Chapter 5

Implementation of a grammar fragment

5.1 Introduction

So far, this study has been concerned with a general description of a fragment of a Northern Sotho grammar. Chapters 2 and 3 provide the units and information on how they form morphosyntactic constellations, while chapter 4 contains an introduction to parsing and some basics for a generalisation from these constellations as a first step to developing a linguistic model of the language.

In this chapter, some of the Northern Sotho "groups of consecutive words" (Jurafsky and Martin, 2000, p. 421) or constituents are described more formally in a context-free grammar (CFG, see paragraph 1.4.4). For sake of demonstration, we opted for an implementation of some core parts of the grammar fragment, i.e. the basic verbal phrase, the imperative and indicative constellations in the constraint-based Lexical-Functional Grammar formalism (LFG[1]). LFG has been successfully utilised to model morphosyntactic phenomena for a number of languages, Bresnan (2001, pp. 148 to 160) describes - amongst others - phenomena of the Malawi Bantu language Chichewa, a language with a number of phenomena similar to Northern Sotho. This section contains a brief introduction to theory (paragraph 5.1.1) and formalisms (paragraph 5.1.2) of LFG, including a description of the environment used for implementation and testing (paragraph 5.1.2.4) and, lastly, a description of the partial implementation itself (section 5.2).

[1]See e.g. `http://www-lfg.stanford.edu/lfg`

5.1.1 Lexical-Functional Grammar (LFG)

In this paragraph, ideas and background to the theory of Lexical-Functional Grammar (LFG) are introduced, as described by e.g. Asudeh and Toivonen (2009), who state that

> "LFG is a theory of generative grammar, in the sense of Chomsky (1957, 1965). The goal is to explain the native speaker's knowledge of language by specifying a grammar that models the speaker's knowledge explicitly and which is distinct from the computational mechanisms that constitute the language processor."

They refer to Kaplan and Bresnan (1982), who introduced this formal system to describe the grammar of a language. LFG, according to Kaplan and Bresnan (1982, p. 173), supports the expression and explanation of generalisations that concern syntactic issues. It manages information on two levels: the lexicon, where semantic arguments are mapped to grammatical functions appearing at sentence level, and the syntactic rules that identify these functions with "particular morphological and constituent structure configuations" (Kaplan and Bresnan, 1982, p. 174). A constituent and a functional structure form the result of analysis, which together represent the knowledge of the system about a specific sentence, i.e. a surface form. To map these structures to surface sentences, the structure needs to be sufficiently well-formed: the necessary requirements for this will be explained in greater detail below.

5.1.2 The LFG formalism

5.1.2.1 Representations: constituent structure and functional structure

LFG provides two levels of analytic representation, the constituent (or c-)structure, and the functional (or f-)structure, while (predicate-)argument structure is an input from the lexicon. Butt et al. (1999, p. 3) describe phrasal dominance and precedence relations as being encoded in c-structure, while f-structure encodes syntactic predicate argument structure. C-structure is made visible as a tree-structure, f-structure as an attribute-value matrix. The grammar files themselves are (roughly) divided into a lexicon and a rules-section. Figure 5.1, based on (Bresnan, 2001, (13), p. 19) demonstrates the parallel structures of LFG.

5.1.2.2 Predicate argument structure versus syntactic structure

In LFG, predicate argument structure is disassociated from syntactic structure. The predicate argument structure, e.g. a verb's valency, is assigned to the lexicon entry. The

rules-section on the other hand, contains certain constituents relating to grammatical functions.

The lexicon would therefore contain information such as the verb stem *bolela* '[to] speak' requiring a subject in its cotext when used intransitively. (bolela ⟨SUBJ⟩), the specific constituent that carries this function, however, is found in the rules-section, where a sentence is e.g. defined as an NP carrying the subject function and a VP (S → $_{\text{SUBJ}}$NP VP). Any NP described in the rules section, e.g. a single noun (NP → N), may then possibly fill this subject slot. If the sentence in example (80) were to be grammatically analysed , *bolela* will have *monna* assigned in the subject role[2].

(80) *monna*$_{\text{N01}}$ *o*$_{\text{1CS01}}$ *a*$_{\text{MORPH\_pres}}$ *bolela*$_{\text{Vitr}}$
man `subj-cl1` `pres` speak
'(a) man speaks'

In paragraph 1.4.4 (page 13), it was argued that attribute-value pairs (functional schemata or functions) and the unification principle should be utilised in order to reduce the number of rules necessary for developing a grammar. LFG extends this principle alongside two others: completeness and coherence, summarised as the three principles of well-formedness, cf. (Kaplan and Bresnan, 1982, p. 211 et seq.). According to the grammaticality condition

[2]A Northern Sotho grammar rules section would also describe instances where the NP is omitted and the subject concord acquires the subject function, or the imperative case, where the subject does not appear at all.

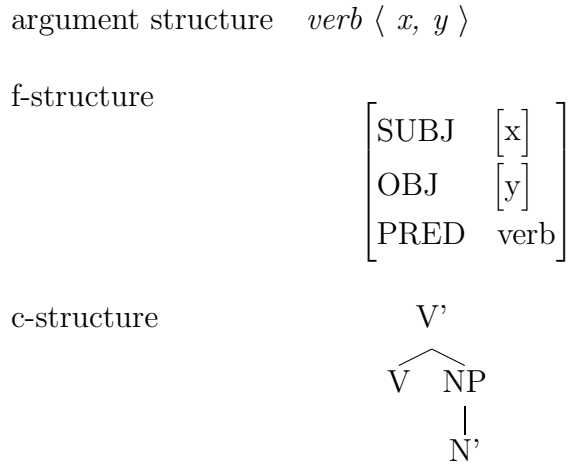

Figure 5.1: Parallel structures of LFG

(ibid. p. 212), "A string is grammatical only if it is assigned a complete and coherent f-structure."

Grammatical functions are assigned to constituents in the rules section in order to generate both a c(onstituent)-structure, and a f(unctional)-structure . The grammatical functions according to Bresnan (2001, p. 97 et seq.) are SUBJ(ect), OBJ(ect), OBL(ique), COMPL(ement) and ADJUNCT, TOP(ic) and FOC(us). Of these, ADJUNCT, TOP and FOC are non-argument functions and thus allow multiple instances, i.e the appearance of a zero element or other such items not governed by any other element. Any other grammatical function may only appear once per f-structure (functional uniqueness principle) and only if the sentence in question contains a unit that the lexicon states as being required (coherence principle). All of the grammatical functions assigned by the units contained in the sentence, i.e. the constituents that are governed by others, must however appear (completeness principle). The three LFG-principles may be summarised as follows (taken from Butt et al. (1999)):

- **Functional Uniqueness**: In a given f-structure, a particular attribute has a maximum of one value;
- **Completeness**: An f-structure is *locally complete* if and only if it contains all the governable grammatical functions that its predicate governs. An f-structure is *complete* if and only if it and all its subsidiary f-structures are locally complete;
- **Coherence**: An f-structure is *locally coherent* if and only if all the governable grammatical functions it contains are governed by a local predicate. An f-structure is *coherent* if and only if it and all its subsidiary f-structures are locally coherent.

5.1.2.3 An example analysis

Lexicon entries. The LFG formalism is best explained by an example analysis[3]. Therefore, we come back to the sentence *monna o reka apola* '(a) man buys (an) apple' mentioned in chapter 1 and begin with a description of the necessary LFG lexicon entries.

The singular noun *monna* 'man' is of noun class 1 and of the third person. These data are found in its lexicon entry (the meaning of ↑ will be explained in the paragraph on constructing c- and f-structure). The noun *apola* 'apple' is described similarly.

[3]We heavily rely on Wescoat (1989) when explaining the LFG formalism.

```
monna N *  (↑ PRED)='monna'
           (↑ CLASS)= 1
           (↑ NUM)= sg
           (↑ PERS) = 3.
apola N *  (↑ PRED)='apola'
           (↑ CLASS)= 9
           (↑ NUM)= sg
           (↑ PERS) = 3.
```

A subject concord is comparable to an inflectional prefix, it belongs to the verb, supplies agreement information (subject-verb agreement, cf. paragraph 2.4.2), and thus is not described with a PRED-value in the lexicon. The concord *o* occurs not only as a subject concord of the noun classes 1, 1a (1st set) and 3 (1st and 2nd set) and of the 2nd person singular (1st and 2nd set), it may also occur as a pronominal object concord of class 3. Person and number information on the subject are usually provided by the subject itself, however, as the subject concord may indeed acquire the subject's function when the respective NP is omitted, the appropriate grammar rule contains a disjunction describing these properties as well. The ambiguity of *o* is mirrored in the lexicon (the copulative use of *o* is not contained in the following lexicon entries):

```
o  1CS *  (↑ SUBJ CLASS)= 1 (↑ SUBJ NUM) = sg;
   1CS *  (↑ SUBJ CLASS)= 1a (↑ SUBJ NUM) = sg;
   1CS *  (↑ SUBJ PERS)= 2 (↑ SUBJ NUM) = sg;
   2CS *  (↑ SUBJ PERS)= 2 (↑ SUBJ NUM) = sg;
   1CS *  (↑ SUBJ CLASS)= 3 (↑ SUBJ PERS) = 3 (↑ SUBJ NUM) = sg;
   2CS *  (↑ SUBJ CLASS)= 3 (↑ SUBJ PERS) = 3 (↑ SUBJ NUM) = sg;
   CO *   (↑ CLASS)= 3 (↑ PERS) = 3 (↑ NUM) = sg.
```

The verb stem *reka* '[to] buy' is transitive, it therefore requires a subject and an object to appear in the sentence. As described in paragraph 2.7.2, certain verbal constellations may be solely identified by the verbal ending which need not be identical to the last letter(s) of the verb stem, like it is the case for *reka*. Verb stems like *re* '[to] say', however, appear as well. These behave syntactically like verbs ending in *–a*. Therefore, a specific attribute, Vend, was introduced which is added to each verb stem entry contained in the lexicon.

```
reka Vtr *  (↑ PRED)='reka ⟨(↑ SUBJ)(↑ OBJ)⟩'
            (↑ VEND)= a.
```

The rules section. LFG grammar rules look very similar to the CFG rules shown in paragraph 1.4.4, however, functional schemata may be added to their right side. Sentence (80) is a positive indicative of the present tense (as described in paragraph 3.2.5.1, page 97), it therefore requires the VIE to be annotated alongside[4]. The constraining equation '(↑VEND) =c a' means that in any such sentence, the attribute VEND must be *a*, i.e. only verb stems in the lexicon that have the attribute VEND defined and set to '−*a*' may appear in it, i.e. this equation must be satisfied by a f-structure (cf. (Kaplan and Bresnan, 1982, p. 207)). The rules may be constructed directly into an annotated tree (cf. Figure 5.2).

```
S    →  NP:   (↑ SUBJ) =↓;
        VP:   ↑=↓.

NP   →  N.

VP   →  VIE:  (↑ TNS-ASP MOOD)= indicative
              (↑ TNS-ASP TENSE)= pres
              (↑ TNS-ASP POL)= pos
        VBP:  (↑VEND) =c a.

VIE  →  1CS.
VBP  →  Vtr
        NP:   (↑ OBJ) =↓.
```

Constructing c- and f-structure When a sentence is analysed, the lexicon entries fill the predefined functions according to the well-formedness principles. The uniqueness principle avoids selecting the wrong lexicon entry of *o* ('CLASS 01' is defined by the lexicon entry of the subject noun *monna*). The completeness principle is fulfilled because all necessary arguments described by the lexicon entry of the predicate are present. The coherence principle is fulfilled because there are no constituents present that would unnecessarily add other arguments.

[4]The tense-aspect (TNS-ASP) attributes mood and tense appearing here are inspired by grammars developed in the pargram project, cf. section 5.1.2.4.

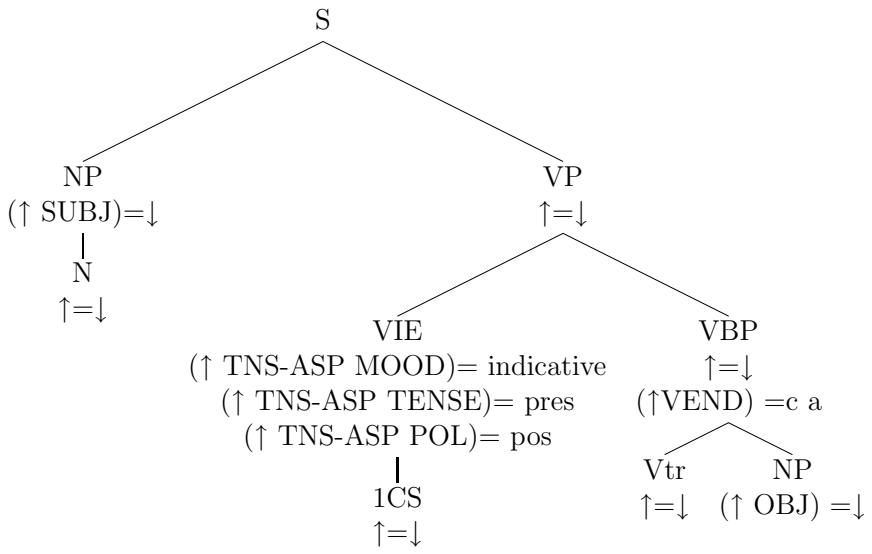

Figure 5.2: Basic sentence, consisting of a NP and a VP

The annotated c-structure shown in Figure 5.3 forms the basis for building an f-structure. This **instantiation** (cf. (Wescoat, 1989, p. 7)) "transforms the schemata into functional equations". Firstly, it is assumed that each node of the tree corresponds to one partial f-structure. Arbitrarily, one can give them numbers, as done in Figure 5.4. The arrows play an important role in identifying the referents, as they show correspondences between the f-structures. "↑" points to the tree node in which the respective node is contained, while ↓ points to the nodes contained in the respective node. As all nodes are instantiated in f-structures, "↑=↓" therefore means that the node's f-structure in which the node is contained is identical with the one the node's f-structure contains itself. In our example, $f2$=$f3$, $f4$=$f5$=$f6$=$f7$=$f8$, $f9$=$f10$.

The equation (↑ SUBJ)=↓ means that the node(s) below, i.e. the node(s) dominated by this node contains the subject of the node above. In our example, this means that the node found below the node $f2$ is the subject of S, i.e. ($f1$ SUBJ)=$f2$(=$f3$). Respectively, the object of S is found ($f7$(=$f4$=$f1$) OBJ)=$f9$(=$f10$), see Figure 5.5. Note that up-arrows in the lexicon entries of the words point to the respective node (and hence, the respective f-structure) where the word is to be contained in.

The set of all instantiated functional equations or **functional description** (cf. (Wescoat, 1989, p. 14)), see Table (5.1), is the only data for constructing the f-structures. For ease of

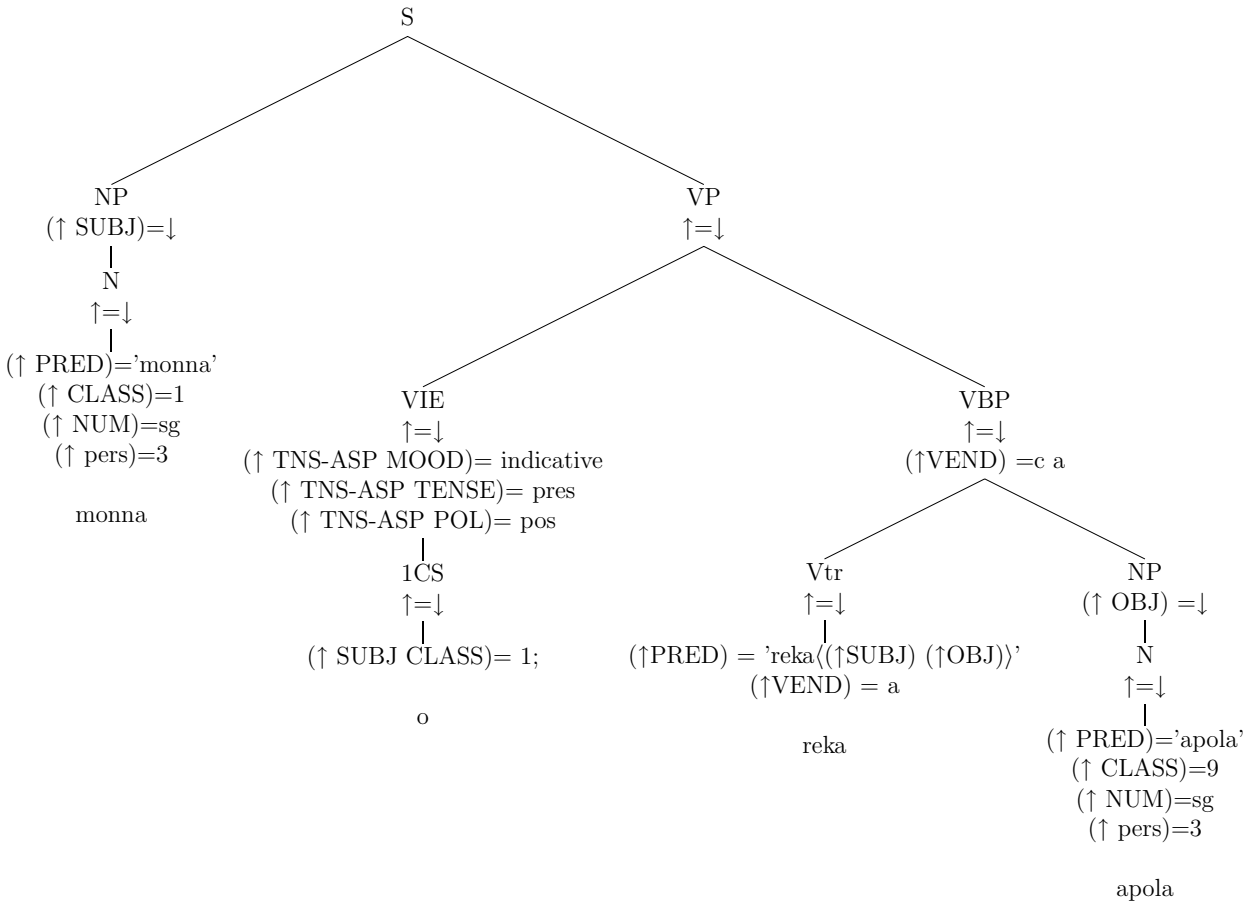

Figure 5.3: The c-structure of *monna o reka apola*

demonstration, we reduce the functional description by replacing the numbers of f-structures which are equated each with the lowest respective number, see Table (5.2). The reduction demonstrates that there are only four f-structures resulting from our analysis: *f*1 which contains a subject in f-structure *f*2 ((*f*1 SUBJ)=*f*2), an object in f-structure *f*9 ((*f*1 OBJ) =*f*9), and the f-structure TNS-ASP *f*4 created by the 2-part equations *f*1 TNS-ASP MOOD/TENSE/POL. All these data are contained in the thus simplified attribute-value matrix shown in Figure 5.6.

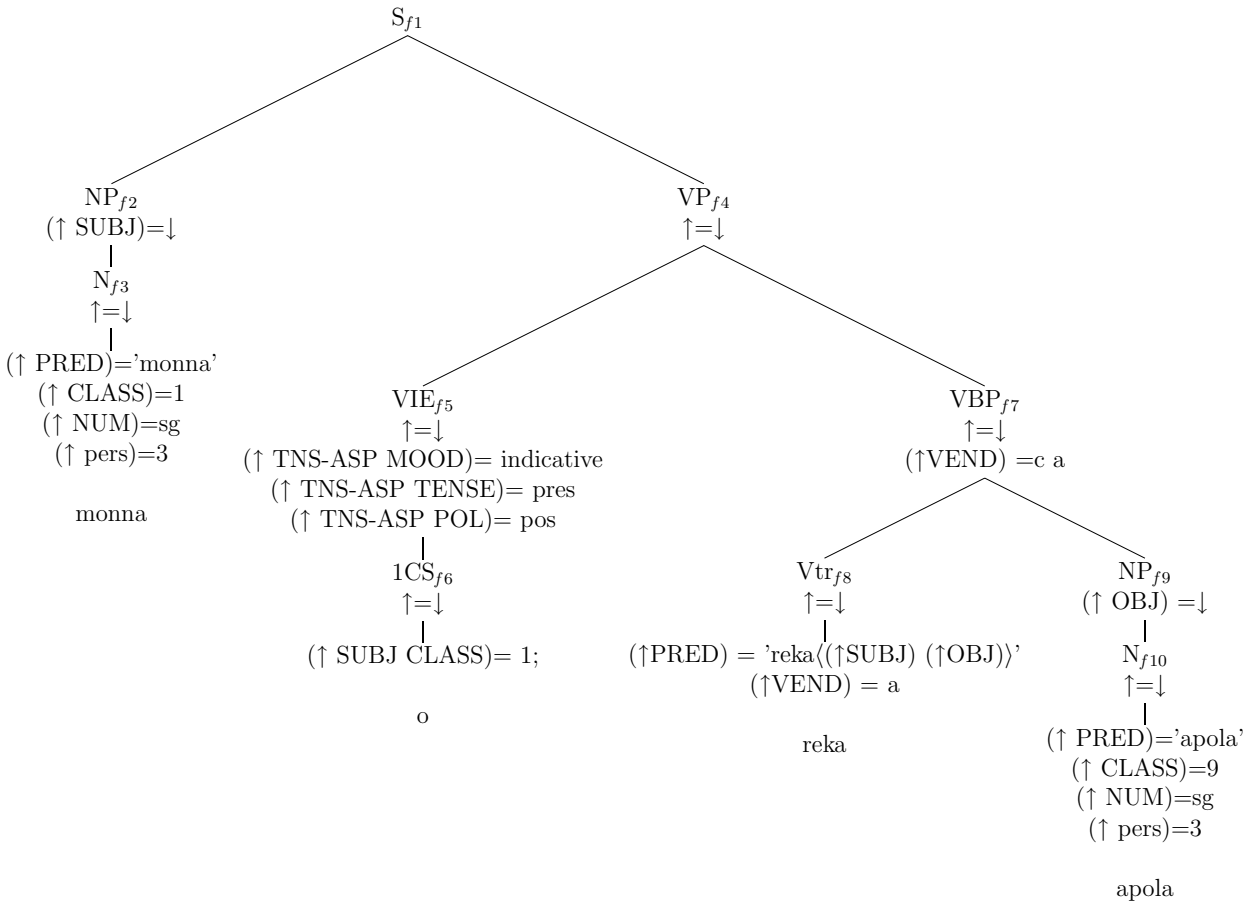

Figure 5.4: Numbered nodes at c-structure of *monna o reka apola*

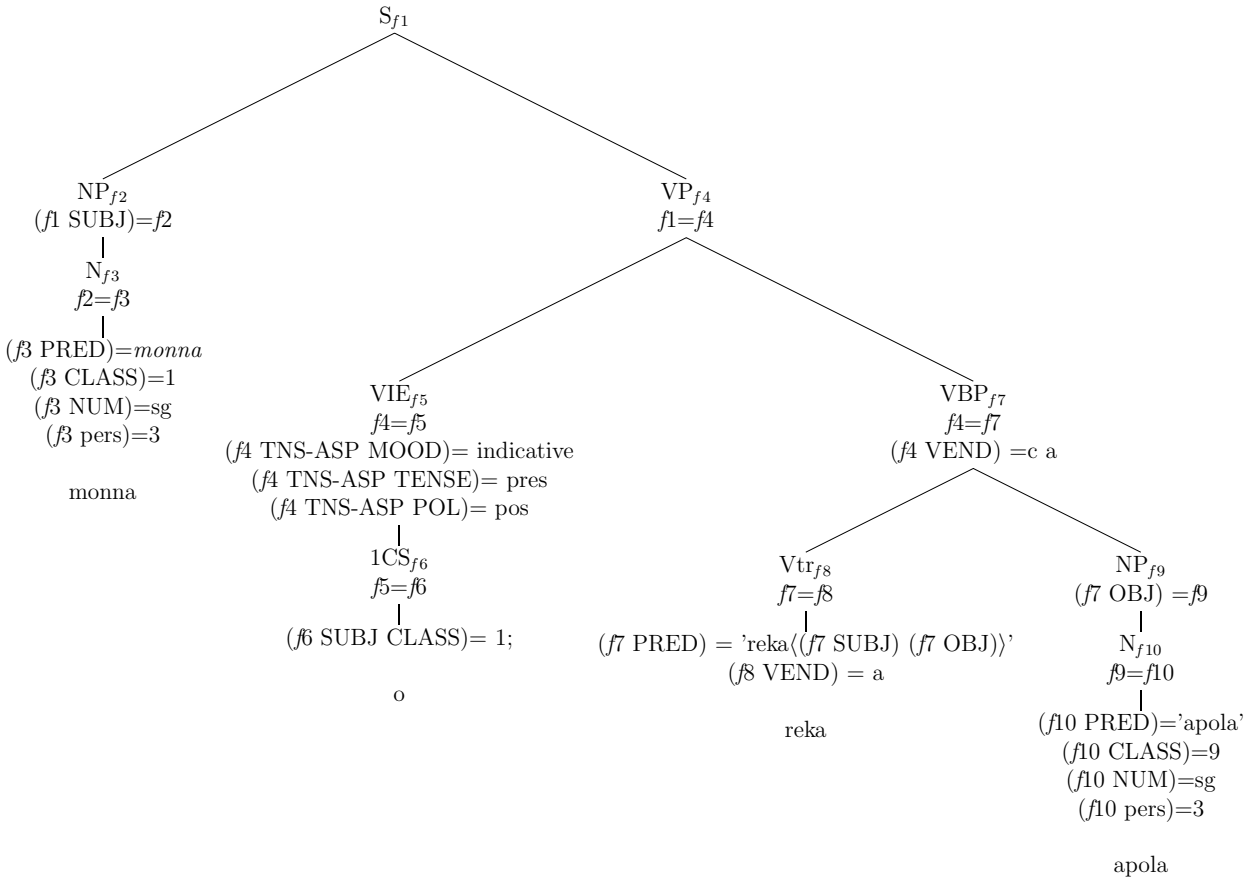

Figure 5.5: Instantiation of metavariables (↑ and ↓) of f-structures in the c-structure of *monna o reka apola*

a. (f1 SUBJ)=f2
b. f2=f3
c. (f3 PRED)=*monna*
d. (f3 CLASS)=1
e. (f3 NUM)=sg
f. (f3 pers)=3
g. f1=f4
h. f4=f5
i. (f4 TNS-ASP MOOD)= indicative
j. (f4 TNS-ASP TENSE)= pres
k. (f4 TNS-ASP POL)= pos
l. f5=f6
m. (f6 SUBJ CLASS)= 1
n. f4=f7
o. (f4 VEND) =c a
p. f7=f8
q. (f7 PRED) = 'reka⟨(f7 SUBJ) (f7 OBJ)⟩'
r. f8 VEND) = a
s. (f7 OBJ) =f9
t. f9=f10
u. (f10 PRED)='apola'
v. (f10 CLASS)=9
x. (f10 NUM)=sg
y. (f10 pers)=3

Table 5.1: Functional equations of *monna o reka apola*

a. (f1 SUBJ)=f2
c. (f2 PRED)=*monna*
d. (f2 CLASS)=1
e. (f2 NUM)=sg
f. (f2 pers)=3
i. (f1 TNS-ASP MOOD)= indicative
j. (f1 TNS-ASP TENSE)= pres
k. (f1 TNS-ASP POL)= pos
m. (f1 SUBJ CLASS)= 1
p. (f1 VEND) =c a
q. (f1 PRED) = 'reka⟨(f1 SUBJ) (f1 OBJ)⟩'
r. f1 VEND) = a
s. (f1 OBJ) =f9
u. (f9 PRED)='apola'
v. (f9 CLASS)=9
x. (f9 NUM)=sg
y. (f9 pers)=3

Table 5.2: Abbreviated version of functional equations of *monna o reka apola*

$$
\begin{bmatrix}
\text{PRED} & \text{'reka}\left\langle (f1\ \text{SUBJ})\ (f1\ \text{OBJ})\right\rangle\text{'} \\
\text{TNS-ASP} & \begin{bmatrix} \text{MOOD} & \text{indicative} \\ \text{TENSE} & \text{pres} \\ \text{POL} & \text{pos} \end{bmatrix} \\
\text{SUBJ} & \underset{f2}{\begin{bmatrix} \text{PRED} & \text{'monna'} \\ \text{CLASS} & 1 \\ \text{NUM} & \text{sg} \\ \text{PERS} & 3 \end{bmatrix}} \\
\text{OBJ} & \underset{f9}{\begin{bmatrix} \text{PRED} & \text{'apola'} \\ \text{CLASS} & 9 \\ \text{NUM} & \text{sg} \\ \text{PERS} & 3 \end{bmatrix}} \\
\underset{f1}{\text{VEND a}} &
\end{bmatrix}
$$

Figure 5.6: The f-structures of *monna o reka apola*.

5.1.2.4 The Pargram project

Concerning the implementation of the LFG formalism, we make use of the Xerox Linguistic Environment (XLE) provided by the Xerox Palo Alto Research Centre (PARC) in the framework of a research license. As foreseen in the LFG theory, XLE mainly provides two levels of descriptions: c-structure and f-structure. It also offers packed representations of ambiguous f-structures, the "fcharts". It may utilise a hand written full form lexicon or a morphological analyser. Rules are written in the LFG-formalism. Within the framework of the **Parallel Grammar** project (Pargram, cf. `http://www2.parc.com/isl/groups/nltt/pargram`) guidelines are provided that help to produce parallel representations of different languages. We will try to adhere to them from the start in order to provide the basics for the next step ahead (cf. section 6), the machine translation from Northern Sotho to English. Only a full form lexicon is however described, as there is ongoing work on automated morphological analysis of Northern Sotho (e.g. Anderson and Kotzé (2006)). At a later stage, such a morphological analyser may become part of a more substantial XLE implementation of Northern Sotho morphosyntax.

The implementation described in this chapter, adheres to the attributes suggested by the Pargram project, only few attributes were added, like, e.g. VEND to indicate the verbal ending of the verb stem in question. Another attribute added is TNS-ASP pol(arity), allowing for the two values positive and negative. TNS-ASP pol assists XLE in producing analyses containing the polarity information, hence making them consistent to the Northern Sotho modal system as described in Table (3.2) of paragraph 3.2.1.3 (Introduction to the modal system).

5.2 Implementation

5.2.1 The lexicon

The small, full form lexicon built for this study contains a number of verbs, nouns and other elements. In order to simplify the reading of the lexicon, some templates were defined[5]. Instead of having to add the attribute-value pair (↑ PERS) = 3 to all lexicon entries of nouns of the third person, one can define a template like e.g. PERS(_P) = (↑ PERS) =

[5]More information on XLE templates may be found at `http://www2.parc.com/isl/groups/nltt/xle/doc/walkthrough.html`.

_P. All respective lexicon entries may then have the person information added in a more human-readable form: '@(PERS 3)'. The same applies for other attributes occurring in this implementation, like noun class or number.

The lexicon, part 1 in (81) shows intransitive verb stems; the grammar fragment lexicon moreover contains saturated transitive, transitive, half saturated double transitive and double transitive verb stems (cf. (82)), as described in paragraph 3.2.1.6. Some past tense forms were added, all have the attribute VEND marked accordingly.

```
"intransitive:  [to] walk"
sepela    Vitr *  (↑ PRED) = 'sepela ⟨ (↑ SUBJ)⟩'  @(VEND a).
sepelang  Vitr *  (↑ PRED) = 'sepela ⟨ (↑ SUBJ)⟩'  @(VEND ang).
sepele    Vitr *  (↑ PRED) = 'sepela ⟨ (↑ SUBJ)⟩'  @(VEND e).
sepeleng  Vitr *  (↑ PRED) = 'sepela ⟨ (↑ SUBJ)⟩'  @(VEND eng).

"intransitive:  [to] speak"
bolela    Vitr *  (↑ PRED) = 'bolela ⟨ (↑ SUBJ)⟩'  @(VEND a).
bolelang  Vitr *  (↑ PRED) = 'bolela ⟨ (↑ SUBJ)⟩'  @(VEND ang).
bolele    Vitr *  (↑ PRED) = 'bolela ⟨ (↑ SUBJ)⟩'  @(VEND e).
boleleng  Vitr *  (↑ PRED) = 'bolela ⟨ (↑ SUBJ)⟩'  @(VEND eng).
boletše   Vitr *  (↑ PRED) = 'bolela ⟨ (↑ SUBJ)⟩'  @(VEND ile)
                  (↑ TNS-ASP TENSE) = past.

"intransitive:  [to] flee"
tšhaba    Vitr *  (↑ PRED) = 'tšhaba ⟨ (↑ SUBJ)⟩'  @(VEND a).
tšhabe    Vitr *  (↑ PRED) = 'tšhaba ⟨ (↑ SUBJ)⟩'  @(VEND e).
tšhabile  Vitr *  (↑ PRED) = 'tšhaba ⟨ (↑ SUBJ)⟩'  @(VEND ile)
                  (↑ TNS-ASP TENSE) = past.
```

(81) The intransitive verb stem entries of the lexicon (part 1)

In Northern Sotho, reflexivity is not explicitly expressed with a constituent carrying the object function, like in other languages ('oneself'), but encoded in the verb stem. The saturated transitive verb *ipshina* 'enjoy oneself' therefore does not require any such external object to appear. A correct analysis must however show that the object of the (originally transitive) verb is identical to the subject, like in (5.7) on page 240. According to the XLE "walkthrough"-page[6], however, the basic ontology of f-structures defined by Kaplan and Bresnan (1982) did not include the definition of an absent element carrying a grammatical

[6]The "walkthrough"-page gives comprehensive practical advice on how to develop a grammar in XLE, http://www2.parc.com/isl/groups/nltt/xle/doc/walkthrough.html.

function. An equation of the type (↑ SUBJ) = (↑ OBJ) is not acceptable, as the f-structure would become indeterminate. Therefore, in the lexicon, the subject's attributes like number, person and class are set equal to the object's respective attributes in the lexicon. The entry is additionally marked with the attribute 'TYPE' = refl(exive). In the rules section (cf. paragraph 5.2.2), objects of such verbs are declared as 'PRO(noun)'.

```
"monna o a ipshina."

    17[PRED    'ipshina<[1:monna], [3-OBJ:pro]>'      ]
    18[       1[PRED 'monna'                          ]
    14[SUBJ   2[CLASS 1, NUM sg, PERS 3               ]
    15[      29[                                      ]
    54[
     7[         [PRED 'pro'                           ]
     9[OBJ      [PRON [TYPE null]                     ]
     3[         [CLASS 1, NUM sg, PERS 3              ]
     6[
    51[TNS-ASP [FORM long, MOOD indicative, TENSE pres]
    44[CLAUSE-TYPE decl, TYPE refl, VEND a
    45[
    46[
    47[
    50[                                               ]
```

Figure 5.7: F-structure containing a saturated transitive verb *monna o a ipshina.* '(a) man enjoys himself.'

Lexicon entries describing a half-saturated double transitive verb stem, e.g. *mpha/mphe* '[to] give me', where the object concord of the first person singular (*N-*) is fused to the stem of *fa* '[to] give', are handled similarly, as shown in (82). As person, class and number of the verb's oblique is known, this data is contained in the entry.

```
"transitive or saturated transitive:  [to] enjoy (oneself)"
ipshina  Vsattr *     (↑ PRED) = 'ipshina ⟨ (↑ SUBJ) (↑ OBJ)⟩'  @(VEND a)
                      (↑ TYPE) = refl
                      (↑ OBJ CLASS) = (↑ SUBJ CLASS)
                      (↑ OBJ NUM) = (↑ SUBJ NUM)
                      (↑ OBJ PERS) = (↑ SUBJ PERS).

"transitive:  [to] buy"
reka     Vtr *        (↑ PRED)='reka ⟨(↑ SUBJ)(↑ OBJ)⟩'         @(VEND a).
rekang   Vtr *        (↑ PRED)='reka ⟨ (↑ SUBJ)(↑ OBJ)⟩'        @(VEND ang).
reke     Vtr *        (↑ PRED)='reka ⟨ (↑ SUBJ)(↑ OBJ)⟩'        @(VEND e).
rekeng   Vtr *        (↑ PRED)='reka ⟨ (↑ SUBJ)(↑ OBJ)⟩'        @(VEND eng).
rekile   Vtr *        (↑ PRED)='reka ⟨ (↑ SUBJ)(↑ OBJ)⟩'        @(VEND ile)
                      (↑ TNS-ASP TENSE) = past.

"transitive:  [to] close"
bula     Vtr *        (↑ PRED)='bula ⟨(↑ SUBJ)(↑ OBJ)⟩'         @(VEND a).
bule     Vtr *        (↑ PRED)='bula ⟨(↑ SUBJ)(↑ OBJ)⟩'         @(VEND e).
bulang   Vtr *        (↑ PRED)='bula ⟨(↑ SUBJ)(↑ OBJ)⟩'         @(VEND ang).
buleng   Vtr *        (↑ PRED)='bula ⟨(↑ SUBJ)(↑ OBJ)⟩'         @(VEND eng).

"half saturated double transitive:  [to] give me"
mpha     Vhsatdtr *   (↑ PRED)='mpha ⟨(↑ SUBJ)(↑ OBJ)(↑ OBL)⟩'  @(VEND a).
                      (↑ OBL NUM) = sg
                      (↑ OBL PERS) = 1
mphe     Vhsatdtr *   (↑ PRED)='mpha ⟨(↑ SUBJ)(↑ OBJ)(↑ OBL)⟩'  @(VEND e).
                      (↑ OBL NUM) = sg
                      (↑ OBL PERS) = 1

"double transitive:  [to] give"
fa       Vdtr *       (↑ PRED)='fa ⟨(↑ SUBJ)(↑ OBL)(↑ OBJ)⟩'    @(VEND a).
fe       Vdtr *       (↑ PRED)='fa ⟨(↑ SUBJ)(↑ OBL)(↑ OBJ)⟩'    @(VEND e).
```

(82) Other verb stem entries of the lexicon (part 2)

The lexicon also contains a number of nouns of all noun classes appearing in Northern Sotho, cf. (83) . Each noun is entered with its noun class, person and number. Note that the quoted translations in the last column of (83) are comments (marked by double quotes in XLE).

```
tate       N *   (↑ PRED) = 'tate' @(CLASS 1a) @(PERS 3) @(NUM sg).      "father"
monna      N *   (↑ PRED) = 'monna' @(CLASS 1) @(PERS 3) @(NUM sg).      "man"
banna      N *   (↑ PRED) = 'banna' @(CLASS 2) @(PERS 3) @(NUM pl).      "men"
bana       N*    (↑ PRED) = 'bana' @(CLASS 2) @(PERS 3) @(NUM pl).       "children"
motato     N *   (↑ PRED) = 'motato' @(CLASS 3) @(PERS 3) @(NUM sg).     "wire"
mothepa    N *   (↑ PRED) = 'mothepa' @(CLASS 3) @(PERS 3) @(NUM sg).    "young woman"
mmutla     N *   (↑ PRED) = 'mmutla' @(CLASS 3) @(PERS 3) @(NUM sg).     "hare"
merako     N *   (↑ PRED) = 'merako' @(CLASS 4) @(PERS 3) @(NUM pl).     "stone wall"
lemati     N *   (↑ PRED) = 'lemati' @(CLASS 5) @(PERS 3) @(NUM sg).     "door"
lengwalo   N *   (↑ PRED) = 'lengwalo' @(CLASS 5) @(PERS 3) @(NUM sg).   "letter"
lesogana   N *   (↑ PRED) = 'lesogana' @(CLASS 5) @(PERS 3) @(NUM sg).   "young man"
mangwalo   N *   (↑ PRED) = 'mangwalo' @(CLASS 6) @(PERS 3) @(NUM pl).   "letters"
masogana   N *   (↑ PRED) = 'masogana' @(CLASS 6) @(PERS 3) @(NUM sg).   "young men"
meetse     N *   (↑ PRED) = 'meetse' @(CLASS 6) @(PERS 3).               "water"
malekere   N *   (↑ PRED) = 'malekere' @(CLASS 6) @(PERS 3)@(NUM pl) .   "sweets"
sepetlele  N *   (↑ PRED) = 'sepetlele' @(CLASS 7) @(PERS 3) @(NUM sg).  "hospital"
ditho      N *   (↑ PRED) = 'ditho' @(CLASS 8) @(PERS 3) @(NUM pl).      "limbs"
puku       N *   (↑ PRED) = 'puku' @(CLASS 9) @(PERS 3) @(NUM sg).       "book"
dipuku     N *   (↑ PRED) = 'dipuku' @(CLASS 10) @(PERS 3) @(NUM pl).    "books"
```

(83) Noun entries of the lexicon

Next, some of the concords and morphemes and, finally, punctuation are listed in (84) to (87) . Note that object concords occur in a pronominal function replacing an omitted (or topicalised) object (cf. e.g. paragraph 3.2.1.1), therefore these entries have a PRED-value defined. Not all of these items' possible parts of speech are entered, *go*, for example is fourteenfold ambiguous[7]. Here, only six of the possible parts of speech are listed. Again, templates are used for a clearer overview, e.g. @C6P3Npl, an abbreviated entry of (↑ SUBJ CLASS) = 6 (↑ SUBJ PERS) = 3 (↑ SUBJ NUM) = pl. It is not possible to foresee the value of the attribute 'person' for entries of class 1, hence this attribute is not mentioned for these entries.

In paragraph 3.2.5.1, Table 3.15 describes the long form of the indicative. Only this constellation contains the present tense morpheme, namely *a*. As this form may only appear if the clause ends after the verb stem, the grammar rules described in paragraph 5.2.4.2 will make use of the attribute TNS-ASP FORM (with the value 'long').

[7] *go* seems to be the most ambiguous linguistic unit of Northern Sotho: it may occur as an object concord of class 15, object concord of the locative classes, object concord of the second person singular, subject concord of class 15 (set 1 and set 2), indefinite subject concord (set 1 and set 2), subject concord of the locative classes (set 1 and set 2), class prefix of class 15, locative particle, copulative indicating either an indefinite subject or a subject of class 15 or a locative subject.

```
"a is a subject concord of classes 1,6, an object concord of class 6"
"and a present or past tense morpheme."
a     2CS *     @C1Nsg;
      3CS *     @C1Nsg;
      1CS *     @C6NplP3;
      2CS *     @C6NplP3;
      3CS *     @C6NplP3;
      CO *      (↑ PRED) = 'pro' @(CLASS 6) @(NUM pl) @(PERS 3) ;
      MORPH *   (↑ TNS-ASP FORM) = long
                (↑ TNS-ASP TENSE) = pres;
      MORPH *   (↑ TNS-ASP TENSE) = past.
"ba is a subject concord of class 2, and an object concord of class 2"
ba    1CS *     @C2NplP3;
      2CS *     @C2NplP3;
      3CS *     @C2NplP3;
      CO *      (↑ PRED) = 'pro' @(CLASS 2) @(NUM pl) @(PERS 3) .
"bja is a subject concord of class 14 (3rd set)"
bja   3CS *     @C14NsgP3.
"bo is a subject concord of class 14, and an object concord of class 14"'
bo    1CS *     @C14NsgP3;
      2CS *     @C14NsgP3;
      CO *      (↑ PRED) = 'pro' @(CLASS 14) @(NUM sg) @(PERS 3).
"di is a subject and an object concord of classes 8 and 10"
di    1CS *     @C8NplP3;
      2CS *     @C8NplP3;
      1CS *     @C10NplP3;
      2CS *     @C10NplP3;
      CO *      (↑ PRED) = 'pro' @(CLASS 8) @(NUM pl) @(PERS 3) ;
      CO *      (↑ PRED) = 'pro' @(CLASS 10) @(NUM pl) @(PERS 3) .
```

(84) Concords: entries of the lexicon (part 1)

```
"e is a subject concord of classes 4 and 9, a neutral subject concord,"
"and an object concord of classes 4 and 9"
e    1CS *       @C4NplP3;
     2CS *       @C4NplP3;
     1CS *       @C9NsgP3;
     2CS *       @C9NsgP3;
     1CSNEUT *   (↑ SUBJ PERS) = 3;
     2CSNEUT *   (↑ SUBJ PERS) = 3;
     CO *        (↑ PRED) = 'pro' @(CLASS 4) @(NUM) = pl @(PERS 3) ;
     CO *        (↑ PRED) = 'pro' @(CLASS 9) @(NUM) = sg @(PERS 3) .
"ga is a negation morpheme, occurring alone and"
"as the first part of the negation cluster ga se"
ga   MORPH *     (↑ NEG) = ga (↑ TNS-ASP POL) = neg;
     MORPH *     (↑ NEG1) = ga (↑ TNS-ASP POL) = neg.
"go is a subject and an object concord of classes 15 and LOC"
"Note that a number of parts of speech of go are not listed here"
"Class 15 is the infinitite class, no person or number"
"Class LOC contains locatives which often are used adverbially"
go   1CS *       (↑ SUBJ CLASS) = 15;
     2CS *       (↑ SUBJ CLASS) = 15;
     1CS *       (↑ SUBJ CLASS) = LOC;
     2CS *       (↑ SUBJ CLASS) = LOC;
     CO *        (↑ PRED) = 'pro' @(CLASS 15);
     CO *        (↑ PRED) = 'pro' @(CLASS LOC).
"gwa is a subject concord of classes 15 and LOC (3rd set)"
gwa  3CS *       (↑ SUBJ CLASS) = 15;
     3CS *       (↑ SUBJ CLASS) = LOC.
"ka is the subject concord of the 3rd set of the 1st person"
"and a potential morpheme"
ka   3CS *       (↑ SUBJ PERS) = 1 (↑ SUBJ NUM) = sg;
     MORPHpot *  .
"la is a subject concord of class 5 and one of the 2nd person plural"
la   3CS *       @C5NsgP3;
     3CS *       (↑ SUBJ PERS) = 2 (↑ SUBJ NUM) = pl.
```

(85) Concords: entries of the lexicon (part 2)

```
le    1CS *       @C5NsgP3;
      2CS *       @C5NsgP3;
      1CS *       (↑ SUBJ PERS) = 2 (↑ SUBJ NUM) = pl;
      2CS *       (↑ SUBJ PERS) = 2 (↑ SUBJ NUM) = pl;
      CO *        (↑ PRED) = 'pro' @(CLASS 5) @(NUM sg) @(PERS 3).

o     1CS *       @C1Nsg;
      1CS *       @C1aNsgP3;
      1CS *       (↑ SUBJ PERS)= 2 (↑ SUBJ NUM) = sg;
      2CS *       (↑ SUBJ PERS)= 2 (↑ SUBJ NUM) = sg;
      1CS *       @C3NsgP3;
      2CS *       @C3NsgP3;
      CO *        (↑ PRED) = 'pro' @(CLASS 3) @(NUM sg) @(PERS 3).

mo    CO *        (↑ PRED) = 'pro' @(CLASS 1) @(NUM sg) @(PERS 3).

ra    3CS *       (↑ SUBJ PERS) = 1 (↑ SUBJ NUM) = pl.

re    1CS *       (↑ SUBJ PERS) = 1 (↑ SUBJ NUM) = pl;
      2CS *       (↑ SUBJ PERS) = 1 (↑ SUBJ NUM) = pl.

sa    3CS *       @C7NsgP3.

se    1CS *       @C7NsgP3;
      2CS *       @C7NsgP3;
      CO *        @C7NsgP3;
      MORPH *     (↑ NEG) = se (↑ TNS-ASP POL) = neg.
      MORPH *     (↑ NEG2) = se (↑ TNS-ASP POL) = neg.

tša   3CS *       @C8NplP3;
      3CS *       @C10NplP3.

wa    3CS *       @C3NsgP3;
      3CS *       (↑ SUBJ PERS) = 2 (↑ SUBJ NUM) = sg.

ya    3CS *       @C4NplP3;
      @C9NplP3;
      3CSNEUT *   (↑ SUBJ PERS) = 3.
```

(86) Concords: entries in the lexicon (part 3)

```
"future tense morphemes"
tlo   MORPH *       (↑ TNS-ASP TENSE) = fut.
tla   MORPH *       (↑ TNS-ASP TENSE) = fut.
"Punctuation"
.     PERIOD *.
,     COMMA *.
!     EXCLMARK *.
?     QUEMARK *.
```

(87) future tense morphemes and punctuation in the lexicon

5.2.2 The basic verbal phrase (VBP)

Table 5.3: Constellations forming the VBP

description	**VBP**			
	pos-1	**pos 0**	**pos+1**	**pos+2**
VBP		$\mathbf{V}^{\text{itr}}$		
VBP		$\mathbf{V}^{\text{tr}}$		$_{\text{OBJ}}\mathbf{NP}$
VBP		$\mathbf{V}^{\text{dtr}}$	$_{\text{OBJ}-\text{TH}}\mathbf{NP}$	$_{\text{OBJ}}\mathbf{NP}$
$\mathbf{VBP}^{\text{p}}$	$_{\text{OBJ}}\mathbf{CO}_{\text{categ}}$	$\mathbf{V}^{\text{tr}}$		
VBP	$_{\text{OBJ}-\text{TH}}\mathbf{CO}_{\text{categ}}$	$\mathbf{V}^{\text{dtr}}$		$_{\text{OBJ}}\mathbf{NP}$

In paragraph 3.2.2, the core element of all verbal phrases, the basic verbal phrase was defined[8]. Butt et al. (1999, p. 50) state that in English grammar, secondary objects are to be called OBL (not OBJ2 or OBJind), because they cannot undergo passivization. The XLE implemenation of an English grammar available from XEROX PARC however makes use of the argument OBJ-TH(ematic) for secondary objects subcategorised by double transitive verbs, e.g. 'I gave **him** a book' while other secondary objects, like the prepositional phrase in e.g. 'I am looking **for the book**' remain labeled as OBL. When preparing for a machine translation, similarity to the English grammar should be aimed for, the indirect object OBJind is hence renamed to OBJ-TH in Table 5.3.

Our VBP coding in XLE contains a number of options contained in braces ({,}); each

[8]As a reminder the contents of Table 3.14 are repeated in Table 5.3. Note that (as described in paragraph 3.2.5.1) according to our understanding, a sentence border should appear following the intransitive basic verbal phrase and the transitive basic verbal phrase where the object appears as an object concord.

option is separated from the other by the disjunction "|". Some verbal endings indicate a certain mood which is thus entered right away (the functional uniqueness principle then prohibits other analyses than those indicated).

The grammar rule describing the VBP can be split into three parts: verb stems subcategorising no arguments, verb stems subcategorising one and, lastly, two arguments, here described as NP[9]. As described in paragraph 3.2.2, the first case of a VBP processes the intransitive verbs. This VBP option firstly distinguishes the two cases where no object NP appears: both solely consist of the verb stem. Such a VBP may either consist of an intransitive verb or a saturated transitive verb. For the latter case, a pronominal object is defined ((↑OBJ PRED) = 'pro', (↑OBJ PRON TYPE)=null). This object will be added to the f-structure (see Figure 5.7), its attributes are stored with the verb entry in the lexicon. The verbal endings *-ang* or *eng* prescribe an imperative, in order to allow processing of the other verbal endings, these must be mentioned, too (the uniqueness principle would otherwise only allow for *-ang* or *eng* to appear with the attribute VEND).

```
VBP --> { { Vitr   : { (^ VEND) = ang  (^ TNS-ASP MOOD)= imperative
                     | (^ VEND) = eng  (^ TNS-ASP MOOD)= imperative
                     | (^ VEND) = a
                     | (^ VEND) = e
                     | (^ VEND) = ile
                     }
          | Vsattr : (^ OBJ PRED)= 'pro'
                     (^ OBJ PRON TYPE)= null
                     { (^ VEND) = ang  (^ TNS-ASP MOOD)= imperative
                     | (^ VEND) = eng  (^ TNS-ASP MOOD)= imperative
                     | (^ VEND) = a
                     | (^ VEND) = e
                     | (^ VEND) = ile
                     }
          }
...
```

The rule continues its description of the VBP with the two cases where one object is available, that of a transitive verb and that of a half saturated double transitive verb[10]. These are described similarly to the first two cases. The next item to be described is the case of an object concord preceding the transitive verb stem. The object concord has the object function assigned.

[9]At a later stage of the project, subcategorised clauses and adverbial attributes will be added.

[10]We will refer again to the line "e: (↑ TNS-ASP FORM)~= long" in paragraph 5.2.4.2.

```
...
        | e : (^ TNS-ASP FORM) ~= long;
          { Vtr    : { (^ VEND) = ang  (^ TNS-ASP MOOD)= imperative
                     | (^ VEND) = eng  (^ TNS-ASP MOOD)= imperative
                     | (^ VEND) = a
                     | (^ VEND) = e
                     | (^ VEND) = ile
                    }
          | Vhsatdtr: (^ OBJ-TH PRED)= 'pro'
                      (^ OBJ-TH PRON TYPE)= null
                      { (^ VEND) = ang  (^ TNS-ASP MOOD)= imperative
                      | (^ VEND) = eng  (^ TNS-ASP MOOD)= imperative
                      | (^ VEND) = a
                      | (^ VEND) = e
                      | (^ VEND) = ile
                       }
          }
          NP: (^ OBJ)=!

          | CO: (^ OBJ)=!;

          Vtr :  { (^ VEND) = ang  (^ TNS-ASP MOOD)= imperative
                 | (^ VEND) = eng  (^ TNS-ASP MOOD)= imperative
                 | (^ VEND) = a
                 | (^ VEND) = e
                 | (^ VEND) = ile
                 }
...
```

The third part of the VBP-rule describes the two cases of the double transitive verbs with two object NPs or with object concord and object NP. For the first case, the order of objects is predefined as thematic (indirect) object preceding the direct object (cf. paragraph 3.2.1.1). As the object concord may in theory stand in for each of them (though a preference for the indirect object has been observed), the latter case would have to be divided into two further options. Note that such a rule leads to a double analysis of all double transitive verbs occurring with an object concord. In this implementation, however, only the case expected to occur more frequently is described: the object concord replacing an indirect (thematic) object.

```
...
        | e : (^ TNS-ASP FORM) ~= long;
          Vdtr:  { (^ VEND) = ang  (^ TNS-ASP MOOD)= imperative
```

```
            | (^ VEND) = eng  (^ TNS-ASP MOOD)= imperative
            | (^ VEND) = a
            | (^ VEND) = e
            | (^ VEND) = ile
            };
      NP: (^ OBJ-TH)=!;
      NP: (^ OBJ)=!

    | e : (^ TNS-ASP FORM) ~= long;
      CO: (^ OBJ-TH)=!;
      Vdtr
      NP: (^ OBJ)=!
    }.
```

In this paragraph, we have implemented a set of optional rules for the VBP that mirror the summarised VBP in Table 3.14. Paragraphs 5.2.3 and 5.2.4 show how the VBP appears in the rules defining the verbal phrases.

5.2.3 The imperative

The imperative is the simplest form of a verbal phrase (cf. paragraph 3.2.3). The non-predicative 'VPimp' either contains solely a VBP for the imperative positive, or additionally an imperative VIE (VIEimp containing the negation morpheme *se*) used for the negative form of this VP for which the condition is set that the verb stem must be described in the lexicon as ending in '*–e*' or '*–eng*' (parameter "Vend"). Sentences 88 to 92 are to be analysed by the grammar.

(88) *Bolela !*
speak !
'speak!'

(89) *Se bolele !*
speak !
'Do not speak!'

(90) *Bulang lemati !*
Close door !
'Close (the) door!'

(91) *Efa monna puku !*
Give man book !
'Give (a) man (a) book!'

(92) *Se fe monna puku !*
neg give man book !
'Do not give (a) man (a) book!'

The rule describing the positive imperative VP (VPimp) extends the VBP only to the extent that the number of the subject (↑SUBJ NUM) is described. The suffix *-ng* is used when several people are addressed (cf. paragraphs 2.7.2 and 3.2.3), therefore the respective verb stems are marked with (↑SUBJ NUM)=pl. The negated form of the imperative additionally uses the imperative verbal inflectional phrase (VIEimp) which solely contains the negation *se*. Note that the mood's polarity (TNS-ASP POL) is set to "neg(ative)" by a respective entry in the lexicon describing *se*.

```
VPimp --> { VBP:(^ TNS-ASP POL) = pos
                { (^ VEND) =c a (^ SUBJ NUM) = sg
                | (^ VEND) =c e (^ SUBJ NUM) = sg
                | (^ VEND) =c ang (^ SUBJ NUM) = pl
                | (^ VEND) =c eng (^ SUBJ NUM) = pl
                }
          | VIEimp: ^=!;
            VBP : { (^ VEND) =c e (^ SUBJ NUM) = sg
                  | (^ VEND) =c eng (^ SUBJ NUM) = pl
                  }
          }.

VIEimp -->
          MORPH: (^ NEG) =c se.
```

In the current implementation, imperative and indicative VPs are described, cf. paragraph 5.2.4). The imperative VP is contained in S. An overall root node, "Root", is defined, containing S and possibly punctuation (the parenthesis "()" occuring in the rule defining ROOT indicate optionality of the described punctuation) .

```
VP -->
   { VPimp: (^ TNS-ASP MOOD)= imperative " imperative"
   | VPpred
   }.

S -->
    VP: (^ SUBJ PRED)='pro'                " imperative "
        (^ SUBJ PRON TYPE)= null
```

```
        (^ SUBJ PERS)= 2
        (^ TNS-ASP MOOD)=c imperative.
ROOT -->
       S
       ({PERIOD
        |QUEMARK: (^ TNS-ASP MOOD)= question
        |EXCLMARK
        }).
```

Figures 5.8 and 5.9 show a XLE-analysis of a positive imperative, making use of the VBP definition of the intransitive verb in (88). The following figures, 5.10 and 5.11 demonstrate the analysis of a transitive verb of example (90). Next, figures 5.12 and 5.13 demonstrate a double transitive VBP contained in the imperative (91), and its negated form in (92), cf. figures 5.14 and 5.15.

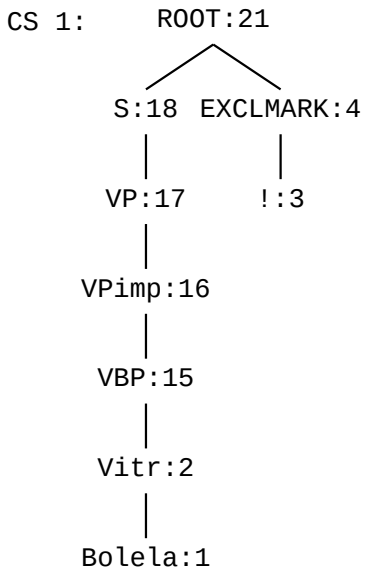

Figure 5.8: C-structure of a positive imperative intransitive *Bolela!* 'Speak!'

```
"Bolela!"

 3 PRED     'bolela<[1-SUBJ:null_pro]>'
 4          [PRED 'null_pro'          ]
 1 SUBJ     [PRON [TYPE null]         ]
 2          [NUM sg, PERS 2           ]
15
16 TNS-ASP  [MOOD imperative, POL pos]
17
18 VEND     a
21
```

Figure 5.9: F-structure of a positive imperative intransitive *Bolela!* 'Speak!'

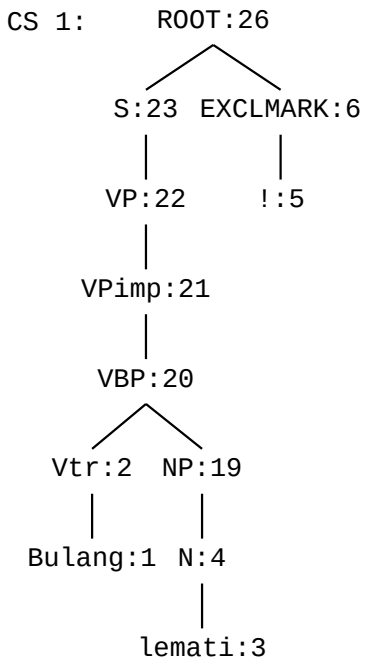

Figure 5.10: C-structure of a positive imperative transitive *Bulang lemati!* 'Close (the) door!'

```
"Bulang lemati!"

 ⎡PRED    'bula<[1-SUBJ:null_pro], [3:lemati]>'⎤
 ⎢        ⎡PRED 'null_pro'                     ⎤
5⎢SUBJ    ⎢PRON [TYPE null]                    ⎥
6⎢        ⎣NUM pl, PERS 2                      ⎦
1⎢
2⎢          3⎡PRED 'lemati'                    ⎤
20⎢OBJ      4⎢CLASS 5, NUM sg, PERS 3          ⎥
21⎢        19⎣                                ⎦
22⎢
23⎢TNS-ASP  [MOOD imperative, POL pos]
26⎣VEND     ang                                ⎦
```

Figure 5.11: F-structure of a positive imperative transitive *Bulang lemati!* 'Close (the) door!'

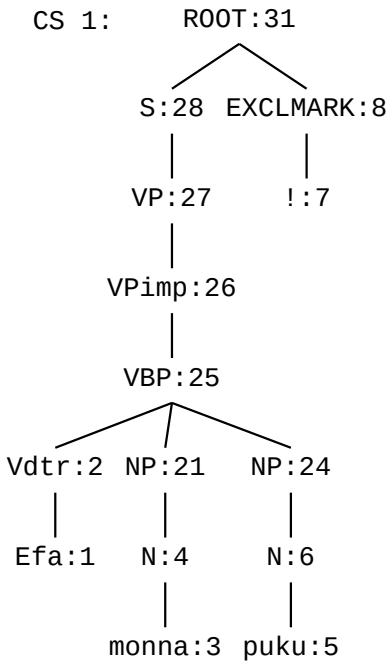

Figure 5.12: C-structure of a positive imperative double transitive *Efa monna puku!* 'Give (a) man (a) book!'

```
"Efa monna puku!"

⎡PRED     'fa<[1-SUBJ:null_pro], [3:monna], [5:puku]>'⎤
⎢         ⎡PRED 'null_pro'                            ⎤
⎢SUBJ     ⎢PRON ⎡TYPE null⎤                           ⎥
⎢         ⎣     ⎣NUM sg, PERS 2 ⎦                     ⎦
⎢       3 ⎡PRED 'monna'                               ⎤
⎢OBJ-TH 4 ⎢CLASS 1, NUM sg, PERS 3                    ⎥
⎢      21 ⎣                                           ⎦
⎢       5 ⎡PRED 'puku'                                ⎤
⎢OBJ    6 ⎢CLASS 9, NUM sg, PERS 3                    ⎥
⎢      24 ⎣                                           ⎦
⎢TNS-ASP  ⎡MOOD imperative, POL pos⎤
⎣VEND     ⎣a                       ⎦
```

Figure 5.13: F-structure of a positive imperative double transitive *Efa monna puku!* 'Give (a) man (a) book!'

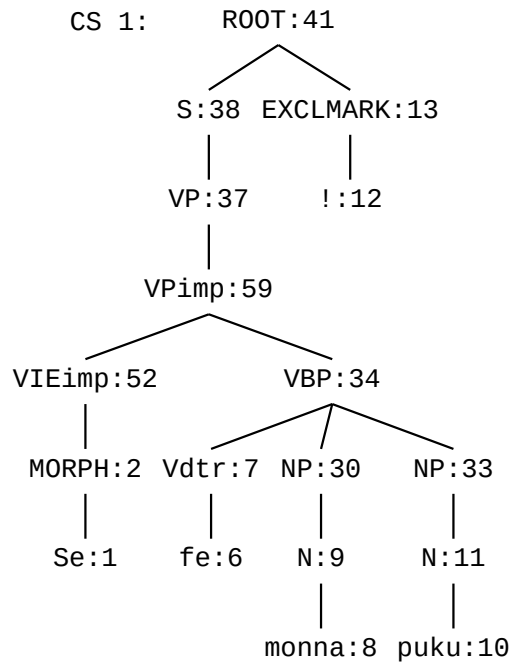

Figure 5.14: C-structure of a negated imperative double transitive *Se fe monna puku!* 'Do not give (a) man (a) book!'

```
"Se fe monna puku!"

    [PRED     'fa<[1-SUBJ:null_pro], [8:monna], [10:puku]>'
               [PRED 'null_pro'                              ]
12 SUBJ        [PRON [TYPE null]                             ]
13             [NUM sg, PERS 2                               ]
 6
 7           8 [PRED 'monna'                                 ]
34 OBJ-TH    9 [CLASS 1, NUM sg, PERS 3                      ]
 1          30 [                                            ]
 2          10 [PRED 'puku'                                 ]
52 OBJ      11 [CLASS 9, NUM sg, PERS 3                     ]
59         33  [                                            ]
37
38 TNS-ASP    [MOOD imperative, POL neg                     ]
41 [NEG se, VEND e                                          ]
```

Figure 5.15: F-structure of a negated imperative double transitive *Se fe monna puku!* 'Do not give (a) man (a) book!'

5.2.4 The predicative independent indicating mood: the indicative

5.2.4.1 General rules for the indicative

The indicative is a predicative mood, it therefore always contains (at least) a subject concord which is found in the VIE. The VPind is defined to be of the clause type declarative ((↑ CLAUSE-TYPE) = decl) in order to conform with English grammar terminology.

```
VPpred -->
       VPind: (^ CLAUSE-TYPE) = decl
              (^ TNS-ASP MOOD)= indicative.

VPind -->
       VIE
       VBP.
```

5.2.4.2 The imperfect indicative

The imperfect indicative may consist of the positive short present tense and long present tense, or its negated form. In paragraph 3.2.5.1, where this indicative is described, we modified the VBP in order to indicate that its long form ends with the verb stem. The XLE parsing algorithm however does not need such markers as the respective rule does not allow any other use of *a*. This case is determined by the constraint (↑ TNS-ASP FORM)=c long. Secondly, the VBPs where one or more NPs follow the verb stem, are exluded from the long form with the constraint "e : (↑ TNS-ASP FORM) ∼= long" ("∼=" is read as "not equal").

Furthermore, a continuation of the rule is indicated with "...", as the perfect tense and the future forms will be described in the paragraphs (5.2.4.3 and 5.2.4.4). There is no element in the short present tense form that indicates that the constellation is of the present tense. The subject concord of the first set, which is the only element of this constellation, can occur with other tenses as well (e.g. the perfect positive, cf. Table 4.7 in paragraph 4.4 on page 210 for an overview). Such missing indication is a quite regular phenomenon in Northern Sotho, as usually the VIE as a whole is seen as indicating information, like, e.g., tense. Therefore, XLE rules often contain symbolic empty elements ("e"), these are inserted in the grammar wherever a surface marker cannot be identified. These empty elements hence contain constraints that cannot be assigned to single elements.

```
"Verbal Inflectional Elements : VIE"
VIE -->   " short present tense form"
       {  1CS : (^ TNS-ASP FORM) = short;
          e : (^ TNS-ASP TENSE) = pres
              (^ VEND) =c a
          "long form "
       | 1CS
          MORPH: (^ TNS-ASP FORM) =c long
                 (^ TNS-ASP TENSE) =c pres;
          e : (^ VEND) =c a
          " negated present tense"
       |  MORPH:  (^ NEG) =c ga;
          e : (^ TNS-ASP TENSE) = pres
              (^ VEND) =c e;
          2CS
       ...
       }.
```

Sentences where the subject is not present often have several readings, as the subject concords are often ambiguous. Here, XLE offers a packed f-structure that indicates all possible readings, e.g. *o a bolela* '(s)he/it speaks' or 'you speak', where the subject concord *o* may refer to a omitted noun of the noun classes 1, 1a or class 3, or to a 2nd person singular, cf. Figure 5.16. The c-structure, however, is not affected by this ambiguity (cf. Figure 5.17).

5.2.4.3 The perfect indicative

The perfect indicative occurs in five different constellations, a positive and four negated forms (cf. paragraph 3.2.5.2). The VIE-rule of the previous paragraph is hence extended respectively.

```
"Verbal Inflectional Elements : VIE"
VIE -->
       { ...
          " perfect tense "
       |  1CS
          e : (^ VEND) =c ile
        " negated perfect tense constellations"
       |  MORPH : (^ NEG1) =c ga;
          MORPH : (^ NEG2) =c se;
          e : (^ TNS-ASP TENSE) = past
              (^ VEND) =c a;
```

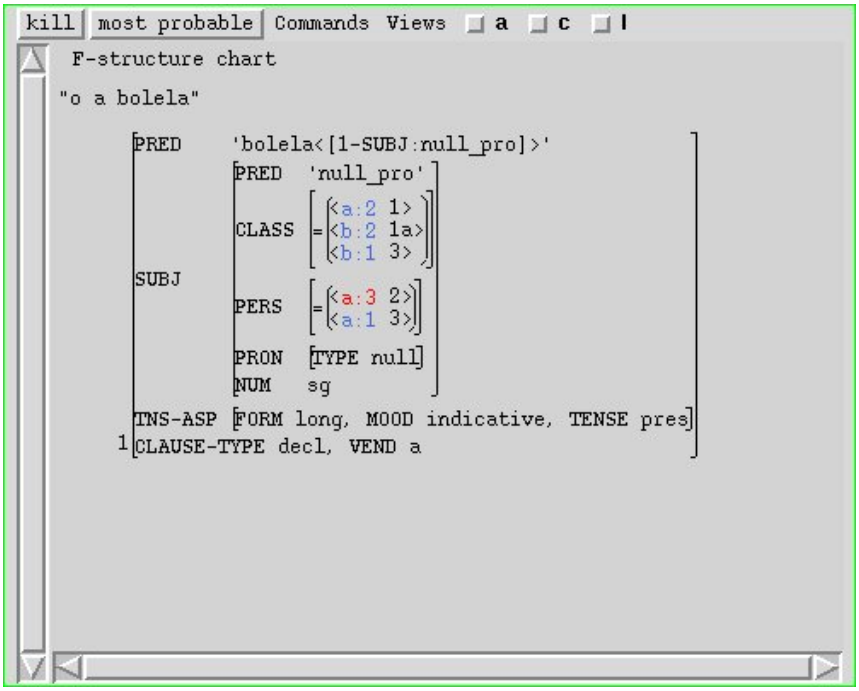

Figure 5.16: Packed f-structure of a positive indicative, present tense, subject not present *o a bolela.* '(s)he/it speaks / you speak.'

```
   3CS
|  MORPH : (^ NEG1) =c ga;
   MORPH : (^ NEG2) =c se;
   e : (^ TNS-ASP TENSE) = past
       (^ VEND) =c e;
   2CS
|  MORPH : (^ NEG) =c ga;
   e : (^ TNS-ASP TENSE) = past
       (^ VEND) =c a;
   3CS
|  MORPH : (^ NEG) =c ga;
   1CS
   MORPH:  (^ TNS-ASP TENSE) = past;
   e : (^ VEND) =c a;
...
}.
```

The following c- and f-structures show a number of example sentences of the perfect tense. The positive form is demonstrated by *lesogana le e rekile* '(a) young man bought it/them', an ambiguous sentence, for the object concord *e* may either belong to noun class 4 (plural)

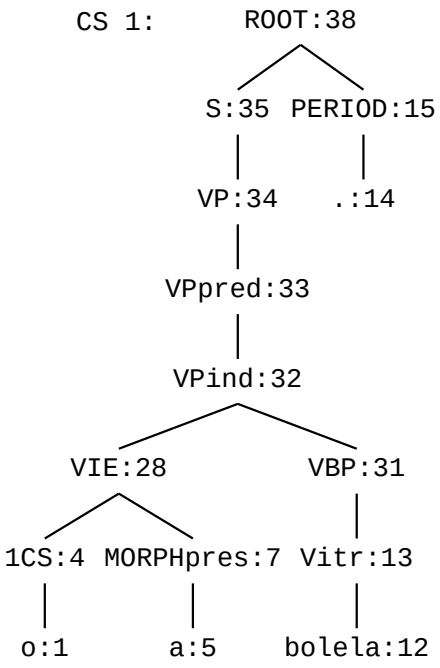

Figure 5.17: C-structure of a positive indicative, present tense, subject not present *o a bolela.* '(s)he/it speaks / you speak.'

or 9 (singular). Hence, the analysis results in two f-structures, packed in Figure 5.19. However, as the constituent analysis is not affected by this issue, only one c-structure results (cf. Figure 5.18). The negated examples are *lesogana ga se la bolela. / lesogana ga se le bolele. / lesogana ga la bolela. / lesogana ga le a bolela.* '(a) young man did not speak.' in figures 5.20/5.21, 5.22/5.23, 5.24/5.25, and 5.24/5.25.

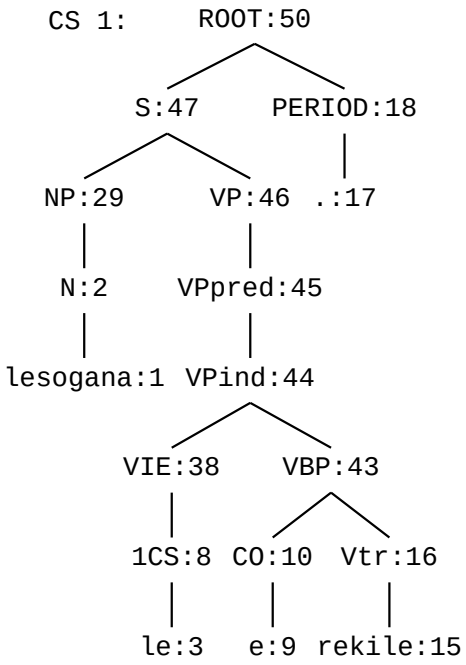

Figure 5.18: C-structure of a positive indicative, perfect tense *lesogana le e rekile.* '(a) young man bought it/them.'

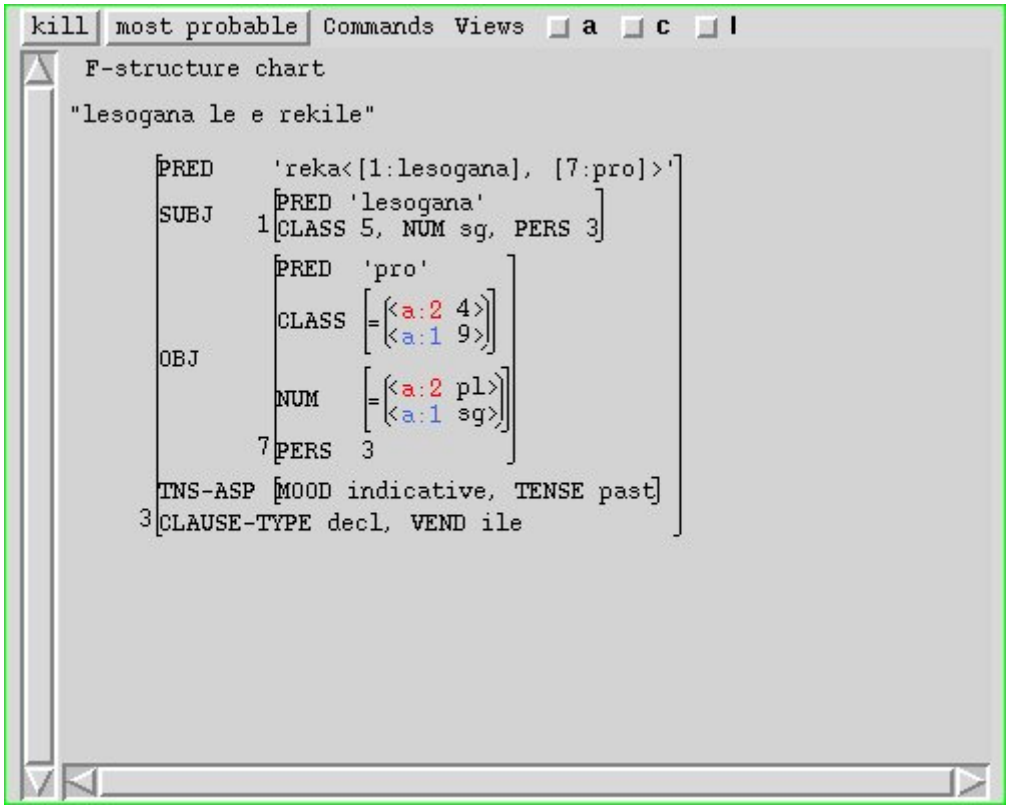

Figure 5.19: Packed f-structure of a positive indicative, perfect tense *lesogana le e rekile.* '(a) young man bought it/them.'

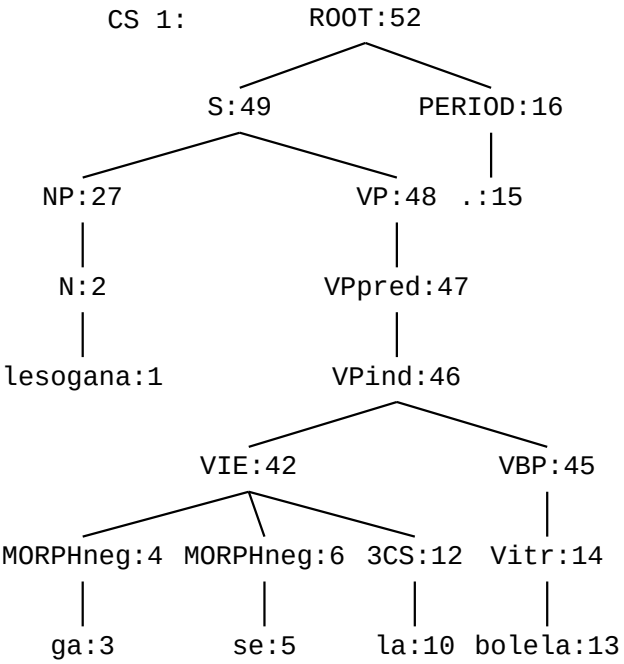

Figure 5.20: C-Structure of a negated indicative, perfect tense *lesogana ga se la bolela.* '(a) young man did not speak.'

```
"lesogana ga se la bolela."

 15 PRED     'bolela<[1:lesogana]>'
 16        1 PRED 'lesogana'
 13 SUBJ   2 CLASS 5, NUM sg, PERS 3
 14       27
 45
 10 TNS-ASP [MOOD indicative, POL neg, TENSE past]
 12 CLAUSE-TYPE decl, NEG1 ga, NEG2 se, VEND a, VTYPE main
  5
  6
  3
  4
 42
 46
 47
 48
 49
 52
```

Figure 5.21: F-Structure of a negated indicative, perfect tense *lesogana ga se la bolela.* '(a) young man did not speak.'

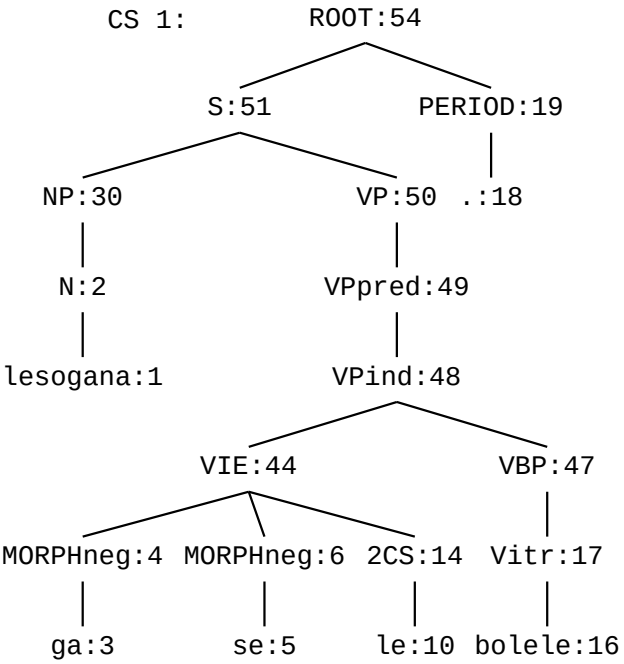

Figure 5.22: C-Structure of a negated indicative, perfect tense *lesogana ga se le bolele.* '(a) young man did not speak.'

```
"lesogana ga se le bolele."

 18 PRED     'bolela<[1:lesogana]>'
 19        1 PRED 'lesogana'
 16 SUBJ   2 CLASS 5, NUM sg, PERS 3
 17       30
 47
 10 TNS-ASP [MOOD indicative, POL neg, TENSE past]
 14 CLAUSE-TYPE decl, NEG1 ga, NEG2 se, VEND e, VTYPE main
  5
  6
  3
  4
 44
 48
 49
 50
 51
 54
```

Figure 5.23: F-Structure of a negated indicative, perfect tense *lesogana ga se le bolele.* '(a) young man did not speak.'

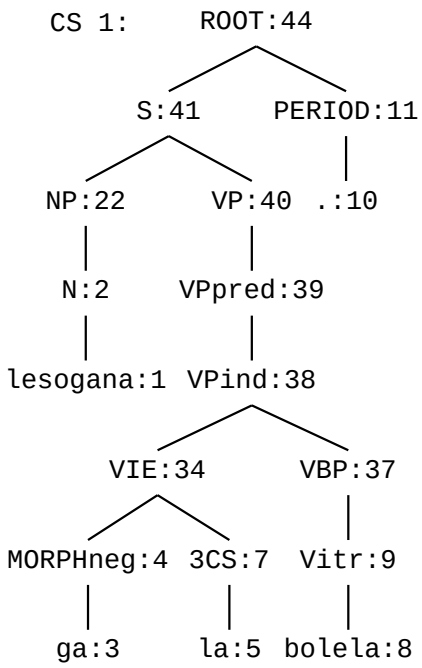

Figure 5.24: C-Structure of a negated indicative, perfect tense *lesogana ga la bolela.* '(a) young man did not speak.'

```
"lesogana ga la bolela."

    10[PRED    'bolela<[1:lesogana]>'                    ]
    11[       1[PRED 'lesogana'                          ]
     8[SUBJ   2[CLASS 5, NUM sg, PERS 3                  ]
     9[      22[                                         ]
    37[
     5[TNS-ASP [MOOD indicative, POL neg, TENSE past]
     7[CLAUSE-TYPE decl, NEG ga, VEND a, VTYPE main
     3[
     4[
    34[
    38[
    39[
    40[
    41[
    44[                                                  ]
```

Figure 5.25: F-Structure of a negated indicative, perfect tense *lesogana ga la bolela.* '(a) young man did not speak.'

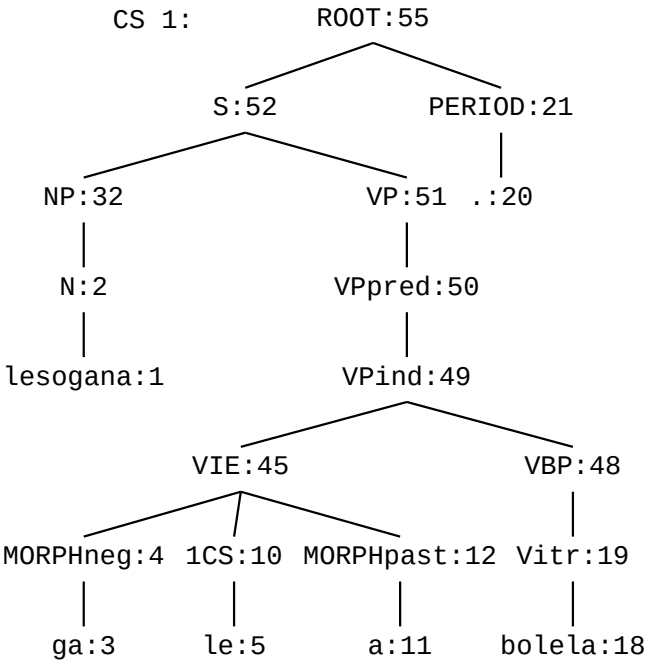

Figure 5.26: C-Structure of a negated indicative, perfect tense *lesogana ga le a bolela.* '(a) young man did not speak.'

```
"lesogana ga le a bolela."

 20 [PRED     'bolela<[1:lesogana]>'                          ]
 21 |        1 [PRED 'lesogana'                                |
 18 |SUBJ     2 [CLASS 5, NUM sg, PERS 3                      |
 19 |        32 [                                             |
 48 |
 11 |TNS-ASP [MOOD indicative, POL neg, TENSE past]
 12 |CLAUSE-TYPE decl, NEG ga, VEND a, VTYPE main
  5 |
 10 |
  3 |
  4 |
 45 |
 49 |
 50 |
 51 |
 52 |
 55 |
```

Figure 5.27: F-Structure of a negated indicative, perfect tense *lesogana ga le a bolela.* '(a) young man did not speak.'

5.2.4.4 The predicative independent indicating mood: The future indicative

Finally, we add the future indicative, as described in paragraph 3.2.5.3. The positive form contains one of the future tense morphemes *tlo* or *tla*, the negated form contains the potential morpheme *ka* and the negation *se*. Concerning the negated form, none of its elements can clearly be identified to indicate the future tense, therefore, again an empty element "e" is defined that inserts this information into the f-structure.

```
"Verbal Inflectional Elements : VIE"
VIE -->
       { ...

         " future tense "
       |  1CS
          MORPH : (^ TNS-ASP TENSE) = fut
                  (^ TNS-ASP POL) = pos
          " negated future tense "
       |  2CS
          MORPHpot
          MORPH : (^ NEG) =c se;
          e: (^ TNS-ASP TENSE) = fut
       }.
```

Example analyses of *mmutla o tlo tšhaba.* '(a) hare will flee.' (figures 5.28 and 5.29) and its negated form *mmutla o ka se tšhabe* '(a) hare will not flee.' (figures 5.30 and 5.31) conclude this section on implementation.

5.2.5 Summary

The purpose of this chapter was to show the possibility of an implementation of part of the Northern Sotho grammar fragment defined in chapter 3. Our "toy"-grammar will be extended in the future. We have defined the imperative and indicative constellations making use of a full form lexicon. At a later stage, this full form lexicon may be replaced by a morphological analyser. The current implementation, however, may already form the basis for an experimental machine translation into English, which will be the subject of the next chapter.

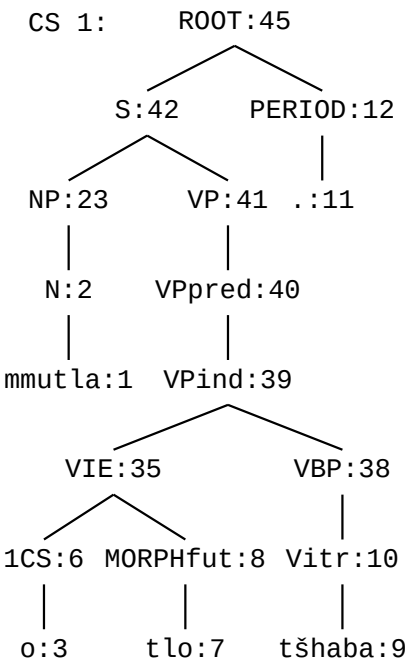

Figure 5.28: C-structure of a positive indicative, future tense *mmutla o tlo tšhaba.* '(a) hare will flee.'

"mmutla o ka se tšhabe."

```
17 PRED     'tšhaba<[1:mmutla]>'
18        1 [PRED 'mmutla'                          ]
15 SUBJ   2 [CLASS 3, NUM sg, PERS 3                ]
16       29 [                                        ]
45
10 TNS-ASP [MOOD indicative, POL neg, TENSE fut]
11 CLAUSE-TYPE decl, NEG se, VEND e
 7
 8
 3
 5
42
46
47
48
49
52
```

Figure 5.29: F-structure of a positive indicative, future tense *mmutla o tlo tšhaba.* '(a) hare will flee.'

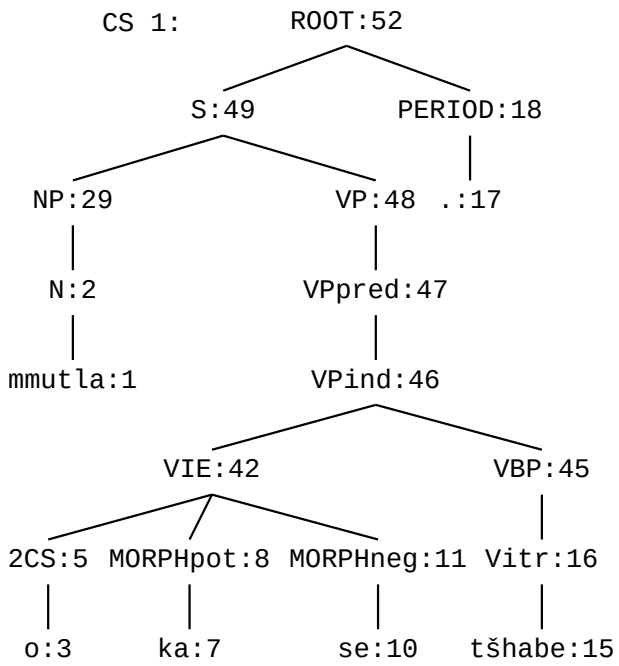

Figure 5.30: C-structure of a negated indicative, future tense *mmutla o ka se tšhabe.* '(a) hare will not flee.'

```
"mmutla o ka se tšhabe."

17 [PRED     'tšhaba<[1:mmutla]>'                          ]
18 [        1[PRED 'mmutla'                                ]
15 [SUBJ    2[CLASS 3, NUM sg, PERS 3                      ]
16 [       29[                                             ]
45 [
10 [TNS-ASP [MOOD indicative, POL neg, TENSE fut          ]
11 [CLAUSE-TYPE decl, NEG se, VEND e
 7 [
 8 [
 3 [
 5 [
42 [
46 [
47 [
48 [
49 [
52 [                                                       ]
```

Figure 5.31: F-structure of a negated indicative, future tense *mmutla o ka se tšhabe.* '(a) hare will not flee.'