Abstract

This paper desribes four areas in which grammar engineers and theoretical linguists can interact. These include: using grammar engineering to confirm linguistic hypotheses; linguistic issues highlighted by grammar engineering; implementation capabilities guiding theoretical analyses; and insights into architecture issues. It is my hope that we will see more work in these areas in the future and more collaboration among grammar engineers and theoretical linguists. This is an area in which HPSG and LFG have a distinct advantage, given the strong communities and resources available.

1 Introduction

LFG and HPSG are in the privileged position of having not only a community of theoretical linguists but also of grammar engineers, with significant crossover between the theoretical and grammar-engineering communities. Grammar engineering involves the implementation of linguistically-motivated grammars so that natural language utterances and text can be processed to produce deep syntactic, and sometimes semantic, structures. In this paper, I outline four areas in which grammar engineering and theoretical linguistics interact (see also King (2011)). These are areas in which significant contributions have already been made and in which I foresee the possibility of even greater impact in the future.

Both LFG and HPSG have large-scale grammar engineering projects which span across institutions and across typologically-distinct languages. The projects test the underlying tenets of the theories, especially their universality across a broad range of languages. In addition, the projects build resources to support applications requiring the structures provided by the theories such as machine translation, question answering, summarization, and language teaching. The LFG-based ParGram and ParSem projects began with English, French, and German and have been expanded to include Japanese, Norwegian, Hungarian, and Indonesian among others. The ParGram grammars (Butt et al., 1999, 2002) are written within the LFG linguistic framework and with a commonly-agreed-upon set of grammatical features, using XLE (Crouch et al., 2011) as a grammar development platform. ParSem develops semantic structures based on the ParGram syntactic structures; most of the ParSem systems use the XLE XFR (transfer) system (Crouch and King, 2006), although some use a Glue Semantics implementation (Dalrymple, 2001; Asudeh et al., 2002). There are two HPSG-based grammar engineering projects which share these same goals: DELPH-IN and CoreGram. DELPH-IN and the LinGO Grammar Matrix (Bender et al., 2002, 2010) is a framework for the development

[†]I would like to thank the audience of HeadLex 2016 for lively discussion, the two anonymous reviewers for their detailed comments, and especially the HeadLex 2016 organizers for inviting me and for bringing together the theoretical and grammar engineering communities of HPSG and LFG. I would also like to thank the ParGram community who have been such an important part of my life for two decades.

of broad-coverage, precision, implemented grammars for diverse languages. The project builds from experience with the broad-coverage implemented HPSG grammars of English (the LinGO ERG (Flickinger, 2000)), German (DFKI's grammar (Müller and Kasper, 2000)), and Japanese (the JACY grammar (Siegel and Bender, 2002)) to extract components that are common across these grammars and therefore may be useful in the development of new grammars. They facilitate the development of grammars for different languages, producing semantic representations in a common format (MRS (Copestake et al., 2005)). The CoreGram (Müller, 2013, 2015) project is a multilingual grammar engineering project that develops HPSG grammars for typologically diverse languages, including German, Chinese, Danish, Maltese, and Persian. These grammar share a common core and are implemented in TRALE.

In the remainder of this paper I discuss four areas in which grammar engineering interacts with theoretical linguistics, illustrating these with examples from the LFG-based ParGram project. These include: using grammar engineering to confirm linguistic hypotheses (section 2); linguistic issues highlighted by grammar engineering (section 3); implementation capabilities guiding theoretical analyses (section 4); and insights into architecture issues (section 5). It is my hope that we will see more work in these areas and more collaboration among grammar engineers and theoretical linguists. This is a domain in which HPSG and LFG as a distinct advantage compared to many other linguistic theories, given the strong communities and resources available.

2 Grammar Engineering to Confirm Hypotheses: Indeterminacy by Underspecification

Grammar engineering can be used to confirm linguistic hypotheses (Butt et al., 1999; Bender, 2008; Bender et al., 2011; Fokkens, 2014). By implementing a fragment of a grammar that focuses on the hypothesis in question, the linguist can explore the details of the analysis and understand whether key issues or corner cases have been missed in the initial analysis. One caveat for this approach is that limitations of the grammar engineering platform may limit what types of hypotheses can be tested, e.g. in XLE there is no implementation of standard Lexical Mapping Theory (see LMT references in Dalrymple (2001)) and so testing hypotheses about LMT via grammar engineering is difficult.

In general this approach has been taken by linguists who work both in theoretical and computational linguistics and hence are able to straightforwardly test theoretical hypotheses through grammar engineering. However, this is a fruitful area for collaboration between theoretical linguists and grammar engineers. Some examples of LFG and HPSG work which used grammar engineering to confirm theoretical hypotheses include: Bender (2010)'s work on Wambaya which takes Nordlinger (1998a,b)'s detailed LFG analysis of Wambaya (morpho-)syntax and implements it in HPSG; Butt et al. (1997)'s work on extensions of LFG's Linking

Theory; Asudeh (2004)'s work on the analysis of resumptive pronouns using Glue semantics; Crysmann (2015)'s work on Hausa resumption and extraction and Crysmann (2016)'s work on Hausa tone and the phonology-syntax interface in HPSG; Beyaev et al. (2015)'s work on adjective coordination in LFG; Sadler et al. (2006) and Villavicencio et al. (2005)'s work on agreement with coordinated nouns in Brazilian Portuguese; Müller (1999)'s work on German syntax.

In this section, I review Dalrymple et al. (2009)'s proposal for handling indeterminacy by the underspecification of features (see Ingria (1990) on indeterminacy in general, Dalrymple and Kaplan (2000) on indeterminacy in LFG, and Crysmann (2005) and references therein on indeterminacy in HPSG). Dalrymple et al. (2009) examines the formal encoding of feature indeterminacy, focussing on case. Forms that are indeterminately specified for the value of a feature can simultaneously satisfy conflicting requirements on that feature and thus are a challenge to constraint-based formalisms which model the compatibility of information carried by linguistic items by combining or integrating that information. Dalrymple et al. (2009) views the value of an indeterminate feature as a complex and possibly underspecified feature structure. This complex feature structure allows for the incremental, monotonic refinement of case requirements in particular contexts. The proposed structure uses only atomic boolean-valued features. It covers the behaviour of both indeterminate arguments and indeterminate predicates (i.e. predicates placing indeterminate requirements on their arguments).

German has four cases (nominative, accusative, dative, genitive). Many nouns are fully specified for case; that is, they can only be interpreted as being a single case. However, some nouns are indeterminant for case. (1) shows an example of a noun which is indeterminate for all four German cases.

(1) Papageien
parrots
NOM/ACC/DAT/GEN
'parrots' (nominative, accusative, dative, or genitive) [deu]

In (2a) the indeterminate noun is the object of a verb which requires accusative case, while in (2a) the same noun is the object of a verb which requires dative case.

(2) a. Er findet Papageien.
he finds parrots
OBJ=ACC NOM/ACC/DAT/GEN
'He finds parrots.' [deu]

b. Er hilft Papageien.
he helps parrots
OBJ=DAT NOM/ACC/DAT/GEN
'He helps parrots.' [deu]

The data in (2) could be indicative of indeterminate case on a noun or of an ambiguously case-marked form. What distinguishes indeterminate forms is that they can simultaneously satisfy more than one case. This can be seen in (3). The indeterminate form can be the object of coordinated verbs, one which requires accusative case on its object (as in (2a)) and one which requires dates case on its object (as in (2b)).

(3)	Er	findet	und	hilft	Papageien.
	he	finds	and	helps	parrots
		OBJ=ACC		OBJ=DAT	NOM/ACC/DAT/GEN

'He finds and helps parrots' [deu]

The question is how to analyze indeterminate case so that the shared object can simultaneously satisfy the requirements to be dative and accusative. The proposal is to have case be a feature structure which for some nouns is indeterminate. For nouns like *Papageien* the only information that is available as to the case of the noun in the underlying form is that it must have a case, similar to all nouns heading noun phrases in German. It is only within a specific linguistic construction that the case values are specified. Determinate forms would have case feature values such as in (4), while indeterminate nouns have case features values such as in (5).[1]

(4) Determinate accusative case: Determinate dative case:

$$\left[\text{CASE}\ \begin{bmatrix}\text{NOM} & - \\ \text{ACC} & + \\ \text{GEN} & - \\ \text{DAT} & -\end{bmatrix}\right] \qquad \left[\text{CASE}\ \begin{bmatrix}\text{NOM} & - \\ \text{ACC} & - \\ \text{GEN} & - \\ \text{DAT} & +\end{bmatrix}\right]$$

(5) Indeterminate case:

$$\left[\text{CASE}\ \begin{bmatrix}\text{NOM} \\ \text{ACC} \\ \text{GEN} \\ \text{DAT}\end{bmatrix}\right]$$

Consider the situation with an indeterminate noun occuring with the same dative assigning verb in (6). The noun initially has no case specification and so within the f(unctional)-structure it appears with only CASE DAT=+ due to the case assignment from the verb. This works similarly for a verb taking an accusative object when the noun is indeterminate, only with an ACC=+ specification.

(6)	a.	Er	hilft	Papageien.
		he	helps	parrots
			OBJ=DAT	NOM/ACC/GEN/DAT

'He helps parrots.' [deu]

[1]Partially indeterminate forms would have some values of the case features specified, but not all.

b. $$\left[\begin{array}{ll} \textsc{obj} & \left[\begin{array}{ll} \textsc{pred} & '\text{parrots}' \\ \textsc{case} & \left[\begin{array}{ll} \textsc{nom} & \\ \textsc{acc} & \\ \textsc{gen} & \\ \textsc{dat} & + \end{array}\right] \end{array}\right] \end{array}\right]$$

When coordinating a verb taking a dative object with a verb taking an accusative object there is no clash, as shown in (7). If the nouns had been specified as having CASE=ACC or CASE=DAT there would be a clash of feature values. Similarly, a fully determinate noun would be ungrammatical because the DAT=− specification would clash with DAT=+ and vice versa for ACC. (The indices on the f-structures, e.g. the *1* in (7), indicate re-entrancy, i.e. an f-structure shared across two or more parts of the larger f-structure.)

(7) a.

Er	findet	und	hilft	Papageien.
he	finds	and	helps	parrots
	OBJ=ACC		OBJ=DAT	NOM/ACC/DAT/GEN

‘He finds and helps parrots’ [deu]

b. $$\left\{\begin{array}{l} \left[\begin{array}{ll} \textsc{pred} & '\text{find}' \\ \textsc{obj} & \left[\begin{array}{ll} \textsc{pred} & '\text{parrots}' \\ \textsc{case} & \left[\begin{array}{ll} \textsc{nom} & \\ \textsc{acc} & + \\ \textsc{gen} & \\ \textsc{dat} & + \end{array}\right] \end{array}\right]_{1} \end{array}\right] \\ \left[\begin{array}{ll} \textsc{pred} & '\text{help}' \\ \textsc{obj} & [\ ]_{1} \end{array}\right] \end{array}\right\}$$

Next consider how adjectival modification interacts with indeterminacy. An unambiguously dative modifier like *alten* ‘old’ imposes a negative specification for ACC in addition to the positive specification for DAT. This ACC − clashes with the ACC + of the accusative-taking verb *findet*, as in (8).

(8) a.

*Er	findet	und	hilft	alten	Papageien.
he	finds	and	helps	old	parrots
	OBJ=ACC		OBJ=DAT	DAT	NOM/ACC/DAT/GEN

‘He finds and helps old parrots.’ [deu]

b. Ill-formed f-structure:

$$\begin{bmatrix} \textsc{pred} & \text{'parrots'} \\ \textsc{adjunct} & \left\{\begin{bmatrix}\textsc{pred} & \text{'old'}\end{bmatrix}\right\} \\ \textsc{case} & \begin{bmatrix}\textsc{nom} & - \\ \textsc{acc} & +/- \\ \textsc{gen} & - \\ \textsc{dat} & + \end{bmatrix}\end{bmatrix}$$

In contrast to determinate adjectives like *alt* 'old', the adjective *rosa* 'pink' is fully indeterminate, and imposes no case restrictions on the noun it modifies.

(9) rosa: [no case restrictions]

This means that the noun *Papageien* can be modified by *rosa* and still satisfy simultaneous accusative and dative requirements, as in (10).

(10)	a.	Er	findet	und	hilft	rosa	Papageien.
		he	finds	and	helps	pink	parrots
			OBJ=ACC		OBJ=DAT		NOM/ACC/DAT/GEN

'He finds and helps pink parrots.' [deu]

b.

$$\begin{bmatrix} \textsc{pred} & \text{'parrots'} \\ \textsc{adjunct} & \left\{\begin{bmatrix}\textsc{pred} & \text{'pink'}\end{bmatrix}\right\} \\ \textsc{case} & \begin{bmatrix}\textsc{nom} & \\ \textsc{acc} & + \\ \textsc{gen} & \\ \textsc{dat} & + \end{bmatrix}\end{bmatrix}$$

This appeared to be a plausible analysis, but before proposing it, to confirm the analysis, we implemented a grammar fragment in XLE and developed a testsuite (Chatzichrisafis et al., 2007) with one instance of each adjective, determiner, noun, and verb type. We then ran all the (un)grammatical sentences composed of these lexical items in order to see whether the predictions held. Grammatical sentences should be accepted by the grammar and ungrammatical ones rejected. In this case, the implementation confirmed that our proposed analysis captured the data. This was especially helpful for untangling how adjectives and determiners combinations with indeterminate nouns in different syntactic positions.

3 Linguistic Issues Highlighted by Grammar Engineering: Copulas, Adjectives, and Subjects

Implementing a broad coverage grammar requires, by definition, analyzing a wide range of syntactic phenomena in such a way that they interact correctly with one another. This contrasts with theoretical linguistics which tends to focus on a phenomenon in isolation. Analyzing a wide range of phenomena simultaneously highlights interesting facts about the language. These are often obvious in hindsight, but they fall out from implementing each part of the analyses and from working on corpora where constructions are often more complicated than they originally seemed (Bender et al., 2011). These interactions are indicative of how the formalism and theory need to be structured. As an example of where implementing a broad coverage grammar highlights linguistic issues, consider the interaction of copular constructions, predicate adjectives, and subjecthood. The topics have been debated for decades in the theoretical literature, but implementing them unearthed interactions not captured in the standard LFG analyses.

In many languages, copular constructions show predicate adjective agreement between the adjective and the subject of the copular clause, as in the French example in (11). This leads to the question of whether adjectives have subjects given their predicative nature and the agreement facts. If they do, then the adjective can agree with its subject, which in turn can be identified with the subject of the copular clause, as in (12). If they do not, then the adjective must agree with the subject of the copula, as in (13). (See Dalrymple et al. (2004a), Butt et al. (1999) and references therein for more details on the copular construction in LFG; see van Eynde et al. (2016) on using treebanks to inform theoretical HPSG analyses of copular constructions.)

(11) Elle est petite.
she.F.Sg is small.F.Sg
'She is small.' [fra]

(12) a.
$$\begin{bmatrix} \textsc{pred} & \text{'be}<\textsc{xcomp}>\textsc{subj}\text{'} \\ \textsc{subj} & \begin{bmatrix} \textsc{pred} & \text{'pro'} \\ \textsc{num} & \text{sg} \\ \textsc{gend} & \text{fem} \end{bmatrix} \mathit{1} \\ \textsc{xcomp} & \begin{bmatrix} \textsc{pred} & \text{'small}<\textsc{subj}>\text{'} \\ \textsc{subj} & [\ ]\mathit{1} \end{bmatrix} \end{bmatrix}$$

b. petite $(\uparrow \textsc{pred}) = \text{'small}<\textsc{subj}>\text{'}$
$(\uparrow \textsc{subj num}) =_c \text{sg}$
$(\uparrow \textsc{subj gend}) =_c \text{fem}$

c. est $(\uparrow \textsc{pred}) = \text{'be}<\textsc{xcomp}>\textsc{subj}\text{'}$
$(\uparrow \textsc{subj}) = (\uparrow \textsc{xcomp subj})$

(13) a.
$$\begin{bmatrix} \textsc{pred} & \text{'be<}\textsc{subj,predlink}\text{>'} \\ \textsc{subj} & \begin{bmatrix} \textsc{pred} & \text{'pro'} \\ \textsc{num} & \text{sg} \\ \textsc{gend} & \text{fem} \end{bmatrix} \\ \textsc{predlink} & \begin{bmatrix} \textsc{pred} & \text{'small'} \end{bmatrix} \end{bmatrix}$$

b. petite $(\uparrow \textsc{pred}) = \text{'small'}$
$((\textsc{predlink}\ \uparrow)\ \textsc{subj num}) =_c \text{sg}$
$((\textsc{predlink}\ \uparrow)\ \textsc{subj gend}) =_c \text{fem}$

The open complement (XCOMP) analysis with the adjectives with subjects shown in (12) makes it easy to capture agreement of predicate adjectives with their subjects, the semantic predication relation between the adjective and the subject, and the control relations for raising adjectives, as in (14) (Dalrymple et al. (2004a), p194). This is easy to encode because the subject information is passed up with standard function application and hence becomes local, as in (14). Because of these facts, this analysis was implemented in the English ParGram grammar.

(14) a. It is likely/bound/certain to rain.

b. They are eager/foolish/loathe to leave.

(15)
$$\begin{bmatrix} \textsc{pred} & \text{'be<}\textsc{xcomp}\text{>}\textsc{subj}\text{'} \\ \textsc{subj} & \begin{bmatrix} \textsc{pron-form} & \text{it} \end{bmatrix}_1 \\ \textsc{xcomp} & \begin{bmatrix} \textsc{pred} & \text{'likely<}\textsc{xcomp}\text{>}\textsc{subj}\text{'} \\ \textsc{subj} & [\ ]_1 \\ \textsc{xcomp} & \begin{bmatrix} \textsc{pred} & \text{'rain<>}\textsc{subj}\text{'} \\ \textsc{subj} & [\ ]_1 \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

However, as the ParGram grammar was used to parse large corpora and served as the input to further semantic and abstract knowledge representations (Crouch and King, 2006; Bobrow et al., 2007), it was discovered that this analysis fails when the post-copular element already has a subject, as in (16) (Dalrymple et al. (2004a), p193).

(16) a. The problem is that they appear.

b. The problem is their appearing.

c. The problem is (for them) to leave before 6.

Constructions like those in (16) are incompatilbe with analyses where copulas are analyzed as taking an open complements due to the conflict as to the subject of the complement. This is shown in (17) for (16a) where the subject of *appear* should be *they*, but the open complement construction also assigns the subject of the copula *the problem* to be the subject of *appear*.[2]

(17) Open Complement

$$\left[\begin{array}{ll} \textsc{pred} & \text{'be<}\textsc{xcomp}\text{>}\textsc{subj}\text{'} \\ \textsc{subj} & \left[\begin{array}{ll} \textsc{pred} & \text{'problem'} \end{array}\right] \\ \textsc{xcomp} & \left[\begin{array}{ll} \textsc{pred} & \text{'appear<}\textsc{subj}\text{>'} \\ \textsc{subj} & \left[\begin{array}{ll} \textsc{pred} & \text{*'they/problem'} \end{array}\right] \end{array}\right] \end{array}\right]$$

When the object of the copular is a closed complement, there is no conflict for what the subject is, as shown in (18). However, this open complement analysis of this construction requires more machinery for the simple adjectival copular and raising adjective cases.

(18) Closed Complement

$$\left[\begin{array}{ll} \textsc{pred} & \text{'be<}\textsc{predlink}\text{>}\textsc{subj}\text{'} \\ \textsc{subj} & \left[\begin{array}{ll} \textsc{pred} & \text{'problem'} \end{array}\right] \\ \textsc{predlink} & \left[\begin{array}{ll} \textsc{pred} & \text{'appear<}\textsc{subj}\text{>'} \\ \textsc{subj} & \left[\begin{array}{ll} \textsc{pred} & \text{'they'} \end{array}\right] \end{array}\right] \end{array}\right]$$

How best to analyze copular constructions including their interactions with the argument structure of adjectives is still not resolved and continues to be the topic of debate among theoretical linguists and grammar engineers. The issues raised by these competing analyses could have been pursued in a purely theoretical setting, but implementing these constructions in a broad-coverage grammar clarified some of the issues, especially the interaction amongst constructions, even if a satisfactory solution has not yet been found.

4 Implementation Guiding Analysis: Complex Predicates via Restriction

Complex predicates are a major area of study in LFG (Butt, 1995; Mohanan, 1994; Butt et al., 2009) and HPSG (Müller, 2002). The fundamental issue with complex predicates is that it is not possible to exhaustively list them in the lexicon. Instead, they must be formed productively through the combination of main verbs and light verbs. There have been many theoretical linguistic proposals for how to analyze

[2]With an open complement analysis, the grammar has to create a dummy layer with a dummy PRED to protect the subject of the lower clause: an unsatisfying, unintuitive analysis. Details of this analysis are not discussed to space limitations.

complex predications, but most of them are not currently implementable in XLE (for the LFG proposals) because they involve Lexical Mapping Theory and complex operations in argument-structure. One analysis of complex predicates which is not used theoretically is to employ the restriction operator. By implementing complex predicates via restriction, it is possible to determine whether restriction is a theoretically-feasible option.

Consider the complex predicate in (19): (19) is a N(oun)-V(erb) complex predicate in which *kahAnI* 'story' is an argument which is contributed (i.e. licensed) by the noun, yet functions as the direct object of the clause. The finite verb *kI* 'did' has two arguments: Nadya and *yAd* 'memory'. The noun *yAd* 'memory' plays a double role: it is an argument of the finite verb, yet it also contributes to the overall predication of the clause (Mohanan, 1994).

(19) nAdiyah nE kahAnI yAd k-I
Nadya.F.Sg Erg story.F.Sg.Nom memory.F.Sg.Nom do-Perf.F.Sg
'Nadya remembered a/the story.' [urd]

In theoretical analyses, complex predicates comprise a single PRED form as in (20a) constructed from two underlying forms as in (20b) and (20c). This composition is generally argued to take place in argument structure or as a pre-syntactic operation over the lexicon.

(20) a. Standard LFG complex PRED: $(\uparrow \text{PRED}) = {}'\text{memory-do}<\text{SUBJ,OBJ}>'$

b. $(\uparrow \text{PRED}) = {}'\text{memory}<\text{OBJ}>'$

c. $(\uparrow \text{PRED}) = {}'\text{do}<\text{SUBJ,OBJ}>'$

Given the prevalence of complex predicates in Urdu, when implementing the Urdu ParGram grammar (Butt and King, 2002, 2007) it was imperative to analyze complex predicates from the outset. However, the XLE platform has no implementation of argument structure. Instead, the choices were (1) to include all complex predicates in the lexicon, (2) to use the lexical rules standardly used for passive formation, or (3) to explore the restriction operator (Kaplan and Wedekind, 1993). The lexicon and lexical rules options were not viable due to the productive nature of complex predicates and the types of argument changes they require. However, restriction can construct predicates on the fly, forming a new predicate form and altering the argument structure. Thus it was decided to explore this option.

First consider how the restrition operator works (Kaplan and Wedekind, 1993). The restriction operator allows for f-structure features to be "restricted out", i.e., to cause the grammar to function as if these features did not exist. A restricted f-structure is identical to the original f-structure except that it does not contain the restricted attribute. Monotonicity, which is fundamental to LFG, is still preserved since the original, non-restricted f-structure still exists.[3] An example of restricting case from the f-structure of a noun phrase is shown in (21).

(21) Original f-structure: F-structure with case restricted out:

$$\begin{bmatrix} \text{PRED} & \text{'Nadya'} \\ \text{NUM} & \text{sg} \\ \text{PERS} & 3 \\ \text{CASE} & \text{ERG} \end{bmatrix} \qquad\qquad \begin{bmatrix} \text{PRED} & \text{'Nadya'} \\ \text{NUM} & \text{sg} \\ \text{PERS} & 3 \end{bmatrix}$$

For complex predicates, this operation can construct complex predicate-argument structures dynamically (Butt et al., 2003a, 2009). The resulting PRED contains the same information as proposed in theoretical analyses, but arranged differently. Contrast the two PREDs in (22). In the theoretical analysis in (22a) there is a single predicate *memory-do* which takes two arguments, a SUBJ and an OBJ. In the restriction analysis in (22b) the PRED is spread across *do* and *memory* but there are again two arguments, a SUBJ and an OBJ. The f-structure for (19) is shown in (23).

(22) a. Standard LFG PRED: $(\uparrow \text{PRED}) = \text{'memory-do} < \text{SUBJ,OBJ} > \text{'}$

b. Restriction-based PRED: $(\uparrow \text{PRED}) = \text{'do} < \text{SUBJ,'memory} < \text{OBJ} > \text{'} > \text{'}$

(23) $$\begin{bmatrix} \text{PRED} & \text{'do} < \text{SUBJ,'memory} < \text{OBJ} > \text{'} > \text{'} \\ \text{SUBJ} & [\ \text{PRED 'Nadya'}\ ] \\ \text{OBJ} & [\ \text{PRED 'story'}\ ] \end{bmatrix}$$

This f-structure is achieved by a dynamic composition of the subcategorization frames contributed by *kar* 'do' and *yAd* 'memory'. The restriction operator is invoked as part of the f-structure annotations on the c(onstituent)-structure rules and is represented by a backslash. Grammatical functions and attributes listed after the backslash are restricted out of the f-structure when forming the new f-structure. Any grammatical functions or attributes not mentioned are inherited by the new f-structure. (24) shows the annotated c-structure rule which creates a complex predicate from a noun and a light verb. As is standard with LFG complex predicate analyses, the N and the Vlight are both heads of the Vcp, as indicated by $\uparrow=\downarrow$, since they both contribute to form the single, complex predicate in the f-structure. For the $\uparrow=\downarrow$ on the N, the PRED is restricted out ($\backslash$PRED) and instead its PRED becomes the second argument (ARG2) of the Vcp PRED (($\uparrow$ PRED ARG2)=($\downarrow$ PRED)).

(24) $$\text{Vcp} \rightarrow \begin{array}{cc} \text{N} & \text{Vlight} \\ \uparrow\backslash\text{PRED}=\downarrow\backslash\text{PRED} & \uparrow=\downarrow \\ (\uparrow \text{PRED ARG2})=(\downarrow \text{PRED}) & \end{array}$$

[3] This multiplicity of f-structures is often considered aesthetically unstatisfying, especially in theoretical linguistics.

In the lexicon the light verb's subcategorization contributes a subject but its second argument is incomplete, as in (25) where %Pred represents a variable to be filled in. This predicate is provided by the N in (24), e.g. the noun *yAd* 'memory'. The annotation ($\uparrow$ PRED ARG2)=($\downarrow$ PRED) in (24) substitutes the PRED value of *yAd* as the second argument of the light verb. The subcategorization frame of *yAd* is lexically specified to contribute an object, as in (26).

(25) $(\uparrow \text{PRED}) = {}'\text{do}< \text{SUBJ}\ \%\text{Pred} >'$

(26) $(\uparrow \text{PRED}) = {}'\text{memory}<\text{OBJ}>'$

To reiterate, the restriction operator restricts out those pieces of information which are "changed" as part of complex predication, namely the PRED. When the light verb and the noun are combined, they create the PRED in (27), and the annotated c-structure rules create the f-structure in (29) from the f-structures of the N and Vlight in (28).

(27) $(\uparrow \text{PRED}) = {}'\text{do}<\text{SUBJ},'\text{memory}<\text{OBJ}>'>'$

(28)

$$\begin{array}{cc}
\text{N} & \text{Vlight} \\
\downarrow \backslash\text{PRED}{=}\uparrow \backslash\text{PRED} & \uparrow{=}\downarrow \\
(\uparrow \text{PRED ARG2}){=}(\downarrow \text{PRED}) & \\
\left[\begin{array}{ll}
\text{PRED} & '\text{memory}<\text{OBJ}>' \\
\text{NUM} & \text{sg} \\
\text{GEND} & \text{fem} \\
\text{OBJ} & [\ ]
\end{array}\right] &
\left[\begin{array}{ll}
\text{PRED} & '\text{do}<\text{SUBJ},\ \%\text{Pred}>' \\
\text{PERF} & + \\
\text{NUM} & \text{sg} \\
\text{GEND} & \text{fem} \\
\text{SUBJ} & [\ ]
\end{array}\right]
\end{array}$$

(29)

$$\begin{array}{c}
\text{Vcp} \\
\left[\begin{array}{ll}
\text{PRED} & '\text{do}<\text{SUBJ},\ '\text{memory}<\text{OBJ}>'>' \\
\text{PERF} & + \\
\text{NUM} & \text{sg} \\
\text{GEND} & \text{fem} \\
\text{SUBJ} & [\ ] \\
\text{OBJ} & [\ ]
\end{array}\right]
\end{array}$$

Based on experiences with the Urdu ParGram grammar, experiences which were driven largely out of implementational necessity, Butt et al. (2003a) demonstrated that the restriction operator is indeed able to model different types of complex predicates in the Urdu grammar and can even model cases of stacked complex predicates (Butt et al., 2009). Having a complex predicate analysis for these constructions is necessary to meet the linguistic requirements of Urdu and to allow the

Urdu ParGram analysis of these constructions to be parallel to those of the other ParGram grammars. This implementation via the restriction operator opens a new theoretical approach to complex predicates. The jury is still out as to whether this analysis is superior to existing ones, but the theory is richer for having restriction as a possible formal device for complex predicate formation.

5 Insights Into Architecture Issues: Passive-Causative Interactions

A final area in which grammar engineering informs theoretical linguistics is by providing insights into architectural issues. This arose in ParGram in the interaction of passives and causatives where complex predication via restriction occurred in conjunction with passive sublexical rules. This interaction was observed in the Turkish (Çetinoğlu, 2009; Çetinoğlu and Oflazer, 2009) and Urdu grammars. In this section the focus is on Urdu, but the same issue arises in Turkish and was noticed there first.

Causatives in Urdu are formed morphologically. The causative morpheme *-A* adds an argument, the causer, to the PRED of the verb, as in (30). With a transitive verb, the subject of the transitive is realized as the causee and is marked with the dative/accusative *kO*.

(30) a. yassIn=nE kHAnA kHa-yA
Yassin=Erg food.M.Sg.Nom eat-Perf.M.Sg
'Yassin ate food.' [urd]

b. nAdyA=nE yassIn=kO kHAnA kHil-A-yA
Nadya=Erg Yassin=Dat food.M.Sg.Nom eat-Caus-Perf.M.Sg
'Nadya had Yassin eat (fed Yassin).' [urd]

Causatives can be analyzed as complex predicates: the overall argument structure is co-determined by more than one predicational element (Alsina, 1993). The Urdu grammar treats morphologically formed causatives on a par with syntactically formed complex predicates (Butt et al., 2003b; Butt and King, 2006). The predicate-argument structure is calculated dynamically based on the information contributed by each of the predicational parts. The final subcategorization frame is created by the restriction operator. For causatives, as shown in (32), both the PRED and the SUBJ are restricted out from the verb; the causative morpheme will provide the subject for the causativized verb ((33a)). With morphological causatives, this plays out at the level of sublexical rules (see Frank and Zaenen (2004) for discussion of sublexical rules in XLE and LFG). The morphological analyzer provides the analysis in (31) for the verb *kHilvAyA* 'made to eat'. The tags are terminal nodes of sublexical rules,[4] and the *+Cause* tag provides a phrase-structure locus for the restriction operator.

[4]The _BASE notation indicates a sublexical rule. The only difference between sublexical rules

(31) eat.Causative
kHilvAyA $\Leftrightarrow$ kHA +Verb +Cause +Perf +Masc +Sg

(32) $$\text{V} \rightarrow \quad \begin{array}{c} \text{V\_BASE} \\ \downarrow\backslash\text{PRED}\backslash\text{SUBJ}{=}\uparrow\backslash\text{PRED}\backslash\text{SUBJ} \\ (\downarrow\text{SUBJ}){=}\ (\uparrow\text{OBJ-GO}) \\ (\uparrow \text{PRED ARG2}){=}(\downarrow \text{PRED}) \end{array} \quad \begin{array}{c} \text{CAUSE\_BASE} \\ \uparrow{=}\downarrow \\ \\ \\ \end{array}$$

(33) a. +Cause: ($\uparrow$ PRED) = $'$cause$<$ SUBJ %Pred $>'$

b. eat ($\uparrow$ PRED) = $'$eat$<$ SUBJ OBJ $>'$

The causative lexical entry in (33a) is that of a complex predicate light verb. The variable is filled by the PRED value of the main verb *kHA* 'eat' and the original subject of 'eat' is realized as the causative OBJ-GO (i.e. a goal thematic object). (34) shows the f-structures for the main verb and the causative morpheme. The resulting causative verb's f-structure is in (35).

(34) V_BASE CAUSE_BASE

$$\begin{bmatrix} \text{PRED} & '\text{eat}{<}\text{SUBJ,OBJ}{>}' \\ \text{SUBJ} & [\ ]\mathit{1} \\ \text{OBJ-GO} & [\ ]\mathit{2} \end{bmatrix} \quad \begin{bmatrix} \text{PRED} & '\text{cause}{<}\ \text{SUBJ}\ \%\text{Pred}\ {>}' \\ \text{AGR} & \begin{bmatrix} \text{PERF} & + \\ \text{GEND} & \text{masc} \\ \text{NUM} & 3 \end{bmatrix} \\ \text{SUBJ} & [\ ]\mathit{3} \end{bmatrix}$$

(35) V

$$\begin{bmatrix} \text{PRED} & '\text{cause}{<}\text{SUBJ},'\text{eat}{<}\text{OBJ-GO,OBJ}{>}'{>} \\ \text{AGR} & \begin{bmatrix} \text{PERF} & + \\ \text{GEND} & \text{masc} \\ \text{NUM} & 3 \end{bmatrix} \\ \text{SUBJ} & [\ ]\mathit{3} \\ \text{OBJ-GO} & [\ ]\mathit{1} \\ \text{OBJ} & [\ ]\mathit{2} \end{bmatrix}$$

Passives in Urdu are formed by combining the verb *jA* 'go' with the perfect form of the main verb. The agent of the verb is realized as an adjunct and is marked with *se* 'with/from', as shown in (36).

---

and standard c-structure rules is how they are displayed by the XLE system. By default sublexical rules do not show the internal structure, e.g. the V_BASE and CAUSE_BASE in (31), in the visual display produced by XLE.

(36) a. yassIn=nE kHAnA kHa-yA
Yassin=Erg food.M.Sg.Nom eat-Perf.M.Sg
'Yassin ate food.' [urd]

b. kHAnA yassIn=sE kHa-yA ga-yA
food.M.Sg.Nom Yassin=Inst eat-Perf.M.Sg go-Perf.M.Sg
'The food was eaten by Yassin.' [urd]

Now consider the interaction of the causative with passivization. When a causative is passivized, the agent of the causative is realized as an adjunct and is marked with *se*, as in (37). That is, the causative applies first and then the passive.

(37) yassIn=kO nAdyA=sE kHAnA kHil-A-yA ga-yA
Yassin=Dat Nadya=Inst food.M.Sg.Nom eat-Caus-Perf.M.Sg go-Perf.M.Sg
'The food was fed to Yassin by/through Nadya.' [urd]

However, although the rules for causatives and passives worked independently in the grammar, they did not interact properly to provide an analysis of cases where a verb was both causativized and passivized. In addition the grammar could parse the ungrammatical constructions where the indirect object (OBJ-GO) *Yassin*, which was the agent of the main verb but not the agent of the causative form, was realized as an agentive adjunct, as in (38).

(38) *nAdyA=nE yassIn=sE kHAnA kHil-A-yA ga-yA
Nadya=Erg Yassin=Inst food.M.Sg.Nom eat-Caus-Perf.M.Sg go-Perf.M.Sg
'Nadya made the food be eaten by/through Yassin.' [urd]

That is, the implemented grammar of Urdu could not analyze grammatical combinations and incorrectly provided analyses for ungrammatical ones. The underlying problem is an architectural one. Passivization has traditionally been handled by lexical rules in LFG (Bresnan, 1982). These lexical rules apply in the lexicon and hence are applied directly to the specification of subcategorization frames. For example, as shown in (39a), the transitive verb *kHA* 'eat' states that there is a predicate 'P' (eat) which has a subject and an object and which can optionally be subject to passivization. The '@' sign signals a template call to the template PASSIVE, shown in (39b) which effects the passivization via a lexical rule (see Dalrymple et al. (2004b) and Asudeh et al. (2008) on templates in LFG).

(39) a. $\textsc{trans(p)} = @(\textsc{passive}\ (\uparrow \textsc{pred}){=}'\textsc{p}{<}(\uparrow \textsc{subj})\ (\uparrow \textsc{obj}){>}')$

b. $\textsc{passive(p)} = \begin{array}{l}(\uparrow \textsc{subj}) \longrightarrow \textsc{null} \\ (\uparrow \textsc{obj}) \longrightarrow (\uparrow \textsc{subj})\end{array}$

Since the passive lexical rule is specified in the lexical entry of *kHA* 'eat', passivization always applies before causativization. That is, the lexical rule is applied to the V_BASE. This is followed by the application of the causativization restriction operator in the sublexical rules in the syntax. The intuitive order of application in the original implementation is shown in (40) with passivization occuring in the lexicon and causativization in the grammar.

(40) Ungrammatical derivation of passive+causative:

a. Original Predicate: $(\uparrow\text{PRED}) = {}'\text{eat}<(\uparrow\text{SUBJ})\ (\uparrow\text{OBJ})>'$

b. Lexical Rule Passive: $(\uparrow\text{PRED}) = {}'\text{eat}<\text{NULL}\ (\uparrow\text{SUBJ})>'$

c. Restriction Causative: $(\uparrow\text{PRED}) = {}'\text{cause}<\text{SUBJ},'\text{eat}<\text{NULL},\text{OBL-GO}>'>$

However, passivization should operate on the entire complex predicate, i.e. passivization applies to the causativized verb. Once the source of the issue was identified, passivization was moved to be part of the sublexical rules and analyzed via the restriction operator. This allowed for the intuitive order of operations in (41) since the passive sublexical rule (not shown here) applies to the causative sublexical rule ((32)) in the c-structure.

(41) Grammatical derivation of passive+causative:

a. Original Predicate: $(\uparrow\text{PRED}) = {}'\text{eat}<(\uparrow\text{SUBJ})\ (\uparrow\text{OBJ})>'$

b. Restriction Causative: $(\uparrow\text{PRED}) = {}'\text{cause}<\text{SUBJ},'\text{eat}<\text{OBL-GO},\text{OBJ}>'>$

c. Restriction Passive: $(\uparrow\text{PRED}) = {}'\text{cause}<\text{NULL},'\text{eat}<\text{OBL-GO},\text{SUBJ}>'>$

While the solution outlined above in which both causative and passive are handled via restriction captures the linguistic generalization, many theoretical linguists do not consider it a satisfactory analysis. In theoretical LFG, argument alternations occur in a(rgument)-structure and are independent of particular morphological or syntactic realizations. However, there is an architectural flaw in how argument alternations are treated within the computational grammar implementation. In the ParGram grammars, passivization continues to be treated via lexical rules, as per classic LFG (but see Wedekind and Ørsnes 2003) and a-structure is not implemented in the grammars. Instead, predicate-arguments are modeled solely via subcategorization frames pertaining to grammatical functions. The interaction between causativization and passivization at the morphology-syntax interface highlights how traditional lexical rules make incorrect predictions when causativization is morphological but passivization is part of the syntax.

6 Conclusion

In this paper, I explored four areas in which grammar engineers and theoretical linguists can interact. These include: using grammar engineering to confirm linguistic hypotheses (section 2); linguistic issues highlighted by grammar engineering (section 3); implementation capabilities guiding theoretical analyses (section 4); and insights into architecture issues (section 5). These are all areas in which significant contributions have been made and in which I foresee the possibility of greater impact in the future. This is an area in which HPSG and LFG as a distinct advantage, given the strong communities and resources available.

References

Alsina, Alex. 1993. *Predicate Composition: A Theory of Syntactic Function Alternations*. Ph. D.thesis, Stanford University.

Asudeh, Ash. 2004. *Resumption as Resource Mangement*. Ph. D.thesis, Stanford University.

Asudeh, Ash, Crouch, Dick and Dalrymple, Mary. 2002. The Syntax-Semantics Interface: Theory and Implementation, NASSLLI-02 course notes.

Asudeh, Ash, Dalrymple, Mary and Toivonen, Ida. 2008. Constructions with Lexical Integrity: Templates as the LexiconSyntax Interface. In *Proceedings of the LFG08 Conference*, CSLI On-Line Publications.

Bender, Emily M. 2008. Grammar Engineering for Linguistic Hypothesis Testing. In Nicholas Gaylord, Alexis Palmer and Elias Ponvert (eds.), *Proceedings of the Texas Linguistics Society X Conference: Computational Linguistics for Less-Studied Languages*, pages 16–36, CSLI Publications.

Bender, Emily M. 2010. Reweaving a Grammar for Wambaya: A Case Study in Grammar Engineering for Linguistic Hypothesis Testing. *Linguistic Issues in Language Technology* 3, 1–34.

Bender, Emily M., Drellishak, Scott, Fokkens, Antske, Poulson, Laurie and Saleem, Safiyyah. 2010. Grammar Customization. *Research on Language & Computation* 8(1), 23–72.

Bender, Emily M., Flickinger, Dan and Oepen, Stephan. 2002. The Grammar Matrix: An Open-Source Starter-Kit for the Rapid Development of Cross-Linguistically Consistent Broad-Coverage Precision Grammars. In John Carroll, Nelleke Oostdijk and Richard Sutcliffe (eds.), *Proceedings of the Workshop on Grammar Engineering and Evaluation at the 19th International Conference on Computational Linguistics*, pages 8–14.

Bender, Emily M., Flickinger, Dan and Oepen, Stephan. 2011. Grammar Engineering and Linguistic Hypothesis Testing: Computational Support for Complexity in Syntactic Analysis. In Emily M. Bender and Jennifer E. Arnold (eds.), *Language from a Cognitive Perspective: Grammar, Usage and Processing*, pages 5–29, CSLI Publications.

Beyaev, Oleg, Dalrymple, Mary and Lowe, John. 2015. Number Mismatches in Coordination: An LFG Analysis. In *Proceedings of the LFG15 Conference*, pages 26–46, CSLI On-line Publications.

Bobrow, Daniel G., Cheslow, Bob, Condoravdi, Cleo, Karttunen, Lauri, King, Tracy Holloway, Nairn, Rowan, de Paiva, Valeria, Price, Charlotte and Zaenen, Annie. 2007. PARC's Bridge and Question Answering System. In *Proceedings of the Grammar Engineering Across Frameworks (GEAF07) Workshop*, pages 46–66, CSLI Publications.

Bresnan, Joan (ed.). 1982. *The Mental Representation of Grammatical Relations*. MIT Press.

Butt, Miram and King, Tracy Holloway. 2007. Urdu in a Parallel Grammar Development Environment. In T. Takenobu and C.-R. Huang (eds.), *Language Resources and Evaluation: Special Issue on Asian Language Processing: State of the Art Resources and Processing*, volume 41, pages 191–207.

Butt, Miriam. 1995. *The Structure of Complex Predicates in Urdu*. CSLI Publications.

Butt, Miriam, Dalrymple, Mary and Frank, Anette. 1997. An Architecture for Linking Theory in LFG. In *Proceedings of the LFG97 Conference*, CSLI On-line Publications.

Butt, Miriam, Dyvik, Helge, King, Tracy Holloway, Masuichi, Hiroshi and Rohrer, Christian. 2002. The Parallel Grammar Project. In *Proceedings of COLING2002 Workshop on Grammar Engineering and Evaluation*, pages 1–7.

Butt, Miriam and King, Tracy Holloway. 2002. Urdu and the Parallel Grammar Project. In *Proceedings of COLING-2002 Workshop on Asian Language Resources and International Standardization*, pages 39–45.

Butt, Miriam and King, Tracy Holloway. 2006. Restriction for Morphological Valency Alternations: The Urdu Causative. In Miriam Butt, Mary Dalrymple and Tracy Holloway King (eds.), *Intelligent Linguistic Architecturs: Variations on Themes by Ronald M Kaplan*, pages 235–258, CSLI Publications.

Butt, Miriam, King, Tracy Holloway and III, John T. Maxwell. 2003a. Complex Predicates via Restriction. In Miriam Butt and Tracy Holloway King (eds.), *The Proceedings of the LFG '03 Conference*, pages 92–104, CSLI Publications.

Butt, Miriam, King, Tracy Holloway and Maxwell III, John T.2̇003b. Complex Predicates via Restriction. In *Proceedings of the LFG09 Conference*, CSLI Online Publications.

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

Butt, Miriam, King, Tracy Hollowy and Ramchand, Gillian. 2009. Complex Predication: Who made the child pinch the elephant? In Linda Uyechi and Lian Hee Wee (eds.), *Reality Exploration and Discovery: Pattern Interaction in Language and Life*, pages 231–256, CSLI Publications.

Çetinoğlu, Özlem. 2009. *A Large Scale LFG Grammar for Turkish*. Ph. D.thesis, Sabanci University.

Çetinoğlu, Özlem and Oflazer, Kemal. 2009. Integrating Derivational Morphology into Syntax. In N. Nicolov et al. (ed.), *Recent Advances in Natural Language Processing V*, John Benjamins.

Chatzichrisafis, Nikos, Crouch, Dick, King, Tracy Holloway, Nairn, Rowan, Rayner, Manny and Santaholma, Marianne. 2007. Regression Testing For Grammar-Based Systems. In *Proceedings of the Grammar Engineering Across Frameworks (GEAF07) Workshop*, pages 128–143, CSLI On-line Publications.

Copestake, Ann, Flickinger, Dan, Sag, Ivan A. and Pollard, Carl J. 2005. Minimal Recursion Semantics: An Introduction. *Research on Language and Computation* .

Crouch, Dick, Dalrymple, Mary, Kaplan, Ron, King, Tracy, Maxwell, John and Newman, Paula. 2011. XLE Documentation, on-line documentation from the Palo Alto Research Center.

Crouch, Dick and King, Tracy Holloway. 2006. Semantics via F-Structure Rewriting. In *Proceedings of LFG06*, pages 145–165, CSLI On-line publications.

Crysmann, Berthold. 2005. Syncretism in German: A Unified Approach to Underspecification, Indeterminacy, and Likeness of Case. In Stefan Müller (ed.), *Proceedings of the 12th International Conference on Head-Driven Phrase Structure Grammar*, pages 91–107, CSLI On-line Publications.

Crysmann, Berthold. 2015. Resumption and extraction in an implemented HPSG of Hausa. In *Proceedings of Grammar Engineering Across Frameworks*, ACL-IJCNLP.

Crysmann, Berthold. 2016. Representing Morphological Tone in a Computational Grammar of Hausa. *Journal of Language Modelling* pages 463–512.

Dalrymple, Mary. 2001. *Lexical Functional Grammar*. Academic Press, syntax and Semantics Volume 34.

Dalrymple, Mary, Dyvik, Helge and King, Tracy Holloway. 2004a. Copular Complements: Closed or Open? In *Proceedings of the LFG'04 Conference*, pages 188–189, CSLI Publications.

Dalrymple, Mary and Kaplan, Ronald M. 2000. Feature Indeterminacy and Feature Resolution. *Language* 76, 759–798.

Dalrymple, Mary, Kaplan, Ronald M. and King, Tracy Holloway. 2004b. Linguistic Generalizations over Descriptions. In *Proceedings of the LFG04 Conference*, CSLI On-Line Publications.

Dalrymple, Mary, King, Tracy Holloway and Sadler, Louisa. 2009. Indeterminacy by Underspecification. *Journal of Linguistics* 45, 31–68.

Flickinger, Dan. 2000. On Building a More Efficient Grammar by Exploiting Types. *Natural Language Engineering* 6 (1) (Special Issue on Efficient Processing with HPSG), 15 – 28.

Fokkens, Antske Sibelle. 2014. *Enhancing Empirical Research for Linguistically Motivated Precision Grammars*. Ph. D.thesis, Department of Computational Linguistics, Universität des Saarlandes.

Frank, Anette and Zaenen, Annie. 2004. Tense in LFG: Syntax and Morphology. In Louisa Sadler and Andew Spencer (eds.), *Projecting Morphology*, pages 23–66, CSLI Publications.

Ingria, Robert J. P. 1990. The Limits of Unification. In *Proceedings of the 28th Annual Meeting of the ACL*, pages 194–204.

Kaplan, Ronald M. and Wedekind, Jürgen. 1993. Restriction and correspondence-based translation. In *Proceedings of the 6th Conference of the Association for Computational Linguistics European Chapter (EACL)*, pages 193–202.

King, Tracy Holloway. 2011. (Xx*-)Linguistics: Because We Love Language. *Linguistic Issues in Language Technology: Interaction of Linguistics and Computational Linguistics* 6.

Mohanan, Tara. 1994. *Argument Structure in Hindi*. CSLI Publications.

Müller, Stefan. 1999. *Deutsche Syntax deklarativ. Head-Driven Phrase Structure Grammar für das Deutsche*. Max Niemeyer Verlag.

Müller, Stefan. 2002. *Complex Predicates: Verbal Complexes, Resultative Constructions, and Particle Verbs in German*. Studies in Constraint"=Based Lexicalism, No. 13, Center for the Study of Language and Information.

Müller, Stefan. 2013. The CoreGram Project: A Brief Overview and Motivation. In Denys Duchier and Yannick Parmentier (eds.), *Proceedings of the Workshop on High-level Methodologies for Grammar Engineering (HMGE 2013), Düsseldorf*, pages 93–104.

Müller, Stefan. 2015. The CoreGram Project: Theoretical Linguistics, Theory Development and Verification. *Journal of Language Modelling* 3(1), 21–86.

Müller, Stefan and Kasper, Walter. 2000. HPSG Analysis of German. In Wolfgang Wahlster (ed.), *Verbmobil: Foundations of Speech-to-Speech Translation*, Artificial Intelligence, pages 238 – 253, Springer.

Nordlinger, Rachel. 1998a. *Constructive Case: Evidence from Australian languages*. CSLI Publications.

Nordlinger, Rachel. 1998b. *A Grammar of Wambaya*. Pacific Linguistics.

Sadler, Louisa, Arnold, Doug and Villavicencio, Aline. 2006. Portuguese: Corpora, Coordination and Agreement. In *Proceedings of Linguistic Evidence: Empirical, Theoretical, and Computational Perspectives*.

Siegel, Melanie and Bender, Emily M. 2002. Efficient Deep Processing of Japanese. In *Proceedings of the 3rd Workshop on Asian Language Resources and International Standardization at the 19th International Conference on Computational Linguistics*.

van Eynde, Frank, Augustinus, Liesbeth and Vandeghinste, Vincent. 2016. Number Agreement in Copular Constructions: A Treebank-based Investigation. *Lingua* pages 104–126.

Villavicencio, Aline, Sadler, Louisa and Arnold, Doug. 2005. An HPSG Account of Closest Conjunct Agreement in NP Coordination in Portuguese. In *Proceedings of the 12th International Conference on Head-Driven Phrase Structure Grammar*.

Wedekind, Jürgen and Ørsnes, Bjarne. 2003. Restriction and Verbal Complexes in LFG: A Case Study for Danish. In Miriam Butt and Tracy Holloway King (eds.), *Proceedings of LFG03*, CSLI On-line Publications.