of elements of weak bidirectionality. The analysis could not be modelled in a grammaticality/preference model or in strong bidirectionality. (It is however possible to model it using the comprehension-based/production-based sequence of optimization, which does not suffer from the overgeneration problem just mentioned.)

5.3.4 Deriving recoverability through bidirectional optimization

Null subjects

Recall from sec. 3.2.3 the discussion of null subjects in Italian, based on the OT analysis of Grimshaw and Samek-Lodovici (1998) and Samek-Lodovici (1996). As the example in (178) illustrates in some more detail, the subject is dropped in Italian when it is coreferent with the topic of the preceding sentence: in (178a), *Gianni* is the subject and thus the topic, so in (178b), no pronoun appears. In (179) however, *Gianni* is the focus or within the focus of the first sentence (the topic being *la mostra* 'the exhibition'). So (179b) is ill-formed with a null subject.

(178) a. Questa mattina, Gianni$_i$ ha visitato la mostra.
this morning G. has visited the exhibition

b. Più tardi, ϵ_i/?egli$_i$/??lui$_i$ ha visitato l'università.
more late (he)/he has visited the university

(179) a. Questa mattina, la mostra è stata visitata da Gianni$_i$.
this morning the exhibition was visited by G.

b. Più tardi, *ϵ_i/egli$_i$/lui$_i$ ha visitato l'università.
more late (he)/he has visited the university

Grimshaw and Samek-Lodovici (1998) derive these facts with the three constraints (180) in an ordinary comprehension-based optimization. PARSE is a different formulation for our constraint MAX-IO. To avoid terminological confusion, I will keep to the terminology of this book, calling the constraint MAX-IO.

(180) (Grimshaw and Samek-Lodovici, 1998, 194)

a. DROPTOPIC
Leave arguments coreferent with the topic structurally unrealized. Failed by overt constituents which are coreferential with the topic.

b. SUBJECT
The highest A-specifier in an extended projection must be filled. Failed by clauses without a subject in the canonical position.

c. PARSE [MAX-IO]
Parse input constituents. Failed by unparsed elements in the input.

The ranking Grimshaw and Samek-Lodovici (1998) assume for Italian is the following:

(181) DROPTOPIC ≫ MAX-IO [or PARSE] ≫ SUBJECT

Furthermore assuming that the topicality status of the pronominal's antecent is encoded in the input, we get the following tableau for *ha cantato* ('he has sung')—as a simplified version of the second sentence in (178) (candidate b., which is maximally MAX-IO-unfaithful to the input is called the *null parse*).

(182) (Grimshaw and Samek-Lodovici, 1998, 202)

Input: $\langle \text{cantare}(x), x = \text{topic}, x = \text{lui} \rangle$	DROPTOPIC	MAX-IO [or PARSE]	SUBJECT
a. ☞ ha cantato		*	*
b.		*!*	
c. lui ha cantato	*!		
d. ha cantato lui	*!		*

Candidates c. and d. violate DROPTOPIC, since the subject pronoun is coreferent with the topic, but nevertheless realized in these candidates. Since the DROPTOPIC outranks the faithfulness constraint MAX-IO, candidate a. with the unexpressed subject pronoun is thus most harmonic.

When the subject pronoun is not coreferent with the topic (as in (179)), candidates c. and d. satisfy DROPTOPIC, so their faithfulness to the input pays off this time, and c. becomes the winner:[110]

(183) (Grimshaw and Samek-Lodovici, 1998, 203)

Input: $\langle \text{cantare}(x), x = \text{lui} \rangle$	DROPTOPIC	MAX-IO	SUBJECT
a. ha cantato		*	*
b.		*!*	
c. ☞ lui ha cantato			
d. ha cantato lui			*

For English, Grimshaw and Samek-Lodovici (1998) assume the ranking MAX-IO ≫ DROPTOPIC ≫ SUBJECT. Hence, the effect of DROPTOPIC is neutralized by the faithfulness requirement, and no null subjects are predicted.

Deriving recoverability from constraint interaction

The constraint DROPTOPIC that Grimshaw and Samek-Lodovici (1998) assume is a very effective constraint, but its formulation is also a fairly complex. It contains a condition checking the discourse status of a pronoun's antecedent—within the antecendent's sentence; and sensitive to this condition the non-realization of the argument is rewarded.

Following the discussion in chapter 2 and sec. 3.1, such a constraint poses two questions: (i) is there an independent motivation and (ii) is the constraint really primitive, given its complex logical structure? (Cf. also the commitment to logically simple constraints in Grimshaw (1998).) To answer question (i): What could be a functional motivation for leaving underlying material unexpressed? At first this seems to thwart the most basic communicative principles. But it can be reconstructed straightforwardly from economy principles. To keep utterances brief, it is essential to avoid unnecessary structure. Of course, economy can only be complied with to the degree that the content of the utterance can still be conveyed. In other words, only material that

[110]Candidate d. may only win if an additional constraint ALIGNFOCUS is assumed, ranked between MAX-IO and SUBJECT. With the subject marked as focus, only d. will satisfy this constraint and thus win.

can be easily reconstructed by the hearer should be dropped. This is exactly what the constraint DROPTOPIC specializes on: avoid the repetition of (a certain type of) material that is already contextually given. So there *is* independent motivation. But, addressing question (ii), is the constraint really a primitive one?

I would like to argue that in a bidirectional optimization model, the interaction of communicative forces just discussed as a way of motivating DROPTOPIC can be captured more perspicuously as an instance of constraint interaction, based on simpler constraints. As an additional point, this derivation of the effect of DROPTOPIC is to demonstrate the independence of this interaction from the faithfulness constraint MAX-IO (or PARSE). Even with a low-ranking MAX-IO, we can get a language that does not allow null subjects. This is important since otherwise, we would have very limited possibilities of explaining ellipsis phenomena in non-*pro-drop*-languages as an effect of constraint interaction (involving MAX-IO too).

The economy of expression principle can be formulated as a constraint *XP (184). With this constraint, the basic pattern favouring null subjects is very simple (185) (note the low ranking of MAX-IO).

(184) *XP
Avoid structure. Violated by maximal projections.

(185)

Input: $\langle \text{cantare}(x), x = \text{lui} \rangle$	*XP	SUBJECT	MAX-IO
a. ☞ [$_{\text{IP}}$ ha [$_{\text{VP}}$ cantato]]	**	*	*
c. [$_{\text{IP}}$ [$_{\text{NP}}$ lui] ha [$_{\text{VP}}$ cantato]]	***!		

Candidate a. wins over c. simply because it contains less structure. For English, we would have SUBJ $\gg$ *XP, so c. would win. This pattern alone certainly does not give us the correct results. It predicts that the null parse (182b) is even more harmonic than a., and it does not predict any difference between the topic context (178) and the non-topic context (179).

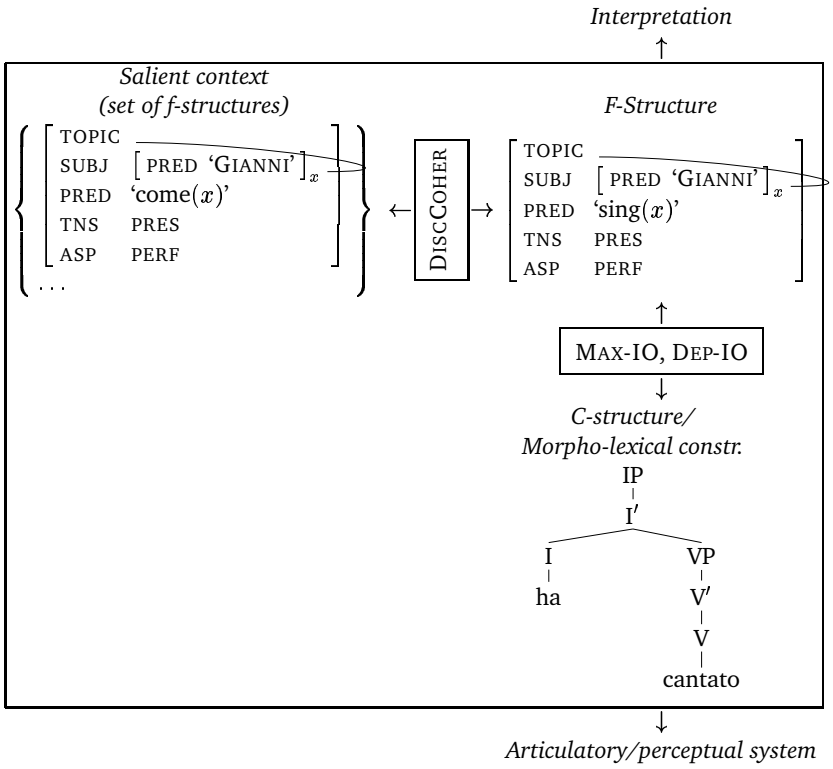
The intuition is that material can only be dropped as long as it is contextually recoverable. One way of formalizing this intuition would be to limit the candidate generation function $Gen_{G_{inviol}}$ accordingly, assuming a *recoverability principle*. Then the null parse would no longer

appear in the tableaux, and the two tableaux (182) and (183) would differ in that a. would be missing from the latter one.

However, the recoverability effects can be explained by assuming a comprehension-based optimization which complements the production-based part just sketched. (Subsequently, I will focus on the problem of topic/non-topic context, leaving aside the null parse problem; once the context problem is solved, a generalization to the ull parse problem follows.)

To make the bidirectional optimization explicit, we need an architecture with a representation of context (compare the diagram in (186)).

(186) *Architecture for a simple context-dependent OT model*



For making the case, I will here assume a very simple context representation: a set of the f-structures of the sentences in the salient context. Of course, what is actually required is some discourse semantic representation, for instance a discourse representation structure (cf. Kamp

(1981, 1990); Kamp and Reyle (1993)). It is essential that we have discourse referents for the individuals that can be referred to, rather than syntactic representations of surface phrases. But to avoid notational overhead, I will pretend that we can use the PRED value 'Gianni' in the f-structure as a discourse referent that is anchored to the individual the speaker and hearer know by the name Gianni.

As a further simplification I will assume that the feature TOPIC in the f-structure represents exactly the relevant information-structural notion (which is not quite true for LFG's actual discourse function TOPIC—this is the grammaticalized version of the information-structural notion).

A discourse coherence constraint DISCCOHER checks the f-structure of the currently uttered sentence against the salient context. Finding a good formulation for this constraint is not trivial, but for our purposes, the very simple formulation in (187) one will do.[111] In Italian, DISCCOHER is ranked above *XP.[112]

(187) DISCCOHER
For atomic f-structure values in the current f-structure, the current f-structure and an f-structure from the salient context set contain the same feature path.

$$\textit{atomic-f-str}(\star) \rightarrow \exists f, P.[\textit{f-str}(f) \wedge (f \text{ CURRENT } P) = \star \ \wedge \ (f \text{ CONTEXT} \ni P) = \star]$$

Coreference with topic: null subject

Consider the following simplified version of dialogue (178):

(188) a. Gianni è venuto.
G. has come
b. Ha cantato
has sung

[111] Just like the faithfulness constraints MAX-IO and DEP-IO, DISCCOHER should probably be seen as a familiy of constraints, parametrized for particular features.

The formula in the constraint description language assumes that the current f-structure is represented under a feature CURRENT and the set of contextual f-structures under CONTEXT.

[112] Presumably this ranking must hold universally, but it should not be required to stipulate this fact; a re-ranking would lead to an absurd system, which should be precluded by the learning scheme. '$\ni$' is used to reference one set member (rather than distributing the information over the entire set).

We want to derive that in this context, the null subject in the second sentence is correct. In a symmetrical bidirectional system, we have to check the following: is *ha cantato* the optimal way of expressing the underlying content *sing*(*Gianni*)? And vice versa, is *sing*(*Gianni*) the optimal interpretation of the string *ha cantato*? Ignoring the null parse at this point, this turns out to be straightforward in the given context. The relevant optimizations are shown in (189) and (190) below.

(189) *Production-based optimization*

Context	*Input F-Structure*	DISCCOHER	*XP	SUBJECT	MAX-IO
{ [TOPIC ; SUBJ [PRED 'GIANNI']$_x$; PRED 'come(x)' ; TNS PRES ; ASP PERF] ... }	[TOPIC ; SUBJ [PRED 'GIANNI']$_x$; PRED 'sing(x)' ; TNS PRES ; ASP PERF]				
a. ☞ [$_{IP}$ ha [$_{VP}$ cantato]]		*	**	*	*
b. [$_{IP}$ [$_{NP}$ lui] ha [$_{VP}$ cantato]]		*	***!		

For the production-based optimization (189), DISCCOHER takes no effect, since both the f-structure input and the context are fixed, and so all candidates incur the same DISCCOHER violations (every informative utterance will violate discourse coherence to a certain degree). Thus, as in (185), economy gives us *ha cantato* as the winner.

In comprehension-based optimization (190), different interpretations of the string *ha cantato* compete. For the syntactic markedness constraints (*XP and SUBJECT), these candidates will all have the same constraint profile, since the string is fixed.[113] We do get differences in the MAX-IO violations, as candidate c. shows: this candidate assumes that the string is elliptical for *Maria claimed that Gianni has sung*. So the amount of material left unrealized in the c-structure/l-structure varies with the interpretation assumed as underlying in the candidates. Note however that the number of violations of DISCCOHER is decisive for the outcome. Since *Gianni* is the topic in the context, candidate a. incurs fewest violations of DISCCOHER and is thus the winner in (190).[114]

[113]Generally, of course different syntactic analyses are possible for a given string, but here we may assume that all analyses other than [$_{IP}$ ha [$_{VP}$ cantato]] are wildly more marked.

[114]Strictly speaking, the candidate set contains also candidates with a more heavily unfaithful c- and l-structure. This way *ha cantato* can in fact be combined with an f-structure that is 100% coherent with respect to the context. Given the high ranking

(190) *Comprehension-based optimization*

Context / *Input String*	DISCCOHER	*XP	SUBJECT	MAX-IO
{ TOPIC, SUBJ [PRED 'GIANNI'], PRED 'come(x)', TNS PRES, ASP PERF, ... } ha cantato				
a. ☞ TOPIC, SUBJ [PRED 'GIANNI']$_x$, PRED 'sing(x)', TNS PRES, ASP PERF	*	**	*	*
b. TOPIC, SUBJ [PRED 'MARIA']$_x$, PRED 'sing(x)', TNS PRES, ASP PERF	**!	**	*	*
c. SUBJ [PRED 'MARIA']$_u$, PRED 'claim(u, v)', COMP [TOPIC, SUBJ [PRED 'GIANNI']$_x$, PRED 'sing(x)', TNS PRES, ASP PERF]$_v$	**!*...	**	*	***

Taking together the two directions of optimization (189) and (190), we get the clear prediction that *ha cantato* is the correct form in the context under discussion.

Coreference with non-topic: overt subject

Let us now investigate the alternative context, with *Gianni* not being the topic in the previous sentence. We want to predict that a null subject is unacceptible in that context.

(191) a. Questa mattina, la mostra è stata visitata da Gianni.
this morning the exhibition was visited by G.

b. *ϵ/lui ha cantato
(he)/he has sung

of DISCCOHER such a candidate is the actual winner of this optimization. However, as soon as the opposite optimization direction is considered, the irrecoverability of them becomes obvious. I ignore these candidates in this exposition, since it is more illustrative to demonstrate the irrecoverability with more intuitive examples.

We get (roughly) the following context representation in (192). Note that *la mostra* 'the exhibition' is marked as the topic.

(192)
$$\left\{\begin{array}{l} \left[\begin{array}{ll} \text{TOPIC} & \\ \text{SUBJ} & \left[\text{PRED} \quad \text{'EXHIBITION'}\right]_y \\ \text{PRED} & \text{'visit}(x,y)\text{'} \\ \text{OBL} & \left[\text{PRED} \quad \text{'GIANNI'}\right]_x \\ \text{TNS} & \text{PRES} \\ \text{ASP} & \text{PERF} \end{array}\right] \\ \ldots \end{array}\right\}$$

This time, the assumption of a weak bidirectional model proves essential. Recall that the candidate sets in weak bidirectionality are defined by mutual reference to the opposite concept of optimality. Abstractly, one should think of all dependencies as applying simultaneously. But if we want to verify the predictions we are forced to proceed in some sequence. This means that initially, we can only work with *preliminary* candidate sets, which are potentially too large since some candidates may not actually be optimal in the opposite direction (in fact, very many of them will not).

Let us start with comprehension-based optimization for the string *lui ha cantato*:

(193) *Comprehension-based optimization*—preliminary candidate set

Context	*Input String*	DISCCOHER	*XP	SUBJECT	MAX-IO
$\left\{\begin{array}{l} \left[\begin{array}{ll} \text{TOPIC} & \\ \text{SUBJ} & \left[\text{PRED 'EXHIBITION'}\right]_y \\ \text{PRED} & \text{'visit}(x,y)\text{'} \\ \text{OBL} & \left[\text{PRED 'GIANNI'}\right]_x \\ \text{TNS} & \text{PRES} \\ \text{ASP} & \text{PERF} \end{array}\right] \\ \ldots \end{array}\right\}$	lui ha cantato				
a. ☞ $\left[\begin{array}{ll} \text{TOPIC} & \\ \text{SUBJ} & \left[\text{PRED} \quad \text{'GIANNI'}\right]_x \\ \text{PRED} & \text{'sing}(x)\text{'} \\ \text{TNS} & \text{PRES} \\ \text{ASP} & \text{PERF} \end{array}\right]$		**	***		
b. $\left[\begin{array}{ll} \text{SUBJ} & \left[\text{PRED} \quad \text{'MARIA'}\right]_u \\ \text{PRED} & \text{'claim}(u,v)\text{'} \\ \text{COMP} & \left[\begin{array}{ll} \text{TOPIC} & \\ \text{SUBJ} & \left[\text{PRED} \quad \text{'GIANNI'}\right]_x \\ \text{PRED} & \text{'sing}(x)\text{'} \\ \text{TNS} & \text{PRES} \\ \text{ASP} & \text{PERF} \end{array}\right]_v \end{array}\right]$		***!...	***		**

The direction of optimization

Since the subject is realized, there is a smaller space of possibilities for filling out the missing part (the coreferent element has to be third person masculine). And although due to the different context, the discourse coherence is now lower than it was in (190), we get candidate (193a.)—*sing*(*Gianni*)—as a clear winner.[115]

We might now expect that the production-based optimization will confirm *lui ha cantato* as optimal for *sing*(*Gianni*). This is not the case however. Without the specially tailored DROPTOPIC constraint, there is nothing substantial that makes competition (194) (for the non-topic context) different from (189) above (for the topic context).

(194) *Production-based optimization*—preliminary candidate set

Context	*Input F-Structure*	DISCCOHER	*XP	SUBJECT	MAX-IO
$\left\{\begin{bmatrix} \text{TOPIC} & \\ \text{SUBJ} & \begin{bmatrix}\text{PRED 'EXHIBITION'}\end{bmatrix}_y \\ \text{PRED} & \text{'visit}(x,y)\text{'} \\ \text{OBL} & \begin{bmatrix}\text{PRED 'GIANNI'}\end{bmatrix}_x \\ \text{TNS} & \text{PRES} \\ \text{ASP} & \text{PERF} \\ \ldots & \end{bmatrix}\right\}$	$\begin{bmatrix} \text{TOPIC} & \\ \text{SUBJ} & \begin{bmatrix}\text{PRED 'GIANNI'}\end{bmatrix}_x \\ \text{PRED} & \text{'sing}(x)\text{'} \\ \text{TNS} & \text{PRES} \\ \text{ASP} & \text{PERF} \end{bmatrix}$				
a. ☞ [$_{\text{IP}}$ ha [$_{\text{VP}}$ cantato]]		**	**	*	*
b. [$_{\text{IP}}$ [$_{\text{NP}}$ lui] ha [$_{\text{VP}}$ cantato]]		**	***!		

Had we assumed a strong bidirectional model, this would be a problem. Since the two optimizations do not agree on a single f-structure/string pair, the underlying input f-structure would be predicted to be ineffable under the ranking for Italian.

In a weak bidirectional model we are not finished yet however. We do not know yet whether we were initially using the correct candidate sets for the two optimizations. So we have to check whether (194a) really should have been in the candidate set for the respective underlying input f-structure (i.e., *sing*(*Gianni*)). For this to be the case, *sing*(*Gianni*) would have to be optimal for the string of (194a)—*ha cantato*—in the present context. This comprehension-based competition is checked in (195).

[115] The same proviso as in footnote 114 on page 164 applies.

(195) *Comprehension-based optimization*—preliminary candidate set

Context	*Input String*	DISCCOHER	*XP	SUBJECT	MAX-IO
$\left\{\begin{bmatrix} \text{TOPIC} & \\ \text{SUBJ} & [\text{PRED 'EXHIBITION'}]_y \\ \text{PRED} & \text{'visit}(x,y)\text{'} \\ \text{OBL} & [\text{PRED 'GIANNI'}]_x \\ \text{TNS} & \text{PRES} \\ \text{ASP} & \text{PERF} \end{bmatrix} \ldots\right\}$	ha cantato				
a. ☞ $\begin{bmatrix} \text{TOPIC} & \\ \text{SUBJ} & [\text{PRED 'EXHIBITION'}]_x \\ \text{PRED} & \text{'sing}(x)\text{'} \\ \text{TNS} & \text{PRES} \\ \text{ASP} & \text{PERF} \end{bmatrix}$		**	**	*	*
b. $\begin{bmatrix} \text{TOPIC} & \\ \text{SUBJ} & [\text{PRED 'GIANNI'}]_x \\ \text{PRED} & \text{'sing}(x)\text{'} \\ \text{TNS} & \text{PRES} \\ \text{ASP} & \text{PERF} \end{bmatrix}$		***!	**	*	*
c. $\begin{bmatrix} \text{TOPIC} & \\ \text{SUBJ} & [\text{PRED 'MARIA'}]_x \\ \text{PRED} & \text{'sing}(x)\text{'} \\ \text{TNS} & \text{PRES} \\ \text{ASP} & \text{PERF} \end{bmatrix}$		***!	**	*	*

This competition starts out similar to (190), but note the difference in the context. For this reason, *Gianni* is no longer the most coherent choice for filling out the subject. It is more coherent to use the topic of the previous sentence as the subject, i.e., the exhibition. So we have found an instance of irrecoverability: in the present context, *sing*(*exhibition*) is optimal for *ha cantato* and vice versa.[116] This shows that (194a) is not contained in the true candidate set for (194), according to weak bidirectional optimization and we get b. as the winner of the corrected tableau:

[116]This means that the actual prediction for the data in (191) and (179) is the following: the null subject version of the second sentence (in both examples) is not syntactically ungrammatical, but it has the semantically anomalous reading in which the exhibition is the agent of an action. This seems a reasonable account of the data.

(196) *Production-based optimization*—corrected candidate set

Context	*Input F-Structure*	DISCCOHER	*XP	SUBJECT	MAX-IO
$\left\{ \begin{array}{ll} \text{TOPIC} & \\ \text{SUBJ} & [\text{PRED 'EXHIBITION'}]_y \\ \text{PRED} & \text{'visit}(x,y)\text{'} \\ \text{OBL} & [\text{PRED 'GIANNI'}]_x \\ \text{TNS} & \text{PRES} \\ \text{ASP} & \text{PERF} \\ \ldots & \end{array} \right\}$	$\left[\begin{array}{ll} \text{TOPIC} & \\ \text{SUBJ} & [\text{PRED 'GIANNI'}]_x \\ \text{PRED} & \text{'sing}(x)\text{'} \\ \text{TNS} & \text{PRES} \\ \text{ASP} & \text{PERF} \end{array} \right]$				
b. ☞ $[_{\text{IP}}\ [_{\text{NP}}$ lui$]$ ha $[_{\text{VP}}$ cantato$]]$		**	***!		

Hence we have a match between the two directions of optimization (193) and (196): in the non-topic context, *lui ha cantato* is predicted to be the correct form for *sing*(*Gianni*)).

Note finally, that although the null subject analysis incurs a MAX-IO violation, this constraint is neither involved in deriving the right context effects, nor in the cross-linguistic contrast with non-*pro-drop*-languages like English (English follows if SUBJECT outranks *XP). This is desirable since it leaves space for ellipsis analyses in non-*pro-drop*-languages.

The argumentation in this section showed several things:

- the constraint set assumed by Grimshaw and Samek-Lodovici (1998) for the analysis of null subjects can indeed be simplified by moving to a bidirectional optimization model: the complex formulation of the DROPTOPIC constraint is no longer required;
- in a bidirectional optimization account, the interaction between constraints and the directions of optimization derive the concept of *recoverability*, which is usually assumed as a meta-restriction on syntactic analyses;
- optimization based on a fixed context permits a fine-grained account of the empirical data, while at the same time working with highly general constraints (using the same constraint set for both directions);
- finally, the notion of *weak* bidirectionality (as opposed to the straightforward strong model) is crucial for such an account.[117]

[117] As pointed out at the end of sec. 5.3.3, it is however possible to model the facts using the comprehension-based/production-based sequence of optimization.

5.3.5 Ineffability and strong vs. weak bidirectionality

As pointed out briefly in sec. 5.3.2, a bidirectional OT model provides a solution to the puzzle that Language-Particular Ineffability poses to the OT approach (cf. sec. 3.3.3) without recurring to the assumption of LF-unfaithful candidates (as Legendre et al. (1998) do).[118] Recall that LF-unfaithful candidates are problematic from the point of view of learnability (sec. 3.3.4).

Ineffability and strong bidirectionality

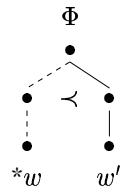
With strong bidirectionality, the mechanism that makes a certain logical form ineffable in a particular language is quite straightforward: only meaning-form pairs that are optimal in both directions are defined to be included in the language of the OT system. Now, suppose the optimal candidate for a certain underlying representation Φ has surface string w. But in the comprehension-based optimization of the alternative candidate analyses with string w, we get a different winner: some candidate with the underlying form Φ' ($\Phi' \neq \Phi$). This means that Φ is ineffable in that language. One may say that this meaning is irrecoverable for the hearer since she must always assume that when a speaker utters w he means Φ'. In other words, Φ' blocks Φ, making Φ ineffable. We can illustrate this by the following schematic picture (following a visualization scheme of Johnson (1998)); I use the * on an f-structure representation to mark ineffability and $\prec$ for the less-harmonic-than relation:

(197)

Note that ungrammaticality and ineffability are exact mirror images in a strong bidirectional model: Ungrammaticality of a string w results if the optimal among all analyses for a string w has the underlying form Φ, but in production-based optimization for Φ we get an optimal candidate with a different surface string w' ($w' \neq w$). So, w' blocks w, making w ungrammatical.

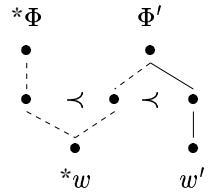
[118] This application of bidirectional optimization was brought to my attention by Joan Bresnan (p.c., March 2000). The same general idea underlies the proposal by Smolensky (1998) (see also the application of bidirectional OT-LFG in Lee (2001a)).

(198)



With the simplicity of blocking configurations, it is easy to see that we can have a situation where an underlying form Φ is blocked by an alternative reading Φ' of string w, which is itself ungrammatical due to another string w' for Φ'. String w may for example be a multiple *wh*-question like *Who ate what*, which is ungrammatical in Italian, with the underlying multiple *wh*-question (Φ) being ineffable. An alternative interpretation of *Who ate what*, with an indefinite object (Φ'), blocks Φ. The string w itself is blocked by a different string w'—the Italian equivalent of *Who ate anything*.

(199)

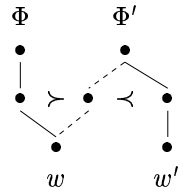


For the present purposes it is of subordinate relevance what exact constraints are involved in giving rise to the sketched harmony relation among the candidates. The following *ad hoc* interpretive constraints under ranking $C^1 \gg_{Italian} C^2$ would for instance have the intended effect of making Φ' more harmonic than Φ:

(200) a. C^1: non-topicalized pronouns get an indefinite interpretation
b. C^2: *what* is interpreted as a *wh*-word

In English, the opposite constraint ranking would lead to a different picture, with both the form-meaning pair $\langle w, \Phi \rangle$ and $\langle w', \Phi' \rangle$ being well-formed:

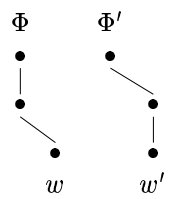
(201)



Ineffability and weak bidirectionality

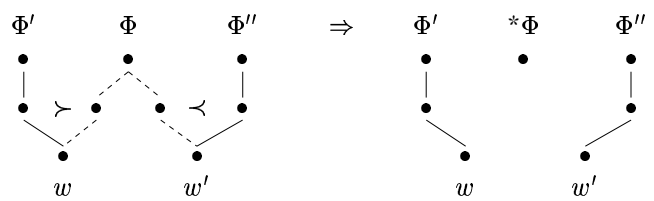
How can ineffability be modelled in the weak bidirectional model discussed in sec. 5.3.3 and 5.3.4? The direct blocking account of the strong model does not carry over, since with weak bidirectionality the unoptimal candidates of the first round may "get a second chance". As the bidirectional winners are removed from the preliminary candidate sets, losers of a strong bidirectional competition may still become winners under the weaker regime. For the picture in (199), we would for instance exclude the $\langle w, \Phi \rangle$ candidate from the preliminary candidate set of w: since we know that the optimal form for Φ, we can be sure that w cannot be its optimal form too.

(202)



But this would change the situation in such a way that $\langle w, \Phi \rangle$ is predicted to be bidirectionally optimal too—both in English and Italian. This shows that with weak bidirectionality a more sophisticated account of ineffability is required. An underlying representation will be predicted to be ineffable only if all possible realizations are bidirectionally optimal for some other representation, as sketched in the schematic picture in (203) (with preliminary candidate sets on the left and the final candidate sets—singletons—on the right). The competition for the ineffable meaning Φ ends up with an empty candidate set since all preliminary candidates turned out to be optimal for some other meaning.

(203)



Note however that all *strings* in the original preliminary candidate set for Φ are predicted to be grammatical (with some other meaning). This poses an empirical problem for ineffability examples like the Italian **Who ate what*. Of course, the context-sensitive set-up proposed in sec. 5.3.4 provides further means to derive the unacceptability of

examples—all possible contexts may be semantically anomalous, but we have to note that the strong bidirectional model allows for a more elegant account of Language-Particular Ineffability. Hence, both the strong and the weak model have conceptual and empirical advantages and disadvantages; clearly more research is required before one can decide which one should be favoured—or how a third alternative should look like, combining the advantages of both.

Interestingly, some of the computational considerations in the following chapter will depend on design decisions in the bidirectionality question too.

5.4 Summary

In this chapter I discussed various issues around the directionality question in a formal OT model of syntax. At this point it would be premature to draw any definite conclusions. A striking observation is that although the formal means make it straightforward to define a comprehension-based optimization in analogy to the standard production-based model, it is fairly intricate to judge whether the behaviour of such a system is in line with linguistic intuitions. Future work has to clarify the character of the individual optimizations and their combination further.

Sec. 5.3.4 showed however that the context-oriented application of a weak bidirectional model gives rise to a very promising account of syntactic phenomena in context. The analysis uses a set of well-motivated constraints with a simple structure, deriving the recoverability condition on syntactic structures as an effect of constraint interaction. On the other hand, sec. 5.3.5 revealed advantages of a strong bidirectional OT model in the derivation of Language-Particular Ineffability. The decision between the two types of bidirectionality has to be considered open and will require further research.

6

Computational OT Syntax

In this chapter, I address the computational side of the OT syntax formalization. The main issues, to be reviewed in sec. 6.1, are the infinity of the candidate set and directionality of processing. As we have seen in sec. 4.2.2, unrestricted OT systems are undecidable in the general case; i.e., it may be undecidable whether a given string is in the language generated by such a system. Here, I will argue however that the conceptually and empirically well-motivated formalization of chapter 4 provides a sufficiently restricted basis for a computational account. In sec. 6.2, I show that decidability results on LFG generation can be extended to the OT-LFG generation model; in sec. 6.3 I discuss the consequences for the other processing tasks.

It is important to note that the computational considerations in this chapter are no attempt to specify a *psycholinguistic* processing model of the cognitive comprehension and production capabilities. By design, the OT concepts of candidate set generation and harmony evaluation are abstract devices which are intended to model universal aspects of the language faculty at a symbolic level. The OT hypothesis is that the constraint ranking system captures the linguistically relevant aspects for a high-level model of typology, grammaticality and learnability. The OT architecture abstracts away from particularities of implementation and processing, for which one should ultimately assume a complex connectionist architecture (compare e.g., Smolensky (2000)).

Thus, a computational model for the high-level, symbolic OT system should not (necessarily) be taken to make any predictions for human comprehension or production—many different implementations result-

ing in the same input-output mapping are conceivable. A rigorous computational account of the abstract symbolic system is nevertheless very important to ensure that the symbolic framework is well-behaved. It will also provide the basis for computational systems which can be used to test the empirical predictions of non-trivial OT grammar fragments.

6.1 Processing issues for OT-LFG

Given the formal OT-LFG system defined in chapter 4 and the generalization to the other direction of optimization discussed in chapter 5, can one devise computational procedures for tasks like recognition, parsing and generation? Here, I will mainly look at the individual unidirectional models, addressing the question of bidirectional models only in passing (but I will come back to bidirectionality in sec. 6.3). Due to the "directionality" (production-based vs. comprehension-based optimization) inherent to the abstract definition of OT systems, some of the processing tasks are intuitively simpler for a certain type of OT models. Let us start considerations with the generation task for a production-based optimization system: given an underlying form (an input f-structure), what is the optimal candidate according to an OT system? I will call this task (A1).

The initial idea of how to approach this task is quite obvious: we could follow the definition of the OT-LFG system illustrated in (145) on page 123, using standard LFG processing techniques: (i) generate candidate analyses from the input f-structure, using the LFG grammar G_{inviol} 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:

(204)

$$\begin{array}{lll} \textit{generation} & \textit{marks} & \\ & \langle T_1, \Phi_1 \rangle \longrightarrow \langle n_{1,1}, n_{2,1}, \ldots, n_{k,1} \rangle & \\ & \langle T_2, \Phi_2 \rangle \longrightarrow \langle n_{1,2}, n_{2,2}, \ldots, n_{k,2} \rangle & \\ \Phi_{in} & \ldots \longrightarrow \ldots & \\ & \langle T_j, \Phi_j \rangle \longrightarrow \langle n_{1,j}, n_{2,j}, \ldots, n_{k,j} \rangle & \end{array} \Bigg\} \begin{array}{l} \textit{Eval}_{\mathcal{L}} \\ \\ \longrightarrow \langle T_{opti}, \Phi_{opti} \rangle \end{array}$$

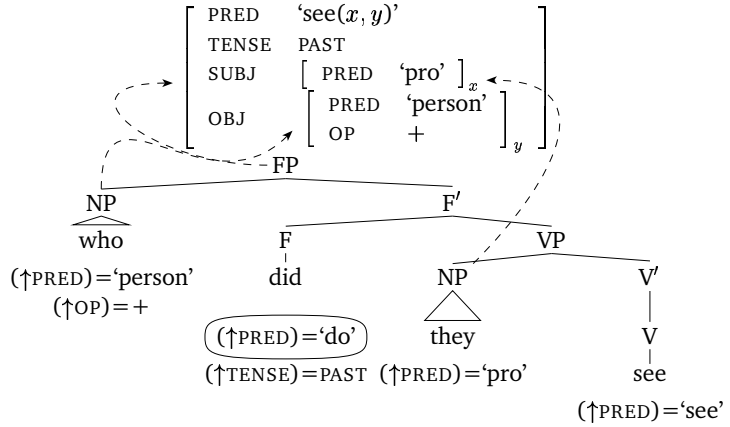
The parsing task with a comprehension-based optimization system is symmetrical (see (152) on page 129). I will call this task (B1). For a computational account the same set-up suggests itself, only starting with a string and applying standard LFG parsing rather than generation as the initial step.

Are these obvious computational approaches possible with the system formalized in chapter 4? It is clear that the naive sequential procedure just sketched will fail if there is an infinite set of candidates: step (i) can never be finished, so the procedure will not ever progress to step (ii). One of the crucial issues is whether there is some way of exploring the relevant part of the candidate set without generating an infinite number of losing candidates.

6.1.1 Infinite candidate sets in processing

For each of the two directions of optimization, one of the faithfulness constraints gives rise to an infinite number of candidates and thus poses a processing issue (when violated). For convenience, (134) and (136) are repeated as examples of unfaithful analyses (details of the λ-projection based representation of sec. 4.5.2 are irrelevant here).

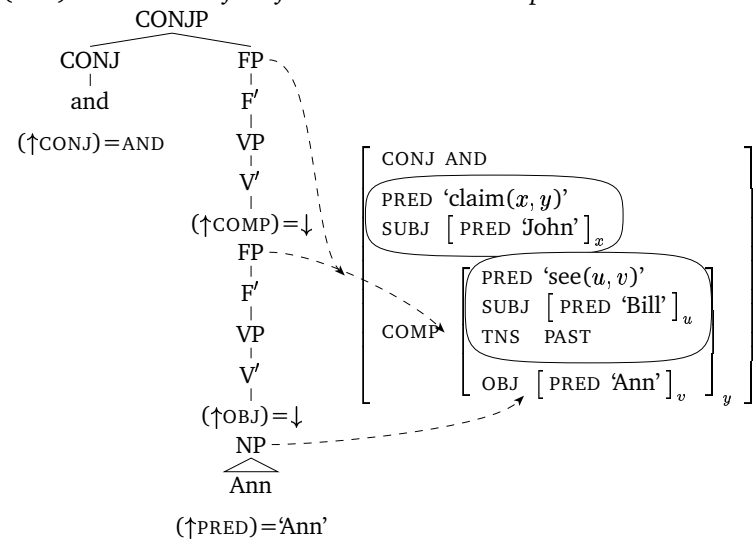
(134) *Violation of* DEP-IO



DEP-IO violations—epentheses—like (134) can create an infinite number of possibilities for generation with a production-based optimization (task (A1)): As the notation using the general category symbol FP for the functional projection suggests (cf. footnote 87 on page 113), the c-structure rules in G_{inviol} allow us to stack arbitrarily many functional projections in an extended projection. Furthermore, the f-structure contribution of all of them can be unified, since we can ignore the PRED values (under a DEP-IO violation). In other words, candidate [$_{\text{FP}}$ who did [$_{\text{FP}}$ did [$_{\text{VP}}$ they see]]] is available too for the same input f-structure as in (134), etc. So, according to the OT intuitions we indeed get infinitely many candidates. And this is cer-

tainly true for any input. Note that in the (A1) generation task MAX-IO violations—deletions—like (136) are no problem, since there is only a finite amount of information in the input to be deleted.

(136) MAX-IO-*unfaithful candidate in an ellipsis account*



For the (B1) task—parsing with a comprehension-based optimization—the situation is the mirror image. MAX-IO violations create an infinite number of possibilities, as suggested by candidate (136), which may be used in an ellipsis analysis. Here, DEP-IO violations are unproblematic, as there is only a finite number of string elements that could have been inserted.

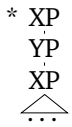
(Against the background of the context-oriented setup of a bidirectional OT system discussed in sec. 5.3.4, one expects that deep stacking of repeated MAX-IO violations is empirically more relevant than repeated DEP-IO violations. But there should be a general way of deriving the effects from constraint interaction.)

Given that the formalization of OT syntax is based on the LFG formalism, why does the infinity problem not arise in classical LFG parsing and generation? Here, the details in the interpretation of G_{inviol} as a "formal LFG-style grammar" (sec. 4.2.1 and 4.2.2) come into play. The bare LFG formalism—the 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 struc-

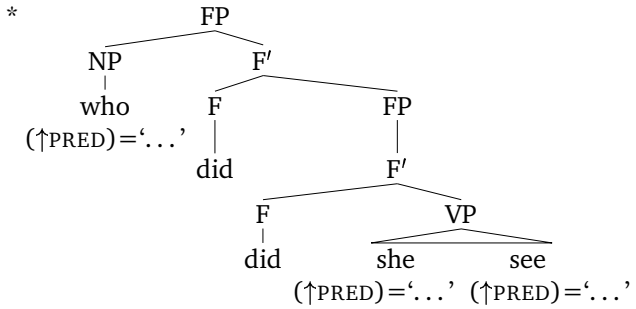
tures over a given terminal string: if the context-free grammar contains a rule recursion that can be used in a non-branching way, there are arbitrarily many c-structures, including zero to n recursions. To ensure decidability of the parsing task, such recursions are excluded in LFG by definition: the 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 (205a), which was briefly discussed in sec. 4.1 on page 64).

For the classical generation task (see Wedekind (1995, 1999), Kaplan and Wedekind (2000)), there is a parallel issue to be taken care of: in a unification-based framework, the same feature information can arise from arbitrarily 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, vacuous is used with reference to resourced feature structures (cf. the example in (205b), assuming that *did* does not introduce a resourced PRED value).[119]

(205) a. *Parsing*



b. *Generation*



[119]The use of an offline generability condition has been explored by the XLE group at Xerox PARC (John Maxwell, p.c., July 2000). The idea is that recursive parts of the grammar may only be passed if some resourced—or instantiated—feature (e.g., a PRED value) is introduced along the way.

Now, coming back to OT processing, there is a choice:

- One could decide to inherit the favourable computational properties of the standard LFG model and define G_{inviol} to be an LFG grammar in the narrow sense, including the offline parsability and offline generability condition. So, in certain cases the computed candidate set would be restricted *a priori*, where the formal OT model actually predicts restriction through constraint interaction. Note that this does not exclude the possibility of faithfulness violations altogether. It only excludes repeated faithfulness violations based on recursion using the same rule(s). So, empirically this restriction might not have a negative effect (in the sense of excluding candidates that should be winners)—at least for the DEP-IO violations in the (A1) task.

  For all practical purposes involving the implementation of non-trivial fragments, this is presumably the adequate choice, in particular as it means that existing parsing/generation systems like XLE[120] can be applied straightforwardly.

- Alternatively, G_{inviol} could be defined as a more relaxed formal grammar than a classical LFG grammar. The limiting effect of constraint interaction on the degree of unfaithfulness would then have to be reflected in the processing system too. This computational account would meet the methodological principle (25) of constraint interaction as *the* explanatory device also in the processing approach. Thus this choice is superior on theoretical and aesthetic grounds. For practical application on a larger scale, it is presumably less adequate.

  To be able to control the constraint profile of an infinite set of candidates, one needs an approach with interleaved candidate generation and constraint checking. The sequential set-up (204) cannot work, since the first step would not terminate. Such a procedure, using a chart for generation and parsing, was proposed in Kuhn (2000b, 2001a),[121] and the construction in sec. 6.2 builds on this insight.

Before exploring the second option in more detail, a second processing issue should be addressed.

[120] `http://www.parc.xerox.com/istl/groups/nltt/xle/`

[121] The chart-based parsing algorithm is also described in section 6.4 of (Kuhn, 2001b), which was not included in this book.

6.1.2 Directionality in processing

Independent of the decision on the formal character of G_{inviol}, discussed at the end of the previous section, there is a further processing issue that either of the approaches has to address. This is introduced in the current subsection.

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 other important tasks. With (A1), we have the generation task for production-based optimization, but the respective parsing and recognition task is still missing: we want a procedure telling us whether a given *string* is contained in the string language defined by the OT model (the recognition task). We may also want to know what the correct structure for this string is, which is what (Johnson, 1998, sec. 4) calls the *universal parsing task*. [122] Recognition and parsing (for production-based optimization) are closely related. Let us refer to both tasks as (A2).

According to the definition of the language generated by an OT system ((87) on page 70), the (A2) tasks amount to checking whether there is *some* underlying input representation for which the string is the optimal candidate.[123] In parsing, we are typically interested in all possible analyses, so the task is to find *all* underlying input representations which have the given string as the terminal string of the optimal candidate.

At this point it is probably already clear that the (A2) task is crucially different from (B1)—although both tasks are parsing tasks (for more discussion see Johnson (1998) and (Kuhn, 2001c, sec. 4.1)): in

[122] Johnson's definition is:

(i) *The universal parsing problem for OT-LFG:*
Given a phonological string s and an OT-LFG G as input, return the input-candidate pairs $\langle i, c \rangle$ generated by G such that the candidate c has phonological string s and c is the optimal output for i with respect to the ordered constraints defined in G.

[123] For comprehension-based optimization (B), the parallel task—(B2)—is 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.

the formal model underlying (B1), comprehension-based optimization, it is the string that defines the candidate set; hence, when we are given a particular string, it is straightforward to compute the corresponding candidate set $Gen_{G_{inviol}}(w)$ by parsing with G_{inviol} and then apply harmony evaluation on the analyses. In contrast to this, the formal model underlying the (A2) task is production-based. So, grammaticality can only be determined with an f-structure-based candidate set. When we are given a string w, we first have to find underlying input f-structures Φ_j which could have defined the candidate sets $Gen_{G_{inviol}}(\Phi_j)$ that made an analysis with string w optimal. So, although (A2) is the parsing task for production-based OT, the crucial step is a generation step, very much like in (A1): the generation from underlying f-structures gives us the set of candidates on which the constraints are applied.

How do we arrive at the possible underlying f-structures Φ_j (for which the generation task will be computed)? Since input f-structure subsume candidate f-structures it suffices to look for those analyses in G_{inviol}, which have string w as their terminal string. From these candidates' f-structures we can extract the relevant input information. So, given our string w we are looking for the set $\{\langle T_j, \Phi_j \rangle \in L(G_{inviol}) \mid w \text{ is the terminal string of } T_j\}$. This is in fact a parsing task quite similar to the initial step of (B1), but recall that it will be followed by (A1)-style "backwards" generation steps to complete the (A2) parsing task.

The scheme in fig. 1 on page 182 (taken from Kuhn (2001c)) illustrates the entire (A2) process graphically for an abstract example (parsing the string 'a b c'). Note that a given string may have no or many grammatical readings. Quite obviously, a computational procedure for task (A2) along these lines involves processing in two directions, so one may call it a *bidirectional processing* approach.[124] However one should not confuse this with the concept of bidirectional *optimization* models. Note that optimization is applied only in one direction. One could of course apply another optimization at the very end of the procedure sketched in fig. 1, comparing the strings that survive the comparison step (v). This extended procedure would then model the bidirectional grammaticality/preference model of sec. 5.3.1, sketched in (170) on page 148. So rather interestingly, the implementation of this simple bidirectional model adds nothing to the complexity of the processing task—neither does it reduce the complexity. If a strong bidirectional

[124]Compare the implementation discussed in Kuhn (2000b, 2001a), which uses Earley deduction parsing and generation, following Neumann (1994, 1998), Varges (1997).

model (as sketched in (173) on page 152) is implemented, the processing task actually becomes simpler: one can apply comprehension-based optimization already during the parsing step (i), so only the optimal candidate's underlying form has to be considered for step (iii)-(v). (For instance, the parse $A_{p,1}$ in fig. 1 may be optimal for step (i), then (ii), (iii) and (iv) would only have to be applied to this candidate, and failure of the comparison (v) would tell us immediately that the string parsed is not well-formed according to strong bidirectional optimization.)

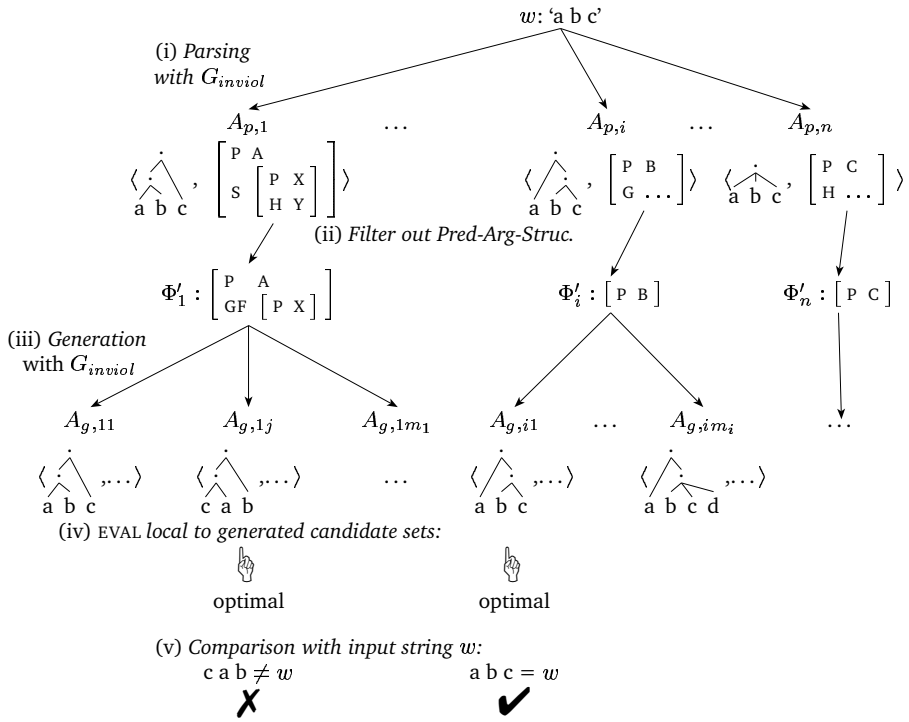
(206) *Processing steps for fig. 1 on page 182*
- (i) parse string to determine possible candidates with the string
- (ii) filter out the part of the candidates' f-structure that reflects the input
- (iii) "backward generation" from input f-structures
- (iv) harmony evaluation: optimal candidate in a generation-based competition determines grammaticality
- (v) string in optimal candidate has to match the initial string; otherwise, initial string is not grammatical for this particular underlying input f-structure

But let us go back to the (A2) task, i.e., parsing for a unidirectional, comprehension-based optimization model. To close off this section let us come back to the issue of unfaithful candidates discussed in sec. 6.1.1. There I noted that only DEP-IO violations are problematic for the (A1) task. So one might have thought that production-based optimization can be modelled computationally without worrying about unpleasant side-effects of MAX-IO violations (e.g., for null subjects or ellipsis).

But implementing (A1) alone is not sufficient, since this task does not tell us for a given string whether it is contained in the language of the OT system; the recognition/parsing task (A2) *is* required. And with (A2) we *do* get the issue of MAX-IO violations for the comprehension-based optimization model too. Formally, there is an infinite set of possible candidates for each given string, so step (i) from fig. 1 may not terminate.

Of course the problem does not arise if G_{inviol} is defined as a strict LFG grammar, since then offline parsability guarantees that the set of analyses for a given string is finite. With the alternative choice however (unrestricted G_{inviol}), the situation is more intricate. In the direction

FIGURE 1 *Task (A2) as parsing and "backward generation"*



of optimization (generation for (A1) and (A2)), we have the possibility of exploiting constraint interaction for controlling the degree of unfaithfulness. This possibility is missing for the opposite direction, i.e., parsing in (A2). This issue is discussed in more detail in the following section.

6.2 Decidability of OT-LFG generation

In this section, I show that the OT-LFG formalization established in chapter 4 provides a sufficiently restricted basis for a computational account. Sec. 6.2.1 reviews the result of Kaplan and Wedekind (2000), which establishes that the result of standard LFG generation (from a given fully specified f-structure) can be constructed as a context-free grammar. In sec. 6.2.2, I exploit this result to show that the generation for a production-based OT-LFG system (task (A1) from sec. 6.1) is decidable. Sec. 6.3 will discuss the consequences for the remaining processing tasks.

6.2.1 Generation with LFG grammars

The result of Kaplan and Wedekind (2000)—that LFG generation produces context-free languages—constitutes the crucial ingredient for the decidability argument for OT-LFG I will make in the following section. Before going into Kaplan and Wedekind's construction, I will briefly review some of the longer-established results on decidability of generation with LFG grammars.

Decidability of LFG generation from fully specified f-structures

Wedekind (1995) proves the decidability of generation from *fully* specified f-structures; in contrast, Wedekind (1999) presents a proof that generation from f-structures which are only partially specified is undecidable in the general case (see also Dymetman (1991)). However, as (Wedekind, 1999, 280f) discusses briefly, it seems that undecidability arises only with certain non-linguistic grammars in which "semantic" features are used that are structurally unrelated to the remaining f-structure representations and where arbitrarily large portions of recursive f-structure can trigger a single "semantic" feature distinction. The use of features for encoding underlying semantic forms in natural language grammars is more limited: the semantic features will always correspond to some non-recursive portion of other f-structure information in a way defined by the grammar. This property will guarantee

that generation from such a semantic feature structure, lacking a full f-structure specification, is decidable after all: the size of the unknown full f-structure corresponding to a given semantic representation is bounded by the size of the semantic representation multiplied with a constant factor determined by the grammar; thus, in adding f-structure features to the underspecified feature structure, only a bounded space of structural possibilities has to be checked against the grammar.

What remains to be established is a straightforwardly checkable criterion that a grammar has to satisfy in order to be sure that the semantic features are well-behaved.[125] A safe (though possibly overrestrictive) criterion is to allow underspecification only for those input features which have a finite range of values. From such mildly underspecified input f-structures it is very easy to get to generation from fully specified input f-structures: filling in all combinations of possible values for the unspecified features leads to a finite set of f-structures; so for each of these, generation from fully speficied f-structures can be performed. The overall result is then the union of all individual generation results.

The Kaplan and Wedekind (2000) paper is also based on the assumption that no underspecification beyond this mild type is allowed in generation from f-structures. (Anticipating the application in OT-LFG candidate generation, note that this restriction is generally unproblematic: as argued at length in chapter 4, all candidates share the same interpreted part of f-structure, which directly reflects the input. Thus, only morphosyntactic features may be added in candidate generation—an example might be the declension class feature of adjectives in German. Note that such features generally have a finite range of values.[126])

[125] As discussed in sec. 6.1.1, an offline generability condition to this effect is being explored in ongoing work (see footnote 119 on page 177 in particular).

[126] Of course, l-structure (sec. 4.5.2) has to be taken into account too. Under a DEP-IO violation, it is possible that l-structure material occurs in a candidate that is not contained in the input f-structure. Note however that such additions can only occur in lexical insertion. Thus the range of possibilities is restricted by the size of the lexicon. Since each lexicon entry has a finite set of "lexical constraints" (i.e., lexical f-annotations), and there is a finite lexicon, the set of possible l-structure extensions for a given f-structure is (large but) finite. So the decidability of candidate generation is not affected. (Note that this means that productive word formation has to be triggered by the input f-structure, so it can also be based on a finite set of building blocks. Furthermore, it has to be assumed that l-structures introduced by copies of the same lexical item are unified.)

LFG generation as the construction of a context-free grammar

While the general decidability results for LFG generation have been established for some time, Kaplan and Wedekind (2000) provide a new perspective on the generation problem, which proves very fruitful for the understanding of candidate generation and constraint interaction in OT-LFG.

In an LFG generation task, we are given an LFG grammar, including a lexicon, and a particular ("input") f-structure. The problem is to find the set of strings s such that the grammar assigns an LFG analysis $\langle T, \Phi \rangle$ to s, with s the terminal string of T and Φ the "input" f-structure[127] that is being generated from. The innovation of Kaplan and Wedekind (2000) is to move attention somewhat away from the input f-structure, focusing on the LFG grammar that has to be able to provide an analysis for the strings we are interested in. What is effectively performed in their reconstruction of the generation task is a specialization of this LFG grammar with the outcome of getting a grammar that will not accept arbitrary strings from the original language, but only the strings that have the input f-structure as their f-structure. What makes this reconstruction so interesting is that the resulting specialized grammar is no longer a full-fledged LFG grammar (in which case the enterprise would have been quite useless), but rather a simple context-free grammar. A formal precondition for this is that the input f-structure may contain no cycles, which is a very reasonable assumption for linguistic representations.

(207) *Kaplan and Wedekind's construction*
Given an arbitrary LFG grammar G and a cycle-free f-structure Φ, a context-free grammar G' can be constructed that generates exactly the strings to which G assigns the f-structure Φ.

I will occasionally refer to the resulting context-free grammar as $KW(G, \Phi)$. The (structure) language $L(KW(G, \Phi))$ is a set of trees; the string language $\mathcal{L}(KW(G, \Phi))$ is a context-free (string) language.

The context-freeness result is reached by folding all f-structural contributions of lexical entries and LFG rules into the c-structural rewrite rules. Of course, this is only possible since we know in advance the range of f-structural objects that can play a role in the derivation: they

[127] I will continue using the term "input" f-structure in the following, since it seems quite suggestive, even though we are not in an OT setting at the moment.

must all be substructures of the input f-structure, i.e., either the root f-structure itself, or partial f-structures that can be reached through some feature path from the root f-structure. There is a finite set of such substructures, and we know that every metavariable in the f-annotations of rules and lexicon entries in the original LFG grammar have to end up instantiated to one of these substructures.

So, Kaplan and Wedekind (2000) construct multiple copies of each LFG rule, each for a particular instantiation of the metavariables. While the general idea is quite straightforward, the details of the construction are rather sophisticated: as all f-annotations are removed in the final step, one has to make sure that all their restricting effects are mimicked by the context-free grammar, i.e., the c-structure symbols have to be specialized sufficiently to make all relevant distinctions. The presentation of the construction in the Kaplan and Wedekind (2000) paper is rather dense; here I attempt to point out the underlying ideas and illustrate them with examples. For the exact specification, the reader is referred to the original paper. In sec. 6.2.2, I will propose an extension of Kaplan and Wedekind's construction to show decidability of OT-LFG generation based on the formalization of chapter 4.

Category specialization I will illustrate the specialization steps with a sample grammar (208) and lexicon (209). A sample analysis illustrating the use of functional projection FP in this grammar is given in (210).

(208)

$$\text{ROOT} \rightarrow \begin{array}{c}\text{FP}\\ \uparrow=\downarrow\end{array}$$

$$\text{FP} \rightarrow \left\{\begin{array}{c|c} \begin{array}{c}\text{NP}\\ (\uparrow\text{TOPIC}) = \downarrow\\ (\uparrow\text{COMP}^* \text{ OBJ}) = \downarrow\end{array} & \begin{array}{c}\text{FP}\\ \uparrow = \downarrow\end{array} \\ \begin{array}{c}(\text{NP})\\ (\uparrow\text{SUBJ}) = \downarrow\end{array} & \begin{array}{c}\text{F}'\\ \uparrow = \downarrow\end{array}\end{array}\right\}$$

$$\text{F}' \rightarrow \begin{array}{c}\text{F}\\ \uparrow = \downarrow\end{array} \quad \left\{\begin{array}{c|c}\text{FP} & \text{VP}\\ \uparrow = \downarrow & \uparrow = \downarrow\end{array}\right\}$$

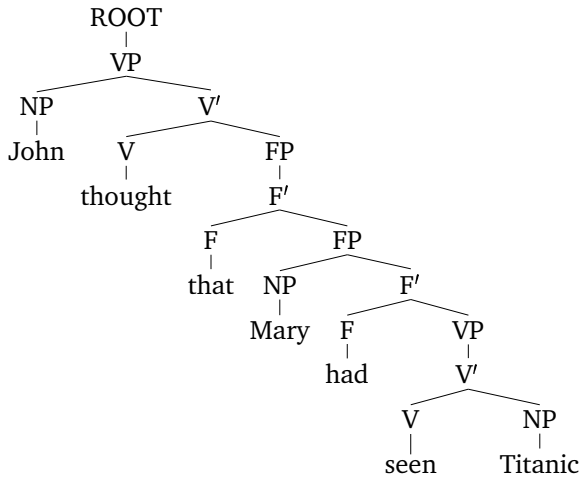
$$\text{VP} \rightarrow \begin{array}{c}(\text{NP})\\ (\uparrow\text{SUBJ})=\downarrow\end{array} \quad \begin{array}{c}\text{V}'\\ \uparrow=\downarrow\end{array}$$

$$\text{V}' \rightarrow \begin{array}{c}\text{V}\\ \uparrow = \downarrow\end{array} \quad \left\{\begin{array}{c|c}\text{NP} & \text{FP}\\ (\uparrow\text{OBJ}) = \downarrow & (\uparrow\text{COMP}) = \downarrow\end{array}\right\}$$

(209)

Mary	NP	(↑PRED)=‘Mary’ (↑NUM)=SG
John	NP	(↑PRED)=‘John’ (↑NUM)=SG
Titanic	NP	(↑PRED)=‘Titanic’ (↑NUM)=SG
that	F	
has	F	(↑TNS)=PRES (↑SUBJ NUM)=SG
have	F	(↑TNS)=PRES (↑SUBJ NUM)=PL
had	F	(↑TNS)=PAST
see	V	(↑PRED)=‘see⟨ (↑SUBJ) (↑OBJ) ⟩’
saw	V	(↑PRED)=‘see⟨ (↑SUBJ) (↑OBJ) ⟩’ (↑TNS)=PAST
seen	V	(↑PRED)=‘see⟨ (↑SUBJ) (↑OBJ) ⟩’
thought	V	(↑PRED)=‘think⟨ (↑SUBJ) (↑COMP) ⟩’ (↑TNS)=PAST
laughed	V	(↑PRED)=‘laugh⟨ (↑SUBJ) ⟩’ (↑TNS)=PAST

(210) a. c-structure



b. f-structure

$$\begin{bmatrix} \text{PRED} & \text{'think}\langle(\uparrow\text{SUBJ})\ (\uparrow\text{OBJ})\rangle\text{'} \\ \text{TNS} & \text{PAST} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{'John'} \\ \text{NUM} & \text{SG} \end{bmatrix} \\ \text{COMP} & \begin{bmatrix} \text{PRED} & \text{'see}\langle(\uparrow\text{SUBJ})\ (\uparrow\text{OBJ})\rangle\text{'} \\ \text{TNS} & \text{PAST} \\ \text{ASP} & \text{PERF} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{'Mary'} \\ \text{NUM} & \text{SG} \end{bmatrix} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{'Titanic'} \\ \text{NUM} & \text{SG} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Before the generation-specific conversion from LFG rules to context-free rules is performed, the generalized format of right-hand sides in LFG rules is converted to the standard context-free notation (as a kind of "preprocessing" step). So rules containing disjunctions and optionality brackets are replaced by a list of rules with simple category strings as right-hand sides. (Kleene stars, which do not appear in the sample grammar, are removed by introducing new category symbols and recursive rules. Disjunctions appearing within the f-annotations are also propagated up to the rule level.)

The result of this preprocessing step is shown in (211). Note that the only effect this conversion has on the resulting c-structure/f-structure analyses arises from the replaced Kleene star (and this effect can be systematically recovered from the analysis).

(211) a. ROOT → FP
$\uparrow=\downarrow$

b. FP → NP FP
$(\uparrow\text{TOPIC})=\downarrow$ $\uparrow=\downarrow$
$(\uparrow\text{COMP}^*\ \text{OBJ})=\downarrow$

c. FP → NP F′
$(\uparrow\text{SUBJ})=\downarrow$ $\uparrow=\downarrow$

d. FP → F′
$\uparrow=\downarrow$

g.	VP	→	NP (↑SUBJ)=↓	V′ ↑=↓
h.	VP	→	V′ ↑=↓	
i.	V′	→	V ↑=↓	NP (↑OBJ)=↓
j.	V′	→	V ↑=↓	FP (↑COMP)=↓
k.	V′	→	V ↑=↓	

The next conversion step depends on the particular input f-structure that is being generated from. Let us assume we want to generate from the f-structure in (210b). This means we have five substructures: the root f-structure, plus the embedded f-structures that can be reached by the paths SUBJ, COMP, COMP SUBJ, and COMP OBJ. We know that any relevant metavariable in the rules and lexicon must end up instantiated to one of these, although we do not know yet which are the correct ones. So for each path in the f-structure, from the root f-structure to some substructure, a distinct variable is introduced that can be used in the category specialization. As the variable names, we can simply use v subscripted with the (abbreviated and possibly empty) feature path from the root f-structure, so we get the variables $v, v_s, v_c, v_{cs}, v_{co}$.

As a first augmentation of the category names, the corresponding f-structure variable is added. So for the original category FP, we get FP:v, FP:v_s, FP:v_c, FP:v_{cs}, and FP:v_{co}. In principle, the rules are multiplied out to cover all possible combinations of augmented categories. However, the combinations are restricted by the original f-annotations relating the mother's f-structure projection to each of the daughters' f-structure projection (↑=↓, (↑SUBJ)=↓, etc.). For the FP:v_c variant of rule (211c), we get only NP:v_{cs} and F′:v_c, while the FP:v variant maps to NP:v_s and F′:v.

This diversification of the category symbols provides the basis for remodelling the effect of all the f-annotations in the rules and in the lexicon entries, including lexical contributions like (↑PRED)='Mary' or (↑NUM)=SG, as well as potential restrictions on rules with defining or constraining equations (not shown in the sample grammar). The diversified category symbols tell us what particular f-structure to assume as the instantiation of ↑ or ↓. For instance, in the NP:v_s variant of the lexicon entry for *Mary*, the f-annotation (↑NUM)=SG is instantiated as

$(v_s\ \text{NUM})=\text{SG}$, while in the NP:$v_{cs}$ variant, it surfaces as $(v_{cs}\ \text{NUM})=\text{SG}$.

It seems extraordinarily difficult to keep track of the interactions between these f-constraints within the c-structural skeleton, but Kaplan and Wedekind (2000) apply a very effective "trick". Rather than computing the effect of the various f-constraints at intermediate levels of the tree structure, sets of instantiated f-constraints are treated as an unanalyzed part of the category names; they are then threaded through the tree in a bottom-up fashion.

So the fully diversified category name for two instances of the lexicon entry *Mary* look as follows (now written as standard context-free rules, for lexical insertion):

(212) $$\text{NP}{:}v_s{:}\left\{\begin{array}{c}(v_s\ \text{PRED})=\text{'Mary'}\\(v_s\ \text{NUM})=\text{SG}\end{array}\right\} \rightarrow \text{Mary}$$
$$\text{NP}{:}v_{cs}{:}\left\{\begin{array}{c}(v_{cs}\ \text{PRED})=\text{'Mary'}\\(v_{cs}\ \text{NUM})=\text{SG}\end{array}\right\} \rightarrow \text{Mary}$$

The rules are again multiplied out to cover all possible combinations, with the constraint that the instantiated f-constraints of all daughters, plus the appropriately instantiated rule-specific annotations have to be unioned together to form the set of f-constraints of the mother. So, for rule (211g) based on the categories $\text{NP}{:}v_{cs}{:}\left\{\begin{array}{c}(v_{cs}\ \text{PRED})=\text{'Mary'}\\(v_{cs}\ \text{NUM})=\text{SG}\end{array}\right\}$ and $\text{V}'{:}v_c{:}\left\{\begin{array}{c}(v_c\ \text{PRED})=\text{'laugh'}\\(v_c\ \text{TNS})=\text{PAST}\\v_c = v_c\end{array}\right\}$, we would get the rule

$$\text{VP}{:}v_c{:}\left\{\begin{array}{c}(v_c\ \text{SUBJ}) = v_{cs}\\(v_{cs}\ \text{PRED})=\text{'Mary'}\\(v_{cs}\ \text{NUM})=\text{SG}\\(v_c\ \text{PRED})=\text{'laugh'}\\(v_c\ \text{TNS})=\text{PAST}\\v_c = v_c\end{array}\right\} \rightarrow \text{NP}{:}v_{cs}{:}\left\{\begin{array}{c}(v_{cs}\ \text{PRED})=\text{'Mary'}\\(v_{cs}\ \text{NUM})=\text{SG}\end{array}\right\}\ \text{V}'{:}v_c{:}\left\{\begin{array}{c}(v_c\ \text{PRED})=\text{'laugh'}\\(v_c\ \text{TNS})=\text{PAST}\\v_c = v_c\end{array}\right\}$$

The topmost f-constraint in the VP node's set arises from instantiating the $(\uparrow\text{SUBJ})=\downarrow$ annotation of the NP daughter: $\uparrow$ is instantiated as v_c (from VP:v_c), and $\downarrow$ is instantiated as v_{cs} (from NP:v_{cs}). The $\uparrow=\downarrow$ annotation of V′ in rule (211g) does not add an instantiated f-constraint that is not already present from the union of the daugthers.

With this bottom-up construction it is ensured that at the new categories corresponding to the root symbol of the original LFG grammar (i.e., categories of the form ROOT:v:{ ... }), the full collection of

instantiated f-constraints is available. The check whether or not the resulting sets of f-constraints have the input f-structure as their minimal model is effectively made external to the context-free grammar: all those variants of the original root symbol whose f-constraint union does *not* have the intended f-structure model[128] are excluded from the context-free grammar. Technically, this is done through the selective addition of rules for a new start symbol, above the original root symbol: only a small number of rules of the form

(213) ROOT′ → ROOT:v:{ . . . }

are introduced to the new context-free grammar. One reason for not introducing the ROOT′ rule for a particular variant of the original ROOT symbol is inconsistency of the collected f-constraints. For instance, the combination of a third person singular subject (like *Mary*) and the auxiliary *have* goes all the way through—up to the ROOT symbol, but then the f-constraint set contains (v SUBJ NUM)=PL from the lexical f-annotations of *have*, plus (v_s NUM)=SG from the NP, plus finally (v SUBJ)=v_s from rule (211c). This set of equations is not satisfied by any model.

But even the vast majority of consistent (and complete and coherent) sets of f-constraints fails to satisfy the criterion of having exactly the input f-structures as their minimal model. This is because in the construction, contributions from the entire lexicon and all rules are combined freely, and the check is only made at the ROOT′ level. It is obvious that the purpose of this construction is not a computationally useful approach, but rather the principled solvability result (which may increase the understanding of various algorithmic approaches).

Being able to specify the result of LFG generation as a context-free grammar has the advantage that standard techniques for context-free grammars can be applied. For example, if there are infinitely many possible strings for a given f-structure, the shortest one can be produced, based on the pumping lemma for context-free languages. (Essentially, the infinite number of possible solutions must be due to a recursion in the resulting context-free grammar; so, if the grammar is applied without allowing recursion, only a finite number of strings are generated.)

[128]The task of determining which f-constraint sets have the intended model is decidable since the set of all possible instantiated descriptions is finite (recall that the instantiation of metavariables is limited to the substructures of the input f-structure)—i.e., there is a (large but) finite number of subsets to be checked.

In the case of the sample grammar (211), we do indeed get infinitely many strings for the input f-structure. The grammar overgenerates in several respects. For instance, the functional projection FP can be stacked, and since the lexicon entry for *that* (as an F^0) does not contribute any PRED value it can be introduced over and over again. (If a PRED value was introduced, the several instances could not be unified, due to the instantiated symbol character of PRED values.[129]) The specialized context-free grammar contains recursions, based on rules (211d) and (211e), one example being the following (irrelevant morphosyntactic features are skipped in the f-constraints):

(214)
$$\text{FP}{:}v_c{:}\left\{\begin{array}{c}(v_c\ \textsc{pred})=\text{`see}\langle\ \ldots\rangle\text{'}\\(v_c\ \textsc{tns})=\textsc{past}\\(v_c\ \textsc{subj})=v_{cs}\\(v_{cs}\ \textsc{pred})=\text{`Mary'}\\(v_c\ \textsc{obj})=v_{co}\\(v_{co}\ \textsc{pred})=\text{`Titanic'}\\v_c=v_c\end{array}\right\}\rightarrow \text{F}'{:}v_c{:}\left\{\begin{array}{c}(v_c\ \textsc{pred})=\text{`see}\langle\ \ldots\rangle\text{'}\\(v_c\ \textsc{tns})=\textsc{past}\\(v_c\ \textsc{subj})=v_{cs}\\(v_{cs}\ \textsc{pred})=\text{`Mary'}\\(v_c\ \textsc{obj})=v_{co}\\(v_{co}\ \textsc{pred})=\text{`Titanic'}\\v_c=v_c\end{array}\right\}$$

$$\text{F}'{:}v_c{:}\left\{\begin{array}{c}(v_c\ \textsc{pred})=\text{`see}\langle\ \ldots\rangle\text{'}\\(v_c\ \textsc{tns})=\textsc{past}\\(v_c\ \textsc{subj})=v_{cs}\\(v_{cs}\ \textsc{pred})=\text{`Mary'}\\(v_c\ \textsc{obj})=v_{co}\\(v_{co}\ \textsc{pred})=\text{`Titanic'}\\v_c=v_c\end{array}\right\}\rightarrow \text{F}{:}v_c{:}\emptyset$$

$$\text{FP}{:}v_c{:}\left\{\begin{array}{c}(v_c\ \textsc{pred})=\text{`see}\langle\ \ldots\rangle\text{'}\\(v_c\ \textsc{tns})=\textsc{past}\\(v_c\ \textsc{subj})=v_{cs}\\(v_{cs}\ \textsc{pred})=\text{`Mary'}\\(v_c\ \textsc{obj})=v_{co}\\(v_{co}\ \textsc{pred})=\text{`Titanic'}\\v_c=v_c\end{array}\right\}$$

F:v_c:$\emptyset$ is one of the "diversified" categories we get fot the lexicon entry *that* in (209), so the context-free grammar will indeed generate an arbitrary number of *that*s on top of any FP. So for the sample f-structure in (210b), we would get the following strings (among infinitely many others):

[129]Technically, this effect is not brought out by the part of the construction described here, since it requires a special treatment of the f-annotations introducing instantiated symbols. Kaplan and Wedekind (2000) do not go into these details either, but they point out (fn. 2) that instantiated symbols have to receive a similar treatment as set-valued f-structures (which I skipped in the presentation above).

(215) a. John thought that Mary had seen Titanic.
b. John thought that that Mary had seen Titanic.
c. That John thought that Mary had seen Titanic.

The effect does not depend on the emptiness of the set of f-constraints contributed by the F^0 element: since the mother's f-constraint set is formed by set union over the daughters, adding the same f-constraints several times will not change the result either, so the following rule in the constructed context-free grammar is also effective:

(216)
$$F':v_c:\left\{\begin{array}{c}(v_c\ \text{PRED})=\text{`see}\langle\ \ldots\rangle\text{'}\\(v_c\ \text{TNS})=\text{PAST}\\(v_c\ \text{SUBJ})=v_{cs}\\(v_{cs}\ \text{PRED})=\text{`Mary'}\\(v_c\ \text{OBJ})=v_{co}\\(v_{co}\ \text{PRED})=\text{`Titanic'}\\v_c=v_c\end{array}\right\}$$
$$\rightarrow F:v_c:\left\{(v_c\ \text{TNS})=\text{PAST}\right\}\ FP:v_c:\left\{\begin{array}{c}(v_c\ \text{PRED})=\text{`see}\langle\ \ldots\rangle\text{'}\\(v_c\ \text{TNS})=\text{PAST}\\(v_c\ \text{SUBJ})=v_{cs}\\(v_{cs}\ \text{PRED})=\text{`Mary'}\\(v_c\ \text{OBJ})=v_{co}\\(v_{co}\ \text{PRED})=\text{`Titanic'}\\v_c=v_c\end{array}\right\}$$

In this case, the F^0 category is matched by the lexicon entry for *had*, leading to

(217) a. John thought that Mary had had seen Titanic.
b. John thought that Mary had had had seen Titanic.

Other choices in generation arise from the freedom of generating the subject in the specifier of VP or FP and from the possibility of (unbounded) topicalization of the object (rule (211a) contains a functional-uncertainty equation). So, we will also get the following generation alternatives:

(218) a. John thought that Titanic, Mary had seen.
b. Titanic, John thought that Mary had seen.

Note that some of these indeterminisms are due to insufficient restriction of the sample grammar (e.g., the multiple *that*s or *had*s), others could be argued to be a linguistically justified model of "real" generation alternatives (as in the topicalization case). With classical LFG

grammars it seems to be a safe assumption that vacuous recursions, as in the *that* example, are not intended by grammar writers. Hence, the strategy of not passing such cycles (or possibly passing them at most once) is sensible.[130] As we will see in the next subsection, the situation with G_{inviol} in OT-LFG is slightly different.

LFG generation in OT-LFG

We can now also think of candidate generation in OT-LFG as the construction of a context-free grammar producing the set of (terminal strings in the) candidate analyses. This means that the OT constraints have to help us determine the subset of candidates (often just a singleton) that is most harmonic with respect to the given constraint ranking. Note that in this context, it is certainly *not* justified to generally exclude recursive application of the rules. While with classical grammars, the presence of recursions could be assumed to arise from unintended underrestriction of the grammar, the lack of restriction in the G_{inviol} grammar in OT-LFG is absolutely intentional. It is the job of the constraints to add the required restrictions, but G_{inviol} has to ensure that all candidates are generated in the first place. For instance, DEP-IO violations as in *Who did you see* will arise by passing a recursion in the context-free grammar constructed during generation.

As discussed extensively in chapter 4, a candidate containing such a vacuous cycle has still the chance to become the winner of the competition. If the DEP-IO constraint is outranked by some other constraint that is violated by the non-recursive structure but satisfied by the larger, recursive structure, the resulting harmony is increased by going through the recursion a certain number of times. It is for this very reason, that *Who did you see* is predicted to be grammatical in English.

Intuitively, there seems to be an upper limit to the number of useful passes through such a recursion: the best one can do is to avoid all the constraint violations that would have otherwise arisen. Beyond this point, continued application of the recursion will not have any positive effect on harmony. This intuitive interdependence is brought out so clearly thanks to the construction of Kaplan and Wedekind (2000). In the following section, an extension of their system is presented that establishes the conditions under which this OT-constraint control over candidate generation is formally guaranteed.

[130] The "offline generability" condition employed in the XLE system is based on this strategy; cf. fn. 119 on page 177 above.

6.2.2 OT-LFG generation

The most straightforward approach to the OT-LFG scenario would probably be the following: (i) apply Kaplan and Wedekind's construction on G_{inviol} directly (i.e., given an input f-structure Φ_{in} construct $KW(G_{inviol}, \Phi_{in})$), and (ii) use the resulting context-free grammar to generate candidates and check them for constraint violations. However, it is less straightforward how generation with the context-free grammar can be restricted in a way that avoids eternal expansion of recursions (the "pumping" situation) while at the same time ensuring that all relevant candidates are generated.

The strategy I propose instead is the following: the observations of sec. 4.4.4—which suggested that a rule-local formulation of OT constraints is sufficient for modelling the linguistic intuitions behind OT syntax—are exploited to convert the LFG grammar G_{inviol} to a different form $O_{\mathcal{C}}(G_{inviol})$ (depending on the constraint set $\mathcal{C}$), which is however still an LFG grammar. Now, when Kaplan and Wedekind's construction is applied to $O_{\mathcal{C}}(G_{inviol})$, all "pumping" structures generated by the context-free grammar $KW(O_{\mathcal{C}}(G_{inviol}), \Phi_{in})$ can indeed be ignored since all OT-relevant candidates are already contained in the finite set of non-recursive structures—i.e., the structures in which no c-structure tree path from the root symbol to a terminal category contains the same nonterminal symbol more than once. So, as a final step (which is trivial in terms of decidability), the ranking of the constraints is taken into consideration in order to determine the harmony of all the candidates in this finite subset of the full candidate set and thus finally find the optimal candidate.

The conversion $O_{\mathcal{C}}(G_{inviol})$

The extra conversion of the candidate generation grammar G_{inviol} is similar in spirit to the construction that Kaplan and Wedekind apply. The category representation is augmented in order to take more fine-grained distinctions into account. In our case, the relevant distinctions arise from the number of constraint violations incurred locally. I assume that at the level of c-structure, the category representation in G_{inviol} is already fine-grained enough to encode the distinctions checked for in the OT constraints (cf. sec. 4.4.4). So, each OT constraint schema has one of the following implicational forms:

(219) a. $\begin{array}{ccc} N & \Rightarrow & N' \\ S[\star] & & S'[\star] \end{array}$

where N, N' are descriptions of nonterminal symbols of G_{inviol}; S, S' is a standard LFG f-annotation of constraining equations with $\star$ as the only f-structure metavariable (interpreted as $\phi(N)$).[131]

b. $\begin{array}{ccc} N & & N' \\ \overbrace{\rho \quad M \quad \sigma} & \Rightarrow & \overbrace{\rho' \quad M' \quad \sigma'} \\ S & & S' \end{array}$

where N, N', M, M' are descriptions of nonterminal symbols of G_{inviol}, N and N' refer to the mother in a local subtree configuration, M and M' refer to the same daughter category; $\rho, \rho', \sigma, \sigma'$ are regular expressions over nonterminals of G_{inviol}; S, S' are standard LFG f-annotations of constraining equations.

Recall from sec. 4.4.4 that I assume structured representation for c-structure categories ("complex category symbols"), i.e., the *descriptions* of nonterminal symbols may specify some properties (e.g., bar level), leaving certain other dimensions (e.g., lexical class) underspecified.

So, HEAD LEFT (in its non-scalar interpretation; compare (118) on page 94) may for example be formulated as an instance of (219b):

$$\begin{array}{ccc} \mathrm{X}^{n+1} & & \mathrm{X}^{n+1} \\ \overbrace{?^* \quad \mathrm{X}^n \quad ?^*} & \Rightarrow & \overbrace{\epsilon \quad \mathrm{X}^n \quad ?^*} \\ \uparrow = \downarrow & & \end{array}$$

X^{n+1} abbreviates 'one bar level higher than X^n'. $?^*$ is a regular expression denoting zero or more occurrences of arbitrary categories; ϵ denotes the empty string. So, if a local subtree matches the left-hand side (i.e., X^n—with annotation $\uparrow = \downarrow$—is projected to X^{n+1}), then it also has to satisfy the right-hand side (or it will violate HEAD LEFT): no category may be to the left of X^n.

As a special case, the category descriptions may be entirely underspecified. Thus the constraint OPSPEC (compare (119) on page 95) is expressed as an instance of (219a), as follows:

$$\begin{array}{ccc} \mathrm{X} & \Rightarrow & \mathrm{X} \\ (\star\ \mathrm{OP}) = + & & (\mathrm{DF}\ \star) \end{array}$$

[131] More generally, explicit reference to the category-level metavariable ($*$ in classical LFG) should be made, i.e., other projections than ϕ are also permitted.

Preprocessing Like Kaplan and Wedekind (2000), I assume that the grammar G_{inviol} has undergone an initial preprocessing step in which (i) the c-structure part of rules has been converted into standard context-free form, i.e., the right-hand side is a category string rather than a regular expression.[132] Lexical entries are transformed to standard rule format to (e.g., "V → see"). Furthermore (ii), f-structural constraint schemata which contain disjunctions have been transformed in such a way that the disjunctions are propagated to the rule level—leading to alternative rules, each with non-disjunctive f-annotations.

Format conversion (i) ensures that for a given local subtree, each constraint can be applied only a finite number of times: if l is the arity of the longest right-hand side of a rule, the maximal number of local violations is l (since some constraints of type (219b) can be instantiated to all daughters). Conversion (ii) ensures that we can keep track of the violation of f-structural constraints at the level of rules.

Grammar conversion With the number of local violations bounded, we can encode all candidate distinctions with respect to constraint violations at the local-subtree level with finite means; in particular, we can augment the category representation in such a way that the number of local violations for all constraints is explicitly represented. The set of categories in the newly constructed LFG grammar $O_{\mathcal{C}}(G_{inviol})$ is the finite set

(220) $\mathcal{N}_{O_{\mathcal{C}}(G_{inviol})}$: *the set of categories in* $O_{\mathcal{C}}(G_{inviol})$
$\{\, N{:}\langle n^1, n^2, n^3 \ldots n^k\rangle \mid N$ a nonterminal symbol of G_{inviol},
k the size of the constraint set $\mathcal{C}$,
$0 \leq n^i \leq l$,
l the arity of the longest rhs in rules of G_{inviol} $\}$

The rules in $O_{\mathcal{C}}(G_{inviol})$ are constructed that for each rule

$\underset{T_1}{\mathrm{X}_1} \ldots \underset{T_m}{\mathrm{X}_m} \rightarrow \mathrm{X}_0$ in G_{inviol},

and each sequence $\langle n_0^1, n_0^2 \ldots n_0^k\rangle$, $0 \leq n_0^i \leq l$, all rules of the form

$$\underset{T'_1}{\mathrm{X}_1{:}\langle n_1^1, n_1^2 \ldots n_1^k\rangle} \ldots \underset{T'_m}{\mathrm{X}_m{:}\langle n_m^1, n_m^2 \ldots n_m^k\rangle} \rightarrow \mathrm{X}_0{:}\langle n_0^1, n_0^2 \ldots n_0^k\rangle, \ (0 \leq n_j^i \leq l)$$

[132] In this context, I also assume that the c-structural OT constraints are adjusted to this format, i.e., the conversion of Kleene stars to rule recursions is also reflected in the constraint schemata.

are included such that n_0^i is the number of violations of constraint C^i incurred local to the rule (and the f-annotations $T'_1 \ldots T'_m$ are appropriately specified). Concretely this means[133]

(221) for C^i of form (219a) $\begin{bmatrix} N & \Rightarrow & N' \\ S[\star] & & S'[\star] \end{bmatrix}$:

a. $n_0^i = 0$; $T'_j = T_j$ $(1 \leq j \leq m)$
if X_0 does not match the condition N;

b. $n_0^i = 0$; $T'_1 = T_1 \wedge \neg S[\star/\uparrow]$; $T'_j = T_j$ $(2 \leq j \leq m)$
if X_0 matches N;

c. $n_0^i = 0$; $T'_1 = T_1 \wedge S[\star/\uparrow] \wedge S'[\star/\uparrow]$; $T'_j = T_j$ $(2 \leq j \leq m)$
if X_0 matches both N and N';

d. $n_0^i = 1$; $T'_1 = T_1 \wedge S[\star/\uparrow]$; $T'_j = T_j$ $(2 \leq j \leq m)$
if X_0 matches N but not N';

e. $n_0^i = 1$; $T'_1 = T_1 \wedge S[\star/\uparrow] \wedge \neg S'[\star/\uparrow]$; $T'_j = T_j$ $(2 \leq j \leq m)$
if X_0 matches both N and N';

(222) for C^i of form (219b) $\left[\begin{array}{ccc} & N & \\ \rho & M & \sigma \\ & S & \end{array} \Rightarrow \begin{array}{ccc} & N' & \\ \rho' & M' & \sigma' \\ & S' & \end{array}\right]$:

a. $n_0^i = 0$; $T'_j = T_j$ $(1 \leq j \leq m)$
if X_0 does not match the condition N;

b. $n_0^i = \sum_{j=1}^{m} d_j$; $T'_j = \delta(T_j, S, S')$ $(1 \leq j \leq m)$,
where

i. $d_j = 0$; $\delta(T_j, S, S') = T_j$
if X_j does not match M, or $X_1 \ldots X_{j-1}$ do not match ρ, or $X_{j+1} \ldots X_m$ do not match σ;

ii. $d_j = 0$; $\delta(T_j, S, S') = T_j \wedge S \wedge S'$
if X_0 matches both N and N'; X_j matches both M and M'; $X_1 \ldots X_{j-1}$ match ρ and ρ'; $X_{j+1} \ldots X_m$ match σ and σ';

[133] $S[\star/\uparrow]$ is defined as the f-annotation schemata S' resulting from a replacement of $\star$ in S with $\uparrow$. The f-annotations that are added in (221) refer to the mother node's f-structure. So it is arbitrary which daughter is picked for annotating them (the $\uparrow$ will refer to the mother in any of the daughters' f-annotation). Here I use the first daughter throughout.

iii. $d_j = 0$; $\delta(T_j, S, S') = T_j \wedge \neg S$
if X_0 matches both N and N'; X_j matches both M and M'; $X_1 \ldots X_{j-1}$ match ρ and ρ'; $X_{j+1} \ldots X_m$ match σ and σ';

iv. $d_j = 1$; $\delta(T_j, S, S') = T_j \wedge S$
if X_0 matches N, X_j matches M, $X_1 \ldots X_{j-1}$ match ρ, $X_{j+1} \ldots X_m$ match σ, but (at least) one of them does not match the respective description in the consequent (N', M', ρ', σ');

v. $d_j = 1$; $\delta(T_j, S, S') = T_j \wedge S \wedge \neg S'$
if X_0 matches both N and N'; X_j matches both M and M'; $X_1 \ldots X_{j-1}$ match ρ and ρ'; $X_{j+1} \ldots X_m$ match σ and σ'.

Note that for a particular combination of a rule and a constraint, several new rules can result—even with the exact same number of constraint violations. For instance, if the right-hand side of a rule (X_0) matches both the antecedent (N) and the consequent (N') category description of a constraint of form (219a), three clauses apply: (221b), (221c), and (221d). So, we get two new rules with the count of 0 local violations of the relevant constraint and two new rules with count 1. The difference lies in the f-annotations of course. The point of this construction is to make sure that all candidates will still be generated by the modified grammar; the c-structural conflicts can be detected during the construction already, whereas for the f-structural conditions, all options have to be provided in the f-annotations. They will be checked only during generation (or parsing), when a model is constructed from the f-constraints. In some cases, inconsistencies will arise from the newly added annotations, in other cases, different alternative analyses will arise.

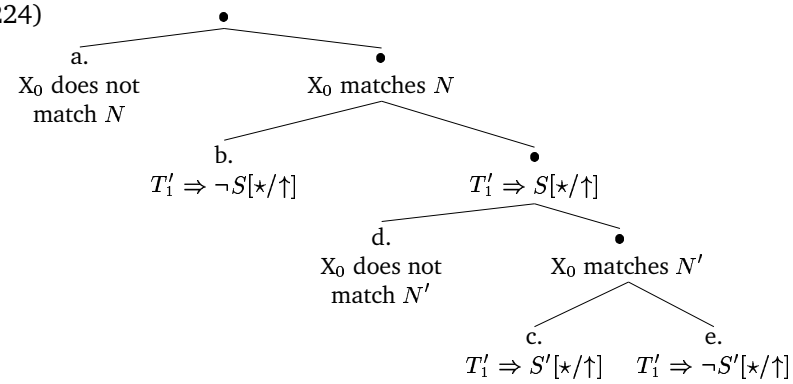
Another point to note is that the constraint profile of the daughter categories does not play any role in the determination of constraint violations local to the subtree under consideration (only the sequences n_0^i are restricted by the conditions (221) and (222); the sequences on the daughter categories n_1^i through n_m^i are unrestricted). This means that for each new rule type, all combinations of constraint profiles on the daughters are constructed. This creates a huge number of rules, but for the present purposes the only relevant aspect is that the number of combinations is finite (since the set of augmented categories is finite). Providing all possible constraint profiles for the daughters ensures that no sentence that can be parsed (or generated) by G_{inviol} is excluded

from $O_{\mathcal{C}}(G_{inviol})$ (as stated by fact (223)): whichever constraint profiles are constructed for the left-hand side of a G_{inviol} rule by (221) and (222), it is guaranteed that the respective augmented category can be used in all other rules.

(223) *Coverage preseveration*
All strings generated by a (preprocessed) LFG grammar G are also generated by $O_{\mathcal{C}}(G)$.

We have just seen that providing all possible combinations of augmented category symbols on the right-hand rule sides in $O_{\mathcal{C}}(G)$ ensures that the newly constructed rules can be reached from the root symbol in a derivation. What remains to be shown is that whenever a rule R in G contributes to an analysis, at least one of the rules constructed from R will contribute to the corresponding analysis in $O_{\mathcal{C}}(G)$. This is ensured iff the subclauses in (221) and (222) cover the full space of logical possibilities. For (221) this is easily seen: we get the following case distinction.

(224)



Disjoining the alternatives leads to a tautology in all cases, i.e., the disjunction of all new rules expresses no restrictions in addition to those already contained in the original rule. The same holds for (222) (note that the cases i.-v. in (222) are modelled after a.-e. in (221)). This proves fact (223).

The original G analysis can be recovered from a $O_{\mathcal{C}}(G)$ analysis by applying a projection function *Cat* to all c-structure categories:

(225) *Definition of Cat*
Cat$(N{:}\langle n^1, n^2 \ldots n^k\rangle) = N$ for every category in $\mathcal{N}_{O_{\mathcal{C}}(G_{inviol})}$ (220)

We can overload the function name *Cat* with a function applying to the set of analyses produced by an LFG grammar G by defining

$Cat(L(G)) = \{\langle T, \Phi\rangle \,|\, \langle T', \Phi\rangle \in L(G)$, T is derived from T' by applying *Cat* to all category symbols $\}$.

Coverage preservation of the $O_{\mathcal{C}}$ construction holds also for the projected c-category skeleton (the same proof applies):

(226) *C-structure level coverage preservation*
For a (preprocessed) LFG grammar G: $Cat(L(O_{\mathcal{C}}(G))) = L(G)$

But the analyses of $O_{\mathcal{C}}(G)$ provide additional information about the OT constraint violations: each category encodes for all the constraints how many violations of them are incurred in the local subtree it dominates. Since the constraint format is assumed to be restricted to locally evaluable constraints, *all* constraints that can be violated by a candidate analysis have to be incurred local to some subtree. Hence the total number of constraint violations incurred by a candidate can be computed by simply summing over all category-encoded local violation profiles:

(227) *Total number of constraint violations*
When $Nodes(T)$ is the bag (or multiset) of categories occurring in the c-structure tree T, then the total number of violations of constraint C^i incurred by an analysis $\langle T, \Phi\rangle \in L(O_{\mathcal{C}}(G_{inviol}))$ is
$\#_{C^i} = \sum_{N:\langle \ldots n^i \ldots\rangle \in Nodes(T)} n^i$
Define
$Total_{\mathcal{C}}(T) = \langle \#_{C^1}, \#_{C^2}, \ldots \#_{C^k}\rangle$

Applying KW on the new LFG grammar:
OT-LFG Generation Produces Context-free Languages

The grammar $O_{\mathcal{C}}(G_{inviol})$ is a standard LFG grammar, and it generates the same language as G_{inviol}. So, we can apply Kaplan and Wedekind's 2000 construction directly to $O_{\mathcal{C}}(G_{inviol})$. This produces a context-free grammar for a given f-structure Φ_{in}—what I referred to as $KW(O_{\mathcal{C}}(G_{inviol}), \Phi_{in})$ above. Note that the application of KW fixes the f-structure (which has to be the minimal model for all sets of f-constraints gathered by the context-free grammar), so the f-structural OT constraints—which were only annotated in $O_{\mathcal{C}}(G_{inviol})$ for all logically possible combinations—are now effectively checked. All rule com-

binations for which the spelled out f-structural constraint clauses lead to an inconsistency are thus filtered out by the KW construction.

After applying both $O_{\mathcal{C}}$ and KW, the category symbols now have the form X:$\langle n^1, \ldots, n^k \rangle$:$v$:$D$, with v and D arising from the KW construction. We can overload the projection function *Cat* again such that *Cat*(u:v:w:x) $= u$ for all augmented category symbol of the new format and *Cat*($L(G)$) is also defined when G is a context-free grammar.

Since the $O_{\mathcal{C}}$ construction preserves the language generated, coverage preservation holds also after the application of KW to $O_{\mathcal{C}}(G_{inviol})$ and G_{inviol}, respectively:

(228) $\mathit{Cat}(L(KW(O_{\mathcal{C}}(G_{inviol}), \Phi_{in}))) = \mathit{Cat}(L(KW(G_{inviol}, \Phi_{in})))$

But again, the context-free trees are more diverse in the analyses of the $O_{\mathcal{C}}$-based grammar. As pointed out above, they include information about violations of the constraints in $\mathcal{C}$. Since the computation of the total constraint violations (227) depends only on the augmented c-structure, it can be performed on the resulting context-free grammar as well. For example, the constraint violation information could be exploited when the context-free grammar is used to generate the actual strings: one could first try to use only rules with no constraint violations, and if that is not successful proceed to rules violating only low-ranked constraints etc.

But the crucial point for our purposes here is the following: when Kaplan and Wedekind's construction is applied on the $O_{\mathcal{C}}$-version of G_{inviol}, all *instances of recursion* in the resulting context-free grammar *create candidates that are at most as harmonic as their non-recursive counterparts*. This follows directly from the fact that all possible constraint violations are encoded in the context-free categories.

Assuming a projection function *CatCount*(u:v:w:x) $= u$:v, we can state:

(229) If T_1 and T_2 are *CatCount* projections of trees produced by the context-free grammar $KW(O_{\mathcal{C}}(G_{inviol}), \Phi_{in})$, using exactly the same rules, and T_2 contains a superset of the nodes that T_1 contains, then
$n_1^i \leq n_2^i$, for all n_1^i, n_2^i $(i = 1..k)$ from $\langle n_1^1 \ldots n_1^i \ldots n_1^k \rangle = \mathit{Total}_{\mathcal{C}}(T_1)$, and $\langle n_2^1 \ldots n_2^i \ldots n_2^k \rangle = \mathit{Total}_{\mathcal{C}}(T_2)$.

This fact follows from definition of *Total* (227): the violations counts in the additional nodes in T_2 will add to the total of constraint violations

(and if none of the additional nodes contains any local constraint violation at all, the total will be the same as in T_1). Intuitively, the effect of the augmentation of the category format is that certain recursions in the pure KW construction (which one may think of as a loop) are unfolded, leading to a longer loop. While the original loop collapsed certain distinctions relevant for the OT constraints, the new loop is sufficiently large to make all relevant distinctions.

This result can be directly exploited in processing: if all non-recursive analyses are generated (of which there are only finitely many) it is guaranteed that a subset of the optimal candidates is among them. If the grammar does not contain any *violation-free* recursion, we even know that we have generated all optimal candidates.

(230) A recursion with the derivation path $A \Rightarrow \ldots \Rightarrow A$ is called *violation-free* iff all categories dominated by the upper occurrence of A, but not dominated by the lower occurrence of A have the form $N\langle n^1, n^2 \ldots n^k\rangle$ with $n^i = 0, i = 1..k$

Note that with a violation-free recursion, the set of optimal candidates is infinite, so if the constraint set is set up properly in a linguistic analysis, one would assume that violation-free recursion should not arise. In Kuhn (2000a) and Kuhn (2000b, 2001a) I assume that the application of such recursions is excluded in the same way as offline parsability excludes vacuous recursions over a string in parsing. Hence I call this condition the *relaxed offline parsability/generability* condition:

(231) *Relaxed offline parsability/generability*
A rule recursion may only be applied if at least one constraint violation is incurred by the recursive structure.

But with Kaplan and Wedekind's 2000 construction, this condition is not necessary for making the generation task with OT-LFG systems solvable. The context-free grammar produced by KW can be transformed further to only generate the optimal candidates according to the constraint ranking $\gg_{\mathcal{L}}$ of the OT system $\mathcal{O} = \langle G_{inviol}, \langle \mathcal{C}, \gg_{\mathcal{L}} \rangle\rangle$, eliminating all but the violation-free recursions in the grammar:

(232) *Creating a context-free grammar that produces the optimal candidates*

a. Define
$\mathcal{T}^{NR}_{\Phi_{in}} = \{T \in L(KW(O_{\mathcal{C}}(G_{inviol}), \Phi_{in})) \mid T$ contains no recursion $\}$.
$\mathcal{T}^{NR}_{\Phi_{in}}$ is finite and can be easily computed, by keeping track of the rules already used in an analysis.

b. Redefine *Eval*$_{\langle \mathcal{C}, \gg_{\mathcal{L}} \rangle}$ to apply on a set of context-free analyses with augmented category symbols with counts of local constraint violations:
Eval$_{\langle \mathcal{C}, \gg_{\mathcal{L}} \rangle}(\mathcal{T}) = \{T \in \mathcal{T} \mid T$ is maximally harmonic in $\mathcal{T}$, under ranking $\gg_{\mathcal{L}}\}$
Using the function *Total* defined in (227), it is straightforward to compute for finite sets, i.e., in particular *Eval*$_{\langle \mathcal{C}, \gg_{\mathcal{L}} \rangle}(\mathcal{T}^{NR}_{\Phi_{in}})$.

c. Augment the category format further by one index component.[134] Introduce index $h = 0$ for all categories in $KW(O_{\mathcal{C}}(G_{inviol}), \Phi_{in})$ of the form X:$\langle n^1, \ldots n^k \rangle$:$v$:$D$, where $n^i = 0$ for $i = 1..k$. Introduce a new unique index $h > 1$ for each node of the form X:$\langle n^1, \ldots n^k \rangle$:$v$:$D$, where $n^i \neq 0$ for some n^i $(1 \leq i \leq k)$ occurring in the analyses *Eval*$_{\langle \mathcal{C}, \gg_{\mathcal{L}} \rangle}(\mathcal{T}^{NR}_{\Phi_{in}})$ (i.e., different occurrences of the same category are distinguished).

d. Construct the context-free grammar
$G^{OT}_{\Phi_{in}} = \langle N^{OT}_{\Phi_{in}}, T^{OT}_{\Phi_{in}}, \mathrm{S}^{OT}_{\Phi_{in}}, R^{OT}_{\Phi_{in}} \rangle$,
where $N^{OT}_{\Phi_{in}}, T^{OT}_{\Phi_{in}}$ are the indexed symbols of step c.;
$\mathrm{S}^{OT}_{\Phi_{in}}$ is a new start symbol;
the rules $R^{OT}_{\Phi_{in}}$ are (i) those rules from $KW(O_{\mathcal{C}}(G_{inviol}), \Phi_{in})$ which were used in the analyses in *Eval*$_{\langle \mathcal{C}, \gg_{\mathcal{L}} \rangle}(\mathcal{T}^{NR}_{\Phi_{in}})$—with the original symbols replaced by the indexed symbols; (ii) the rules in $KW(O_{\mathcal{C}}(G_{inviol}), \Phi_{in})$, in which the mother category and all daughter categories are of the form X:$\langle n^1, \ldots n^k \rangle$:$v$:$D$, $n^i = 0$ for $i = 1..k$ (with the new index 0 added); and (iii) one rule $\mathrm{S}^{OT}_{\Phi_{in}} \rightarrow \mathrm{S}_j$:$h$ for each of the indexed versions S_j:h of the start symbols of $KW(O_{\mathcal{C}}(G_{inviol}), \Phi_{in})$.

With the index introduced in step (232c), the original recursion in the context-free grammar is eliminated in all but the violation-free cases.

The grammar *Cat* projection of the grammar $G^{OT}_{\Phi_{in}}$ is the set of c-structures of the optimal candidate analyses for the input Φ_{in}:[135]

(233) $Cat(L(G^{OT}_{\Phi_{in}})) = \{T \mid \langle T, \Phi_{in}\rangle \in Eval_{\langle \mathcal{C}, \gg_{\mathcal{L}}\rangle}(Gen_{G_{inviol}}(\Phi_{in}))\}$, i.e., the set of c-structures for the optimal candidates for input f-structure Φ_{in} according to the OT system $\mathcal{O} = \langle G_{inviol}, \langle \mathcal{C}, \gg_{\mathcal{L}}\rangle\rangle$.

To prove fact (233) I will show that the c-structure of an arbitrary candidate analysis generated from Φ_{in} with G_{inviol} is contained in $Cat(L(G^{OT}_{\Phi_{in}}))$ if and only if all other candidates are equally or less harmonic.

Let us take an arbitrary candidate c-structure T generated from Φ_{in} with G_{inviol} such that $T \in Cat(L(G^{OT}_{\Phi_{in}}))$. We have to show that all other candidates T' generated from Φ_{in} are equally or less harmonic than T. Assume there were a T' that is more harmonic than T. Then there must be some constraint $C^i \in \mathcal{C}$, such that T' violates C^i fewer times than T does, and C^i is ranked higher than any other constrain in which T and T' differ. Constraints are restricted in such a way that violations have to be incurred within some local subtree; so T must contain a local violation configuration that T' does not contain, and by the construction (221)/(222) the $O_{\mathcal{C}}$-augmented analysis of T—call it $O_{\mathcal{C}}(T)$—must make use of some violation-marked rule not used in $O_{\mathcal{C}}(T')$. Now there are three possibilities:
(i) Both $O_{\mathcal{C}}(T)$ and $O_{\mathcal{C}}(T')$ are free of recursion. Then the fact that $O_{\mathcal{C}}(T')$ avoids the highest-ranking constraint violation excludes T from $Cat(L(G^{OT}_{\Phi_{in}}))$ (by construction step (232b)). This gives us a contradiction with our assumption.
(ii) $O_{\mathcal{C}}(T)$ contains a recursion and $O_{\mathcal{C}}(T')$ is free of recursion. If the recursion in $O_{\mathcal{C}}(T)$ is violation-free, then there is an equally harmonic recursion-free candidate T''. But this T'' is also less harmonic than $O_{\mathcal{C}}(T')$, such that it would have been excluded from $Cat(L(G^{OT}_{\Phi_{in}}))$ too. This again means that $O_{\mathcal{C}}(T)$ would also be excluded (for lack of the relevant rules in the non-recursive part). On the other hand, if it were the recursion in $O_{\mathcal{C}}(T)$ that incurred the additional violation (as compared to $O_{\mathcal{C}}(T')$), then there would be a more harmonic recursion-free

[135] Like Kaplan and Wedekind (2000), I made the assumption that the input f-structure in generation is fully specified (i.e., all the candidates have the form $\langle T, \Phi_{in}\rangle$), but the result can be extended to the case where in generation a finite amount of information can be added to the input f-structure. In this case, the specified routine has to be computed separately for each of the possible f-structural extensions and the result is compared in the end.

candidate T'''. However, this T''' would exclude the presence of $O_{\mathcal{C}}(T)$ in $L(G^{OT}_{\Phi_{in}})$ by construction step (232c,d) (only violation-free recursion is possible). So we get another contradiction to the assumption that $T \in \mathit{Cat}(L(G^{OT}_{\Phi_{in}}))$.
(iii) $O_{\mathcal{C}}(T')$ contains a recursion. If this recursion is violation-free, we can pick the equally harmonic candidate avoiding the recursion to be our $O_{\mathcal{C}}(T')$, and we are back to case (i) and (ii). Likewise, if the recursion in $O_{\mathcal{C}}(T')$ does incur some violation, not using the recursion leads to an even more harmonic candidate, for which again cases (i) and (ii) will apply.
Hence, all possible cases lead to a contradiction with the assumptions. So there can be no candidate T' that is more harmonic than our $T \in \mathit{Cat}(L(G^{OT}_{\Phi_{in}}))$.

We still have to prove that if the c-structure T of a candidate analysis generated from Φ_{in} with G_{inviol} is equally or more harmonic than all other candidates, then it is contained in $\mathit{Cat}(L(G^{OT}_{\Phi_{in}}))$. We can construct an augmented version T' of T, such that $\mathit{Cat}(T') = T$ and then show that there is a homomorphism mapping T' to some analysis $T'' \in L(G^{OT}_{\Phi_{in}})s$ with $\mathit{Cat}(T'') = T$.
We can use the constraint marking construction $O_{\mathcal{C}}$ and Kaplan and Wedekind's 2000 construction to construct the tree T' with augmented category symbols of the analysis T. Kaplan and Wedekind's result plus (228) guarantee that $\mathit{Cat}(T') = T$. Now, there has to be a homomorphism from the categories in T' to the categories of some analysis in $G^{OT}_{\Phi_{in}}$. $G^{OT}_{\Phi_{in}}$ is also based on $KW(G_{inviol}, \Phi_{in})$ (with an additional index h on each category and some categories and rules of $KW(G_{inviol}, \Phi_{in})$ having no counterpart in $G^{OT}_{\Phi_{in}}$).
Since we know that T is equally or more harmonic than any other candidate generated from Φ_{in}, we know that the augmented tree T' either contains no recursion or only violation-free recursion. If it does contain such violation-free recursions we map all categories N on the recursion paths to the indexed form N:0, and furthermore consider the variant of T' avoiding the recursion(s). For our tree, there is guaranteed to be a counterpart in the finite set of non-recursive trees in $L(G^{OT}_{\Phi_{in}})$ with all categories pairwise identical apart from the index h in $G^{OT}_{\Phi_{in}}$. We pick this tree and map each of the categories in T' to the h-indexed counterpart. The existence of this homomorphism guarantees that an analysis $T'' \in L(G^{OT}_{\Phi_{in}})$ exists with $\mathit{Cat}(T'') = \mathit{Cat}(T') = T$. QED

So it could be shown that for OT-LFG systems in which all constraints can be expressed local to a local subtree in c-structure (as discussed in sec. 4.4.4), the generation task from non-cyclic f-structures is solvable.

The infinity of the conceptually underlying candidate set does not preclude a computational approach. It is obvious that the construction proposed here has the purpose of bringing out the principled computability, rather than suggesting a particular algorithm for implementation.

6.3 Recognition and parsing for OT-LFG

As discussed in sec. 6.1.2, generation from an input f-structure and the evaluation of the most harmonic candidates solves only one of the tasks associated with a production-based optimization system—what I called (A1) above. For determining whether a given string is contained in the string language of an OT system (the recognition task (A2)) further processing is required.

This circumstance is underlined by the construction in the previous section. The crucial construction step taken over from Kaplan and Wedekind (2000), which produces the context-free grammar, presupposes a particular underlying f-structure, i.e., a particular input. When we start out with a string, it would be nice if context-free parsing techniques could be applied, but it is not clear what context-free grammar to use since the underlying f-structure is not yet known, so we do not have any context-free KW-grammar yet. Hence, the initial processing step for (A2) has to be more powerful than context-free parsing.

Essentially, we have to apply LFG parsing to determine all possible underlying input f-structures and then perform the (A1) task of ("backward") generation for each of them. In this section we have to ask: is this combined task solvable? I already addressed the problem briefly in sec. 6.1.1: with the possibility of deletion—MAX-IO violations—there is an infinite set of possible underlying f-structures. The offline parsability condition of standard LFG would guarantee that there are only finitely many f-structures for a given string, but according to OT intuitions this condition is too strong.

6.3.1 Undecidability of the unrestricted recognition problem

As the construction in sec. 6.2.2 showed, infinity issues do not generally preclude a computational approach. Generation of infinitely many generation alternatives in the context of (A1) could be controlled since they were standing in competition. The most harmonic candidate according to the constraints had to be found. However, the situation in the (A2) task—parsing with an OT system using production-based optimization to define grammaticality—is different. The different analyses

of the input string are real alternatives, without any mutual interdependence (this would be different in a comprehension-based optimization context). So there is almost no exploitable structure in the infinite set of parses for the given string.

Even with the restrictions on the OT-LFG system which guarantee that the generation problem is solvable, we can construct an undecidability argument. Modifying the second construction discussed in sec. 4.2.3 ($\mathcal{O}_2$), we get the following system:[136]

(234) *Constructed OT-LFG system (schema) $\mathcal{O}_3$*
Assume an LFG G_1 for which the emptiness problem is undecidable (i.e., it is undecidable whether $\mathcal{L}(G_1) = \emptyset$).
Construct G_1' from G_1 by replacing all occurrences of all terminal symbols in the rules (i.e., typically just the LFG lexicon entries) with the empty string ϵ. Note that $\epsilon \in \mathcal{L}(G_1')$ if and only if $\exists s$ s.t. $s \in \mathcal{L}(G_1)$. Hence, $\mathcal{L}(G_1') = \emptyset$ is also undecidable.[137]
Now construct G_{inviol} (with start symbol S) from G_1' (with start symbol S_1') by adding new nonterminal symbols Y and B, new terminals *yes* and *better*, and the following productions:

S → Y S_1'
(↑PRED)='yes' (↑OTHER)=↓
S → Y
(↑PRED)='yes'
S → B
(↑PRED)='yes'
Y → *yes*
B → *better*

Assume a single constraint (in addition to the faithfulness constraint MAX-IO which is violated whenever the G_1' rules introduce an empty string):

$C^1(\Phi, \langle T, \Phi' \rangle) = 0$ if T contains the symbol B
1 otherwise

The ranking of C^1 with respect to MAX-IO does not play a role.

Now, what happens when we want to determine whether *yes* $\in \mathcal{L}(\mathcal{O}_3)$? After (incomplete) parsing of the string *yes* with G_{inviol}, we get an analysis with the second S production, giving us the f-structure

[136] This system is again related to Johnson's 1998 sketch of an undecidability argument for OT parsing.

[137] Johnson (1988) shows the undecidability of parsing with a similar grammar by construction of a Turing Machine.

Φ_1 =[PRED 'yes']. We have to apply backward generation to this underlying form. This gives us the following production-based optimization:

(235)

Input F-Structure: [PRED 'yes']	C^1	MAX-IO
a. [$_Y$ *yes*]	*!	
b. ☞ [$_B$ *better*]		

The string *yes* is less harmonic than the alternative candidate *better*. So for the underlying representation Φ_1, the string *yes* is not a grammatical surface form. But this does not exclude that *yes* is grammatical for some other underlying representation. We can still use the first S production in the initial parsing step. The terminal symbol *yes* may be "followed" by an empty string introduced by the S'_1 symbol. If this is the case, the f-structure produced for S'_1 is introduced under the feature OTHER, so we get a different underlying representation Φ_2 and we have to perform a different backward generation task. So, in this case no alternative candidate satisfying C^1 can be generated (the only way of satisfying C^1 is by using the third S production, which provides no way of creating the f-structure material under OTHER). Thus the MAX-IO violations that the candidate(s) incur would play no role and some candidate with the string *yes* would be optimal—this means *yes* $\in \mathcal{L}(\mathcal{O}_3)$. However, in order to determine whether or not an empty string follows the symbol *yes* in the string parsed inititally the emptiness problem for G'_1 would have to be solved, which is undecidable by assumption. So it is also undecidable whether *yes* $\in \mathcal{L}(\mathcal{O}_3)$.

However again something about this problem appears to be counterintuitive. G'_1 displayed instances of massive ellipsis (the introduction of the empty string ϵ always goes along with a MAX-IO violation). And linguistically, it does not make much sense to reason about such massive ellipsis without taking the utterance context and/or world knowledge into account (recall the discussion in sec. 5.2.2). If such violations are possible in optimal candidates then they are justified by (contextual) recoverability for the hearer. To model this intuition we have to adjust the formal framework to allow for reference to context and restrict the initial parsing step by context.

Another possibility for ensuring decidability of the parsing task is to shift attention from production-based optimization models to a strong bidirectional optimization model. Both approaches are discussed in the following section.

6.3.2 Decidability of variants of the parsing task

Context as a bounding factor for analyses in parsing

In this subsection, I discuss a strategy of ensuring decidability of the recognition/parsing task by building a kind of contextual recoverability condition into the parsing step of task (A2) (with a simple production-based optimization model).

Let us assume a context representation that is input to the parsing process—in addition to the string. Like in sec. 5.3.4, we can assume a set of f-structures as the context representation. The relevant salient context for a given utterance can be safely assumed to be finite. We also need a structurally controlled way in which the contextual contribution is (re-)introduced to the f-structure of the present utterance. Here, the idea of "pseudo-lexical constraints" discussed in sec. 4.5.2 on page 120 can be used. A generalized form of the sample rule (143) for introducing such f-annotations at the level of maximal projections is (236).

(236) *Pseudo-lexical constraint introduction*

$$\mathrm{X} \rightarrow \left\{ \begin{array}{c|c} \mathrm{X} & \cdots \\ (\uparrow F) = V & \end{array} \right\}$$

for some category X, feature F and value V.

Using such a rule once goes along with one violation of MAX-IO (as in (136) on page 114). Let us assume that rules of this format are the only source of MAX-IO violations.

Decidability of parsing with classical LFG grammar is guaranteed by the offline parsability condition. Essentially, non-branching dominance chains of categories are only produced up to the point where a category would occur for the second time. As discussed in (Kuhn, 1999b, 10), this condition is too strict even for classical LFG. In order to cover all linguistically justified cases, usages of the same category name in the grammar rules with different f-annotations have to be distinguished, as formulated in (237).[138] This relaxation does not affect decidability.

[138] According to an LFG reviewer (for the LFG '99 conference), this was pointed out

(237) *Revised offline parsability condition*
A c-structure derivation is valid if and only if no category appears twice in a nonbranching dominance chain *with the same f-annotation*.

In the following I show that it is quite natural to combine the revised offline parsability with pseudo-lexical constraint introduction as in (236). As long as the pseudo-lexical f-annotations differ, it is now legal to go through a dominance chain with the same category X. Let us assume that (236) is restricted by the following contextual condition:

(238) In the analysis $\langle T, \Phi \rangle$ of a given utterance, each value V introduced in a pseudo-lexical f-annotation has to correspond to a type-identical f-structural object V' in the set of context f-structures.

For instance, if we introduce (↑PRED)='see(x, y)' as a pseudo-lexical f-annotation, there has to be an instance of 'see(x, y)' in the context representation. In the context of *Bill saw Mary*, the analysis for *and John* can thus build on correspondence with the PRED values of *Bill* and *see* in the context representation. Note that semantic forms are treated as instantiated symbols, so the two instances of 'see(x, y)' are automatically distinguished by different indices (e.g., 'see(x, y)'$_{23}$ as V' and 'see(x, y)'$_{78}$ as the newly introduced value in Φ).

I will change the notation for introducing such instantiated symbols in f-annotations. Normally, a neutral notation is used specifying just the *type* of instantiated object. In pseudo-lexical f-annotation, I propose to use the *token* of the corresponding contextual object for specification, e.g., (↑PRED)='see(x, y)'$_{23}$. This does not affect the further treatment of the f-constraint in the f-structural model construction. Once annotation schema gets instantiated (instantiating ↑ to a particular f-structure), a fresh instantiated symbol is created—say 'see(x, y)'$_{78}$ as above.[139] However, for the c-structure-related book-keeping according to revised of-

already in the early 1980's by Lori Levin, and the revised offline parsability condition is used in LFG systems like XLE.

[139] For complex f-structures introduced by pseudo-lexical f-annotations (as they are for example required to derive the subject in the example *Bill saw Mary—and John*), I have to assume a new mechanism creating a type-identical copy of the entire structure referred to in the context. Note however that the described technique of introducing new instantiated objects for semantic forms in the pseudo-lexical f-annotations is a special case of this more general copy mechanism.

Alternative solutions for dealing with the embedded f-structures would be (i) to extend the legal format for pseudo-lexical f-annotations allowing for feature paths rather

fline parsability (237), all references to the same contextual token V' are identical. This means that a non-branching chain with two references to the same contextual object are excluded.

However, if the context contains two instances of the same type of object a non-branching dominance chain may arise that contains a pseudo-lexical f-annotation for each one. This is desirable since an analysis of ellipsis should allow *and John* in the context of (239) to have an f-structure with two instances of the PRED value 'think(x, y)' recovered from the context.

(239) I thought that Mary thought that Bill saw Sue

Without further restrictions on the pseudo-lexical f-annotations, it is quite obvious that the sketched analysis of ellipsis overgenerates considerably. So for instance, one analysis of *and John* in the context of (239) would presumably have the same f-structure as the same utterance can have in the context (240):

(240) Mary thought that Bill thought that I saw Sue

However, what is relevant here is that the formal restrictions assumed are liberal enough to provide the space for a more precise formulation of a theory of ellipsis. Then, further restrictions can be formulated within the formalism provided. This seems to be the case for the proposed approach.

So, it remains to be shown that decidability of the parsing task is ensured with the new formal restrictions. Since the representation of the salient context is finite, there can only be a finite number of different contextual objects used in pseudo-lexical f-annotations. Furthermore there is a finite number of categories in the grammar. This ensures that there is an upper limit for the length of a legal non-branching dominance chain. Hence, for a given string there is only a finite number of different c-structures; for each c-structure, the number of minimal f-structure models is also finite, like in standard LFG parsing, so we can process the f-structures for each of them separately (applying backward generation according to the construction in sec. 6.2.2).

---

than just a single feature; or (ii) to permit pseudo-lexical f-annotations on categories dominating the empty string.

Controlling the set of parses with OT constraints

An alternative way of controlling the infinite set of parsing analyses transfers the OT technique used to control generation alternatives (sec. 6.2.2) to the parsing situation. Constraint interaction can distinguish between the alternatives and exclude certain cycles that are responsible for producing an infinite number of possibilities. In other words, the problem of undecidability of parsing with a production-based optimization model is resolved by moving to a strong bidirectional optimization model (cf. sec. 5.3.2).

For the OT system $\mathcal{O}_3$ (234) constructed to illustrate undecidability of the parsing problem, we would thus get an additional condition on well-formed candidates: they have to be optimal in the parsing direction too (i.e., the most harmonic analysis with the given string). So, we would look at all possible analyses of *yes*. Even not knowing the internals of the part of the grammar "under" S'_1, it is easy to see that no analysis can be constructed for *yes* that satisfies constraint C^1 (which demands the B category of *better* to be present) and that all (if any) analyses of *yes* using the production S $\rightarrow$Y S'_1 would have to incur at least one MAX-IO violation (unless $\mathcal{L}(G_1)$ already contained the empty string, which we can easily exclude by stipulating addition of an extra symbol before constructing G'_1). Hence, the candidate a. in (235)—[$_{\text{Y}}$ *yes*]—must be optimal in the comprehension-based candidate set over the string *yes*. Since the same candidate is unoptimal in the opposite optimization of (235), we know that *yes* is not contained in the language defined by the strong bidirectional version of $\mathcal{O}_3$ (i.e., *yes* $\notin \mathcal{L}_{2/strong}(\mathcal{O}_3)$).

In a certain sense, the control strategy based on constraint interaction is closer to the spirit of OT. On the other hand, to a certain degree it assumes that production-based optimization and comprehension-based optimization in syntax are mirror images and in particular that the strong bidirectional optimization approach is correct. But as the discussion in chapter 5 suggested, it is not entirely clear so far whether these assumptions are justified. Here I just present the formal considerations showing that the control strategy of applying a bidirection OT model can effectively avoid the problem of undecidability.

The $O_{\mathcal{C}}$-construction proposed in sec. 6.2.2 converts an LFG grammar into another LFG grammar with distinctions at the level of c-structure that allow for a distinction of all possible combination of local constraint violations according to constraint set $\mathcal{C}$. In combination with Kaplan and Wedekind's 2000 construction this allowed us to

generate for an input f-structure all distinct types of production-based candidates that have a chance to become the winner.

The same construction can be applied in comprehension-based optimization. The simplest case is an OT system ignoring context in the constraint evaluation—i.e., without context-sensitive constraints like DISCCOHER, assumed in sec. 5.3.4. (Of course such a system suffers from the ambiguity problem pointed out by Hale and Reiss (1998), since almost every string is predicted to have a single, unambiguous interpretation.) We can apply the $O_{\mathcal{C}}$-construction to G_{inviol} including pseudo-lexical f-annotations as in (236). So the task is to find for an input string w the set of pairs $\langle T, \Phi \rangle \in L(G_{inviol})$ that are most harmonic in the set of candidate analyses with that string (i.e., the set $\mathit{Eval}_{\langle \mathcal{C}, \gg_{\mathcal{L}} \rangle}(\mathit{Gen}_{G_{inviol}}(w))$).

If we assume a finite inventory of features and values (including the predicates in semantic forms), we get a large but finite set of possible pseudo-lexical f-annotations. So, when the $O_{\mathcal{C}}$-grammar is run with the standard LFG offline parsability condition, we can be sure that for a given string, only a finite number of c-structures is produced; hence, termination is guaranteed. For each c-structure, the f-annotations have to be evaluated in order to exclude analyses leading to inconsistencies or incompleteness/incoherence (this is part of standard LFG parsing). From the remaining candidates, the optimal one(s) can be determined by simply summing over the local violations encoded in the augmented c-structure category symbols.

(241) a. Parse string w with $O_{\mathcal{C}}(G_{inviol})$, applying standard LFG parsing (obeying the offline parsability condition): $\mathcal{A}_1(w) = \{\langle T, \Phi \rangle \in L(O_{\mathcal{C}}(G_{inviol})) \mid w$ is the terminal string/yield of $T\}$.

b. Pick the set of analyses $\mathcal{A}_{O_{\mathcal{C}}(G_{inviol})}(w)$ from $\mathcal{A}_1(w)$ with optimal (most harmonic) overall constraint profile $\mathit{Total}_{\mathcal{C}}(T)$ for $\langle T, \Phi \rangle \in \mathcal{A}_1(w)$.

The question is: does the $O_{\mathcal{C}}$-construction guarantee for parsing too that all relevant candidates are among the ones produced? We have to show that the constructed set of analyses is identical to the set of optimal candidates according to the definition (recall that we overloaded the function *Cat* appropriately to convert the augmented category symbols in an LFG analysis back to the normal c-structure):

(242) $\mathit{Cat}(\mathcal{A}_{O_{\mathcal{C}}(G_{inviol})}(w)) = \mathit{Eval}_{\langle \mathcal{C}, \gg_{\mathcal{L}} \rangle}(\mathit{Gen}_{G_{inviol}}(w))$

Contrary to the situation with generation where the assumption of violations in all vacuously recursive parts of the grammar (the "relaxed offline parsability/generability" condition (231)) turned out to dispensable, we now have to make such an assumption: recursions in the grammar that can be passed string-vacuously, i.e., creating a non-branching chain have to lead to at least one local constraint violation. The simplest way to guarantee this is to assume an Economy-of-structure constraint *STRUCT which "counts" the number of categories in an analyses and favours the candidates with fewer categories.

To show $\mathit{Cat}(\mathcal{A}_{O_{\mathcal{C}}(G_{inviol})}(w)) \subseteq \mathit{Eval}_{\langle \mathcal{C}, \gg_{\mathcal{L}} \rangle}(\mathit{Gen}_{G_{inviol}}(w))$, assume we have an analysis $\langle T, \Phi \rangle$ in the *Cat* projection of the $O_{\mathcal{C}}$-constructed outcome. We have to show that $\langle T, \Phi \rangle \in \mathit{Eval}_{\langle \mathcal{C}, \gg_{\mathcal{L}} \rangle}(\mathit{Gen}_{G_{inviol}}(w))$. Assume that it were not, then there would have to be a more harmonic competitor $\langle T', \Phi' \rangle$ in $\mathit{Gen}_{G_{inviol}}(w)$. Since all the constraints are locally expressed we can exclude that $\langle T', \Phi' \rangle$ is among the offline parsable candidates in $\mathit{Gen}_{G_{inviol}}(w)$: they are checked directly in $\mathcal{A}_{O_{\mathcal{C}}(G_{inviol})}(w)$. So $\langle T', \Phi' \rangle$ must contain at least one non-branching dominance chain with two occurences of a category X. This string-vacuous recursion must incur a constraint violation along the way (by assumption (231))—say $*C^i$. Since $\langle T', \Phi' \rangle$ is optimal, the non-branching dominance chain must avoid the violation of a higher-ranking constraint (say C^j) incurred by the simpler, competitor $\langle T'', \Phi'' \rangle$, which is the same as $\langle T', \Phi' \rangle$, apart from avoiding the recursion contained in $\langle T', \Phi' \rangle$.

The difference wrt. C^j between $\langle T', \Phi' \rangle$ and $\langle T'', \Phi'' \rangle$ must lie in the local violations for recursion-causing category X, for otherwise avoiding the cycle would definitely be more harmonic. So effectively the local constraint violation profile of the upper X in $\langle T', \Phi' \rangle$ and the single X in $\langle T'', \Phi'' \rangle$ differ. But then the $O_{\mathcal{C}}$-construct will have created different augmented symbols for the two profiles and the longer dominance chain of $\langle T', \Phi' \rangle$ will not have been filtered out in the $O_{\mathcal{C}}$-equivalent ("$O_{\mathcal{C}}(\langle T', \Phi' \rangle)$"). This gives us a contradiction with our assumptions because $O_{\mathcal{C}}(\langle T', \Phi' \rangle)$ would have been more optimal than the original analysis "$O_{\mathcal{C}}(\langle T, \Phi \rangle)$", so the construction (241) would not have included the latter in $\mathcal{A}_{O_{\mathcal{C}}(G_{inviol})}(w)$, but $O_{\mathcal{C}}(\langle T', \Phi' \rangle)$.

For the opposite direction, $\mathit{Eval}_{\langle \mathcal{C}, \gg_{\mathcal{L}} \rangle}(\mathit{Gen}_{G_{inviol}}(w)) \subseteq \mathit{Cat}(\mathcal{A}_{O_{\mathcal{C}}(G_{inviol})}(w))$, we can take the c-structure T from an arbitrary $\langle T, \Phi \rangle \in \mathit{Eval}_{\langle \mathcal{C}, \gg_{\mathcal{L}} \rangle}(\mathit{Gen}_{G_{inviol}}(w))$ and construct an augmented tree T' from it with $\mathit{Cat}(T') = T$. Then we show that $T' \in \mathcal{A}_{O_{\mathcal{C}}(G_{inviol})}(w)$. For the construction of the augmented c-structure T' from T, we apply all constraints locally at each category in the tree T and construct

the violation marks of the augmented category format accordingly (the cases in (221) and (222) can be applied to local subtrees in just the same way as they are applied to rules—the f-annotations have to be evaluated as a check on the existing f-structure model Φ).[140] Note that $Cat(T') = T$. Now, is $T' \in \mathcal{A}_{O_{\mathcal{C}}(G_{inviol})}(w)$? The augmented categories we constructed are in accordance with the $O_{\mathcal{C}}$-construction and the f-annotations were checked against the f-structure model. There cannot be any double occurrences of the same augmented category in a non-branching chain, because then the initial candidate $\langle T, \Phi \rangle$ would not have been optimal (according to a stronger version of the same argument as in the opposite direction—if there is a cycle in the augmented categories, then there must be at least one cycle in the simpler categories). So we know that T' is the c-structure of a successful LFG analysis of the string w in $O_{\mathcal{C}}(G_{inviol})$ (cf. (241a)). Since $\langle T, \Phi \rangle$ is known to be most harmonic in $Gen_{G_{inviol}}(w)$ and all constraints are locally confined, there cannot be any candidate in $\mathcal{A}_1(w)$ with a better total constraint profile than $\langle T', \Phi \rangle$, so $\langle T', \Phi \rangle$ is picked in step (241b), and thus $(Cat(T') =)T \in Cat(\mathcal{A}_{O_{\mathcal{C}}(G_{inviol})}(w))$. QED

Since the "simple" processing tasks for the production-based optimization model (generation, i.e., (A1)) and for the comprehension-based model (parsing, i.e., (B1)) could be shown to be computationally solvable, the combined tasks ((A2) and (B2)) in a strong bidirectional model are also solvable. Depending on the input representation, (A1) or (B1) can be performed first, and the second task can be applied on the output (after filtering out the relevant information, i.e., the "input part" of f-structure, or the surface string).

Weaker bidirectional models (cf. sec. 5.3.1 and 5.3.3) cannot exploit this strategy since the interdependence of the two directions of optimization is more sophisticated. In order to ensure decidability of the parsing task for them, an additional recoverability condition of the kind discussed in the previous subsection seems to be required.

Adding context-sensitive constraints in OT-controlled parsing

So far, the OT-based control of infinity of the set of parsing analyses ignored context-sensitive constraints. This means that the ellipsis analysis for utterances like *and John* used as an example throughout this book could not be modelled in the framework, since it would

[140] If there are several possibilities we construct all of them and pick the most harmonic one based on the total constraint violations.

require string-vacuous cycles of the type that is excluded (in the comprehension-based direction it will generally be more harmonic to avoid ellipsis). To make a multiply embedded ellipsis optimal, a context-sensitive constraint like DISCCOHER (sec. 5.3.4) has to be assumed, outranking the constraints that punish ellipsis. But the approach proposed here can be transferred to a set-up including a context representation. Here, I will only sketch the idea.

We can assume a finite context representation and constraints that are still local to subtrees of depth 1, but which can in addition refer to the context representation (for simplicity, I will still assume f-structures as a context representation; the relevant context is of course not confined to the *linguistic* context). The O_C-construction is then extended to depend on a particular context representation. With a fixed context, the local constraint violation profile can be computed for all rules in the grammar, again producing augmented category symbols. So, the resulting grammar can be applied in the same way as before, but the augmented set of categories is even more diverse; hence, the non-branching chains that can be constructed before a category is included twice can become considerably larger, and analyses that would have been excluded before can become winners.

Note that with the context-sensitive version of the bidirectional optimization strategy, it is only the parsing task with respect to a given context that is guaranteed to be decidable. If we are interested in the string language generated by the OT system independent of context, the extended O_C-construction does not help. I conjecture that the string recognition problem for a contextually sensitive grammar model is semidecidable for principled reasons.

If the context of a given utterance is unknown, the parsing process has to rely on heuristics for the accomodation of a certain amount of contextual information. But at some point, the process has to stop—even 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 semidecidability seems rather plausible. Recall under what circumstances candidates that are heavily unfaithful to MAX-IO (like the one in (136) on page 114) 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,

(243) 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 the type of heuristics we just discussed is at work. Undecidability of the parsing task (for the formal system specified by our—idealized—model of competence grammar) may be unavoidable.

6.4 Summary

The main results of this chapter are the following: there are two main issues in the computational processing of OT systems for syntax, the infinity of the candidates set and the directionality issue—i.e., the fact that the recognition task for production-based OT systems involves processing in the opposite direction. Based on computational results for generation with classical LFG grammars (in particular Kaplan and Wedekind (2000)) I showed that for a suitably restricted OT-LFG system the generation problem is solvable. The restrictions—locally expressed constraints and full specification of the semantic part of f-structures—are in the spirit of the underlying assumptions of OT, as discussed in the previous chapters.

The recognition/parsing task is undecidable for a production-based OT system allowing arbitrary faithfulness violations. However, there are two plausible ways of constraining the system: a recoverability condition on ellided material based on a finite context representation, or the move to a strong bidirectional optimization model. I showed decidability of the parsing problem for both these constrained systems. If in the bidirectional model context-sensitive constraints are assumed, decidability can be guaranteed only based on a known context representation. Allowing for free accomodation of context with an isolated sentence again leads to an undecidable recognition task. This

fact seems cognitively plausible given that native speakers have trouble with grammaticality judgements for non-trivial cases of ellipsis in isolation.

February 4, 2003

7

Conclusion

In this book, I proposed a formal framework for making explicit a number of different OT models for syntax. I discussed the intuitions behind the OT assumptions, identifying the requirements for a faithful rendering in the formal apparatus for syntax. Taking into account learnability considerations and following previous work on OT-LFG, I construed the candidates for OT syntax as non-derivational analyses of an underlying LFG-style grammar, where all members of a candidate are invariant in terms of the interpreted part of f-structure. Constraints can be formalized as simple schemata in a structural description language.

The LFG-based formalization provides a tool for investigations of the intricate interrelations that arise (i) when the dual of standard production-based optimization is assumed: comprehension-based (or interpretive) optimization; and (ii) when moreover the two optimization concepts are combined to form a bidirectional model. I discussed ways of isolating syntactic effects from the role that encyclopaedic knowledge and other non-syntactic sources play in disambiguation. The explanatory limitations of the straightforward strong bidirectional model were addressed, and I demonstrated that a weak bidirectional model working with a fixed context representation across candidates can lead to a simplification and clarification of the constraint set.

Finally, the computational properties of the formal models of OT syntax were discussed, identifying the possibility of arbitrary faithfulness violations and the directionality of processing as the two main challenges. The investigation of computational properties adds interesting ways of differentiating between set-ups of OT systems, in particular with respect to the bidirectionality question. Decidability of a

model with constraint-controlled faithfulness is guaranteed for a strong bidirectional model, but not for a general unidirectional or weak bidirectional model. Processing relative to a fixed context representation is decidable for the weaker models too.

Based on the OT-LFG formalization, it is easy to specify slightly simplified OT systems for syntax, using a standard parsing/generation system for LFG, like the XLE system. Such an approach takes some of the theoretically assumed power of constraint interaction away (making *Gen* somewhat overrestrictive), but for the purpose of developing and assessing a focused empirical OT account, it should practically always be sufficient. In any event, the manipulation of candidates and constraint sets with a computer is faster and less error-prone than the paper-based construction of tableaux (for which the number of candidates that can actually be worked out is considerably smaller than the candidate sets one may run through the computer).

Outlook

As discussed initially in chapter 1, one of the motivations for the enterprise of formalizing OT syntax was the prospect of a linguistic theory with a precisely defined learning scheme that one can apply on realistic empirical data. With the stochastic constraint ranking of Boersma (1998) or a similar account, the OT learning model can in particular take into account the frequency of linguistic phenomena and lexical items in the training data.

As the discussion of bidirectionality in chapter 5 showed, some aspects of the formal models still need clarification, before one can expect learning simulations on real syntactic data to be directly revealing about the syntactic constraint sets under consideration in theoretical syntactic work. The fixed-context approach seems quite promising however. It could be combined with a non-OT probabilistic estimation of the distribution of context factors.

Within the existing formal framework, there are various points that should be addressed by future research. For instance, what constraints should one assume for modelling native speakers' idiosyncratic knowledge, i.e., knowledge about collocational preferences of lexical items etc. (cf. the brief discussion in sec. 4.3.2)? Moreover, a broad range of syntactic phenomena has to be addressed under the OT perspective; with the context awareness of the weak bidirectional model (sec. 5.3.4), notorious phenomena for broad-coverage grammars like

ellipsis and coordination may await a satisfactory analysis that could be validated on corpus data.

One may hope that the theoretically informed bidirectional OT model with its detailed syntactic representations will ultimately improve the results of genuine computational-linguistic tasks like broad-coverage parsing. In the last decade, statistical models have been used and developed to permit corpus-based training of parsers. Recently, also rich syntactic representations have been applied in such models, in particular in the LFG-based accounts of Johnson et al. (1999), Riezler et al. (2000, 2002). There is an interesting parallel between the log-linear models used in the training of these probabilistic LFG grammars and the OT architecture. As Johnson (1998) points out, the comprehension-directed variant of the standard OT model[141] can be viewed as a special case of the log-linear model. The ranked violable constraints of OT (counting instances of a certain configuration in the candidate representations) are the correspondents of the "features" for which a log-linear model provides estimated parameters (these "features" or properties are also functions from configurations to numbers).

This formal parallel at the same time reveals a difference in the typical application of the models. In OT, the original motivation comes from production-based optimization (as a definition of grammaticality), while in statistical natural language processing, so far only the comprehension-based, i.e., parsing direction has been explored and used for disambiguation. The work using rich syntactic representations like LFG structures has either relied on existing classical grammars, which were manually written, or on corpora with a detailed syntactic annotation.

The bidirectional OT approach could attempt to take the scope of corpus-based learning with deep linguistic representations considerably further: both the grammaticality-defining constraint ranking and the disambiguation preferences would be learned from training data. The training data would not have to be annotated with the full syntactic target representations of the grammatical framework adopted, but only with a suitable underlying predicate-argument representation (and using bootstrapping techniques, it may be possible to keep the amount of training data required manageable; some experiments to this end are reported in Kuhn (2002)).

[141] Assuming a fixed limit for the number of multiple violations that can be incurred for the constraints.

February 4, 2003

References

Abraham, Werner. 1986. Word order in the middle field of the German sentence. In W. Abraham and S. de Meij, eds., *Topic, Focus, and Configurationality*, pages 15–38.

Aissen, Judith. 1999. Markedness and subject choice in Optimality Theory. *Natural Language and Linguistic Theory* 17:673–711.

Andrews, Avery D. and Christopher D. Manning. 1999. *Complex Predicates and Information Spreading in LFG*. Stanford, CA: CSLI Publications.

Asudeh, Ash. 1999. Linking, optionality, and ambiguity in Marathi: An Optimality Theory analysis. Ms. Stanford University. Department of Linguistics.

Beaver, David. 2000. The optimization of discourse. Ms., Stanford University.

Beaver, David. 2003. Input-output mismatches in OT. Ms., Stanford University, University of North Carolina at Chapel Hill.

Blackburn, Patrick and Edith Spaan. 1993. A modal perspective on the computational complexity of attribute value grammar. *Journal of Logic, Language, and Information* 2(2):129–169.

Blutner, Reinhard. 2000. Some aspects of optimality in natural language interpretation. Ms., Humboldt University Berlin.

Boersma, Paul. 1998. *Functional Phonology. Formalizing the interactions between articulatory and perceptual drives*. Ph.D. thesis, University of Amsterdam.

Boersma, Paul and Bruce Hayes. 2001. Empirical tests of the gradual learning algorithm. *Linguistic Inquiry* 32(1):45–86.

Bresnan, Joan. 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, Joan. 2000. Optimal syntax. In J. Dekkers, F. van der Leeuw, and J. van de Weijer, eds., *Optimality Theory: Phonology, Syntax, and Acquisition*. Oxford University Press.

Bresnan, Joan. 2001a. The emergence of the unmarked pronoun. In J. Grimshaw, G. Legendre, and S. Vikner, eds., *Optimality Theoretic Syntax*. Cambridge, MA: MIT Press.

Bresnan, Joan. 2001b. *Lexical-Functional Syntax*. Oxford: Blackwell.

Bresnan, Joan and Ashwini Deo. 2001. Grammatical constraints on variation: 'be' in the Survey of English Dialects and (Stochastic) Optimality Theory. Ms., Stanford University.

Bresnan, Joan, Shipra Dingare, and Christopher Manning. 2001. Soft constraints mirror hard constraints: Voice and person in English and Lummi. In M. Butt and T. H. King, eds., *Proceedings of the LFG 01 Conference*. CSLI Publications.

Butt, Miriam, Stefanie Dipper, Anette Frank, and Tracy Holloway King. 1999a. Writing large-scale parallel grammars for english, french, and german. In M. Butt and T. H. King, eds., *Proceedings of the LFG99 Conference, Manchester, UK*, CSLI Proceedings Online.

Butt, Miriam, Tracy King, Maria-Eugenia Niño, and Fréd'erique Segond. 1999b. *A Grammar Writer's Cookbook*. No. 95 in CSLI Lecture Notes. Stanford, CA: CSLI Publications.

Choi, Hye-Won. 1999. *Optimizing Structure in Context: Scrambling and Information Structure*. CSLI Publications.

Chomsky, Noam. 1981. *Lectures on Government and Binding*. Dordrecht: Foris.

Chomsky, Noam. 1995. *The Minimalist Program*. Cambridge, Mass.: MIT Press.

Dalrymple, Mary, ed. 1999. *Semantics and Syntax in Lexical Functional Grammar: The Resource Logic Approach*. Cambridge, MA: MIT Press.

Dalrymple, Mary, ed. 2001. *Lexical Functional Grammar*, vol. 34 of *Syntax and Semantics*. New York: Academic Press.

Dalrymple, Mary, Ronald M. Kaplan, John T. Maxwell, and Annie Zaenen, eds. 1995. *Formal Issues in Lexical-Functional Grammar*. Stanford, CA: CSLI Publications.

Dalrymple, Mary, John Lamping, Fernando Pereira, and Vijay Saraswat. 1997. Quantifiers, anaphora, and intensionality. *Journal of Logic, Language, and Information* 6:219–273.

Dalrymple, Mary, John Lamping, and Vijay Saraswat. 1993. LFG semantics via constraints. In *Proceedings of EACL, University of Utrecht*, pages 97–105.

Dingare, Shipra. 2001. *The Effect of Feature Hierarchies on Frequencies of Passivization in English*. Master's thesis, Stanford University.

Dymetman, Marc. 1991. Inherently reversible grammars, logic programming and computability. In *Proceedings of the ACL Workshop: Reversible Grammar in Natural Language Processing*, pages 20–30. Berkeley.

Eisner, Jason. 1997. Efficient generation in primitive optimality theory. In *Proceedings of the ACL 1997*. Madrid. Rutgers Optimality Archive, `http://ruccs.rutgers.edu/roa.html` (ROA-206-0797).

Ellison, Mark. 1994. Phonological derivation in optimality theory. In *COLING 1994*, pages 1007–1013. Kyoto, Japan. Rutgers Optimality Archive, `http://ruccs.rutgers.edu/roa.html` (ROA-75-0000).

Erbach, Gregor and Hans Uszkoreit. 1990. Grammar engineering: Problems and prospects. Tech. rep., Universität des Saarlandes, Computational Linguistics. CLAUS-Report Nr. 1.

Falk, Yehuda N. 2001. *Lexical-Functional Grammar: An Introduction to Parallel Constraint-Based Syntax*. Stanford: CSLI Publications.

Fanselow, Gisbert, Matthias Schlesewsky, Damir Cavar, and Reinhold Kliegl. 1999. Optimal parsing, syntactic parsing preferences, and optimality theory. Ms., Rutgers Optimality Archive, `http://ruccs.rutgers.edu/roa.html` (ROA-367-1299).

Frank, Anette, Tracy H. King, Jonas Kuhn, and John 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, Anette, Tracy H. King, Jonas Kuhn, and John Maxwell. 2001. Optimality Theory style constraint ranking in large-scale LFG grammars. In P. Sells, ed., *Formal and Empirical Issues in Optimality-theoretic Syntax*, pages 367–397. Stanford: CSLI Publications.

Frank, Robert and Giorgio Satta. 1998. Optimality theory and the generative complexity of constraint violation. *Computational Linguistics* 24(2):307–316.

Gécseg, Ferenc and Magnus Steinby. 1997. Tree languages. In G. Rozenberg and A. Salomaa, eds., *Handbook of Formal Languages*, pages 1–68. Berlin/Heidelberg: Springer Verlag.

Gerdemann, Dale and Gertjan van Noord. 2000. Approximation and exactness in finite state Optimality Theory. In *SIGPHON 2000, Finite State Phonology. Proceedings of the Fifth Workshop of the ACL Special Interest Group in Computational Phonology*. Luxembourg.

Gibson, Edward and Kevin Broihier. 1998. Optimality Theory and human sentence processing. In P. Barbosa, D. Fox, P. Hagstrom, M. McGinnis, and D. Pesetsky, eds., *Is the Best Good Enough?*, pages 157–191. MIT Press and MITWPL.

Grimshaw, Jane. 1991. Extended projection. Unpublished Manuscript, Brandeis University.

Grimshaw, Jane. 1997. Projection, heads, and optimality. *Linguistic Inquiry* 28(3):373–422.

Grimshaw, Jane. 1998. Constraints on constraints in Optimality Theoretic syntax. Manuscript, Rutgers University.

Grimshaw, Jane and Vieri 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?*, pages 193–219. MIT Press.

Haegeman, Liliane. 1994. *Introduction to Government and Binding Theory*. Oxford: Blackwell, 2nd edn.

Hale, Mark and Charles Reiss. 1998. Formal and empirical arguments concerning phonological acquisition. *Linguistic Inquiry* (29):656–683.

Hammond, Michael. 1997. Parsing syllables: modeling OT computationally Rutgers Optimality Archive, `http://ruccs.rutgers.edu/roa.html` (ROA-222-1097).

Heck, Fabian and Gereon Müller. 2000. Successive cyclicity, long-distance superiority, and local optimization. In *Proceedings of WCCFL 19*, pages 218–231.

Heck, Fabian, Gereon Müller, Ralf Vogel, Silke Fischer, Sten Vikner, and Tanja Schmid. 2000. On the nature of the input in Optimality Theory. Manuscript, University of Stuttgart.

Hendriks, Petra and Helen de Hoop. 2001. Optimality Theoretic semantics. *Linguistics and Philosohpy* 24(1):1–32.

Höhle, T. 1982. Explikation für 'normale Betonung' und 'normale Wortstellung'. In W. Abraham, ed., *Satzglieder im Deutschen*. Tübingen: Gunther Narr Verlag.

Jäger, Gerhard. 2002a. Learning constraint sub-hierarchies. the bidirectional gradual learning algorithm. Ms., University of Potsdam.

Jäger, Gerhard. 2002b. Recursion by optimization: On the complexity of bidirectional optimality theory. Ms., ZAS Berlin.

Jäger, Gerhard. 2002c. Some notes on the formal properties of bidirectional Optimality Theory. *Journal of Logic, Language, and Information* 11(4):427–451.

Johnson, Mark. 1988. *Attribute-Value Logic and the Theory of Grammar*. Stanford, CA: CSLI.

Johnson, Mark. 1998. Optimality-theoretic Lexical Functional Grammar. In *Proceedings of the 11th Annual CUNY Conference on Human Sentence Processing, Rutgers University*. To appear.

Johnson, Mark, Stuart Geman, Stephen Canon, Zhiyi Chi, and Stefan Riezler. 1999. Estimators for stochastic "unification-based" grammars. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL'99), College Park, MD*, pages 535–541.

Kager, René. 1999. *Optimality Theory*. Cambridge: Cambridge University Press.

Kamp, Hans. 1981. A theory of truth and semantic representation. In J. Groenendijk, T. Janssen, and M. Stokhof, eds., *Formal Methods in the Study of Language*.

Kamp, Hans. 1990. Prolegomena to a structural theory of belief and other attitudes. In C. A. Anderson and J. Owens, eds., *Propositional Attitudes*. CSLI Lecture Notes 20.

Kamp, Hans and Uwe Reyle. 1993. *From Discourse to Logic. Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*. Dordrecht: Kluwer.

Kaplan, Ronald M. 1995. The formal architecture of lexical-functional grammar. In M. Dalrymple, R. M. Kaplan, J. T. Maxwell, and A. Zaenen, eds., *Formal Issues in Lexical-Functional Grammar*, pages 7–27. Stanford, CA: CSLI Publications. Originally appeared in *Proceedings of ROCLING II*, ed. C.-R. Huang and K.-J. Chen (Taipei, Republic of China, 1989), 1-18.

Kaplan, Ronald M. and Joan W. Bresnan. 1982. Lexical-functional grammar: a formal system for grammatical representation. In J. W. Bresnan, ed., *The Mental Representation of Grammatical Relations*, chap. 4, pages 173–281. Cambridge, MA: MIT Press.

Kaplan, Ronald M. and John T. Maxwell. 1988. An algorithm for functional uncertainty. In *Proceedings of COLING-88*, pages 297–302. Budapest. Reprinted in Dalrymple et al. (1995), pp. 177-197.

Kaplan, Ronald M. and John T. Maxwell. 1996. LFG grammar writer's workbench. Tech. rep., Xerox PARC. `ftp://ftp.parc.xerox.com/pub/lfg/`.

Kaplan, Ronald M. and Jürgen Wedekind. 2000. LFG generation produces context-free languages. In *Proceedings of COLING-2000*, pages 297–302. Saarbrücken.

Kaplan, Ronald M. and Annie Zaenen. 1989/95. Long-distance dependencies, constituent structure, and functional uncertainty. In M. Dalrymple, R. M. Kaplan, J. T. Maxwell, and A. Zaenen, eds., *Formal Issues in Lexical-Functional Grammar*, pages 137–165. Stanford, CA: CSLI Publications. Originally appeared in *Alternative Conceptions of Phrase Structure*, ed. M. Baltin and A. Kroch (Chicago: Chicago University Press, 1989) 17–42.

Karttunen, Lauri. 1998. The proper treatment of optimality in computational phonology. In *Proceedings of the International Workshop on Finite-State Methods in Natural Language Processing, FSMNLP'98*, pages 1–12. To appear, ROA-258-0498.

Keller, Bill. 1993. *Feature Logics, Infinitary Descriptions and Grammar*. Stanford, CA: CSLI.

Keller, Frank. 2000. *Gradience in Grammar: Experimental and Computational Aspects of Degrees of Grammaticality*. Ph.D. thesis, University of Edinburgh.

Keller, Frank and Ash Asudeh. 2001. Probabilistic learning algorithms and Optimality Theory. Ms., Saarbrücken, Stanford University.

King, Tracy Holloway, Stefanie Dipper, Annette Frank, Jonas Kuhn, and John Maxwell. 2000. Ambiguity management in grammar writing. In E. Hinrichs, D. Meurers, and S. Wintner, eds., *Proceedings of the Workshop on Linguistic Theory and Grammar Implementation, ESSLLI-2000, Birmingham, UK*.

Klein, Wolfgang. 1993. Ellipse. In J. Jacobs, A. von Stechow, W. Sternefeld, and T. Vennemann, eds., *Syntax: An International Handbook of Contemporary Research*, pages 763–799. Berlin: de Gruyter.

Koontz-Garboden, Andrew. 2001. A stochastic OT approach to word order variation in Korlai Portuguese. paper presented at the 37th annual meeting of the Chicago Linguistic Society, Chicago, IL, April 20, 2001.

Kuhn, Jonas. 1995. Information packaging in German – some motivation from HPSG-based translation. Ms. Centre for Cognitive Science, University of Edinburgh.

Kuhn, Jonas. 1996a. Context effects on interpretation and intonation. In D. Gibbon, ed., *Natural Language Processing and Speech Technology. Results of the 3rd KONVENS Conference*, pages 186–198. Berlin: de Gruyter.

Kuhn, Jonas. 1996b. On intonation and interpretation in context – is there a unitary explanation for focus and deaccenting? Diplomarbeit [Master thesis], Institut für maschinelle Sprachverarbeitung, Universität Stuttgart.

Kuhn, Jonas. 1996c. An underspecified HPSG representation for information structure. In *Proceedings of COLING-96*, pages 670–675. Copenhagen.

Kuhn, Jonas. 1998. Towards data-intensive testing of a broad-coverage LFG grammar. In B. Schröder, W. Lenders, W. Hess, and T. Portele, eds., *Computers, Linguistics, and Phonetics between Language and Speech, Proceedings of the 4th Conference on Natural Language Processing – KONVENS-98*, pages 43–56. Bonn: Peter Lang.

Kuhn, Jonas. 1999a. The syntax and semantics of split NPs in LFG. In F. Corblin, C. Dobrovie-Sorin, and J.-M. Marandin, eds., *Empirical Issues in Formal Syntax and Semantics 2, Selected Papers from the Colloque de Syntaxe et Sémantique à Paris (CSSP 1997)*, pages 145–166. The Hague: Thesus.

Kuhn, Jonas. 1999b. Towards a simple architecture for the structure-function mapping. In M. Butt and T. H. King, eds., *Proceedings of the LFG99 Conference, Manchester, UK*, CSLI Proceedings Online.

Kuhn, Jonas. 2000a. Faithfulness violations and bidirectional optimization. In M. Butt and T. H. King, eds., *Proceedings of the LFG 2000 Conference, Berkeley, CA*, CSLI Proceedings Online, pages 161–181.

Kuhn, Jonas. 2000b. Processing Optimality-theoretic syntax by interleaved chart parsing and generation. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL-2000)*, pages 360–367. Hongkong.

Kuhn, Jonas. 2001a. Computational optimality-theoretic syntax – a chart-based approach to parsing and generation. In C. Rohrer, A. Roßdeutscher, and H. Kamp, eds., *Linguistic Form and its Computation*, pages 353–385. Stanford: CSLI Publications.

Kuhn, Jonas. 2001b. *Formal and Computational Aspects of Optimality-theoretic Syntax*. Ph.D. thesis, Institut für maschinelle Sprachverarbeitung, Universität Stuttgart.

Kuhn, Jonas. 2001c. Generation and parsing in Optimality Theoretic syntax – issues in the formalization of OT-LFG. In P. Sells, ed., *Formal and Empirical Issues in Optimality-theoretic Syntax*, pages 313–366. Stanford: CSLI Publications.

Kuhn, Jonas. 2001d. Resource sensitivity in the syntax-semantics interface and the German split NP construction. In D. M. andTibor Kiss, ed., *Constraint-Based Approaches to Germanic Syntax*, pages 177–215. Stanford: CSLI Publications.

Kuhn, Jonas. 2002. Corpus-based learning in stochastic OT-LFG—experiments with a bidirectional bootstrapping approach. In M. Butt and T. H. King, eds., *Proceedings of the LFG 2002 Conference, Athens, Greece*, CSLI Proceedings Online.

Kuhn, Jonas and Christian Rohrer. 1997. Approaching ambiguity in real-life sentences – the application of an Optimality Theory-inspired constraint ranking in a large-scale LFG grammar. In *Proceedings of DGfS-CL*. Heidelberg.

Lee, Hanjung. 2000. The emergence of the unmarked order in Hindi. In *Proceedings of NELS 30, October 1999, Rutgers, NJ*. Amherst, MA: Graduate Linguistics Students' Association.

Lee, Hanjung. 2001a. Markedness and word order freezing. In P. Sells, ed., *Formal and Empirical Issues in Optimality-theoretic Syntax*, pages 63–128. Stanford: CSLI Publications.

Lee, Hanjung. 2001b. *Optimization in Argument Expression and Interpretation: A Unified Approach*. Ph.D. thesis, Stanford University.

Legendre, Géraldine, William Raymond, and Paul Smolensky. 1993. An optimality-theoretic typology of case and grammatical voice systems. In *Proceedings of the 19th Meeting of the Berkeley Linguistics Society*.

Legendre, Géraldine, Paul Smolensky, and Colin 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*, pages 249–289. Cambridge, Massachusetts: MIT press.

Lenerz, Jürgen. 1977. *Zur Abfolge nominaler Satzglieder im Deutschen*. Tübingen: Gunther Narr Verlag.

Levin, Lori S. 1982. Sluicing: A lexical interpretation procedure. In J. W. Bresnan, ed., *The Mental Representation of Grammatical Relations*, chap. 9, pages 590–654. Cambridge, MA: MIT Press.

McCarthy, John J. and Alan Prince. 1995. Faithfulness and reduplicative identity. In J. Beckman, L. W. Dickey, and S. Urbanczyk, eds., *University of Massachusetts Occasional Papers, Volume 18: Papers in Optimality Theory*, pages 249–384. Amherst, MA: Graduate Linguistics Students' Association.

Müller, Gereon. 1998. German word order and Optimality Theory. Working paper no. 126, SFB 340 Stuttgart/Tübingen.

Müller, Gereon. 1999. Optionality in Optimality-Theoretic syntax. *GLOT International* 4(5):3–8.

Neumann, Günter. 1994. *A uniform computational model for natural language parsing and generation*. Ph.D. thesis, Universität des Saarlandes, Germany.

Neumann, Günter. 1998. Interleaving natural language parsing and generation through uniform processing. *Artifical Intelligence* 99:121–163.

Oepen, Stephan and Daniel P. Flickinger. 1998. Towards systematic grammar profiling. Test suite technology ten years after. *Journal of Computer Speech and Language* 12 # 4 (Special Issue on Evaluation):411 – 436.

Oepen, Stephan, Klaus Netter, and Judith Klein. 1997. TSNLP — Test Suites for Natural Language Processing. In J. Nerbonne, ed., *Linguistic Databases*, pages 13 – 36. Stanford, CA: CSLI Publications.

Pesetsky, David. 1998. Some optimiality principles of sentence pronunciation. In P. Barbosa, D. Fox, P. Hagstrom, M. McGinnis, and D. Pesetsky, eds., *Is the Best Good Enugh? Optimality and Competition in Syntax*, pages 337–383. Cambridge, Massachusetts: MIT press.

Pollard, Carl J. and Ivan A. Sag. 1994. *Head-Driven Phrase Structure Grammar*. Studies in Contemporary Linguistics. Chicago, London: University of Chicago Press.

Prince, Alan and Paul Smolensky. 1993. Optimality theory: Constraint interaction in generative grammar. Tech. Rep. Technical Report 2, Rutgers University Center for Cognitive Science.

Riezler, Stefan, Dick Crouch, Ron Kaplan, Tracy King, John Maxwell, and Mark Johnson. 2002. Parsing the Wall Street Journal using a Lexical-Functional Grammar and discriminative estimation techniques. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL'02), Pennsylvania, Philadelphia*.

Riezler, Stefan, Detlef Prescher, Jonas Kuhn, and Mark Johnson. 2000. Lexicalized stochastic modeling of constraint-based grammars using log-linear measures and EM training. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL'00), Hong Kong*, pages 480–487.

Samek-Lodovici, Vieri. 1996. *Constraints on Subjects: An Optimality Theoretic Analysis*. Ph.D. thesis, Rutgers University.

Schmid, Tanja. 2001. OT accounts of optionality: A comparison of global ties and neutralization. In G. Müller and W. Sternefeld, eds., *Competition in Syntax*, pages 283–319. Mouton de Gruyter. To appear.

Sells, Peter. 1999. Constituent order as alignment. In S. Kuno et al., eds., *Harvard Studies in Korean Linguistics 8*, pages 546–560. Harvard University.

Sells, Peter. 2001. *Alignment Constraints in Swedish Clausal Syntax*. Stanford: CSLI Publications.

Smolensky, Paul. 1996. On the comprehension/production dilemma in child language. *Linguistic Inquiry* 17:720–731.

Smolensky, Paul. 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.

Smolensky, Paul. 2000. Grammar-based connectionist approaches to language. *Cognitive Science* 23:589–613.

Sternefeld, Wolfgang. 1997. Comparing reference sets. In M. Bierwisch, H.-M. Gärtner, and C. Wilder, eds., *Economy in Linguistic Theory*, pages 81–114. Berlin: Akademieverlag.

Tesar, Bruce. 1995. *Computational Optimality Theory*. Ph.D. thesis, University of Colorado.

Tesar, Bruce B. and Paul Smolensky. 1998. Learnability in Optimality Theory. *Linguistic Inquiry* 29(2):229–268.

Tesar, Bruce B. and Paul Smolensky. 2000. *Learnability in Optimality Theory*. Cambridge, MA: The MIT Press.

Uszkoreit, H. 1987. *Word Order and Constituent Structure in German*. No. 8 in CSLI Lecture Notes. Stanford, CA: CSLI.

Varges, Sebastian. 1997. *Parsing und Generierung in uniformen Architekturen*. Master's thesis, Heinrich-Heine-Universität Düsseldorf.

Wartena, Christian. 2000. A note on the complexity of optimality systems. Ms. Universität Potsdam, Rutgers Optimality Archive, `http://ruccs.rutgers.edu/roa.html` (ROA-385-03100).

Wedekind, Jürgen. 1995. Some remarks on the decidability of the generation problem in LFG- and PATR-style unification grammars. In *Proceedings of the 7th EACL Conference*, pages 45–52. Dublin.

Wedekind, Jürgen. 1999. Semantic-driven generation with LFG- and PATR-style grammars. *Computational Linguistics* 25(2):277–281.

Wedekind, Jürgen and Ronald M. Kaplan. 1993. Type-driven semantic interpretation of f-structures. In *Proceedings of EACL, University of Utrecht*.

Wilson, Colin. 2001. Bidirectional optimization and the theory of anaphora. In J. Grimshaw, G. Legendre, and S. Vikner, eds., *Optimality Theoretic Syntax*. Cambridge, MA: MIT Press.

Zeevat, Henk. 2000. The asymmetry of Optimality Theoretic syntax and semantics. Ms., University of Amsterdam.

February 4, 2003

Constraint Index

February 4, 2003

Subject Index

February 4, 2003

Name Index