

A Computational Grammar for Georgian

Paul Meurer

Aksis, UNIFOB, University of Bergen
paul.meurer@uib.no

Abstract. In this paper, I give an overview of an ongoing project which aims at building a full-scale computational grammar for Georgian in the Lexical Functional Grammar framework and try to illustrate both practical and theoretical aspects of grammar development. The rich and complex morphology of the language is a major challenge when building a computational grammar for Georgian that is meant to be more than a toy system. I discuss my treatment of the morphology and show how morphology interfaces with syntax. I then illustrate how some of the main syntactic constructions of the language are implemented in the grammar. Finally, I present the indispensable tools that are used in developing the grammar system: *fst*; the XLE parsing platform, the LFG Parsebanker, and a large searchable corpus of non-fiction and fiction texts.

Key words: Georgian, Lexical-Functional Grammar, XLE, computational grammar, treebanking.

1 Introduction

In this paper, I give an overview of an ongoing project which aims at building a full-scale computational grammar for Georgian in the Lexical Functional Grammar (LFG) framework [1]. The grammar is part of the international ParGram project ([2], [3], [4]), which coordinates the development of LFG grammars in a parallel manner using the XLE (Xerox Linguistic Environment) grammar development platform developed by the Palo Alto Research Center.¹ In its current state, the grammar has a large lexicon and most of the morphology as well as most basic and some more advanced syntactic constructions are covered.²

In the first part, I describe the lexicon and morphology part of the grammar. I then illustrate how some of the main syntactic constructions of the language are implemented in the grammar and also touch upon some issues of theoretical interest. Finally, I present the indispensable tools that are used in developing the grammar system: *fst*; the XLE parsing platform; and the LFG Parsebanker.

2 Morphology

The standard tool for morphological analysis with the XLE platform is the Xerox finite state tool (*fst*) [5]. *fst* integrates seamlessly with XLE, and it is very fast.

¹ See http://www.parc.com/research/projects/natural_language/

² It is however premature to give coverage figures of the grammar on unrestricted text.

Transducers written in *fst* are reversible, they can be used both for analysis and generation.

The lexical input to the Georgian morphological transducer was taken mainly from a digitized version of Kita Tschenkéli's *Georgisch-deutsches Wörterbuch* [6], which is one of the best Georgian dictionaries, and particularly well-suited as a basis for computational work because of its superb presentation of the verbs. Currently, the base form lexicon of the transducer comprises more than 74,000 nouns and adjectives and 3,800 verb roots.³

A prominent feature of Georgian morphology is long-distance dependencies, in the sense that affixes before the verb root license other affixes after the root. Such long-distance dependencies are difficult to model in finite-state calculus, since in the traversal of a finite-state network, no memory is kept of states traversed earlier (i.e. affixes encountered); transitions at a later stage in a traversal cannot be licensed by earlier steps in the traversal. In order to overcome these difficulties and to enlarge the expressiveness of the calculus, *fst* uses a device called *flag diacritics*. Flag diacritics are named flags that can be set, checked and otherwise manipulated in the course of network traversal; they can be used as a memory of encountered earlier stages and thus are well-suited for the treatment of long-distance dependencies. Flag diacritics can be compiled out of the network, yielding a possibly larger, but pure finite state transducer.

The output of an *fst* parse of a given word form is a set of analyses, each consisting of a lexicon entry form, which serves as a lookup key in the LFG grammar's lexicon (see below), plus LFG-relevant morphosyntactic features. Relevant features for nouns include case, number, full vs. reduced case inflection, double declension case and number, animateness, postpositions and various clitics. Features for verbs include tense/mood, person and number marking (encoded as +Subj/+Obj), and verb class. Examples:

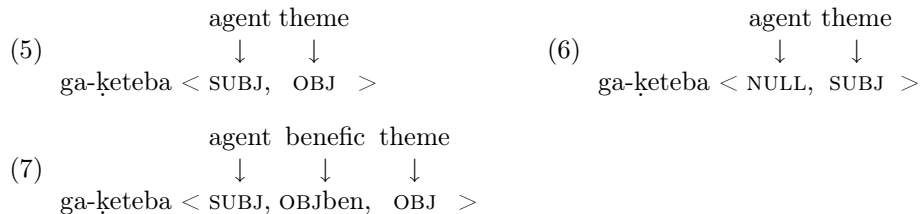
- (1) *ǰvino* 'wine'
→ ǰvino+N+Nom+Sg
- (2) *gogo-eb-isa-tvis-ac* 'for the girls, too'
→ gogo+N+Anim+Full+Gen+Pl+Tvis+C
- (3) *bavšvob-isa-s* 'in childhood'
→ bavšvoba+N+DGen+DSg+Dat+Sg
- (4) *da-mi-xaṭ-av-s* 'I apparently painted it'/'he will paint it for me'
→ { da-xaṭva-3569-5+V+Trans+Perf+Subj1Sg+Obj3
| da-xaṭva-3569-18+V+Trans+Perf+Subj1Sg+Obj3
| da-xaṭva-3569-18+V+Trans+Fut+Subj3Sg+Obj1Sg }

3 Morphosyntax

In Lexical-Functional Grammar, each verb is associated with a set of subcategorization frames (argument structures) and a mapping of each of the arguments

³ I would like to thank Yolanda Marchev, the co-author of the *Georgisch-Deutsches Wörterbuch*, for kindly allowing me to use the material of the lexicon in this project, and Levan Chkhaidze for giving me access to noun and adjective lists.

(thematic roles) in the argument structures to a grammatical function such as subject or object. The argument-to-function mapping is subject to morphosyntactic alternations and hence differs for example between the transitive and the passive form of a verb. Morphosemantic alternations like causativization or formation of the applicative alter the argument structure itself and delete thematic roles or introduce new ones. For example, a basic transitive verb like *ga-v-a-ḳet-eb* ‘I will do it’ (Tschenkéli Class T¹, see below) will have argument structure and associated grammatical functions as displayed in (5), whereas its passive alternation *ga-ḳet-d-eb-a* ‘it will be done’ (Class P²) is described by (6). The mapping of the applicative transitive *ga-v-u-ḳet-eb* ‘I will do it for him/her’ (Class T³) is given in (7).



Verb entries together with argument structure information for the basic alternations are coded in the LFG lexicon, which is consulted by the XLE parser to instantiate the parse chart. The thematic roles themselves that a given verb is associated with are not made explicit in XLE-based LFG grammars; only the grammatical functions they are mapped to are coded in the LFG lexicon. Argument-to-function mappings of morphosyntactic and morphosemantic alternations are derived in the grammar with the help of lexical transformation rules; their application is triggered by morphological features of the surface verb form.

In addition to argument structure, a lexical entry also stores the verb class. In combination with tense information, which is supplied by the tense feature of the morphological analysis, the verb class is needed to determine case alignment and mapping of morphological tense to tense/aspect features. Nouns, adjectives and other word classes are stored in a similar way.

Traditionally, Georgian verbs are classified into four main classes, according to a combination of morphological and case alignment criteria. These criteria can roughly be stated as follows: Verbs in Class I have an ergative subject in the aorist, and they form their future by adding a preverb. Class I verbs are transitive. Class II verbs, too, form their future by the addition of a preverb, but the subject of these verbs is always in the nominative. These verbs are intransitive and mostly passive or unaccusative. Verbs in Class III exhibit the same case alignment as Class I verbs, yet they have no own future forms, but rather recruit their future from related Class I paradigms. These verbs are unergative, or, less often, transitive. The verbs in Class IV are called indirect; their experiencer subject is invariably in the dative. Also this verb class lacks an own future paradigm, it uses forms from related Class II paradigms. (See (15) for details on the alignment patterns of these verb classes.)

The verb classification in Kita Tschenkéli's *Georgisch-deutsches Wörterbuch* follows this classification – the corresponding classes are called T, P (RP), MV (RM) and IV –, yet it is more fine-grained: information about the nature of indirect objects is also coded. Therefore, Tschenkéli's classification could be used directly to automatically derive a preliminary version of the Georgian LFG verb lexicon. For example, Tschenkéli's Class T³ maps to the argument structure P<SUBJ, OBJ, OBJben>, and RP¹ maps to P<SUBJ, OBJth>, where P is an arbitrary predicate.

In many cases, however, the correct frames are not (easily) deducible from Tschenkéli's classification and have to be added or corrected manually. Examples are:

Verbs taking oblique or genitive arguments:

- (8) ča-tvla<SUBJ, OBJ, OBL_{adv}> 'to consider sb. to be sth.'
 še-šineba<SUBJ, OBJ_{gen}> 'to be afraid of sb./sth.'

Class III verbs: Many of them can be transitive and intransitive (unergative), whereas some are only transitive and others only intransitive. This information is not available in the dictionary. For example:

- (9) tamaši<SUBJ, (OBJ)> 'to play (a game)'
 ga-qidva<SUBJ, OBJ> 'to sell sth.'
 ča-svla<SUBJ> 'to go away'

Class II verbs: They can be passives or unaccusatives in the syntactic sense. A passive verb is always related to an active transitive verb via a function-changing lexical transformation; the active and the passive verb have the same set of thematic roles, they merely differ in whether and how the thematic roles are mapped to grammatical functions. Whereas in the active verb, agent and theme (patient) are mapped to SUBJ and OBJ, respectively, in the passive verb, the theme is mapped to SUBJ while the agent is either suppressed or mapped to the oblique function OBL-AG, corresponding to a postpositional phrase with the postposition *mier* 'by' (10). Examples are given in (11) and (12).

- | | | | | | | | | | | | |
|------|----------------|-------|-------|-----|-----------------|-----------|-------|---------|------|---|---|
| | <i>active:</i> | agent | theme | | <i>passive:</i> | agent | theme | | | | |
| | ↓ | ↓ | ↓ | ↔ | ↓ | ↓ | ↓ | | | | |
| (10) | ga-ḱeteba | < | SUBJ, | OBJ | > | ga-ḱeteba | < | NULL, | SUBJ | > | / |
| | | | | | | ga-ḱeteba | < | OBL-AG, | SUBJ | > | |

- (11) *ga-ḱet-d-eb-a* (*mtavrob-is mier*).
 will-be-done.PASS (government.GEN by)
 'It will be done (by the government).'

- (12) *mtavroba ga-a-ḱet-eb-s*.
 government.NOM will-do-it.TRANS
 'The government will do it.'

Unaccusatives, on the other hand, have only one thematic argument which is invariably mapped to SUBJ, and there is no suppressed agent which could optionally resurface as an oblique:

- (13) *unaccusative*: theme
 ↓
 da-bruneba < SUBJ >
- (14) *da-brun-d-eb-a* (* *ded-is* *mier*).
 he-will-return (* mother.NOM by)
 ‘He will return. (/ * He will be returned by the mother.)’

Since Tschenkéli’s classification is primarily a morphological one, it does not explicitly distinguish between passives and unaccusatives of the same morphological shape. The distinction has to be made manually, or could at best be derived from the (non)existence of an active counterpart in the same superparadigm.

For these reasons, the automatically derived lexicon entries had and still have to be refined and corrected later on.

4 Mapping Case and Affixes to Grammatical Functions

Georgian uses both head-marking (mainly 1st and 2nd person affixes) and dependent-marking (case, restricted to 3rd person) to code grammatical functions, where it follows a complex split-ergative scheme that is further complicated due to what is commonly (e.g. in the Relational Grammar literature, [7]) called ‘inversion’.

The dependency of the mapping of person/number affix resp. case to grammatical function on the parameters verbal class and tense group can be read off of the following tables:

- (15) Three case alignment patterns

	SUBJ	OBJ	OBJben
A	ERG	NOM	DAT
B	NOM	DAT	DAT
C	DAT	NOM	- <i>tvis</i>

- (16) Two person/number affix alignment patterns

	SUBJ	OBJ	OBJben
A, B	<i>v-</i>	<i>m-</i>	<i>h-</i>
C	<i>h-</i>	<i>v-</i>	-

- (17) Selection of alignment pattern depending on verb class and tense group

	I	II	III	IV
	<i>trans.</i>	<i>unacc.</i>	<i>unerg.</i>	<i>indir.</i>
present	B	B	B	C
aoarist	A	B	A	C
perfect	C	B	C	C

The mapping of verbal affixes to grammatical functions is coded into the morphology transducer, whereas case alignment is treated in the syntax by f-structure equations attached to the verb lexicon entries. Example (18) shows a simplified version of the equation that codes pro-drop and subject case alignment for Class I and III verbs.

$$(18) \quad \{ (\uparrow \text{SUBJ PRED}) = \text{'pro'} \\ | @(\textit{ifelse} (\uparrow _ \text{TENSEGROUP}) =_c \text{pres} \\ [(\uparrow \text{SUBJ CASE}) = \text{nom}] \\ [@(\textit{ifelse} (\uparrow _ \text{TENSEGROUP}) =_c \text{aor} \\ [(\uparrow \text{SUBJ CASE}) = \text{erg}] \\ [(\uparrow \text{SUBJ CASE}) = \text{dat}])]) \}.$$

5 Syntax: An Overview

In the following, I present the most important grammatical features and some selected construction types of Georgian covered by the grammar and point out how they are dealt with in an LFG setting.

5.1 Word Order, Nonconfigurationality and Discourse Functions

Georgian is traditionally taken to be a language with ‘free word order.’ This is true at the phrase level; there is no VP constituent that would enable one to configurationally distinguish subject position from complement position; the finite verb and other constituents can occur in arbitrary order, or, phrased differently, any permutation of the constituents results in a grammatical sentence. This is what we would expect for a language with full-fledged head- and dependent-marking: since grammatical functions are (mostly unambiguously) coded morphologically, there is no need to repeat the coding of grammatical functions configurationally. Thus one could as a first approximation assume a flat top-level phrase structure:

$$(19) \text{ Initial approximation:} \quad S \rightarrow V, XP^*$$

The Kleene star (*) means that there can be arbitrarily many XP constituents, and the comma means that V and XP constituents can occur in arbitrary order. XP denotes any maximal projection, i.e. NP, DP, AP, POSSP, etc.

Since syntax plays no role in the coding of grammatical functions, word order is available for expressing discourse functions like TOPIC and FOCUS. Although in Georgian, TOPIC and FOCUS do not seem to be coded exclusively configurationally, there is a strong tendency in the language for configurational coding: The TOPIC is mostly sentence-initial, which is very common cross-linguistically. The constituent bearing the FOCUS function normally occupies the position immediately in front of the inflected verb, or, more exactly, the inflected verb complex, which in addition to the verb may contain a negation particle and other modal particles. Heavy focused constituents tend to follow the verb or occupy sentence-final position. Focused verbs mostly precede their arguments.

In addition to the configurational encoding of the FOCUS function (which may be ambiguous if there are constituents both in front of and following the verb), Georgian also uses rising intonation to mark focus, there are a couple of adverbial clitics (*-c*, *ki* etc.) that can be used to mark focus (and topic), and finally, clefting can be utilized to put a constituent into focus.

The position of question words is fully grammaticalized in Georgian: they invariably occupy the position immediately in front of the verb complex. Since the position of a focused word often mirrors the position of the question word in a question–answer–scenario/pair, the pre-verbal focus position follows quite naturally.

The apparent configurational significance of the position immediately in front of the inflected verb motivates a revision of the basic phrase structure rule (19): In compliance with the LFG variant of X' theory ([1] p. 98), I assume that I is the category of the inflected verb (24), and that the specifier position of IP, if present, is occupied by question words (22) or by a potentially focused (or topicalized, if it is sentence-initial) constituent (21). Constituents further to the left are recursively adjoined to IP (20), and the complement of I is the exocentric, non-projecting category S that hosts the material right to the verb (23, 25).

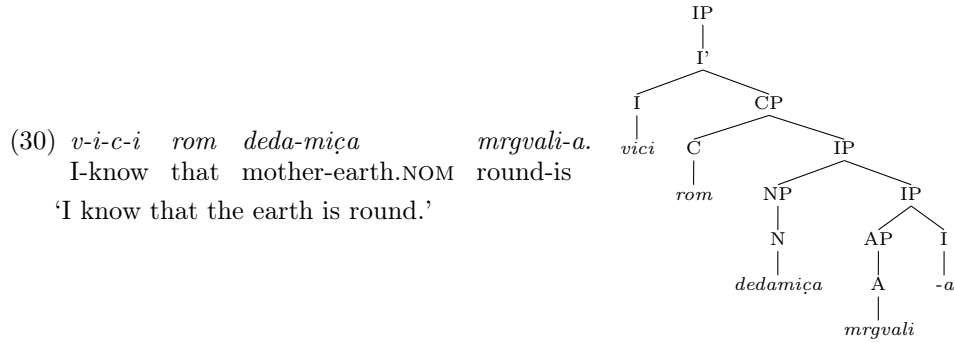
- (20) $IP \rightarrow XP\ IP$ (22) $IP \rightarrow QP+ I'$ (24) $I \rightarrow Vinfl$
 (21) $IP \rightarrow XP\ I'$ (23) $I' \rightarrow I\ (S)$ (25) $S \rightarrow XP+$

Some examples that illustrate these rules follow.

- (26) *çvim-s.* IP
 rains |
 ‘It rains.’ I
 |
 çvims
- (27) *bavšv-i tamaš-ob-s.* IP
 child.NOM plays NP I
 ‘The child is playing.’ | |
 N *tamašobs*
 bavšvi
- (28) *student-i çer-s çeril-s.* IP
 student.NOM writes letter.DAT NP I' S
 ‘The student writes a letter.’ | | |
 N I NP
 studenti *çers* |
 N
 çerils

Subordinate phrases with initial *rom* or *tu* etc. or without overt complementizer are complementizer phrases (CP, 29).

- (29) $CP \rightarrow (C)\ IP$



Noun phrases and the like are normally proper (projective) constituents (e.g., $NP \rightarrow AP\ N$), but modifiers such as relative clauses, adjectives, genitive modifiers and possessive pronouns may be dislocated to the right, that is, they do not need to form a continuous constituent together with the head they modify. An example of a dislocated possessive is (31), whose non-dislocated version is (32).

- (31) *gvar-i ar v-u-txar-i čem-i*
 last-name.NOM not I.told.it.to-him my.NOM.

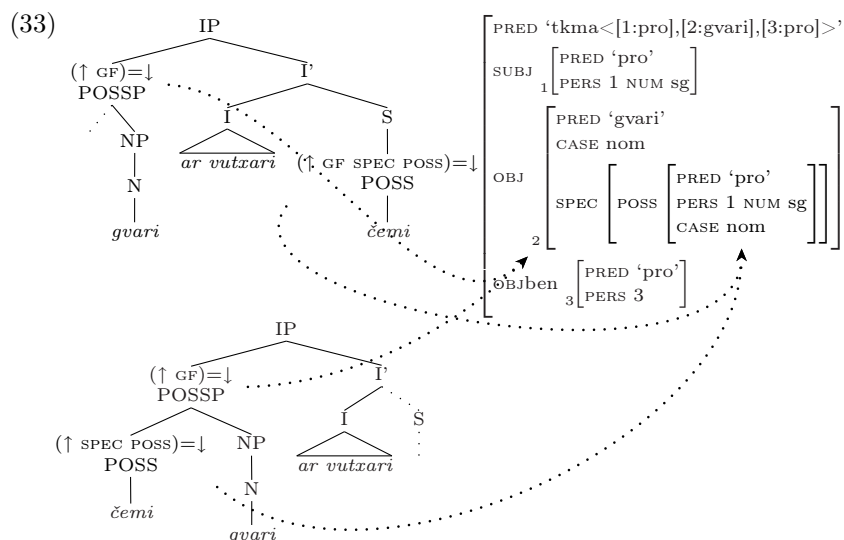
‘I did not tell him my last name.’

- (32) *čem-i gvar-i ar v-u-txar-i*
 my.NOM last-name.NOM not I.told.it.to-him.

‘I did not tell him my last name.’

This example (as well as (34, 35)) illustrates nicely how the separation of c- and f-structure in LFG enables a unified analysis of superficially disparate constructions: Both sentences have the same f-structures, as the analyses in (33) show. In the first c-structure, the dislocated possessive is located below S, which is the normal location of constituents right to the verb. In the c-structure of the non-dislocated version below, the possessive occupies its normal position as a specifier of NP.

It is the annotation of these two nodes which guarantees that both c-structures are mapped to the same f-structure: In the case of the non-dislocated possessive, the straightforward annotation (\uparrow SPEC POSS)= \downarrow makes sure that the possessive is mapped to the value of the path SPEC POSS in the f-structure of the noun it modifies. When the possessive is dislocated, the challenge is to find the nominal it modifies. An eventual candidate has to fulfill three conditions: It has to be located to the left of the possessive, its case has to match the case of the possessive, and it has to be a common noun. In addition, the candidate should correspond to a core grammatical function (GF).



These conditions can be formally stated as equations annotating the POSS node. The main annotation is $(\uparrow \text{GF SPEC POSS})=\downarrow$, which states that the possessive should be mapped to the value of SPEC POSS of *some* grammatical function. Similar annotations make sure that the other conditions are met; in particular, the condition stating that cases have to match picks out exactly one grammatical function in most cases (in the example, it is the OBJ function), leading to an unambiguous attachment of the possessive in the f-structure.

5.2 Pro-drop

Georgian is a pro-drop language: core arguments of the verb are not obligatorily realized as independent morphological words (e.g. personal pronouns) that are syntactic constituents. If an argument is realized, the person/number markers in the verb function as agreement features in the verb, but are providing a pronominal interpretation if the argument is missing ([8]). Dropped pronouns do not figure in the c-structure: the Principle of lexical integrity ([1] p. 92), which formalizes the view that (c-structure) syntax does not have access to word-internal structure, does not allow bound affixes to appear as lexical nodes.⁴ It is the functional annotation of the lexicon entries that makes sure that the grammatical functions of the verb are properly instantiated in the case of pro-drop.

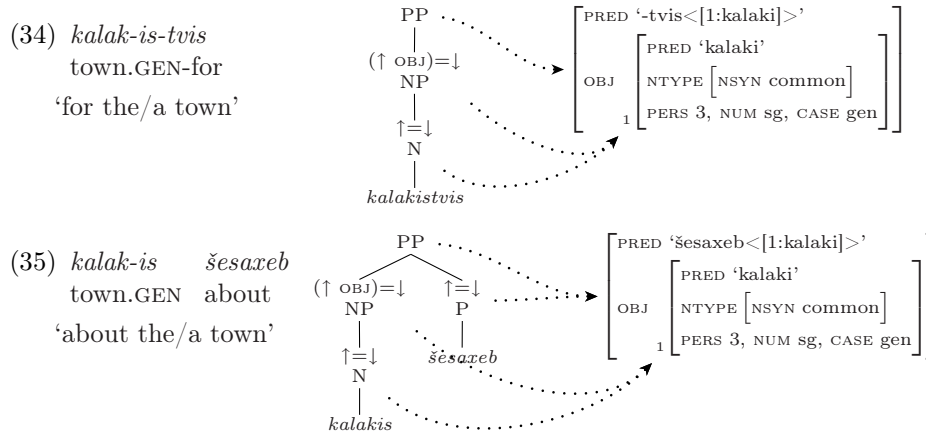
5.3 Postpositions as Phrasal Affixes; Double Declension

Postpositions in Georgian affect whole noun phrases including coordinations, they are phrasal/lexical affixes (not clitics proper).⁵ My implementational choice

⁴ Bound clitics are considered syntactic words and are not affected by this principle; see also section 5.3.

⁵ See [9] for the distinction between clitics and phrasal affixes.

is not to give independent c-structure status to bound postpositions (those that are attached to the word to their left), but to adhere to a strong form of the Lexical integrity principle, maintaining that only morphological words and true clitics can be c-structure lexical nodes, but not bound phrasal affixes, while free postpositions are lexical nodes. The f-structures for bound and free postpositions, however, are not different. Unlike most case endings, virtually all postpositions have semantic content and are predicates on their own that subcategorize for an OBJ.⁶ This is illustrated in (34) and (35).



The second case affix in double declension forms, i.e. nouns that carry two case markers as the result of ellipsis of the head noun, is treated in a similar way. In forms like *bavšvob-is.GEN-a-s.DAT*, 'in childhood', from *bavšvoba* 'childhood', the syntax has access to the inner case, as can be seen from the agreement between possessive pronoun and noun in the phrase (36). This indicates that the second case in a double-case construction has phrasal affix properties.⁷

- (36) [*čem-i bavšvob-isa*]-s
[my.GEN childhood.GEN].DAT
'in my childhood'

⁶ The only exceptions are the postpositions *-tvis*, *-tan* and *-ze*, when they are grammaticalized to mark (oblique) indirect objects in the perfect series. In such constructions, they have no PRED value/semantic content and are treated similarly to case endings.

⁷ As to whether case endings in general have clitic- or phrasal affix-like properties, see the discussion in [10], where the distinction between clitics and phrasal affixes is not explicitly drawn. There, Harris shows that case endings are not clitics. But there is at least one peculiar construction where case endings clearly behave like phrasal affixes: In [11], §103, Šanije discusses 'sentence declension' (*činadadebis bruneba*), by which he denotes the interpretation of a whole phrase or sentence as a cited noun phrase, which as such can be in inflected, case-marked argument position, for example: [*mex-i ki da-g-e-c-a*]-sa-c zed da-a-*tan-da*. '«May lightning strike you», he would also add.'

5.4 *unda* and *šeijleba*

In this section, I discuss in somewhat more detail the implementation I have chosen for constructions involving *unda* ‘must’ and *šeijleba* ‘possibly’, as they have received little and inadequate treatment in the literature.

At first glance, *unda* and *šeijleba* behave like adverbials which put modality restrictions on the verb they are attached to: The verb has to stand in one of the modal tenses (Optative, Pluperfect, Conjunctive Present/Future), but the case syntax of all of the arguments is determined by the (main) verb, as (37) and (38) demonstrate.

- (37) *gia-m çeril-i unda da-çer-o-s.*
 Gia.ERG letter.NOM must write.OPT.
 ‘Gia must write a letter.’
- (38) *gia çeril-eb-s unda çer-d-e-s.*
 Gia.NOM letter.PL.DAT must write.CONJ-PRES.
 ‘Gia must write letters.’

This contrasts to the control constructions (39) and (40) with the homonymous verb form *unda* ‘he wants’, which clearly require a biclausal analysis.

- (39) *gia-s u-nd-a rom çeril-i da-çer-o-s.*
 Gia.DAT wants.PRES that letter.NOM write.OPT.
 ‘Gia wants to write a letter.’
- (40) *gia-s u-nd-a rom çeril-eb-s çer-d-e-s.*
 Gia.DAT wants.PRES that letter.PL.DAT write.CONJ-PRES.
 ‘Gia wants to write letters.’

Harris and Campbell [12] analyze the construction with *unda* ‘must’ (37, 38) as a monoclausal structure with auxiliary and main verb. They interpret the modern construction as the result of a diachronic ‘Clause fusion’ process, in the course of which the construction with the 3rd person singular verb form *unda* (which is cognate to the inflected verb form *u-nd-a* ‘he wants’) underwent a semantic shift, followed by clause fusion, and consequently a change in case syntax. As evidence for a synchronic monoclausal analysis, they more or less implicitly state the case syntax of the construction, which is determined by the main (subordinate) verb alone, the invariability of the modal, and the impossibility of a *rom* complementizer, which is obligatory in the parallel control constructions.

There are, however, several constructions involving *unda* and *šeijleba* which indicate that a biclausal analysis is appropriate also here, both at f-structure and c-structure level. One of those constructions is negation: the negation particle *ar* ‘not’ can be placed either in front of *unda/šeijleba*, or in front of the main verb, or in front of both, as in (41).

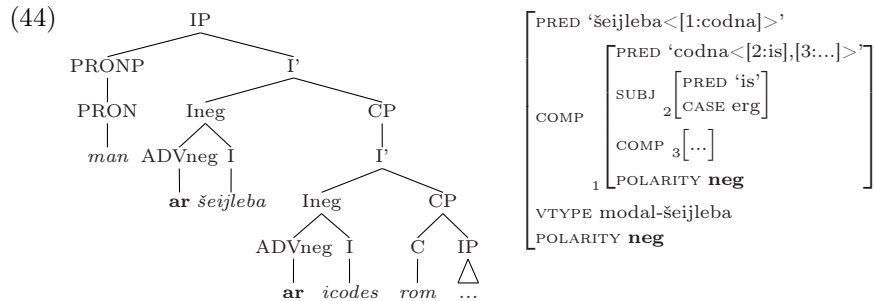
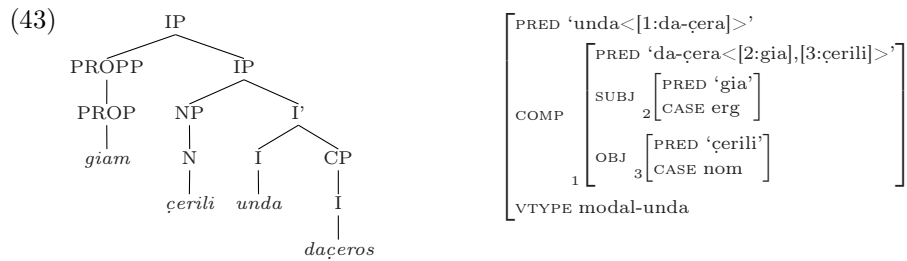
- (41) *man ar šeijleba ar i-cod-e-s, rom ...*
 he.ERG not possible not knows.CONJ-PRES, that ...
 ‘It is not possible that he does not know that ...’

The two negation possibilities can be most naturally accounted for in a biclausal analysis: the first *ar* negates the matrix clause, whereas the second *ar* negates the subordinate clause.

A still stronger argument for a biclausal analysis is verb phrase coordination: *unda* and *šejleba* in front of the first verb normally have scope over both verbs, as in (42):

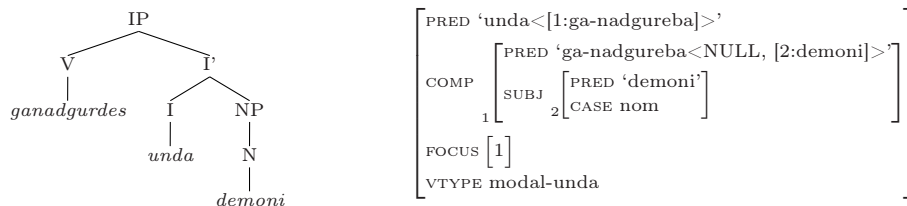
- (42) *unda ga-gv-i-xar-d-e-s da v-i-dgesašcaul-o-t.*
 must rejoice.OPT.1PL and celebrate.OPT.1PL.
 ‘We should be happy and celebrate.’

We can account for these facts if we treat *unda* and *šejleba* syntactically as verbs that occupy the I position like normal inflected verbs, but that, in contrast to other verbs taking phrasal arguments, subcategorize for one single argument, namely the subordinate phrase in COMP function (which, at c-structure level, corresponds to a CP node). Analyses of (37) and (41) are given in (43) and (44).



Finally, treatment of *unda* as a syntactic verb allows for an easy explanation of constructions with postponed *unda* as in (45). In such cases, the verb is in IP specifier position and thus focussed.

- (45) *sxva gza ar aris, ga-nadgur-d-e-s unda demon-i.*
 other way.NOM not is, destroy.PASS.FOCUS must demon.NOM
 ‘There is no other way, the demon has to be *destroyed*.’



6 Tools for LFG Grammar Development

In this part, I present the essential tools that are used in the development of the Georgian grammar.⁸

6.1 XLE and *fst*: The Development Environment for LFG Grammars

XLE (Xerox Linguistic Environment) is at the heart of most computational work with LFG grammars. It is a sophisticated development platform for LFG grammars developed by the Palo Alto Research Center with active participation of some of the inventors of LFG. XLE consists of a parser, a generator and a transfer module. These modules can be used both from Emacs via a Tcl/Tk interface that provides powerful viewing and debugging facilities, and as a shared library, which opens up for integrating XLE into custom software. Tokenization and morphological analysis is normally done with the Xerox finite state tool, *fst*.

6.2 XLE-Web: A Web Interface to XLE

XLE-Web is an easy-to-use pedagogical Web interface to XLE for parsing sentences on the fly. I developed it originally as a tool to facilitate the accomodation of the Norwegian ParGram grammar for use in the Norwegian–English machine translation project LOGON.⁹ The software is now in use for many of the ParGram grammars. Main features of the system are display of c- and f-structures of LFG analyses, visualization of the mapping from c- to f-structure, and display of compact packed representations of c- and f-structures that combine the c- and f-structures of all analyses of a given parse into one c- and one f-structure graph.

6.3 LFG Parsebanker: Grammar Development and Treebanks

When developing a large grammar, it is essential to be able to run the grammar on a set of sample sentences, to store the parse results, and to rerun successive versions of the grammar on the same sentences, in order to monitor progress, to assess coverage and to compare analyses across different grammar versions.

⁸ Some of the tools discussed here as well as the Georgian grammar can be tested online at <http://www.aksis.uib.no/kartuli>.

⁹ See <http://www.emmtee.net/>

Eventually, one might want to run the grammar on a larger set of sentences (perhaps chosen from running text), and let the collection of annotated sentences evolve into a treebank in the sense of a linguistic resource. Since sentences of only moderate complexity often are highly ambiguous, and the desired or correct reading is only one of the analyses offered by the grammar, it should be possible to manually disambiguate the parses in an efficient way.

Together with Rosén and de Smedt ([13], [14]) I have been developing a Web-based treebanking toolkit that suits exactly these needs: the LFG Parsebanker. The LFG Parsebanker is a comprehensive and user-friendly treebanking toolkit for manual disambiguation of a parsed corpus. It supports a process flow involving automatic parsing with XLE, sophisticated querying, and, crucially, efficient manual disambiguation by means of discriminants.

One can characterize discriminants roughly as ‘any elementary linguistic property of an analysis that is not shared by all analyses’ [15]. In LFG grammars, there are often a large number of elementary properties that are not shared by all analyses, such as local c-structure node configurations and labels or f-structure attributes and values. Any such elementary property is a candidate for being a discriminant. In using discriminants, our toolkit is somewhat similar to the Treebanker [15], Alpino [16] and the LinGO Redwoods project’s [incr tsdb()] tool [17]. It is, however, specifically designed for LFG grammars. The underlying design and implementation of our LFG discriminants is described in detail in [18].

6.4 A Georgian Corpus of Fiction and Non-fiction Texts

An indispensable resource for research in Georgian syntax is a searchable text corpus of decent size. There are several collections of Georgian texts available on the Internet which can be used to build up such a corpus. One of them is the electronic newspaper archive Opentext. It comprises more than 100 million words and is by far the largest collection of Georgian texts available online. Another important collection of non-fiction is the text archive of the Georgian service of Radio Free Europe/Radio Liberty with around eight million words. The largest archive of fiction (both prose and poetry) is the UNESCO Project digital collection of Georgian classical literature (both prose and poetry) with three million words.¹⁰

I have harvested the texts of these three archives and imported them into corpus query software based on Corpus Workbench¹¹ which is being developed at Aksis. Although the corpus is not part-of-speech tagged, the versatile query language of Corpus Workbench allows for sophisticated searches.

7 Conclusion

In this paper, I have presented a project that aims at building a linguistically motivated full-scale computational grammar for Georgian in the LFG framework.

¹⁰ See <http://www.opentext.org.ge>, <http://www.tavisupleba.org>,
<http://www.nplg.gov.ge/gSDL/>

¹¹ See <http://www.ims.uni-stuttgart.de/projekte/CorpusWorkbench/>

I have given an overview of the major issues that need to be addressed in this type of project, and I have shown how implementing a grammar in a formal linguistic framework can help solving issues in theoretical linguistics.

References

1. Bresnan, J.: *Lexical-Functional Syntax*. Blackwell Publishers, Oxford (2001)
2. Parallel Grammar Project, <http://www2.parc.com/isl/groups/nlft/pargram/>
3. Butt, M., Dyvik, H., King, T.H., Masuichi, H., Rohrer, C.: *The Parallel Grammar Project*. In: *Proceedings of the COLING Workshop on Grammar Engineering and Evaluation*, 1–7. Taipei (2002)
4. Butt, M., King, T.H., Niño, M.-E., Segond, F.: *A grammar writer’s cookbook*. CSLI Publications, Stanford (1999)
5. Beesley, K. R., Karttunen, L.: *Finite State Morphology*. CSLI Publications, Stanford (2003)
6. Tschenkéli, K., Marchev, Y.: *Georgisch-Deutsches Wörterbuch*. Amirani-Verlag, Zürich (1965–1974)
7. Harris, A.C.: *Georgian Syntax. A study in relational grammar*. Cambridge University Press, Cambridge (1981)
8. Strunk, J.: *Pro-drop in nominal possessive constructions*. In: *Proceedings of the 10th International LFG Conference*. CSLI Publications, Stanford (2005)
9. Halpern, A.: *On the placement and morphology of clitics*. CSLI Publications, Stanford (1995)
10. Harris, A.C.: *Origins of Apparent Violations of the ‘No Phrase’ Constraint in Modern Georgian*. *Linguistic Discovery*, vol. 1 (2), pp. 1–25 (2002)
11. Šanije, A.: *kartuli enis gramaṭikis sapujvlebi (in Georgian)*. Tbilisi University Press, Tbilisi (1973)
12. Harris, A.C., Campbell, L.: *Historical syntax in cross-linguistic perspective*. Cambridge University Press, Cambridge (1995)
13. Rosén, V., Meurer, P., de Smedt, K.: *Constructing a parsed corpus with a large LFG grammar*. In: *Proceedings of the 10th International LFG Conference*. CSLI Publications, Stanford (2005)
14. Rosén, V., Meurer, P., de Smedt, K.: *Towards a toolkit linking treebanking to grammar development*. In: *Proceedings of the Fifth Workshop on Treebanks and Linguistic Theories*, pp. 55–66. Prague (2006)
15. Carter, D.: *The TreeBanker. A Tool for Supervised Training of Parsed Corpora*. In: *Proceedings of the Workshop on Computational Environments for Grammar Development and Linguistic Engineering*. Madrid (1997)
16. Bouma, G., van Noord, G., Malouf, R.: *Alpino. Wide-Coverage Computational Analysis of Dutch*. In: *Computational Linguistics in the Netherlands*, 45–59. Rodopi, Amsterdam (2001)
17. Oepen, S., Flickinger, D., Toutanova, K., Manning, C.D.: *LinGO Redwoods, a rich and dynamic treebank for HPSG*. *Research on Language & Computation* 2 (4), 575–596 (2004)
18. Rosén, V., Meurer, P., de Smedt, K.: *Designing and Implementing Discriminants for LFG Grammars*. In: *Proceedings of the 12th International LFG Conference*. CSLI Publications, Stanford (2007)