Urdu in a Parallel Grammar Development Environment

Miriam Butt and Tracy Holloway King
Universität Konstanz and Palo Alto Research Center

June 2007

Abstract. In this paper, we report on the role of the Urdu grammar in the Parallel Grammar (ParGram) project (Butt et al., 1999; Butt et al., 2002). The Urdu grammar was able to take advantage of standards in analyses set by the original grammars in order to speed development. However, novel constructions, such as correlatives and extensive complex predicates, resulted in expansions of the analysis feature space as well as extensions to the underlying parsing platform. These improvements are now available to all the project grammars.

Keywords: Urdu, deep grammars, grammer engineering, parallel grammar development, LFG

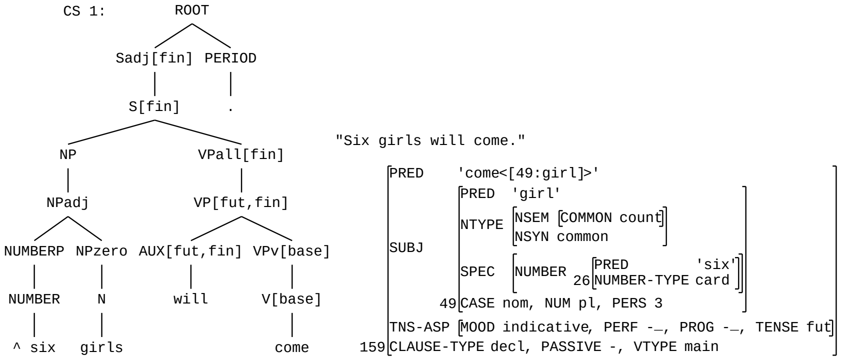
1. Introduction

In this paper, we report on the role of the Urdu grammar in the Parallel Grammar (ParGram) project (Butt et al., 1999; Butt et al., 2002). The ParGram project began with three closely related European languages: English, French, and German. Once grammars for these languages were established, two Asian languages were added: Japanese (Masuichi and Ohkuma, 2003) and Urdu.[1] Here we discuss the Urdu grammar and what special challenges it brought to the ParGram project. We are pleased to report that creating an Urdu grammar within the ParGram standards has been possible and has led to typologically useful extensions to the project and to the underlying grammar development platform.

The ParGram project uses the XLE parser and grammar development platform (Maxwell and Kaplan, 1993; Crouch et al., 2007) to develop deep, broad-coverage grammars for a variety of languages.[2] All of the grammars use the Lexical-Functional Grammar (LFG (Dalrymple, 2001)) formalism which produces constituent-structures (trees) and f(unctional)-structures (AVMs) as the syntactic analysis. The c-structure and f-structure for a simple English sentence is shown in (1); the output is from the broad-coverage English ParGram grammar (Kaplan et al., 2004b). It is the f-structure dependency

[1] The languages now also include Arabic, Chinese, Hungarian, Korean, Malagasy, Norwegian, Vietnamese, and Welsh. Some of these grammars are broad coverage grammars used in applications; some are still at initial stages of development; and some have been developed primarily to test aspects of linguistic theory.

[2] In general, these grammars have focused on edited, written texts such as newspaper text and manuals. The Urdu grammar is also geared towards such texts.



```
"Six girls will come."

PRED     'come<[49:girl]>'
           PRED  'girl'
           NTYPE  NSEM [COMMON count]
                  NSYN common
SUBJ
           SPEC   NUMBER   PRED          'six'
                        26 NUMBER-TYPE card
         49 CASE nom, NUM pl, PERS 3
TNS-ASP  MOOD indicative, PERF -_, PROG -_, TENSE fut
159 CLAUSE-TYPE decl, PASSIVE -, VTYPE main
```

Figure 1. C-structure tree and F-structure AVM for *Six girls will come.*

structure which is used in applications such as machine translation, sentence condensation, CALL, and question answering.[3]

LFG assumes a version of Chomsky's Universal Grammar hypothesis, namely that all languages are governed by similar underlying structures. Within LFG, f-structures are meant to encode a language universal level of analysis, allowing for cross-linguistic parallelism. The ParGram project aims to test the LFG formalism for its universality and coverage limitations and to see how far parallelism can be maintained across languages. Where possible, the analyses produced by the grammars for similar sentences in each language are parallel. This parallelism requires a standard for linguistic analysis. The standardization of the analyses has the computational advantage that the grammars can be used in similar applications and it can simplify cross-language applications.

The conventions developed within the ParGram grammars are extensive. The ParGram project dictates not only the form of the features used in the grammars, but also the types of analyses that are chosen for constructions (Butt et al., 2003a). These conventions are made accessible to the grammar writers by shared templates and feature declarations describing the feature space (Dalrymple et al., 2004b; King et al., 2005) and a few core shared rules (Kaplan et al., 2002), e.g. for coordination.[4] In addition, the XLE platform necessarily provides restrictions on how the grammars can be written. In all cases, the Urdu grammar has successfully incorporated the standards that were originally designed for the European languages. In addition, it has contributed to the formulation of new standards of analysis and implementations of formal devices. Below we discuss several aspects of this: morphology, lex-

[3] These structures can be manipulated via the ordered rewrite systems (transfer component) which is part of the XLE grammar development platform to make them more specialized for a given application.

[4] ParGram does not adopt a more pervasive grammar sharing approach such as that found in (Bender and Flickinger, 2005).

icon, and grammar development for the Urdu grammar within the ParGram project.

2. Morphology

Most of the grammars in the ParGram project depend on two-level finite-state morphologies as input (Beesley and Karttunen, 2002). Without this type of resource, it is extremely difficult to build large-scale grammars, especially for languages with substantial morphology (Kaplan et al., 2004a). For the original three languages (English, French, and German), such morphologies were readily available. As they had been developed for information extraction applications instead of deep grammar applications, there were some minor problems, but the coverage of these morphologies was excellent. An extremely efficient, broad-coverage tokenizer and morphology was also available for Japanese (Asahara and Matsumoto, 2000) and was integrated into the Japanese grammar. This has aided in the Japanese grammar rapidly achieving broad coverage (Masuichi et al., 2003). It has also helped to control ambiguity in the Japanese grammar because the morphology determines the part of speech of each word in the string with very little ambiguity.

No such finite-state morphology was available for Urdu or Hindi. As such, part of the Urdu project is to build a finite-state morphology that will serve as a resource to the Urdu grammar and can later be used in other applications. That is, although such a morphology is crucial to the Urdu grammar, it is independent of the grammar and hence can serve as a resource on its own. The development of the Urdu morphology is a two step process. The first step was to determine the morphological class of words and their subtypes in Urdu. The morphological paradigms which yield the best and most efficient generalizations had to be determined. Once the basic paradigms and morphological classes were identified and understood, the second step is to enter all words in the language with their class and subtype information. These two steps are described in detail below. Currently we are working on the second step.

The finite-state morphologies used in the ParGram project associate surface forms of words with a canonical form (a lemma) and a series of morphological tags that provide grammatical information about that form. An example for English is shown in (1) and for Urdu in (2).

(1) pushes: push +Verb +Pres +3sg
push +Noun +Pl

(2) bOlA bOl +Verb +Perf +Masc +Sg

(1) states the English surface form *pushes* can either be the third singular form of the verb *push* or the plural of the noun *push*. (2) states that the Urdu surface form *bOlA* is the perfect masculine singular form of the verb *bOl*.

The first step of writing a finite-state morphology for Urdu involves determining which tags are associated with which surface forms. As can be seen from the above examples, determining the part of speech (e.g., verb, noun, adjective) is not enough, at least not for writing deep grammars. For verbs, tense, aspect, and agreement features are needed. For nouns, number and gender information is needed, as well as information as to whether it is a common or proper noun. Once the set of relevant tags is chosen, the patterns of how the surface forms map to the stem-tag sets must be determined. For example, in English the stem-tag set *dog +Noun +Pl* corresponds to the surface form *dogs* in which a *s* is added to the stem, while *box +Noun +Pl* corresponds to *boxes* in which an *es* is added. The basic tag set for Urdu has been established, and the morphological paradigms that correspond to these tag combinations have been determined.

The second stage of the process involves greatly increasing the coverage of the morphology by adding in all the stems in Urdu and marking them for which set of tags and surface forms they appear with. This is a very large task. However, by using frequency lists for the language, the most common words can be added first to obtain a major gain in coverage.

In addition, a guesser can be added to guess forms of stems that the morphology does not yet recognize (Chanod and Tapanainen, 1995). This guessing is based on the morphological form of the surface form. For example, if a form ending in *A* is encountered and not recognized, it could be considered a perfect masculine singular form, similar to *bOlA* in (2). For inflecting languages like Urdu, a guesser can add significantly to initial coverage and provide information as to which words that occur in the development corpus still need to be entered into the morphology.

3. Lexicon

One advantage of the XLE system incorporating the large finite-state morphologies is that the lexicons for the languages can then be relatively small (Kaplan et al., 2004a). This is because lexicons are not needed for words whose syntactic lexical entry can be determined based on their morphological analysis. This is particularly true for nouns, adjectives, and adverbs.

Consider the case of nouns. The Urdu morphology provides the following analysis for the proper noun *nadya*.

(3) nAdyA +Noun +Name +Fem

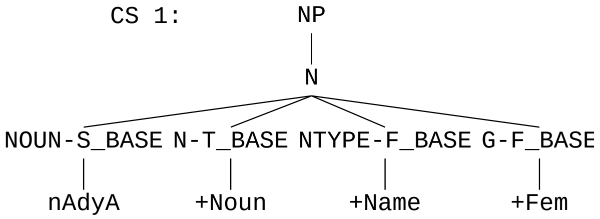
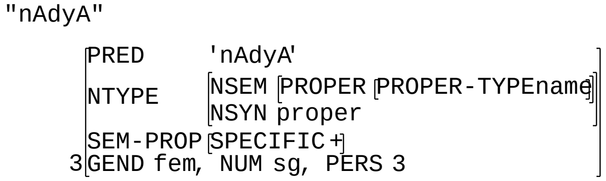


Figure 2. C-structure tree for *nAdyA*



```
"nAdyA"
   [PRED      'nAdyA'                                    ]
   [NTYPE     [NSEM [PROPER [PROPER-TYPE name]]]         ]
   [          [NSYN proper                    ]          ]
   [SEM-PROP  [SPECIFIC +]                               ]
  3[GEND fem, NUM sg, PERS 3                             ]
```

Figure 3. F-structure for *nAdyA*

The tags provide the information that it is a noun, in particular a type of proper noun (a person name), and is feminine. The lexical entries for the tags can then provide the grammar with all of the features that it needs to construct the analysis of *nadya*; this resulting f-structure analysis is seen in Figures 2 and 3. Thus, *nadya* itself need not be in the lexicon of the grammar because it is already known to the morphological analyzer.

Items whose lexical entry cannot be predicted based on the morphological tags need explicit lexical entries. This is generally the case for items whose subcategorization frames are not predictable, primarily for verbs. Currently, the Urdu verb lexicon is hand constructed and only contains a few verbs, generally one for each type of subcategorization frame for use in grammar testing. A sample entry for the verb *kah* 'say' which can be either transitive or take a complement clause is shown in (4).

(4) kah V-S XLE { (V-SUBJ-OBJ %stem) @AGENTIVE
| (V-SUBJ-OBJ-COMP %stem) @AGENTIVE }.

In order to build a broad-coverage Urdu grammar, a more complete verb lexicon is needed. To provide some idea of scale, the current English verb lexicon contains entries for 9,652 verbs; each of these has an average of 2.4 subcategorization frames, some verbs having as many as 15 frames; as such, there are 23,560 verb-subcategorization frame pairs. However, given that Urdu employs the strategy of productive syntactic complex predicate formation for much of its verbal predication, the verb lexicon for Urdu will be significantly smaller than its English counterpart (Rivzi, 2006). On the other hand, writing grammar rules which take care of the productive combinatorial possibilities between adjectives and verbs (e.g., *sAf karnA* 'clean do'='clean'), nouns and verbs (e.g., *yAd karnA* 'memory do'='remember') and verbs and verbs (e.g., *kHa lEnA* 'eat take'='eat up') required significant effort (section 4.2).

There are a number of ways to obtain a broad-coverage verb lexicon. One is to extract the information from electronic dictionaries, as was done for the English verb lexicon. This does not exist for Urdu, as far as we are aware, but see (Rivzi, 2006) for current developments. Another is to extract it from Urdu corpora, as was done for the German verb lexicon. Again, these would have to be either collected or created as part of the grammar development project. A final way is to enter the information by hand, depending on native speaker knowledge and print dictionaries; this option is very labor intensive and has generally been used to supplement the other techniques for high frequency verbs. Fortunately, work is being done on verb subcategorization frames in Hindi.[5] It is hoped that we can incorporate this information into the Urdu grammar verb lexicon.

4. Grammar

The current Urdu grammar is relatively small, comprising 33 rules (left-hand side categories) which compile into a collection of finite-state machines with 274 states and 423 arcs. The size of some other grammars in the ParGram project are shown in (5) for comparison. The number of rules is an arbitrary measure since the grammar writer can decide whether to collapse or break apart a given rule; the states and arcs reflect the size of the compiled grammar and hence give a better indication of grammar size. We are currently expanding the Urdu grammar to provide broad-coverage on standard (grammatical, written) texts. The current smaller size of the Urdu grammar shown in (5) is not a reflection of the difficulty of the language, but rather of the time put into it.[6] That is, comparable coverage is achieved in comparable time, despite typological differences between langauges. Below we discuss the Urdu grammar analyses and how they fit into the ParGram project standardization requirements.

[5] One significant effort is the Hindi Verb Project run by Prof. Alice Davison at the University of Iowa; further information is available via their web site.

[6] Unfortuantely, unlike the other grammars, there has been no full-time grammar writer on the Urdu grammar.

(5)

Language	Rules	States	Arcs
German	444	4883	15870
English	310	4935	13268
French	132	1116	2674
Japanese	50	333	1193
Norwegian	46	255	798
Urdu	33	274	423

To give the reader a feel for LFG grammar rules, one of the simpler rules from the Urdu grammar is shown in (7) for the core modifiers of common nouns; ( ) indicate optionality, * indicates zero or more instances, and @ indicates calls to templates shared across grammars.

```
(6)  Nmod –>  (KPposs)                          possessive
              (Q: @SPEC-QUANT)                  quantifier
              (NUMBER: @SPEC-NUMBER             numeral
                     (ˆ NUM)=(! NUM) )
              AP*: @ADJUNCT                     adjectives
                   (ˆ GEND)=(! GEND)
                   (ˆ NUM)=(! NUM)
                   @(ATYPE_desig ! attributive);
              N: ˆ =!.                          head noun
```

Even within a given linguistic formalism, LFG for ParGram, there is usually more than one way to analyze a construction. Moreover, the same theoretical analysis may have different possible implementations in XLE. These solutions generally differ in efficiency or conceptual simplicity. Whenever possible, the ParGram grammars choose the same analysis and the same technical solution for equivalent constructions. This was done, for example, with canonical imperatives: Imperatives are always assigned a null pronominal subject within the f-structure and a feature indicating that they are imperatives. While Urdu contains syntactic constructions which are not mirrored in the European languages, it does share many of the basic constructions, such as sentential complementation, control constructions, adjective-noun agreement, genitive specifiers, etc. The basic analysis of these constructions was determined in the initial stage of the ParGram project in writing the English, French, and German grammars. These analysis decisions have not had to be radically changed with the addition of typologically distinct Asian languages.

Parallelism, however, is not maintained at the cost of misrepresenting the language. Situations arise in which what seems to be the same construction in different languages cannot have the same analysis. An example of this is predicate adjectives (e.g., *It is red.*) (Dalrymple et al., 2004a). In English, the copular verb is considered the syntactic head of the clause, with the pronoun being the subject and the predicate adjective being an XCOMP. However, in Japanese, the adjective is the main predicate, with the pronoun being the subject. As such, these constructions receive non-parallel analyses.

In addition, many constructions which are stalwarts of English syntax do not exist as such in South Asian languages. Raising constructions with *seem*, for example, find no clear correlate in Urdu: the construction is translated via a psych verb in combination with a *that*-clause. This type of non-correspondence between European and South Asian languages raises quite a few challenges of how to determine parallelism across analyses. A similar example is the use of expletives (e.g., *There is a unicorn in the garden.*): these do not exist in Urdu and even in some European languages.

On the other hand, Urdu contains several syntactic constructions which find no direct correlate in the European languages of the ParGram project. Examples are correlative clauses (these are an old Indo-European feature which most modern European languages have lost), extensive use of complex predication, and rampant pro-drop which is not correlated with agreement or case features in Urdu, unlike in Italian. The analyses of these constructions have not only established new standards within the ParGram project, but have also guided the development of the XLE grammar development platform.

A sample analysis for the sentence in (7) is shown in Figures 4, 5, and 6.

(7) nAdyA kA kuttA AyA
Nadya Gen.M.Sg dog.Nom come.Perf.M.Sg
'Nadya's dog came.'

The parallelism in the ParGram project is primarily across the f-structure analyses which encode predicate-argument structure and other features that are relevant to syntactic analysis, such as tense and number.[7] The Urdu f-structure analysis of (7) is strikingly similar to that of the English equivalent. Both have a PRED for the verb which takes a SUBJ argument at the top level f-structure. This top level structure also has TNS-ASP features encoding tense and aspect information as well as information about the type of sentence (STMT-TYPE) and verb (VTYPE); these same features are found in the English structure. The analysis of the subject noun phrase is also the

[7] The c-structures are less parallel in that the languages differ significantly in their word order possibilities. Japanese and Urdu are SOV languages while English is an SVO language. However, the standards for naming the nodes in the trees and the types of constituents formed in the trees, such as NPs, are similar.

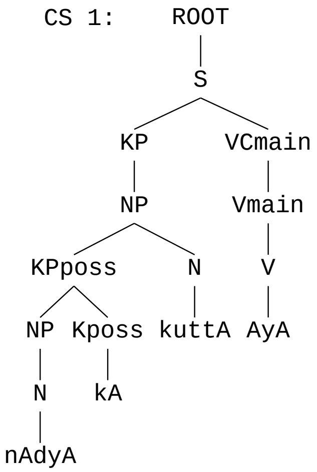


Figure 4. C-structure tree for (7) (sublexical morphology suppressed)

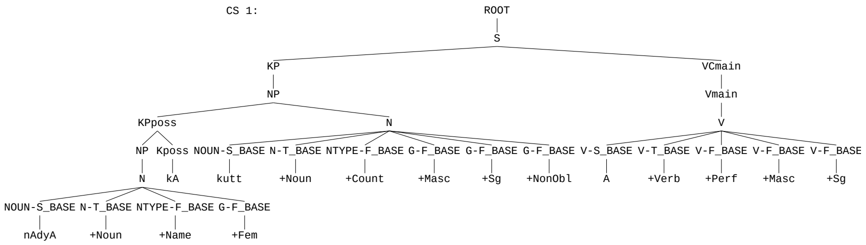
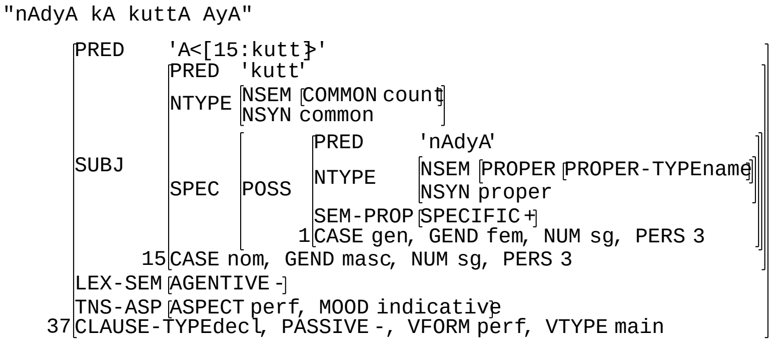


Figure 5. C-structure tree for (7) (sublexical morphology shown)

same as that in English, with the possessive being in the SPEC POSS and with features such as NTYPE, NUM, and PERS. The sentence in (7) involves an intransitive verb and a noun phrase with a possessive; these are both very basic constructions whose analysis was determined before the Urdu grammar



```
"nAdyA kA kuttA AyA"

   PRED    'A<[15:kutt]>'
           PRED  'kutt'
           NTYPE NSEM [COMMON count]
                 NSYN common
                           PRED     'nAdyA'
   SUBJ                    NTYPE    NSEM [PROPER [PROPER-TYPE name]]
           SPEC  POSS               NSYN proper
                           SEM-PROP [SPECIFIC +]
                         1 CASE gen, GEND fem, NUM sg, PERS 3
        15 CASE nom, GEND masc, NUM sg, PERS 3
   LEX-SEM [AGENTIVE -]
   TNS-ASP [ASPECT perf, MOOD indicative]
37 CLAUSE-TYPE decl, PASSIVE -, VFORM perf, VTYPE main
```

Figure 6. F-structure AVM for (7)

was written. Yet, despite the extensive differences between Urdu and the European languages—indeed, the agreement relations between the genitive and the head noun are complex in Urdu but not in English—there was no problem using this standard analysis for the Urdu construction.

4.1. CASE AND INSIDE-OUT FUNCTIONAL UNCERTAINTY

The analysis of case in Urdu posed more of a challenge. Although the ParGram features used in the analysis of case were sufficient for Urdu, there was a problem with implementing it. In Urdu, the case markers constrain the environments in which they occur (Butt and King, 2005b; Butt and King, 2005a). For example, the ergative marker *ne* only occurs on subjects. Note, however, that it is not the case that all subjects are ergative. To the contrary, subjects can occur in the ergative, nominative, dative, genitive, and instrumental cases. As such, we wanted to have the lexical entry for the ergative case state that it applies to a subject and similarly for other cases. This required the use of inside-out functional uncertainty (Kaplan, 1988) which had not been used in any of the other grammars. Inside-out functional uncertainty allows statements about the f-structure that contains an item. The lexical entry for *nE* is shown in (8).

(8) nE K @(CASE erg) `line 1`
(SUBJ ˆ) `line 2`
@VOLITION `line 3`

In (8), the K refers to the part of speech (a case clitic). Line 1 calls a template that assigns the CASE feature the value erg; this is exactly the same as how case is done in the other languages. Line 2 provides the inside-out functional uncertainty statement; it states that the f-structure of the ergative noun phrase, referred to as ˆ, is inside a SUBJ. Finally, line 3 calls a template that assigns volitionality features which ergative noun phrases are associated with. The analysis for (9) is shown in Figures 7 and 8.

(9) nAdyA nE yassin ko mArA
Nadya ERG Yassin ACC hit.Perf.M.Sg
'Nadya hit Yassin.'

There are two interesting points about this analysis of case in Urdu. The first is that although the Urdu grammar processes case differently than the other grammars, the resulting f-structure seen in Figure 8 is strikingly similar to its counterparts in English, German, etc. English would have CASE nom on the subject instead of erg, but the remaining structure is the same: the only indication of case is the CASE feature. The second point is that Urdu tested the application of inside-out functional uncertainty to case both theoretically and computationally. In both respects, the use of inside-out functional uncertainty has proven a success: not only is it theoretically desirable for languages like Urdu, but it is also implementationally feasible, providing the desired output.

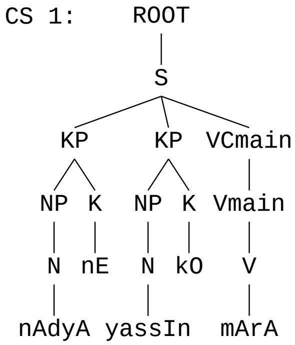


Figure 7. C-structure tree for (9) (sublexical morphology suppressed)

```
"nAdyA nE yassIn kO mArA"
      [PRED     'mAr<[1:nAdyA], [17:yassIn]>'                                ]
      [         [PRED     'nAdyA'                                   ]        ]
      [         [NTYPE    [NSEM [PROPER [PROPER-TYPE name]]]        ]        ]
      [SUBJ     [         [NSYN proper                     ]        ]        ]
      [         [SEM-PROP [SPECIFIC +]                              ]        ]
      [        1[CASE erg, GEND fem, NUM sg, PERS 3                 ]        ]
      [         [PRED     'yassIn'                                  ]        ]
      [         [NTYPE    [NSEM [PROPER [PROPER-TYPE name]]]        ]        ]
      [OBJ      [         [NSYN proper                     ]        ]        ]
      [         [SEM-PROP [SPECIFIC +]                              ]        ]
      [       17[CASE acc, GEND masc, NUM sg, PERS 3                ]        ]
      [LEX-SEM [AGENTIVE +]                                                  ]
      [TNS-ASP [ASPECT perf, MOOD indicative]                                ]
    33[CLAUSE-TYPE decl, PASSIVE -, VFORM perf, VTYPE main                   ]
```

Figure 8. F-structure AVM for (9)

4.2. Complex Predicates and the Restriction Operator

Another interesting case of how Urdu has extended the standards of the ParGram project comes from complex predicates and morphological causatives (these are discussed in detail in (Butt et al., 2003b) and (Butt and King, 2006a) respectively). The English, French, and German grammars had not needed a special complex predicate analysis.[8] However, as complex predicates form an

[8] German and possibly French have some complex predicate constructions. The ParGram grammars for these use a less linguistically satisfying complex clause analysis. The wider range of complex predicate phenomena in Urdu make this approach infeasible.

essential and pervasive part of the Urdu grammar, it was necessary to analyze them in the project. At first, we attempted to analyze complex predicates using the existing XLE tools. However, this proved to be impossible to do in a productive way because XLE did not allow for the manipulation of PRED values outside of the lexicon.[9] Given that complex predicates in Urdu are formed in the syntax and not the lexicon (Butt, 1995), this poses a significant problem. The syntactic nature of Urdu complex predicate formation is illustrated by (10), in which the two parts of the complex predicate *lıkH* 'write' and *dIya* 'gave' can be separated.

(10) a. [nAdyA nE] [saddaf kO] [kitAb] [**likHnE**
Nadya.F=Erg Saddaf.F=Dat book.F.Nom write.Inf.Obl

dI]
give.Perf.F.Sg
'Nadya let Saddaf write a book.'

b. nAdyA nE **dI** saddaf kO [kitAb **likHnE**]

c. nAdyA nE [kitAb **likHnE**] saddaf kO **dI**

The possibility of manipulating predicational structures in the lexicon via lexical rules (as is done for the passive, fn. 9), is therefore inadequate for complex predication. Based on the needs of the Urdu grammar, XLE has been modified to allow the analysis of complex predicates via the restriction operator (Kaplan and Wedekind, 1993) in conjunction with predicate composition in the syntax (Butt et al., 2003b). This restriction-based analysis was then extended to morphological causatives in Urdu which also require predicate composition (Butt and King, 2006b).

From the computational perspective, the problem can be restated as one by which the f-structural subcategorization frame of the main verb needs to be manipulated in order to take the contribution of the light verb into account. Consider the Urdu permissive from the perspective of a restriction analysis. The effect of the permissive light verb is to "add" a new subject to the predication and to "demote" the main verb's subject to a dative-marked indirect object. The sample lexical entries for the light verb 'give' and the main verb 'write' are given in (11) and (12), respectively.

(11) (ˆ PRED) = ′dE<(ˆ SUBJ), %PRED2>′

(12) (ˆ PRED) = ′likH<(ˆ SUBJ), (ˆ OBJ)>′

[9] XLE implements lexical rules which can be used to delete and rename arguments, e.g. for the English passive in which the OBJ becomes the SUBJ and the SUBJ becomes the OBL-AG. However, adding arguments and composing PREDs is not possible.

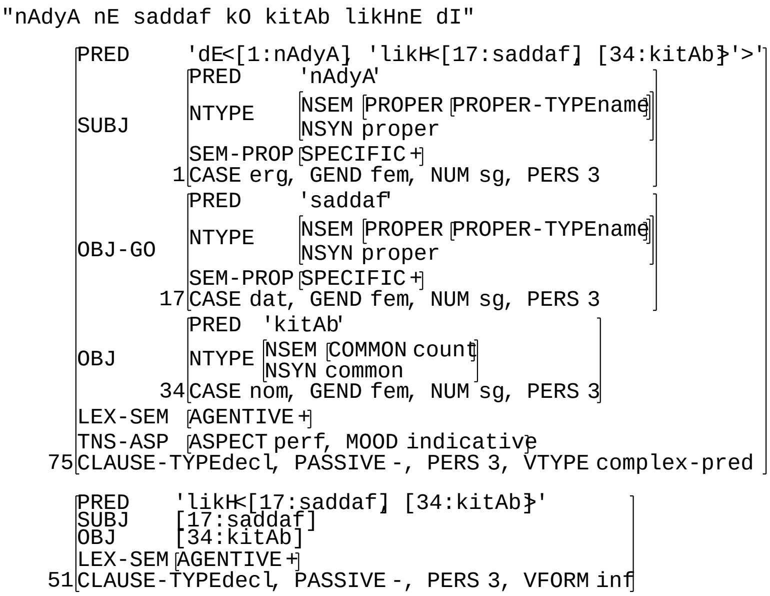
Rather than being analyzed as a three-place predicate, the permissive *dE* ‘give’ is rendered as a two-place predicate, in which the second argument is a local variable, %PRED2 whose value is assigned in the syntax.

In order to compose the two verbs, restriction is used as part of the f-structure annotations on phrase structure rules. The rule in (13) shows the restriction operator within the c-structure rule for a complex predicate. In particular, the restriction on the V node is what allows the composition of the new PRED. The annotation states that the up node (ˆ) comprising the complex predicate is the same as the down node (!) comprising the main verb, except that the SUBJ of the main verb is restricted out, as are the SUBJ and thematic object (OBJ-GO). This allows the former subject of ‘write’ to be identified as an OBJ-GO, via the (ˆ OBJ-GO)=(! SUBJ) equation in (13).

```
(13)                         (likHnE)                    (dI)

     V ⟶                        V                        Vlight
            !\SUBJ\PRED=ˆ\SUBJ\OBJ-GO\PRED               ˆ=!
                   (ˆ PRED ARG2)=(! PRED)
                      (ˆ OBJ-GO)=(! SUBJ)
```

In the final complex f-structure, the predicates *dE* ‘give’ and *likH* ‘write’ have been composed. The “embedded” SUBJ ‘Nadya’ has been restricted out as part of the composition. This is shown in Figure 9.



```
"nAdyA nE saddaf kO kitAb likHnE dI"

   [PRED      'dE<[1:nAdyA], 'likH<[17:saddaf], [34:kitAb]>'>'                ]
   |          [PRED      'nAdyA'                                ]              |
   |          |NTYPE     [NSEM [PROPER [PROPER-TYPE name]]]      |              |
   |SUBJ      |          [NSYN proper                     ]      |              |
   |          |SEM-PROP [SPECIFIC +]                            |              |
   |         1[CASE erg, GEND fem, NUM sg, PERS 3               ]              |
   |          [PRED      'saddaf'                               ]              |
   |          |NTYPE     [NSEM [PROPER [PROPER-TYPE name]]]      |              |
   |OBJ-GO    |          [NSYN proper                     ]      |              |
   |          |SEM-PROP [SPECIFIC +]                            |              |
   |        17[CASE dat, GEND fem, NUM sg, PERS 3               ]              |
   |          [PRED  'kitAb'                          ]                        |
   |OBJ       |NTYPE [NSEM [COMMON count]]            |                        |
   |          |      [NSYN common         ]           |                        |
   |        34[CASE nom, GEND fem, NUM sg, PERS 3     ]                        |
   |LEX-SEM   [AGENTIVE +]                                                     |
   |TNS-ASP   [ASPECT perf, MOOD indicative]                                   |
 75[CLAUSE-TYPE decl, PASSIVE -, PERS 3, VTYPE complex-pred                    ]

   [PRED     'likH<[17:saddaf], [34:kitAb]>'        ]
   |SUBJ     [17:saddaf]                            |
   |OBJ      [34:kitAb]                             |
   |LEX-SEM [AGENTIVE +]                            |
 51[CLAUSE-TYPE decl, PASSIVE -, PERS 3, VFORM inf ]
```

Figure 9. F-structure AVM for (10)

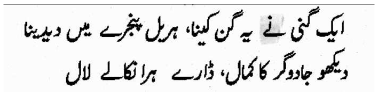
Thus, restriction allows f-structures and predicates to be manipulated in a controlled and detailed fashion, allowing for the implementation of Urdu complex predicates within the ParGram framework. As complex predicates are pervasive across languages, the Urdu implementation is expected to be adopted as other languages join the project.

5. Script

One issue that has not been dealt with in the Urdu grammar is the different script systems used for Urdu and Hindi. As seen in the previous discussions and the Figures, transcription into Latin ASCII is currently being used by the Urdu grammar. Note that this is not a limitation of the XLE system. The Japanese, Chinese, and Arabic grammars have successfully integrated the necessary scripts into their grammar.

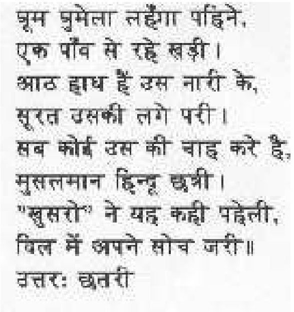
The approach taken by the Urdu grammar is different, largely because two scripts are involved. The Urdu grammar uses the ASCII transcription in the finite-state morphologies and the grammar. At a future date, a version of Malik's finite-state transliteration component will be built onto the grammar system (Malik, 2006). This system takes Urdu (Arabic) and Hindi (Devanagari) scripts and transcribes them for use in the grammar. This component will be written using finite-state technology and hence will be fully compatible with the finite-state morphology used by the grammar. The use of ASCII in the morphology allows the same basic morphology to be used for both Urdu and Hindi. Samples of the scripts are seen in (14a) for Urdu and (14b) for Hindi.

(14) a.



ایک گنی نے یہ گن کینا، ہریل پنجرے میں دیدینا
دیکھو جادوگر کا کمال، ڈارے ہرا نکالے لال

b.



घूम घुमेला लहँगा पहिने,
एक पाँव से रहे खड़ी।
आठ हाथ हैं उस नारी के,
सूरत उसकी लगे परी।
सब कोई उस की चाह करे है,
मुसलमान हिन्दू छत्री।
"खुसरो" ने यह कही पहेली,
दिल में अपने सोच जरी॥
उत्तर: छतरी

6. Discussion and Conclusion

The ParGram project was designed to use a single grammar development platform and a unified methodology of grammar writing to develop large-scale grammars for typologically different languages. At the beginning of the project, three typologically similar European grammars were used to test this idea. The addition of several languages, including Urdu, has shown that the basic analysis decisions made for the European languages can be applied to typologically distinct languages. However, Urdu required the addition of new standard analyses to the project to cover constructions and analysis techniques not found in the European languages, in particular restriction for predicate composition and inside-out functional understainty for case assignment. With this new set of standards, the ParGram project has now been able to be applied to yet other typologically distinct languages.

Once the Urdu grammar is appropriately scaled, a situation largely dependent on the completion of the Urdu FST morphology to improve lexical coverage, then detailed evaluation can be performed. Evaluation of the ParGram LFG grammars has focused on accuracy measures against industry-determined standards such as the Penn Treebank for English and the Tiger Treebank for German. To evaluate against these resources, dependency banks are semi-automatically built for the treebanks (see (Cahill et al., 2005) and references therein for a general approach and (Forst, 2003b; Forst, 2003a; Forst et al., 2004) on German). In addition, gold standard dependency banks, like the PARC700 for English (King et al., 2003), have been built for some languages.[10] The f-structures produced by the grammar are then compared against the dependency bank, giving standard f-score and precision and recall statistics (general technique (Crouch et al., 2002); English (Kaplan et al., 2004b); German (Rohrer and Forst, 2006b; Rohrer and Forst, 2006a))).

The ParGram grammars often produce multiple analyses for a given sentence. For applications that need only a single parse (or n-best parses) as input, stochastic disambiguators using maximum entropy models can be trained for the grammars (Riezler et al., 2002; Forst, 2007). The output of the stochastic disambiguation can then be tested against the dependency gold standard. This allows a measure of how well the parser will perform on open text in applications needing a single parse.

In addition to evaluating accuracy of the ParGram grammars, for many applications speed is also a factor. XLE (Crouch et al., 2007) provides a number of "performance variables" that can be set to limit the time and memory used in different parts of the parser. These can be set for a given corpus to allow for greater efficiency, possibly balanced by a slight lose in accuracy. Experiments

[10] The Japanese grammar (Masuichi and Ohkuma, 2003) was also evaluated against the Japanese *bunsetsu* standard which is a type of dependency measure; see (Masuichi et al., 2003) for details.

on the English grammar show that broad-coverage ParGram grammars can perform similarly to state-of-the-art tree parsers (Kaplan et al., 2004b) in terms of time, while providing more detailed dependency structures. Based on the results for English, German, and Japanese, we hope to develop a similar quality and coverage Urdu grammar which can be evaluated with the same techniques used more generally for dependency parsers.

The parallelism between the grammars in the ParGram project can be exploited in applications using the grammars: the fewer the differences, the simpler a multi-lingual application can be. For example, a translation system that used the f-structures as input and output could take advantage of the fact that similar constructions have the same analysis and same set of features (Frank, 1999; Riezler and Maxwell, 2006). In addition, applications such as sentence condensation (Riezler et al., 2003; Crouch et al., 2004) and CALL (Khader, 2003) which are developed for one language can be more easily be ported to the other languages, as can post-processing of grammars into semantic structures (Crouch and King, 2006; Umemoto, 2006). The standardization also aids further grammar development efforts. Many of the basic decisions about analyses and formalism have already been made in the project. Thus, the grammar writer for a new language can use existing technology to bootstrap a grammar for the new language and can parse equivalent constructions in the existing languages to see how to analyze a construction. This allows the grammar writer to focus on more difficult constructions not yet encountered in the existing grammars.

Acknowledgements

We would like to thank the audience of the COLING Workshop on Asian Languages in which an earlier version of this paper appeared (Butt and King, 2002) and three anonymous reviewers who provided detailed comments. We would also like to thank John Maxwell for extensive discussion and implementation help over the years.

References

Asahara, M. and Y. Matsumoto: 2000, ‘Extended Models and Tools for High-performance Part-of-Speech Tagger’. In: *Proceedings of COLING*.

Beesley, K. and L. Karttunen: 2002, *Finite-State Morphology: Xerox Tools and Techniques*. Cambridge University Press. To Appear.

Bender, E. and D. Flickinger: 2005, ‘Rapid Prototyping of Scalable Grammars: Towards Modularity in Extensions to a Langauge-Independent Core’. In: *Proceedings of IJCNLP-05 (Posters/Demos)*.

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

Butt, M., H. Dyvik, T. H. King, H. Masuichi, and C. Rohrer: 2002, 'The Parallel Grammar Project'. In: *Proceedings of COLING 2002*. pp. 1–7. Workshop on Grammar Engineering and Evaluation.

Butt, M., M. Forst, T. H. King, and J. Kuhn: 2003a, 'The Feature Space in Parallel Grammar Writing'. In: *ESSLLI 2003 Workshop on Ideas and Strategies for Multilingual Grammar Development*.

Butt, M. and T. H. King: 2002, 'Urdu and the Parallel Grammar Project'. In: *Proceedings of COLING 2002*. pp. 39–45. Workshop on Asian Language Resources and International Standardization.

Butt, M. and T. H. King: 2005a, 'Case Systems: Beyond Structural Distinctions'. In: *New Perspectives on Case Theory*. CSLI Publications, pp. 53–87.

Butt, M. and T. H. King: 2005b, 'The Status of Case'. In: V. Dayal and A. Mahajan (eds.): *Clause Structure in South Asian Languages*. Kluwer.

Butt, M. and T. H. King: 2006a, 'Restriction for Morphological Valency Alternations: The Urdu Causative'. In: M. Butt, M. Dalrymple, and T. H. King (eds.): *Intelligent Linguistic Architecturs: Variations on Themes by Ronald M Kaplan*. CSLI Publications, pp. 235–258.

Butt, M. and T. H. King: 2006b, 'Restriction for Morphological Valency Alternations: The Urdu Causative'. In: *Intelligent Linguistic Architectures: Variations on Themes by Ronald M Kaplan*. CSLI Publications, pp. 235–258.

Butt, M., T. H. King, and J. T. Maxwell: 2003b, 'Complex Predicates via Restriction'. In: *Proceedings of the LFG03 Conference*.

Butt, M., T. H. King, M.-E. Niño, and F. Segond: 1999, *A Grammar Writer's Cookbook*. CSLI Publications.

Cahill, A., M. Forst, M. Burke, M. McCarthy, R. O'Donovan, C. Rohrer, J. van Genabith, and A. Way: 2005, 'Treebank-Based Acquisition of Multilingual Unification Grammar Resources'. *Journal of Research on Language and Computation; Special Issue on Shared Representations in Multilingual Grammar Engineering* pp. 247–279.

Chanod, J.-P. and P. Tapanainen: 1995, 'Creating a tagset, lexicon, and guesser for a French tagger'. In: *Proceedings of the ACL SIGDAT Workshop: From Texts To Tags. Issues in Multilingual Language Analysis*. pp. 58–64.

Crouch, D., M. Dalrymple, R. Kaplan, T. H. King, J. Maxwell, and P. Newman: 2007, 'XLE Documentation'. Available on-line at http://www2.parc.com/isl/groups/nltt/xle/doc/xle_toc.html.

Crouch, D. and T. H. King: 2006, 'Semantics via F-structure Rewriting'. In: *Proceedings of LFG06*.

Crouch, R., R. Kaplan, T. H. King, and S. Riezler: 2002, 'A Comparison of Evaluation Metrics for a Broad Coverage Parser'. In: *Workshop on Beyond PARSEVAL at the Language Resources and Evaluation Conference*.

Crouch, R., T. H. King, J. T. Maxwell, S. Riezler, and A. Zaenen: 2004, 'Exploiting F-structure Input for Sentence Condensation'. In: *Proceedings of LFG04*. pp. 167–187.

Dalrymple, M.: 2001, *Lexical Functional Grammar*, Vol. 34 of *Syntax and Semantics*. Academic Press.

Dalrymple, M., H. Dyvik, and T. H. King: 2004a, 'Copular Complements: Closed or Open?'. In: *Proceedings of the LFG04 Conference*.

Dalrymple, M., R. Kaplan, and T. H. King: 2004b, 'Linguistic Generalizations over Descriptions'. In: *Proceedings of the LFG04 Conference*.

Forst, M.: 2003a, 'Treebank Conversion – Creating a German f-structure bank from the TIGER Corpus'. In: *Proceedings of the LFG03 Conference*.

Forst, M.: 2003b, 'Treebank Conversion – Establishing a testsuite for a broad-coverage LFG from the the TIGER Treebank'. In: *Proceedings of the EACL Workshop on Linguistically Interpreted Corpora (LINC '03)*.

Forst, M.: 2007, 'Disambiguation for a Linguistically Precise German LFG Parser'. Ph.D. thesis, IMS Stuttgart. in press.

Forst, M., N. Bertomeu, B. Crysmann, F. Fouvry, S. Hansen-Schirra, and V. Kordoni: 2004, 'Towards a dependency-based gold standard for German parsers - The TiGer Dependency Bank'. In: *Proceedings of the COLING Workshop on Linguistically Interpreted Corpora (LINC '04)*.

Frank, A.: 1999, 'From Parallel Grammar Development towards Machine Translation'. In: *Proceedings of MT Summit VII*. pp. 134–142.

Kaplan, R.: 1988, 'Correspondences and their Inverses'. Presented at the Titisee Workshop on Unification Formalisms: Syntax, Semantics, and Implementation, Titisee, Germany.

Kaplan, R., T. H. King, and J. Maxwell: 2002, 'Adapting Existing Grammars: The XLE Experience'. In: *Proceedings of COLING2002, Workshop on Grammar Engineering and Evaluation*. pp. 29–35.

Kaplan, R., J. T. Maxwell, T. H. King, and R. Crouch: 2004a, 'Integrating Finite-state Technology with Deep LFG Grammars'. In: *Proceedings of the Workshop on Combining Shallow and Deep Processing for NLP (ESSLLI)*.

Kaplan, R. and J. Wedekind: 1993, 'Restriction and Correspondence-based Translation'. In: *Proceedings of the Sixth European Conference of the Association for Computational Linguistics*. pp. 193–202.

Kaplan, R. M., S. Riezler, T. H. King, J. T. Maxwell, A. Vasserman, and R. Crouch: 2004b, 'Speed and Accuracy in Shallow and Deep Stochastic Parsing'. In: *Proceedings of the Human Language Technology Conference and the 4th Annual Meeting of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL'04)*.

Khader, R.: 2003, 'Evaluation of an English LFG-based Grammar as Error Checker'. MSc thesis, UMIST.

King, T. H., R. Crouch, S. Riezler, M. Dalrymple, and R. Kaplan: 2003, 'The PARC700 dependency bank'. In: *Proceedings of the EACL03: 4th International Workshop on Linguistically Interpreted Corpora (LINC-03)*.

King, T. H., M. Forst, J. Kuhn, and M. Butt: 2005, 'The Feature Space in Parallel Grammar Writing'. *In Research on Language and Computation* **3**(2), 139–163.

Malik, A.: 2006, 'Hindi Urdu Machine Transliteration System'. MSc Thesis, University of Paris 7.

Masuichi, H. and T. Ohkuma: 2003, 'Constructing a practical Japanese parser based on Lexical-Functional Grammar'. *Journal of Natural Language Processing* **10**, 79–109. In Japanese.

Masuichi, H., T. Ohkuma, H. Yoshimura, and Y. Harada: 2003, 'Japanese parser on the basis of the Lexical-Functional Grammar Formalism and its Evaluation'. In: *Proceedings of The 17th Pacific Asia Conference on Language, Information and Computation (PACLIC17)*. pp. 298–309.

Maxwell, J. T. and R. Kaplan: 1993, 'The Interface between Phrasal and Functional Constraints'. *Computational Lingusitics* **19**, 571–589.

Riezler, S., T. H. King, R. Crouch, and A. Zaenen: 2003, 'Statistical sentence condensation using ambiguity packing and stochastic disambiguation methods for Lexical-Functional Grammar'. In: *Proceedings of the Human Language Technology Conference and the 3rd Meeting of the North A merican Chapter of the Association for Computational Linguistics*.

Riezler, S., T. H. King, R. Kaplan, D. Crouch, J. Maxwell, and M. Johnson: 2002, 'Parsing the Wall Street Journal using a Lexical-Functional Grammar and Discriminative Estimation Techniques'. In: *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Riezler, S. and J. T. Maxwell: 2006, ‘Grammatical Machine Translation’. In: *Proceedings of Human Language Technology Conference - North American chapter of the Association for Computational Linguistics*.

Rivzi, S. M. J.: 2006, ‘Development of Algorithms and Computational Grammar of Urdu for the Machine Translation between English and Urdu Languages’. Ph.D. thesis, Pakistan Institute of Engineering and Applied Sciences.

Rohrer, C. and M. Forst: 2006a, ‘Broad-coverage Grammar Development – How Far Can It Go?’. In: M. Butt, M. Dalrymple, and T. H. King (eds.): *Intelligent Linguistic Architectures – Variations on Themes By Ronald M. Kaplan*. CSLI Publications.

Rohrer, C. and M. Forst: 2006b, ‘Improving coverage and parsing quality of a large-scale LFG for German’. In: *Proceedings of the Language Resources and Evaluation Conference (LREC-2006)*. Genoa, Italy.

Umemoto, H.: 2006, ‘Implementing a Japanese Semantic Parser Based on Glue Approach’. In: *Proceedings of The 20th Pacific Asia Conference on Language, Information and Computation*.