The Feature Space in Parallel Grammar Writing

Miriam Butt
Centre for Comp. Linguistics
UMIST
PO Box 88
Manchester M60 1QD UK
mutt@ccl.umist.ac.uk

Martin Forst
IMS
Universität Stuttgart
Azenbergstr. 12
70174 Stuttgart GERMANY
forst@ims.uni-stuttgart.de

Tracy Holloway King
NLTT/ISTL
Palo Alto Research Center
3333 Coyote Hill Rd
Palo Alto, CA 94304 USA
thking@parc.com

Jonas Kuhn
Dept. of Linguistics
The University of Texas at Austin
1 University Station, B5100
Austin, TX 78712-1196 USA
jonask@mail.utexas.edu

Abstract

This paper discusses the methodology and tools applied in the Parallel Grammar project (ParGram) to support consistency and parallelism of linguistic representations across multilingual Lexical Functional Grammar (LFG) grammars. A particular issue is that the grammars in the ParGram project are developed at different international sites. The approach that was established over several years relies on (i) a grammar code reviewing committee in which extensions to the existing representations are critically discussed, (ii) a technical tool for checking adherence to the best-practice feature declaration for linguistic representations, and (iii) a coordinated, systematic use of templates for expressing generalizations across lexicon entries and grammar rules. We compare the techniques used in practical LFG development with elements of Head-driven Phrase Structure Grammar (HPSG).

1 Introduction

At the very minimum, multilingual grammar development requires agreement on a common set of representations and broad agreement on the analyses of linguistic phenomena. Discussions about the proper analysis and representation of phenomena such as subject-verb agreement, case marking, relative clauses, or the treatment of adjectives or adverbials can be conducted at a very high linguistic level. Ultimately, though, differences in opinion are reified at a very low level of engineering. Since its inception, the Parallel Grammar (ParGram) project has therefore included a "feature committee," whose job is to find norms for the use and definition of a common multilingual feature space. Adherence to feature committee decisions is technically supported by a routine that checks the grammars for compatibility with a feature declaration. Parallelism at the level of grammatical constraints or descriptions is facilitated by systematic use of means of abstraction in the grammar specification code.

The ParGram project is an international collaboration aimed at producing broad-coverage computational grammars for a variety of languages ((Butt et al., 1999; Butt et al., 2002); see (Riezler et al., 2002) on the coverage of the English grammar). The grammars (to date of English, French, German, Japanese,[1] Norwegian, and Urdu) are written in the framework of Lexical Functional Grammar (LFG) (Kaplan and Bresnan, 1982; Dalrymple, 2001), and they are constructed using a common engineering and high-speed processing platform for LFG grammars: XLE (Maxwell and Ka-

[1] A Korean grammar is currently being ported from the Japanese grammar to determine how quickly a deep grammar can be developed when bootstrapped from the grammar of a typologically similar language (Kim et al., 2003).

plan, 1993).

In keeping with standard LFG practice, these grammars assign two levels of syntactic representation to the sentences of a language: a surface phrase structure tree (called a *constituent structure* or c-structure) and an underlying matrix of features and values (the *functional structure* or f-structure). The c-structure records the order of words in a sentence and their hierarchical grouping into phrases. The f-structure encodes the grammatical functions, syntactic features, and predicate-argument (dependency) relations conveyed by the sentence. F-structures are meant to encode a language universal level of analysis, allowing for crosslinguistic parallelism at this level of abstraction. For example, while the analysis of the English, French, and German versions of a sentence like (1) will necessarily differ at the c-structural level (different word orders, different numbers of auxiliaries), at the f-structure level all the grammars produce something like the dependency structure in Figure 1, in which the main predicate is the verb and the tense and aspect information is collected under the feature TNS-ASP (Butt et al., 1996). The language-particular dependencies between tense inflection, auxiliaries, and verbs are not encoded at f-structure.

(1) a. Tomorrow the letter will have arrived.

b. Demain la lettre sera
tomorrow the letter will-be

arrivée. (French)
arrived

c. Morgen wird der Brief
tomorrow will the letter

angekommen sein. (German)
arrive be

In this paper, we discuss some of the technical and organizational means that have been developed in the ParGram project to establish and enforce cross-grammar standards. Section 2 addresses the definition and comparison of allowable features and their proper values. Section 3 discusses the systematic use of templates in the grammar description language as a way of making generalizations explicit and transparent across grammars. Section 4 provides some discussion and a conclusion.

```
"Morgen wird der Brief angekommen sein."

   PRED      'an#kommen<[41:Brief]>'
   ADJUNCT   { 0[PRED 'morgen', ADV-TYPE temp] }
   SUBJ        [PRED  'Brief'
                NSEM  [COMMON count]
                NTYPE [GRAIN common]
                SPEC  [DET [PRED 'die', DET-TYPE def]]
             41 CASE nom, GEND masc, INFL strong-det, NUM sg, PERS 3]
   TNS-ASP     [ASPECT [FUT +_, PERF +_]
                MOOD indicative, TENSE present]
   TOPIC       [0:morgen]
21 CLAUSE-TYPE declarative, STMT-TYPE declarative, VTYPE main
```

Figure 1: F-structure for (1c)

2 Defining the Feature Space

In this section, we discuss the feature space in the LFG ParGram grammars and how it is defined and regulated.

2.1 The Status of Feature Appropriateness Conditions

One way of ensuring parallelism in f-structure level analysis is to define a crosslinguistically relevant feature space in advance. The idea of theory-driven type/sort definitions for the feature structure representations as used in Head-driven Phrase Structure Grammar (HPSG; (Pollard and Sag, 1994)), for example, would appear to be ideal for such a purpose. The space of valid representations is restricted by the use of typed values for each feature, where the type of a feature-structure object defines exactly what embedded features are *appropriate*. Very elaborate type hierarchies have been proposed (see (Copestake, 2002) for implementation and documentation of typed feature structure grammars).

The grounding of the type hierarchy in the theoretical framework would seem to provide ideal support for a multilingual grammar writing effort (Bender et al., 2002). However, for use in broad-coverage grammars, the central role played by feature appropriateness conditions in HPSG theory can also pose problems. As the discussions in the HPSG community show, even quite fundamental configurations in the feature geometry have been subject to dispute and revision over the years.

The consideration of additional languages or phenomena may lead to insights that justify, or even require, potentially fundamental revisions to the feature space. This can pose considerable problems for a continuing grammar development effort. Moreover, it is not entirely clear whether there is (or should be) a crosslinguistically uniform type hierarchy, although this issue is being tested and explored in the Matrix project (Bender et al., 2002; Flickinger and Bender, 2003).

The methodology for ensuring parallelism and consistency in the ParGram project relies on checking feature appropriateness in a very similar way to HPSG (section 2.3); however, the conditions are established in a less theory-driven manner. The LFG formalism does not enforce feature appropriateness conditions *per se*. They are viewed as an engineering-level construct used for the purpose of consistency-checking. This encourages an explorative comparison of different possible representations: it is especially useful for phenomena not discussed in the theoretical literature. The resulting feature geometry is typically "leaner" than an HPSG feature geometry, which goes along with its lesser theoretical status and makes it less prone to major revision.

In the remainder of this section, we describe two tools used to ensure parallelism in the feature space: feature declarations and the feature table.

2.2 Feature Declarations

While c-structure analyses are subject to language particular variation by definition, the idea behind the f-structures is that they reflect a more language universal analysis. In order to maintain parallel grammar development, it is therefore vital to have identical features playing identical roles in all of the grammars.

For some features, the discussions and decisions to be made in a multilingual context are relatively straightforward. For example, the fact that there should be a CASE feature universally is generally undisputed, as well as the idea that the core values of such a feature should be at the very least NOM(inative), DAT(ive), and ACC(usative). However, one could question whether English does indeed need these features (e.g., (Hudson, 1995)) and whether they should be spelled the English way or, for example, the German way (e.g., AKK(usativ)). While these latter sorts of questions seem relatively low-level, the core of language individual analyses depends on the precise declaration and space of the features. For example, much discussion went into the definition of the tense/aspect features: deciding which attributes and values to use meant engaging in a discussion of the underlying theoretical treatment of tense/aspect.

In feature declarations, two basic types of feature values are distinguished: atomic (or constant) values and complex f-structure values.[2] For atomic values, the feature declaration specifies that the value of a feature FEAT (denoted by FEAT:$\rightarrow$; the right-arrow introduces the definition of the *possible values* of the feature to its left) has to be a member of a set of atomic values:

(2) FEAT:$\rightarrow \in$ {value1 value2 …valueN}.

For complex f-structure values, the declaration specifies the features that the embedded feature structure may contain, using the subsumption operator $\ll$:

(3) FEATA:$\rightarrow \ll$ [FEATB1 …FEATBN].

Consider the TNS-ASP feature and the features it calls in the feature declaration in (4). Values may be atomic, in which case a set of possible atoms is specified, as for MOOD, or they may be complex, in which case appropriate features for the embedded feature structure are specified, as for TNS-ASP. Note that the complex feature structure may consist of only a subset of these features.

(4) TNS-ASP: $\rightarrow \ll$ [MOOD PERF PROG TENSE].
MOOD: $\rightarrow \in$ {imperative indicative subjunctive}.
PERF:$\rightarrow \in$ {+_ −_}.
PROG: $\rightarrow \in$ {+_ −_}.
TENSE: $\rightarrow \in$ {fut null past pres}.

[2] A further type of feature values are closed-set values, as argued for in (Dalrymple and Kaplan, 2000). An indeterminate pronoun may, for example, have the case feature { nom, acc }. The feature declaration for a closed-set-typed case feature would involve using the set subsumption operator:
CASE: $\rightarrow \ll$ { nom gen dat acc }.

Just by examining the features in (4), it is possible to tell that tense-mood-aspect information is grouped together under the TNS-ASP feature (instead of having the individual features directly in the verb's f-structure). In addition, we can see that the grammars do not make use of composite tense/aspect values such as past perfect, pluperfect, or future perfect. Early on in the ParGram project, this type of composite feature value for tense/aspect was attempted for the German, English, and French grammars. However, not only did the cost of determining the right value turn out to be too great (since these tenses are often encoded as periphrastic constructions, a complex system of interdependencies needs to be checked), but it was difficult to establish a coherent crosslinguistic feature space. It was therefore decided to fall back on a more atomic encoding of tense/aspect, in which atomic features are simply registered under the TNS-ASP feature. That is, the English auxiliary *will* contributes the feature FUT, the English auxiliary *have* contributes the PERF +_,[3] etc. The more intricate problem of a precise and crosslinguistically valid semantic analysis of tense/aspect is left to a separate semantic component, which can base its analysis on the information collected under the TNS-ASP feature.

When the grammar is loaded, the feature declaration is checked against the compiled grammar rules. If any undefined features are found or if any undefined values of declared features are found, XLE returns an error message indicating the offending feature and where it appeared. The grammar cannot be loaded until the feature declaration violation is fixed.[4] For example, if the feature declaration allows for ADV-DEGREE `positive` and `comparative` but the grammar also has ADV-DEGREE `superlative`, e.g., as might occur when the grammar writer is adding superlative adverbs to the grammar, then XLE provides a message as in (5).

(5) Feature declaration violation near line 43, column 23 in file /pg/eng/standard/english-templates.lfg:
ADV-DEGREE cannot be equal to superlative

The XLE system allows for grammar adaptation whereby a standard grammar developed for a language can be modified to a specific corpus (Kaplan et al., 2002). Among the possible adaptations is to allow controlled changes of the feature declaration. In particular, the specialized grammar can use the standard features as a default, while overriding only those features which have changed in the specialized version. This way, any changes made to the feature declaration of the standard grammar are automatically incorporated into the specialized grammar. This is done in the grammar configuration file where the feature declaration is stated, as in (6) which gives priority to the EUREKA feature declaration while using the STANDARD one as a default.

(6) FEATURES STANDARD-ENGLISH EUREKA-ENGLISH.

The specialized feature table then only needs to contain newly introduced features, e.g., FIELD in (8a), or features whose values have changed, e.g., NSEM, in (8b).

(7) a. FIELD: $\rightarrow \in$ {cause item problem solution}.

b. NSEM: $\rightarrow \ll$ [PROPER TIME EUREKA-TYPE].

As we will see in the next section, the ability to include multiple feature declarations can also be used in a multilingual context to create common feature declaration which can then be specialized for particular languages.

[3] The use of the underscore in the value '+_' indicates its status as an *instantiated symbol*. This means that it is resource-sensitive in the sense that it can be introduced only once through a *defining* feature equation. In other places, only *constraining* equations can be used to check for the feature value. The most well-known instantiated symbols are the semantic forms used as values of the PRED feature. Here, instantiation excludes a duplication of identical material in various clause positions.

[4] In order to check for feature declaration violations in the lexicon, it is necessary to load the generator for the grammar since only in generation is the entire lexicon indexed. As with feature violations in the grammar, XLE will indicate the feature in question and where it was found; once the violations are fixed, the generator can be loaded.

Feature	ENGLISH	GERMAN	JAPANESE
CASE	acc gen nom	acc dat gen nom	acc gen nom
INF-FORM	bare to	zu	
TNS-ASP	MOOD PERF	FUT MOOD PASS-SEM	MOOD PROG
	PROG TENSE	PERF TENSE	TENSE
VTYPE	copular main modal	copular main modal	copular main

Table 1: Sample FEAT-TABLE

2.3 The Feature Committee and Feature Declaration Checking

Within ParGram, the definition of the multilingually relevant feature space is established via twice-yearly meetings among the grammar writers. Here, new developments in the individual grammars are checked and discussed. New features introduced for the grammar of a particular language are skeptically reviewed and are only sanctioned if they can be shown to have a universal application or if it becomes clear that the individual grammar could not have done without this feature. Every effort is made to keep the feature-space as small as possible and to assimilate new analyses within the existing feature-space. To facilitate this, the feature declarations are combined into a feature table (FEAT-TABLE) which has a column for each language and the features as rows; the cells are the values of the features for a particular language. A sample feature table is shown in Table 1.

The FEAT-TABLE can be used at a fairly high level to determine which grammars cover which contructions and what types of analyses they give to them. The FEAT-TABLE can also be used to determine trivial differences across the grammars, like the spelling of CASE values mentioned above.

One typical issue in which the FEAT-TABLE proved useful was the treatment of noun types. The precise classification of different types of nouns cannot be imported from theoretical linguistics directly into grammar development (unlike, for example, case or tense/aspect). Rather, as the grammars grew and had to parse different kinds of nouns (e.g., as encountered in the Wall Street Journal), individual grammar writers introduced more and more features to constrain the number and type of different analyses. These were all duly recorded in the FEAT-TABLE, enabling the grammar writers to easily compare their analyses and naming conventions and then agree on a standardized approach across the grammars. In particular, a very flat structure had arisen whereby most of the features were immediate attributes of the noun's predicate. Most of these were then rearranged under an NTYPE feature and further subdivided into syntactic (NSYN) and semantic (NSEM) features which in turn could have complex values. The syntactic values include information as to whether the noun is, for example, `common`, `count`, `gerund`, `partitive`, `pronoun`, or `proper`; the semantic values include information as to what type of proper noun it is (e.g. `location`, `name`). This more hierarchical structure allows the grammar writer to more easily envision the possible types of nouns in the grammar and to make reference to noun classes with fewer disjunctions.

The ability to invoke multiple feature tables in an override fashion was discussed in the previous section (Kaplan et al., 2002). In addition to providing a way of specifying multiple "dialects" of a given language/grammar, this idea can also be exploited in a multilingual setting. A "universal" or "common" feature declaration can be created for all languages. This common feature declaration is then invoked by all of the grammars. If a specific language needs additional features or feature values which are not in the common feature table, then these can be specified in the language particular feature table. This language specific table is then given higher priority than the common one, as in (8).

(8) FEATURES COMMON STANDARD-FRENCH.

If, for example, the English grammar needs to have an ACONSTR feature to control adjectives in specific constructions or needs to have a DEIXIS feature as a value of DET, then these features will appear in the English feature table as in (9).

(9) ACONSTR: $\rightarrow \in$ {equi extrapos obj-con raise}.
DET: $\rightarrow \ll$ [DEIXIS DET-TYPE PRED].

Once a language particular feature or feature value has been proposed, the feature committee can then examine the nonconforming specifications and decide whether the feature is actually something that is crosslinguistically relevant (but simply has not come up as yet as a part of the analyses in any of the other grammars) and therefore should be added to the common feature specification, whether the feature is the result of an aberrant analysis on the part of the grammar writer and therefore should be disallowed, or whether the feature indeed reflects a language particular characteristic and should therefore be included as part of the language specific feature declaration. The ability to invoke multiple feature declarations in an override fashion thus allows for more direct multilingual grammar development and highlights differences among the feature spaces of the languages.

3 Capturing Generalizations through Templates

The discussion in the previous section focused on the use of feature *representations* according to an agreed-upon definition of the feature space. This itself does not have necessary implications for the generality in which *descriptions of* or *constraints on these representations* are expressed in the grammars. It would be possible (although difficult) to write a grammar that does not capture any linguistic generalizations, but conforms to the feature declaration in each of its *ad hoc* rule statements.

Much of the grammar development work goes into the identification of suitable generalizations, since they are crucial for making the grammar readable, extensible, and maintainable. Although parallelism across multilingual grammars at the description level is not as indispensable as the representation-level parallelism (which may be sufficient for applications like machine translation), it is possible to avoid a significant amount of duplicate work by using a common specification of language-independent principles across grammars. Ultimately, a systematically organized grammatical description is also the best way to guarantee consistent representations.

An obvious example for a language-independent subsystem required for each of the grammars is the organization of the lexicon, particularly with regard to subcategorization. For example, verbs can universally be classified in terms of notions such as intransitive, transitive, or ditransitive. While languages may differ in terms of case marking or other requirements, there is a level at which this kind of generalization is very useful. When a new grammar is added to the project, the grammar writer can immediately expect to distinguish between intransitive, transitive, and ditransitive verbs. Given a universally defined lexical rule which generates passives from actives, the grammar writer may also expect to find passives in the language. The same goes for templates which define subject-verb agreement.

As was the case for the discussion of the feature space definition in section 2, one possible view to take with respect to the organization of the lexicon is that the linguistic theory should enforce a highly systematic structure. In HPSG, the feature structure entities representing (certain aspects of) words and phrases are organized in an inheritance hierarchy of types/sorts. This makes it possible for general properties holding for a large class of items to be specified once at a high level of abstraction and for the specific instantiations to inherit these properties.

LFG does not assume a sort hierarchy as part of the linguistic theory. However, means of abstraction like templates in the grammar specification make it possible to organize the lexicon in the same general way, following a specialization hierarchy where appropriate: more specific templates can call more general templates as part of their definition. The verbal subcategorization frame templates within most of the ParGram grammars provide a good example.

The central contribution of a verb subcategorization frame is a specification of the verbal stem and the number and type of arguments this verb subcategorizes for. The information about the stem is passed in as a variable, as shown in (10a). However, it is usually the case that other restrictions also apply to verbs with that basic subcategorization frame. These restrictions can be invoked by calling[5] other templates, as in (10b) which invokes templates that block particles and sentential subjects and that require a finite COMP. In turn, this template may be called by other templates that further restrict it, as in (10c) which requires a *that* complement and (10d) which requires a *whether* complement.

(10) a. V-SUBJ-COMP_PRED(_stem) =
(↑PRED)='_stem<(↑SUBJ)(↑COMP)>'.

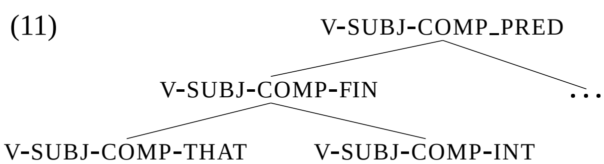
b. V-SUBJ-COMP-FIN(_stem) =
@(V-SUBJ-COMP_PRED _stem)
@COMP-FIN @NO-PRT @NO-CL-SUBJ.

c. V-SUBJ-COMP-THAT(_stem) =
@(V-SUBJ-COMP-FIN _stem)
@(COMP-FORM that).

d. V-SUBJ-COMP-INT(_stem) =
@(V-SUBJ-COMP-FIN _stem)
@(COMP-FORM whether).

The example in (10) results in a partial template hierarchy as in (11).

(11)



Similar dependencies among the templates can be found throughout the template system, e.g., the nominal templates also form a loose hierarchy.

As before, the LFG grammar development philosophy allows the grammar writer of an individual language a large degree of freedom as to whether or not a highly structured subsystem of recursive template calls is used in a specific situation. This has the advantage that less studied linguistic phenomena can be added to the grammar without a major modeling effort; also, ill-understood exceptions to a given generalization can be included in a grammar without having to change the system of representations itself, something which would be required in a strictly inheritance-based framework. At the same time, this large degree of liberty requires a certain discipline in the context of a multi-site multi-language grammar development project. The code reviewing instance of the feature committee has proven to be one helpful way of controlling the freedom offered by the LFG formalism.

[5] Template calls are indicated by an @ before the name of template in question; templates can take arguments, as in @(COMP-FORM that), or not, as in @NO-PRT.

4 Discussion and Conclusions

The ParGram project attempts to test LFG for its universality and coverage and to see how far parallelism can be maintained across languages; previous ParGram work (and much theoretical analysis) has largely confirmed the universality claims of LFG theory. The f-structures produced by the grammars for similar constructions in each language have the same major functions and features, with minor variations across languages (e.g., the f-structures for French nouns have a gender feature but not the English f-structures). This uniformity has the computational advantage that the grammars can be used in similar applications and that machine translation (Frank, 1999) can be simplified. It also has the advantage that when new grammars (such as Urdu or Japanese) are added to the project, they can be bootstrapped relatively efficiently using the existing declared feature space and the existing templates.

The ParGram methodology of establishing consistency and parallelism of representations relies on (i) a feature committee in which extensions to the existing representations are critically reviewed and (ii) technical tools for checking adherence to a feature declaration for linguistic representations. The resulting system of checking feature appropriateness resembles the theory-driven typed feature structure signatures of HPSG; however, the feature appropriateness conditions are not predominantly theory-driven, but are established by best practice. This takes into account the specific needs of multi-site development, and seems to be more appropriate for broad-coverage grammars in which representations often have to be defined for phenomena not yet analyzed in the literature. A similar approach is taken in the systematic structuring of the grammatical descriptions: means of abstraction like templates can be used to implement a hierarchical organization of linguistic generalizations where appropriate (but they can be side-stepped if necessary). Such a structured grammatical subsystem can be used across grammars in a multilingual context.

References

Emily Bender, Dan Flickinger, and Stephan Oepen. 2002. The grammar matrix: An open-source starter-kit for the rapid development of cross-linguistically consistent broad-coverage precision grammars. In *Proceedings of the Workshop on Grammar Engineering and Evaluation*, pages 8–14. COLING 2002 workshop.

Miriam Butt, Maria-Eugenia Niño, and Frédérique Segond. 1996. Multilingual processing of auxiliaries in LFG. In D. Gibbon, editor, *Natural Language Processing and Speech Technology: Results of the 3rd KONVENS Conference*, pages 111–122.

Miriam Butt, Tracy Holloway King, Maria-Eugenia Niño, and Frédérique Segond. 1999. *A Grammar Writer's Cookbook*. CSLI Publications.

Miriam Butt, Helge Dyvik, Tracy Holloway King, Hiroshi Masuichi, and Christian Rohrer. 2002. The Parallel Grammar project. In *COLING 2002: Workshop on Grammar Engineering and Evaluation*, pages 1–7.

Ann Copestake. 2002. *Implementing Typed Feature Structure Grammars*. CSLI Publications, Stanford, California.

Mary Dalrymple and Ronald M. Kaplan. 2000. Feature indeterminacy and feature resolution. *Language*, 76(4):759–798.

Mary Dalrymple. 2001. *Lexical Functional Grammar*. Academic Press, New York. Syntax and Semantics, volume 34.

Dan Flickinger and Emily Bender. 2003. Compositional semantics in a multilingual grammar resource. In *ESSLLI 2003 Workshop on Ideas and Strategies for Multilingual Grammar Development*. This volume.

Anette Frank. 1999. From parallel grammar development towards machine translation. In *Proceedings of MT Summit VII*, pages 134–142.

Richard Hudson. 1995. Does English really have case? *Journal of Linguistics*, 31:375–392.

Ronald M. Kaplan and Joan Bresnan. 1982. Lexical-Functional Grammar: A formal system for grammatical representation. In J. Bresnan, editor, *The Mental Representation of Grammatical Relations*, pages 173–281. The MIT Press.

Ronald M. Kaplan, Tracy Holloway King, and John T. Maxwell III. 2002. The Parallel Grammar project. In *COLING 2002: Workshop on Grammar Engineering and Evaluation*, pages 29–35.

Roger Kim, Mary Dalrymple, Ronald M. Kaplan, Tracy Holloway King, Hiroshi Masuichi, and Tomoko Ohkuma. 2003. Multilingual grammar development via grammar porting. In *ESSLLI 2003 Workshop on Ideas and Strategies for Multilingual Grammar Development*. This volume.

John T. Maxwell, III and Ronald M. Kaplan. 1993. The interface between phrasal and functional constraints. *Computational Linguistics*, 19:571–589.

Carl Pollard and Ivan Sag. 1994. *Head-Driven Phrase Structure Grammar*. Studies in Contemporary Linguistics. University of Chicago Press.

Stefan Riezler, Tracy Holloway King, Ronald M. Kaplan, Richard Crouch, John T. Maxwell III, and Mark Johnson. 2002. Parsing the Wall Street Journal using a Lexical-Functional Grammar and discriminative estimation techniques. In *Proceedings of the ACL*.