Proceedings of the Conference on Language & Technology 2009

Urdu and the Modular Architecture of ParGram

Tina Bogel, Miriam Butt, Annette Hautli and Sebastian Sulger
Universitat Konstanz, Germany
{tina.boegel, miriam.butt, annette.hautli, sebastian.sulger}@uni-konstanz.de

Abstract

This paper reports on the modular architecture for natural language parsing and generation that is a consequence of using Lexical Functional Gram- mar as the linguistic framework in the context of the ParGram (Parallel Grammar) project. In particular, we discuss the following modules: the tokenizer and morphological analyzer, the syntax as implemented in the grammar development platform XLE [15] and the semantics, which is effected through rewrite rules. We also briefly touch upon the ability to allow for extra projections, such as the prosodic projection. Overall, Lexical-Functional Grammar in conjunction with the XLE development platform allows not only for robust and large-scale natural language parsing and generation, but also for the incorporation of deep linguistic insights.

1. Introduction

The Urdu ParGram (Parallel Grammar) Grammar is part of a larger project on grammar development in which a loose alliance of researchers are building large-scale, robust grammars based on common underlying linguistic principles and common technology [13]. Languages for which large grammars exist to date are English, French, German, Japanese, Norwegian and Turkish. Smaller grammars include Arabic, Chinese, Georgian, Malagasy and Welsh. Within the ParGram project, the Urdu grammar currently represents the only South Asian language. As a typologically different language, it has already been able to contribute significantly to the understanding of parallel, multilingual grammar development [22,10]. In this paper we report on our experiences for Urdu grammar development with the type of modular architecture for natural language parsing and generation that is a consequence of using Lexical Functional Grammar (LFG) [5,17] as the underlying linguistic framework. In particular, we discuss the following modules: the tokenizer and morphological analyzer, the syntax as implemented in the grammar development platform XLE [15] and the semantics. We also briefly touch upon the ability to allow for extra projections, such as the prosodic projection. We show that the multilingual nature of the ParGram project provides for an architecture that not only allows for robust and large scale natural language parsing and generation, but does so in a manner that allows for a satisfactory treatment of language-particular phenomena.

2. Overall architecture

The overall architecture of all the ParGram grammars is similar. The first step is tokenization and morphological analysis, all of which is done using state-of-the-art finite state technology (FST) as described in [2]. The output of the morphological analyzer then feeds into the syntactic component, where the analyses are informed by the theoretical linguistic work. At this level, the morphological information interacts with syntactic rules and the relevant morphological information helps to build the c(onstituent)-structure on the one hand and the f(unctional)-structure on the other hand [7]. These two levels of representation, or projections in terms of the LFG architecture, guarantee a transparent representation of all important aspects of a sentence — the c(onstituent)-structure encodes linear precedence and constituency relations as well as information about word classes, whereas the f(unctional)-structure provides functional information about the predicate-argument structure in terms of subject or object, as well as encoding information about voice (passive or not) and tense/aspect.

$$\begin{tikzcd} \text{tokenizer \& morphology (fst)} \arrow[d] & \\ \text{syntax (f- and c-structure)} \arrow[r] \arrow[d] & \text{prosody (p-structure)} \\ \text{semantics (xfr ordered rewriting)} & \end{tikzcd}$$

Figure 1: Overall grammar architecture within ParGram

The syntactic representations then provide the basis of input for semantic analysis. This can be done either in terms of another projection within the LFG/XLE grammar development platform (cf. the Norwegian grammar, which uses Minimal-Recursion Semantics [14], or can be handled via a term-rewriting component, XFR, which works on a Prolog encoding of the syntactic information and maps it into a semantic representation. With respect to the Urdu grammar, we have opted for the XFR term-rewriting component [16], like most of the ParGram grammars. These are the basic pieces of grammar architecture shared within ParGram. However, as the LFG architecture in principle allows for more linguistic levels of representations [17], individual grammars can feel free to experiment. In the Urdu grammar we are currently experimenting with a p(rosodic)-projection (section 6). In the past we have experimented with the representation of notions such as topic and focus at i(nformation)-structure.

The XLE/LFG architecture is suitable both for parsing and generation. Generation is currently performed from (underspecified) f-structures [15], however, generation from a semantic representation back to a string is also possible. Overall, the LFG architecture in combination with XLE/XFR provides a powerful and efficient platform (for possible applications, see Butt and King 2007b). In what follows, we describe some of the individual modules with respect to the Urdu grammar.

3. Tokenization

At the moment, the Urdu grammar uses the default tokenizer provided as part of the ParGram startup kit. This is essentially the tokenizer described in [2]. Input and tokenization is therefore currently only possible in an ASCII format, which is the input required by our morphological analyzer (section 4). However, efforts are currently underway to integrate a transliterator from the Urdu Arabic-based script to ASCII [23]. This is because of the fact that Urdu is structurally almost identical to Hindi. The major difference between the two is that Urdu bears more Persian/Arabic influence on its vocabulary while Hindi is more Sanskrit based. With respect to natural language processing (NLP) the only major difference lies in the script. The Urdu grammar is thus being engineered so as to be able to eventually process both Urdu and Hindi. This means that as part of the tokenizer, a transliteration system will be integrated which transliterates from the Urdu script to ASCII (and back out again for generation purposes) and from the Hindi Devanagari script to ASCII [see 4] for issues that arise because the Urdu and Hindi scripts code pieces of the morphology differently). The morphological analyzer, the syntax and the semantics then all work with the ASCII transliterations [for a similar idea, see [19]).

4. Morphological analyzer

The current morphological analyzer [4] was built and implemented using state-of-the-art finite-state machines [2] to describe the rather complex morphology of Urdu, including non-concatenative phenomenon like reduplication. This morphological analyzer is connected up to the syntax via the interface described in [4]. Morphological information can be extracted relatively easily via this interface and allows a broad vocabulary coverage along with a description of language particular characteristics that can be found in Urdu. One such example is the phenomenon of reduplication (section 4.3).

A positive feature of the very modular approach taken within the ParGram project is that all of the morphologies are independent of the XLE development platform and LFG. That means that from the perspective of XLE and the syntactic analysis, the finite-state morphological analyzer is a black box — it produces usable output in terms of abstract tags (see below), but it could be replaced with some other morphology module. Conversely, since the morphological analyzer is built independently from the grammar, it represents a stand-alone resource [cf. Hussain 2004 for another Urdu finite-state analyzer].

4.1. The basic set-up

Currently, the Urdu morphological analyzer covers all nominal, adjectival and verbal inflectional paradigms. Causativization has been included, but other derivational morphology remains to be integrated. The lexicon is still relatively small and is continuously being expanded. In the interests of transparency and maintainability, there are three independent lexicons for nouns, verbs and adjectives and adverbs.

As described in [2] and Kaplan et al. [21] and as shown below for Urdu, the finite-state morphology associates a surface form with a canonical form, a lemma, and a series of abstract morphological tags. The Urdu form is in our ASCII transliteration, which is based on [18].

(1) laRkA
laRk+Noun+Masc+Sg+Nom

(2) baccHa
baccH+Noun+Masc+Sg+Nom

In (1) and (2) the nouns laRkA 'boy' and baccHa 'male child' are marked for word class, gender, number and case. Note that even though the surface inflectional form varies, both are analyzed in the same way at the abstract level of analysis — the morphological analysis takes place within the same states following the generalized patterns for masculine nouns even if their surface inflectional endings (A/a) differ. Unmarked nouns that is, nouns which do not have one of the overt inflectional endings as in (2) are also provided with abstract tags indicating morphological information. (3) provides an example in which the unmarked noun sher 'lion' must be analyzed as allowing for more than one possibility, namely, a plural or a singular form. Note also that the morphological analyzer registers and passes on the fact that this is an unmarked noun.

(3) sher
sher+Noun+Unmarked+Masc+**Pl**+Nom
sher+Noun+Unmarked+Masc+**Sg**+Nom

The same possibility of multiple analysis holds for verbs, as illustrated in (4) for likH- 'write', where the -A inflection can in principle either be encoding perfect morphology or the causative morpheme.

(4) likH
likH+Verb+**Perf**+Masc+Sg
likH+Verb+**Caus**

Much of Urdu morphology turns out to be similarly ambiguous. However, this does not pose a problem for the morphological analyzer. It also does not pose a problem for syntactic parsing or generation either, as the syntactic context disambiguates between all of the possibilities. Consider (5) vs. (6).

(5) nAdya=ne xat
Nadya.F.Sg=Erg letter.M.Sg.Nom
likH-A
write-Perf.M.Sg
"Nadya wrote a letter."

(6) nAdya=ne yasIn=kO
Nadya.F.Sg=Erg Yassin.M.Sg=Acc
xat
letter.M.Sg
likH-A lI-yA
write-Caus take-Perf.M.Sg
"Nadya made Yassin write a letter."

In (6), the first -A must be the causative and the second -A must encode the perfect. The syntactic rules for Urdu encode the requirement that the last- A in a sequence must always encode tense/aspect information and thus the multiple possibilities passed on by the morphology can be disambiguated quickly and locally within the verbal complex.

4.2. The lexicon

The bulk of the lexicon in the ParGram grammars resides within the finite state morphology (FSM). It is generally necessary only to keep a separate lexicon for the verbs, as these display differing subcategorization frames (i.e., intransitive vs. transitive vs. di-transitive). The lexicon within the FSM is very compact, as just one lexical entry or lemma is needed for each word. In our FSM, this is generally the stem (coded under the "Root" Lexicon), from which the FSM is pointed to the next state, which is a LEXICON named "verb" in (7). Here, the stem points to the generalized inflectional endings in the states and their associated tags, which has the effect of allowing for more than one interpretation, cf. (5) and (6).

(7)
```
LEXICON Root
likH      verb;

LEXICON verb
+Verb+Perf+Masc+Sg:A   #;
+Verb+Caus:A    #;
```

The set up of the FSM thus allows for a very compact rendering of all the morphological information and which word stems it attaches to.

4.3. Reduplication

In order to provide an example of the capabilities of the morphological analyzer, the rather difficult problem of reduplication is demonstrated in this section. Like most South Asian languages, Urdu uses reduplication quite frequently (Abbi, 1991). All content words can generally be reduplicated and the effect is to either strengthen/emphasize the original word or to express something like 'and those kinds of things'.

(8) a. kHAnA vAnA
food.M.Sg. Redup
"food and those kinds of things"

b. tHanDA tHanDA
cold.M.Sg. Redup
"ice cold (cold cold)"

There are two different kinds of reduplication strategies. The one in (8a) is generally described as echo formation or echo reduplication, whereby the onset of the content word is replaced with another consonant. This consonant could be either /v/, /t/ or /ʃ/. Alternatively, as in (8b), the word is simply repeated. Generally, reduplications are written as separate words. Thus, the fundamental problem that the tokenizer faces is the fact that a reduplicated item must be recognized. The transliteration system will yield two words, as in (9), separated by a token boundary.

(9)	calnA valnA
	walk.Inf.M.Sg Redup
	"walking and such things"

Our morphological analyzer basically follows the solution for full word reduplication presented by Beesley and Karttunen [2] for Malay, which takes two tokens separated by white space and allows for them to be processed as one item. Furthermore, the basic lexicon that has already been built independently of reduplication for nouns, verbs, adjectives and other content verbs interacts with regular expressions that allow for reduplication. That is, all of the regularly formed content words in the Urdu morphological analzyer are subject to a regular expression which takes that word and either doubles (duplicates) it, or duplicates it and changes the first consonant. This has the effect of multiplying the size of the morphological analyzer (since it now has to be prepared for reduplications), however, the system is still very efficient (for details see [4]). The final morphological analysis of reduplications as in (9) is as shown in (10).

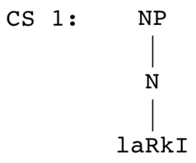
(10)	calnA valnA
	cal+Verb+Inf+Masc+Sg+Redup

That is, within the morphological analyzer, the fact that a reduplicated form has been encountered is simply registered via the tag +Redup and is passed on to the Urdu grammar, which can decide how to use this information, or whether to use the very subtle semantic information implied by reduplication at all.

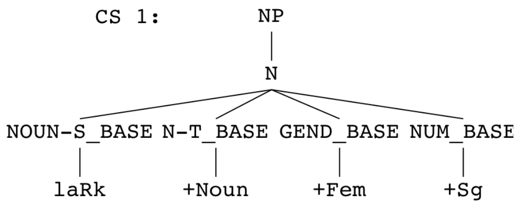
4.4. The morphology-syntax interface

The morphology-syntax interface between FSM and XLE is described in [21]. The methodology encompasses two basic points. For one the abstract morphological tags like +Noun or +Fem must be parsed via a series of sublexical rules within the Urdu grammar. The abstract tags must be parsed in exactly the order provided by the morphology, allowing, for +Fem+Sg, but never +Sg+Fem.

(11) a.



(11) b.



(11) illustrates a sample c-structure along with its sublexical parse in (11b), where the sublexical rules have determined that the stem laRk and the sequence of tags following it are a licit combination for a noun. A noun (N) is thus posited at c-structure and is parsed as part of the phrasal syntax rules, as in (11a) .

Furthermore, the functional information conveyed by the abstract morphological tags must be integrated into the LFG analysis. This is done by writing a lexicon of sublexical information, i.e., a lexicon which lists all of the tags from the morphology along with the information we would like in the syntax. The tag +Fem in (12), for example, is associated with an LFG equation indicating feminine gender.

(12) +Fem xle GEND (^ GEND) = fem.

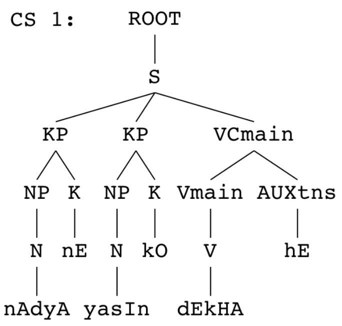
We need this further specification because the abstract tags provided by the morphological analyzer are application independent. The specifications in the sublexical lexicon file allow us to specify what exactly we believe the tags to mean within our application, namely a grammar for Urdu. We can also choose to ignore the tags coming out of the FSM by simply associating no information with them.

5. Syntax: F- and C-structure

Syntax is at the core of the grammar. At this stage, the grammar contains 40 annotated phrase-structure rules. Phenomena treated so far include basic clauses in all different possible word orders (as Urdu is a free word order language), the verbal complex and tense/aspect, causatives and complex predicates of various sorts [12,9] and an implementation of case [8]. Given that the major syntactic analyses for Urdu have been described elsewhere, we only provide a sample c- and f-structure analysis for (13).

(13)	nAdya=ne	yasIn=kO
	Nadya.F.Sg=Erg	Yassin.M.Sg=Acc
	dEkH-A	hE
	see-Perf.M.Sg	be.Pres.3.Sg
	"Nadya has seen Yassin."	

C-structure



The c-structure encodes the basic constituency structure and the linear precedence relations. For example, it encodes that the ergative case nE is part of nAdyA and that hE is combined with dEkhA. However, the c-structure does not include information on the functional characteristics of the tree elements. These are represented at the f-structure, which encodes, for example, that dEkHA 'see' is the main predicate, i.e., the head of the clause and that nAdyA is the subject and yasIn the object. nAdyA furthermore has the functional properties 'name', 'ergative', 'feminine' and 'third person singular'.

F-structure

```
"nAdyA nE yasIn kO dEkHA hE"

    PRED      'dEkH<[1:nAdyA], [18:yasIn]>'
              PRED      'nAdyA'
              CHECK     [_NMORPH obl]
    SUBJ      NTYPE     [NSEM [PROPER [PROPER-TYPE name]]]
                        [NSYN proper                     ]
              SEM-PROP  [SPECIFIC +]
             1CASE erg, GEND fem, NUM sg, PERS 3
              PRED      'yasIn'
              CHECK     [_NMORPH obl]
    OBJ       NTYPE     [NSEM [PROPER [PROPER-TYPE name]]]
                        [NSYN proper                     ]
              SEM-PROP  [SPECIFIC +]
            18CASE acc, GEND masc, NUM sg, PERS 3
    CHECK     [_VMORPH [_MTYPE infl]                       ]
              [GEND masc, NUM sg, PERS 3, _RESTRICTED -    ]
    LEX-SEM   [AGENTIVE +]
    TNS-ASP   [ASPECT [IMPF -, PERF +, PROG -]]
              [MOOD indicative, TENSE pres    ]
  35CLAUSE-TYPE decl, PASSIVE -, VFORM perf, VTYPE main
```

6. P-structure and Ezafe

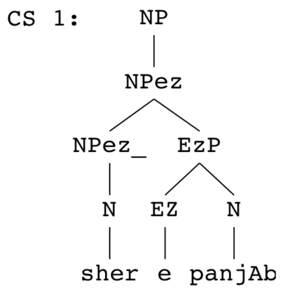
Phonological, especially prosodic information is of great value for the correct understanding of a sen- tence. Intonational material can help with disam- biguation and governs the interaction of differing clausal elements. In the Urdu grammar we are cur- rently experimenting with a p(rosodic)-projection in order to model the complex properties of clitics in general and Urdu Ezafe in particular.

In Urdu, the Ezafe is a loan construction from Persian, which calls for a modifier to the right of the head noun, a radical departure from the usual head-final pattern of the language. The modifier can be either a noun (14a) or an adjective (14b).

(14) a.	sher=e panjAb	b. sadA=e buland
	lion=Ez Punjab	voice=Ez high
	"A/The lion of Punjab"	"high voice"

Our c-structure analysis is shown in (15a). This models the fact that the adjective or noun modifying the head noun (sher 'lion'), is introduced and licensed by the Ezafe -e. The Ezafe is thus the head of n EzP constituent and takes a noun (e.g., panjAb, 'Punjab') or an adjective as a complement. The fact that the complement of Ezafe modifies the head noun is encoded at f-structure via the MOD feature.

(15) a. C-structure



(15) b. F-structure

```
"sher e panjAb"

    PRED    'sher'
    MOD     {   PRED    'panjAb'
                NTYPE   [NSEM [COMMON count]]
                        [NSYN common        ]
              30GEND masc, MOD-TYPE ezafe, NUM sg, PERS 3 }
    CHECK   [_EZAFE +]
    NTYPE   [NSEM [COMMON count]]
            [NSYN common        ]
   1GEND masc, NUM sg, PERS 3
```

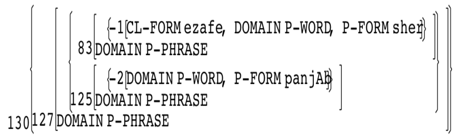
However, the complexities of Urdu Ezafe are not yet exhausted. Examples like the one in (16) from Iqbal show that the =e behaves like a clitic because it can only appear once at the end of a constituent (16), which is not possible for inflectional morphology.

(16) [maal o daulat]=e dunyaa
material and wealth=Ez world
"the material and wealth of the world"

Prosodically, Ezafe is thus part of the head noun to its left: as a clitic it is incorporated into the prosodic phrase to its left. Note also that if the phrase sher e panjAb is pronounced that the intonational break is after the Ezafe and not before it. However, this prosodic fact is modeled at neither c- nor f-structure and, indeed, cannot be modeled at these levels.

We are therefore experimenting with an additional level of representation for prosodic information, the p-structure. Thus, in the p-structure in (17), the Ezafe is encoded as CL-FORM ezafe where CL stands for 'clitic'. As a clitic, Ezafe has been incorporated into the prosodic phrase of the element to its left, sher. The modular architecture of LFG and XLE thus again allows for a powerful and transparent manner of encoding relevant linguistic information.

(17) P-Structure

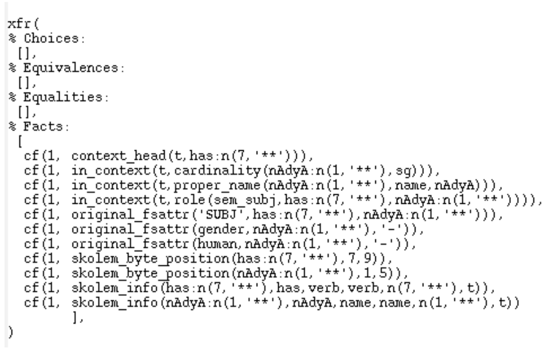

```
                 {-1[CL-FORM ezafe, DOMAIN P-WORD, P-FORM sher]}
              83[DOMAIN P-PHRASE
                 {-2[DOMAIN P-WORD, P-FORM panjAb]}
             125[DOMAIN P-PHRASE
130{127[DOMAIN P-PHRASE
```

7. XFR semantics

In this final section, we take a look at the representation of semantics, which is particularly of interest for NLP applications such as question-answer systems. With respect to the Urdu grammar, we are just beginning to build our first semantic representations. The way this works is to take the Prolog coding of the f-structure analysis provided by the XLE platform and to use this as the input for a semantic rewriting system. This is a reasonable way to proceed, as f-structures have been shown to be equivalent to quasi-logical forms [24]. In the course of the analysis, the input f-structure is thus progressively consumed by the rules and replaced by the output semantic representation. At this level, information about the world can also be included. The English grammar [16,13], for example, integrates information from WordNet. Furthermore, the semantic module can distinguish between the information state of verbs like believe vs. know as in Governor Palin believes/knows that Africa is a country.

As can be seen in the simple example for nAdyA hasI 'Nadya laughed' in (18), the semantic representation contains a list of Facts, which constitute the content of the semantic representation. Each fact is wrapped up in cf(Choice, Fact), where Choice indicates the part of the choice space in which Fact occurs. Because our example is unambiguous, all the semantic facts are to be found under choice 1, e.g. in all possible readings. Furthermore, there is a context (in context(Context, Proposition)), which de- notes a semantic context (possible world/situation). This allows one to evaluate propositions with respect to certain worlds, i.e., a world in which Palin knows something to be true, or a world in which she only believes something to be true. The top level or true context (the real world) is denoted by the term t.

(18) Semantics of nAdyA hasI 'Nadya laughed':


```
xfr(
% Choices:
 [],
% Equivalences:
 [],
% Equalities:
 [],
% Facts:
 [
  cf(1,  context_head(t,has:n(7,'**'))),
  cf(1,  in_context(t,cardinality(nAdyA:n(1,'**'),sg))),
  cf(1,  in_context(t,proper_name(nAdyA:n(1,'**'),name,nAdyA))),
  cf(1,  in_context(t,role(sem_subj,has:n(7,'**'),nAdyA:n(1,'**')))),
  cf(1,  original_fsattr('SUBJ',has:n(7,'**'),nAdyA:n(1,'**'))),
  cf(1,  original_fsattr(gender,nAdyA:n(1,'**'),'-')),
  cf(1,  original_fsattr(human,nAdyA:n(1,'**'),'-')),
  cf(1,  skolem_byte_position(has:n(7,'**'),7,9)),
  cf(1,  skolem_byte_position(nAdyA:n(1,'**'),1,5)),
  cf(1,  skolem_info(has:n(7,'**'),has,verb,verb,n(7,'**'),t)),
  cf(1,  skolem_info(nAdyA:n(1,'**'),nAdyA,name,name,n(1,'**'),t))
         ],
)
```

The semantic representation thus provides yet another module in which information can be encoded and used transparently and straightforwardly.

8. Conclusion

This paper has presented the modular architecture assumed by LFG/XLE. Using examples from Urdu and describing on-going research with respect to the development of an Urdu ParGram grammar, we showed how tokenization, morphological and syntactic analysis and semantic representation is dealt with within a pipe-line architecture. Furthermore, we showed how other possibly relevant information, such as the prosodic structure of a clause can be rep- resented. We conclude that the LFG/XLE methodology used as part of the ParGram grammar development effort is very powerful, versatile and effective.

9. References

[1] A. Abbi. Reduplication in South Asian Languages. An Areal, Topological and Historical Study. New Delhi: Allied, 1991.

[2] K. R. Beesley, and L. Karttunen. Finite State Morphology. CSLI Publications, 2003.

[3] D. G.Bobrow, , B. Cheslow, C. Condoravdi, L. Karttunen, T. H. King, R. Nairn, V. de Paiva, C. Price,and A. Zaenen. PARC's bridge and question answering system. In Grammar Engineering Across Frameworks, CSLI Publications, 2007, pp. 46–66.

[4] T. Bogel, M. Butt, A. Hautli, and S. Sulger. Developing a Finite-State Morphological Analyzer for Urdu and Hindi: Some issues. In FSMNLP07, 2007.

[5] J. Bresnan, Lexical-Functional Syntax. Oxford: Blackwell, 2000.

[6] M. Butt, H. Dyvik, T. H. King, H. Masuichi, and C. Rohrer. The Parallel Grammar Project. In Proceedings of the COLING-2002 Workshop on Grammar Engineering and Evaluation, 2002 pp. 1–7.

[7] M. Butt, and T. H. King. Interfacing phonology with LFG. In M. Butt and T. H. King (Eds.), Proceedings of the LFG98 Conference. CSLI, 1998.

[8] M. Butt, and T. H. King . The status of case. In V. Dayal and A. Mahajan (Eds.), Clause Structure in South Asian Languages,. Berlin: Springer Verlag, 2005, pp. 153–198.

[9] M. Butt, and T. H. King. Restriction for morphological valency alternations: The Urdu causative. In M. Butt, M. Dalrymple, and T. H. King (Eds.), Intelligent Linguistic Architectures: Variations on Themes by Ronald M Kaplan, CSLI Publications., 2006, pp. 235–258.

[10] M. Butt, and T. H. King. Urdu in a parallel grammar development environment. Language Resources and Evaluation, 2007a. 41, 191–207.

[11] M. Butt, and T. H. King. XLE and XFR: A grammar development platform with a parser/generator and rewrite system. ICON Tutorial, http://www2.parc.com/isl/members/-thking/icon-tutorial-2007. ppt, 2007b.

[12] M. Butt, T. H. King, and J. T. Maxwell III. Complex predicates via restriction. In M. Butt and T. H. King (Eds.), On-line Proceedings of the LFG03 Conference - 2003, pp. 92–104. CSLI Publications.

[13] M. Butt, T. H. King, M.-E. Nin˜o, and F. Segond. A Grammar Writer's Cookbook. CSLI.- 1999.

[14] A. Copestake, D. Flickinger, C. Pollard, and I. Sag. Minimal recursion semantics. Research on Language and Computation 3 - 2005, 281–332.

[15] R. Crouch, M. Dalrymple, R. Kaplan, T. H. King, J. Maxwell, and P. Newman . XLE documentation. 2007. http://www2.parc.com/-isl/groups/nltt/xle/doc/xle toc.html.

[16] R. Crouch, and T. H. King. Semantics via f- structure rewriting. In M. Butt and T. King (Eds.), Online Proceedings of LFG06. CSLI.- 2006.

[17] M. Dalrymple,. Lexical Functional Grammar. New York: 2001, Academic Press.

[18] E. H. Glassman,. Spoken Urdu. Lahore: Nirali Kitaben.- 1977.

[19] M. Humayoun, H. Hammarstr¨om, and A. Ranta. Urdu morphology, orthography and lexi- con extraction. In A. Farghaly and K. Megerdoomian (Eds.), Proceedings of the 2nd Workshop on Computational Approaches to Arabic Script-based Languages, 2007, pp. 59–66.

[20]Hussain, S.. Finite-state morphological analyzer for Urdu. Master's Thesis, CRULP, Lahore, 2004.

[21] R. M. Kaplan, J. T. Maxwell III, T. H. King, and R. Crouch. Integrating Finite-State Technology with Deep LFG Grammars. In Proceedings of the ESSLLI 2004 Workshop on Combining Shallow and Deep Processing for NLP, 2004.

[22] T. H. King, , M. Forst, J. Kuhn, and M. Butt. The feature space in parallel grammar writing. Re- search on Language and Computation 3, 2005, 139–163.

[23] A. Malik,. Hindi Urdu machine transliteration system. MSc Thesis, University of Paris 7, 2006.

[24] G. van, J. and R. Crouch. F-structures, QLFs and UDRSs. In M. Butt and T. King (Eds.), On-line Proceedings of the First International Conference on LFG. CSLI Publication, 1996.